A Machine Learning Approach to Reconstructing Signalling Pathways and Interaction Networks in Biology

Frank Dondelinger

Doctor of Philosophy
Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh
2013

Abstract

In this doctoral thesis, I present my research into applying machine learning techniques for reconstructing species interaction networks in ecology, reconstructing molecular signalling pathways and gene regulatory networks in systems biology, and inferring parameters in ordinary differential equation (ODE) models of signalling pathways. Together, the methods I have developed for these applications demonstrate the usefulness of machine learning for reconstructing networks and inferring network parameters from data.

The thesis consists of three parts. The first part is a detailed comparison of applying static Bayesian networks, relevance vector machines, and linear regression with L1 regularisation (LASSO) to the problem of reconstructing species interaction networks from species absence/presence data in ecology (Faisal et al., 2010). I describe how I generated data from a stochastic population model to test the different methods and how the simulation study led us to introduce spatial autocorrelation as an important covariate. I also show how we used the results of the simulation study to apply the methods to presence/absence data of bird species from the European Bird Atlas.

The second part of the thesis describes a time-varying, non-homogeneous dynamic Bayesian network model for reconstructing signalling pathways and gene regulatory networks, based on Lèbre et al. (2010). I show how my work has extended this model to incorporate different types of hierarchical Bayesian information sharing priors and different coupling strategies among nodes in the network. The introduction of these priors reduces the inference uncertainty by putting a penalty on the number of structure changes among network segments separated by inferred changepoints (Dondelinger et al., 2010; Husmeier et al., 2010; Dondelinger et al., 2012b). Using both synthetic and real data, I demonstrate that using information sharing priors leads to a better reconstruction accuracy of the underlying gene regulatory networks, and I compare the different priors and coupling strategies. I show the results of applying the model to gene expression datasets from *Drosophila melanogaster* and *Arabidopsis thaliana*, as well as to a synthetic biology gene expression dataset from *Saccharomyces cerevisiae*. In each case, the underlying network is time-varying; for *Drosophila melanogaster*, as a consequence of measuring gene expression during different developmental stages; for *Arabidopsis thaliana*, as a consequence of measuring gene expression for circadian clock genes under different conditions; and for the synthetic biology dataset, as a consequence of changing the growth environment. I show that in addition to inferring sensible network structures, the model also successfully predicts the locations of changepoints.

The third and final part of this thesis is concerned with parameter inference in ODE models of biological systems. This problem is of interest to systems biology researchers, as kinetic reaction parameters can often not be measured, or can only be estimated imprecisely from experimental data. Due to the cost of numerically solving the ODE system after each parameter adaptation, this is a computationally challenging problem. Gradient matching techniques circumvent this problem by directly fitting the derivatives of the ODE to the slope of an interpolant. I present an inference procedure for a model using nonparametric Bayesian statistics with Gaussian processes, based on Calderhead et al. (2008). I show that the new inference procedure improves on the original formulation in Calderhead et al. (2008) and I present the result of applying it to ODE models of predator-prey interactions, a circadian clock gene, a signal transduction pathway, and the JAK/STAT pathway.

The material within this thesis is partly based on my published papers and book chapters:

- Chapter 2, on reconstructing ecological networks, is based on Faisal et al. (2010).
- Chapters 3 and 4, on reconstructing signalling pathways and gene regulatory networks with information sharing, are based on Dondelinger et al. (2010), Husmeier et al. (2010), Dondelinger et al. (2012a), Lèbre et al. (2012) and Dondelinger et al. (2012b).
- Chapter 5, on inferring parameters in ODE models of biological systems, is partly based on Dondelinger et al. (2012c).
- Appendix C is based on Dondelinger et al. (2011).
- Appendix E is based on part of Lin et al. (2010).

Acknowledgements

This thesis could not have been completed without the unfailing encouragement and support of my supervisor, Professor Dirk Husmeier. I would also like to thank the staff and students at the IANC and at BioSS for providing a productive research environment. Finally, I am grateful for the support of my friends and family, who in all likelihood will never want to hear about networks ever again.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Frank Dondelinger*)

Table of Contents

Chapter 1

Introduction and Background

1.1 Introduction

Networks are ubiquitous in many fields, ranging from engineering over social sciences to biology. In biology, we encounter networks at different levels; most prominently in the form of signalling pathways and gene regulation networks in molecular cell biology, and in the form of food webs and species interaction networks in ecology. In both subfields, trying to construct and analyse networks based on the available data is an active and important area of research. In many cases, knowing the structure and parameters of the network can lead to a better understanding of the system that the network is modelling. In molecular systems biology, this can help identify key factors in genetic diseases, which can lead to the development of better, targeted drugs or treatments (Emilsson et al., 2008). In ecology, it can give an indication of which species are essential to the survival of an ecosystem, and how much environmental strain the system can take before it collapses (Ings et al., 2009).

In systems biology, networks help in understanding gene regulation and molecular signalling at the cell level. The process of gene regulation is quite well understood: Important proteins called transcription factors will bind to the DNA at specific binding sites, and will expedite or inhibit the transcription of a nearby gene into mRNA. This gene will then be translated into a protein, with the protein going on to fulfil its functions in the cell, which might involve acting as a transcription factor to another gene. This process of a transcription factor activating or inhibiting the transcription of a gene is one aspect of gene regulation. Other aspects involve the phosphorylation of proteins, which can enable or disable their activity as a transcription factors or signal carriers, or the methylation of certain sites of the DNA, which can prevent transcription from taking place.

Molecular signalling is the most important mode of communication within and between cells. Receptor-activated signalling, perhaps the most well-known type of signalling, starts when an outside signalling molecule called a ligand binds to a receptor protein at the cell surface. This exposes a binding site on the part of the receptor that is on the inside of the cell. Special messenger proteins (also called signalling proteins) then bind to the receptor and propagate the signal to the intended destination inside the cell. Often, a signalling protein will pass the signal onto another signalling protein, a process which can be repeated to lead to a so-called signalling cascade. Feed-back loops are also possible, whereby the signal can be fed back to an earlier part of the signalling cascade. The ultimate destination of the signal could be a protein in the cell, or a even a transcription factor regulating a gene, thereby linking signalling pathways to gene regulatory networks.

In ecology, networks usually take the form of food webs, where a link between two species indicates that one species is a food source for the other. Such food webs can be arbitrarily complex, with many layers, interdependencies, and even feedback loops, belying the simplified concept of a food chain. The dynamics of food webs have been studied extensively (Cohen et al., 1994; Dunne et al., 2002; Lande et al., 2003). In Chapter 2, I have considered food webs, but I have also introduced the more general concept of a species interaction network. In a species interaction network, a link between two species indicates that the presence of one species influences the presence of the other, which could be through a predator-prey interaction, mutualism, or competition for a common food source.

Given the overwhelming importance of networks in these fields, it is crucial that we develop efficient tools for inferring the networks and analysing them. Generally, we will not be able to observe the interactions in the network directly; in systems biology, many of the processes involved in gene regulation and signalling take place at time and size scales that make observation difficult and often impossible. In ecology, the cost and effort involved in collecting direct observations of species interactions is often prohibitive. This explains the importance of network reconstruction techniques that rely only on data that can be easily observed, such as gene expression data or species presence/absence.

In systems biology, machine learning approaches have been heavily used to reconstruct networks from data, mostly based on microarray data (e.g. Werhli et al., 2006; Tenenhaus et al., 2010; Logsdon et al., 2012), but more recently data from Next-

Generation Sequencing (NGS) has also become available (Werner, 2010). The sudden availability of very high-dimensional data, as well as the need to analyse and integrate datasets of different types and provenance (Yeung et al., 2011), has made sophisticated machine learning techniques essential for learning networks in a principled way. Regression methods (Rogers and Girolami, 2005; van Someren et al., 2006) and Bayesian networks (Friedman et al., 2000; Perrin et al., 2003; Husmeier and McGuire, 2003) have both been applied with great success in this area.

Rather surprisingly, machine learning approaches have been neglected in ecology, where it is common practice to construct models by hand and then analyse them to see if they conform to the data (Memmott, 1999; Vázquez and Simberloff, 2002; Blüthgen et al., 2006). While this works well for small systems with only a few species, it becomes hard to do once the complexity of the system increases.

On the systems biology side, there currently exists a split between regression and graphical models for network inference (e.g. Rogers and Girolami, 2005; van Someren et al., 2006; Fröhlich et al., 2011), and mechanistic methods that model the dynamics of the system as coupled differential equations (e.g. Ashyraliyev et al., 2009; Pokhilko et al., 2010). One can come up with convincing arguments for either approach, and in fact I would argue that they should complement each other. The main goal of my thesis is to present new methods that solve some of the outstanding problems of current techniques in network structure and parameter inference. In doing this, I have contributed new computational tools that will help scientists to reconstruct biological networks. In the rest of this chapter, I will present a brief review of this field, before moving on to describe my contributions in subsequent chapters.

1.2 Types of Biological Networks

Networks are a useful tool for analysing relationships in biology. Their structure reveals which entities interact, and if the dynamics are known, they can provide a useful model for testing hypotheses. In this thesis, I will mainly focus on gene networks and signalling pathways in the cell and species networks in ecology, although I will briefly outline some additional examples of networks in biology below.

1.2.1 Gene Regulatory Networks

The central dogma of molecular biology states that genes in DNA are transcribed into mRNA, which is then translated into proteins that do most of the work inside the cell. However, not all genes in the DNA are turned into mRNA (or 'expressed') all the time, nor at the same rate. The process of controlling which genes are expressed and how much of each gene is turned into mRNA is called gene regulation.

Gene are mostly regulated by other genes (or rather, the proteins that are coded for by other genes). This is because some proteins function as transcription factors, binding to the DNA and facilitating (or in some cases, inhibiting) the transcription of a nearby gene. Because of this mechanism, when we speak of a gene regulatory network, we usually mean a network whose nodes consist of genes, and where the edges indicate that the expression level of one gene will regulate the expression of another gene (either up or down, depending on the type of edge).

Gene regulation can be a bit more complicate than this, because it can also depend on post-transcriptional modifications of the mRNA and post-translational modifications of the proteins. In post-transcriptional regulation, a protein controls the translation rate of another protein by binding to the mRNA strand and interfering with the translation process. Post-translational regulation only involves proteins: After a protein has been translated from mRNA, it may be in an inactive state which does not allow it to function as a transcription factor. The protein can then be modified by other proteins to activate it (for example via phosphorylation). These processes are abstracted away in many network inference methods, but they may be important.

More details on gene regulatory networks can be found in Chapters 3 and 4.

1.2.2 Ecological Networks

In ecology, there are different kinds of networks that one may consider. The most common kind is a so-called food web. In a food web, the relationship between different species in the network is expressed in terms of predator-prey behaviour: a directed edge between species 1 and species 2 means that species 1 preys on species 2.

Antagonistic predator-prey interaction is not the only kind of relationship that can exist between species. Another possible type of interaction is mutualism, where two (or more) species benefit in some way from each other's presence. The most common example is symbiosis, such as the cleaner fish that benefit other fish by feeding on their dead skin.

Another type of interaction that is worth mentioning is competition for common resources. Here, the relationship between two species depends on some shared resource (usually food or living space) which means that one species can only thrive at the expense of another.

What all these interactions have in common is that they influence the population of the species that are involved. This shows the similarity between ecological networks and gene regulation networks. Positive interactions (mutualism in ecology, transcription factor binding in gene regulation) increase the population or concentration, while negative interactions (predator-prey and competition in ecology, inhibitory genes in gene regulation) decrease it.

More details on ecological networks can be found in Chapter 2.

1.2.3 Other Types

Although gene and ecological networks are the main focus of this thesis, it is worth mentioning that these are not the only important types of networks in biology. Other types of networks exist and can be studied. These include:

Protein interaction networks: Gene regulation networks study interactions that are mediated via transcription, and can be found by looking at mRNA expression levels in the cell. But not all protein interactions lead to a change in expression levels. Signalling pathways can be seen as a type of protein interaction network, in which the main activity is the propagation of a signal from the receptors at the cell surface to proteins inside the cell, usually leading to a change of their activation state. Techniques like co-immunoprecipitation, yeast-2-hybrid and others can be used to detect protein-protein interactions directly. Some of these methods are noisy and expensive, so computational statistics and machine learning can still be helpful in constructing protein interaction networks.

Reaction networks: Technically, a reaction network is any network describing how different substrates and products are linked by chemical reactions. In systems biology, the term is most frequently used in relation to metabolic networks (or metabolic pathways) that describe the workings of the cell at the chemical level. Metabolic networks include gene regulatory and protein-protein interactions, but also show metabolites that may play a vital role in the cell.

Cell networks: Most cells communicate by releasing molecules, such as hormones, that are caught by the receptors of other cells. Outside the cells, this kind of signaling is usually undirected, affecting all cells in the immediate vicinity. A more interesting kind of cell network is the network of neurons in the brain. These communicate with other neurons via synapses, where the electric signal of one neuron is passed to its neighbours. Networks of neurons are thought to hold the answers to many of our key questions about the working of the brain and the nervous system.

1.3 Network Models and Inference

Network reconstruction methods can be broadly grouped into three categories: Methods that use relevance scores or regression to determine weights for the edges in the network, probabilistic methods that try to calculate the posterior probability of edges either directly or by sampling networks from a distribution, and mechanistic methods that model the relationship between species as a dynamical system and try to estimate the parameters of the system.

1.3.1 Relevance Networks and Regression Methods

Relevance Networks Relevance networks (Butte and Kohane, 2000) are a relatively simple graphical network model, based on calculating an association score between pairs of nodes. A link between two nodes is added to the network if their score is higher than a certain threshold, indicating that there is a significant association between the nodes. Butte and Kohane suggest using a permutation test to set the threshold (Butte and Kohane, 2003). Scores could be based on correlation coefficients, or on mutual information. More sophisticated pair-wise association scores are also possible; for example, a non-linear correlation measure has recently been proposed in the form of the maximal information coefficient (MIC) (Reshef et al., 2011), which is calculated by finding a grid partition of the data that leads to the highest mutual information.

Regardless of the score that is used, one major drawback of using the relevance network approach is that due to the pairwise approach, it cannot easily distinguish between direct interactions of nodes, and indirect interactions (via a third node).

Gaussian Graphical Models (GGMs) Gaussian Graphical Models (also called Conditional Independence Graphs) are one alternative model that can be used to deal with this problem. GGMs are undirected graphical networks, where we assume that the variables in the network have a multivariate Gaussian distribution. One can then construct a network by calculating the partial correlation coefficient between each pair of nodes in the network, and adding an edge between them if their coefficient is significantly different from zero. The use of partial correlation coefficients highlights the difference between this approach and relevance networks. Unlike correlation coefficients, they describe the correlation between two nodes *conditioned on the rest of the nodes*. This means that only direct interactions between two nodes lead to an edge between them.

The straight-forward GGM construction algorithm as described in Whittaker (1990) works by taking advantage of the fact that the matrix Π of partial correlation coefficients is related to the inverse of the matrix P of correlation coefficients. This means you can calculate Π by taking:

$$\Omega = P^{-1} = (\omega_{ij}) \tag{1.1}$$

and

$$\pi_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}} \tag{1.2}$$

There are some drawbacks with this simple method, which have been resolved in Schäfer and Strimmer (2005b), as will be explained in Chapter 2. More recently, Dobra and Lenkoski (2011) have proposed an extension of GGMs in the form of Copula Gaussian graphical models, which relaxes the multivariate Gaussian assumption by using a Gaussian copula to describe the dependence patterns between the observed variables. However, inference in this model has to be done via a Gibbs sampler, making it more computationally demanding than the method of Schäfer and Strimmer (2005b). Another method that promises to relax the Gaussian assumption is Forest Density Estimation (Liu et al., 2010), where finding the graphical model is reduced to the problem of finding a maximum weight spanning forest. However, while the authors demonstrate that this model is efficient for network inference in high dimensional data, the caveat that the resulting graph has to be a forest (i.e., having at most one path between each pair of vertices) seems quite restrictive.

Linear Regression (Unpenalised) If we assume that one can approximate a gene regulatory network by a linear system, then it makes sense to apply linear regression to determine the gene interactions.

Classic linear regression takes a data set of vectors $\mathbf{x_i}$ and response variables y_i where each $\mathbf{x_i} = (x_{i1}, ..., x_{ip})$ and the goal is to produce a weight vector $\mathbf{w}$ that we can use to predict a value for y_i. Assuming the data has been standardised, that means we want to find:

$$\tilde{y}_i = \sum_j w_j x_{ij} \tag{1.3}$$

such that

$$\sum_i (y_i - \tilde{y}_i)^2 \tag{1.4}$$

is minimised. This is called an ordinary least squares (OLS) estimate, and equation 1.4 is called the residual squared error.

Gardner et al. (2003) used unpenalised regression to infer a subnetwork of the SOS pathway in E. coli. To deal with underdetermination, they imposed a fan-in restriction on the network, so that each gene could only have a fixed number of regulators k. They could then apply the regression for each set of k regulators and choose the set that presented the best fit.

Regularised Linear Regression The problem with the OLS estimate in equation 1.4 is that it has a large variance, and it is hard to interpret which of the predictors (variables in $\mathbf{x}$) have the strongest effect. An alternative way of restricting the search space and dealing with these problems is to add a regularisation term (also sometimes called a penalisation term). The common L2 regularisation takes the form:

$$\mathbf{w} = argmin\{\sum_i (y_i - \sum_j w_j x_{ij})^2 + \lambda \sum_j w_j^2\} \tag{1.5}$$

where λ determines a bound on the norm of the weights. This is also known as ridge regression. A further advantage of regularisation is that is can help with underdetermination in the case where the number of variables p is larger than the number of data point n. It achieves this by reducing the number of parameters that have to be estimated. However, L1-regularised regression only shrinks weights, without setting them to zero. This means that no genes will be completely ruled out as regulators. For that reason, it can be preferable to use L1 penalisation or lasso. LASSO (although usually not capitalised) stands for ”Least Absolute Shrinkage and Selection Operator”, and was first described in Tibshirani (1996). The lasso promises to introduce sparsity by both shrinking the weights and setting some of them to zero, until only the significant ones are left.

Mathematically, the lasso estimate is defined as:

$$\mathbf{w} = argmin\{\sum_i (y_i - \sum_j w_j x_{ij})^2\} \text{ with the constraint that } \sum_j |w_j| \leq t \tag{1.6}$$

where t is a hyperparameter that defines how much shrinkage is applied to the weights. Smaller values of t result in more shrinkage. More details about the lasso can be found in Chapter 2. L1-penalisation using the lasso is a very active research topic, and many variants have been developed, including the group lasso for selecting grouped variables (Yuan and Lin, 2005) and the elastic net, combining L1 and L2 penalties (Zou and Hastie, 2005). Perhaps most relevant for network inference is the graphical lasso, which applies an L1 penalty to the inverse covariance matrix of a Gaussian graphical model in order to infer a sparse undirected graphical model (Meinshausen and Bühlmann, 2006; Friedman et al., 2008).

Relevance Vector Machine Another approach for producing sparser networks with fewer regulators is the relevance vector machine, also known as sparse Bayesian regression (SBR). It is based on the idea of building a sparse regression method similar to a support vector machine, but using a probabilistic model rather than the pre-defined kernel function. This is slower than the SVM approach in general, because it is trained by maximising a marginal likelihood function. However, Tipping and Faul (2003) showed that the run time of the relevance vector machine can be reduced to manageable levels.

See Chapter 2 for the details about the relevance vector machine, as well as a comparison with the lasso.

1.3.2 Probabilistic Methods

Nested Effect Models NEMs are a model for determining the relationship between important regulating genes. In a standard NEM, we try to find the network of regulating genes (called S-genes for signalling genes) by looking at the effect that knocking out each of these genes has on the expression levels of the genes that they regulate (called E-genes for effect reporting genes).

This means that we need two sets of parameters for a NEM: A network hypothesis Φ, that describes the relationship between the S-genes, and a model Θ for the regulation of the E-genes, where $\theta_i = j$ if E-gene i is regulated by S-gene j. We assume that an E-gene can only be regulated by one S-gene and use model averaging to account for all possibilities. Using Bayes' theorem, the score for a network hypothesis given data $\mathbf{D}$ is:

$$P(\Phi|\mathbf{D}) = \frac{P(\mathbf{D}|\Phi)P(\Phi)}{P(\mathbf{D})} \tag{1.7}$$

If we assume that the observations of each E-gene, the parameters θ_i and the knock-out experiments are independent, then the likelihood $P(\mathbf{D}|\Phi)$ for a dataset consisting of m E-genes and n S-genes decomposes as:

$$P(\mathbf{D}|\Phi) = \prod_{i=1}^{m} \sum_{j=1}^{n} \prod_{k=1}^{n} P(D_{ik}|\Phi, \theta_i = j)P(\theta_i = j|\Phi) \tag{1.8}$$

where $P(D_{ik}|\Phi, \theta_i = j)$ is the likelihood of the effect observed at E-gene i when knocking out S-gene k and $P(\theta_i = j|\Phi)$ is the prior probability of E-gene i being regulated by S-gene j. More details can be found in Markowetz et al. (2005) and Fröhlich et al. (2008). I applied NEMs to a case-study on *Pectobacterium atrosepticum*, which can be found in Appendix E, and has also been published in Lin et al. (2010).

Static Bayesian Networks Bayesian networks are a graphical model for representing the relationship between a set of variables. Each variable is a node in the network, and the nodes are linked up by directed links. Often, these links can be interpreted as representing causality, i.e. if A → B then A causes B[1]. By looking at the structure of the network, we can determine which variables are independent of each other (using, e.g. the Bayes' Ball algorithm, Shachter, 1998). Each node is associated with a conditional probability table, which gives the probability of the node taking a particular value, given the values of its parents. Because all Bayesian networks are defined to be acyclic directed graphs, we can decompose the joint probability distribution using the conditional probabilities:

$$P(X_1 ... X_M) = \prod_{i}^{M} P(X_m|\Pi_m) \tag{1.9}$$

where $X_1 ... X_M$ are the nodes in the networks, and Π_m denotes the parents of node X_m.

Pearl (1985) was the first to use the term Bayesian networks. They have frequently been used to successfully model regulatory networks (Friedman et al., 2000; Murphy and Mian, 1999).

[1] Although not every Bayesian network presents a causal ordering; see Pearl (2000) for more on causality.

Many training algorithms exist for constructing Bayesian networks from data and setting their parameters (Heckerman et al., 1995b). Using the data $\mathbf{D}$, we want to find the posterior distribution $P(\mathcal{M}|\mathbf{D})$, where $\mathcal{M}$ ranges over network structures. Heckerman and Geiger (1994) showed that when using a linear Gaussian model with a Normal-Wishart distribution, we can derive a a closed form solution for the posterior probability. They called this the BGe (Bayesian Gaussian likelihood equivalent) score. Once we have determined the posterior probabilities of the edges in the network, we can use them to determine how strong the interactions are between the nodes.

Bayesian networks can be learned using Markov Chain Monte Carlo (MCMC) methods which aim to sample from the posterior distribution over network models. This is done by constructing a Markov chain of network models, where the acceptance probability of each new model ensures that we end up sampling from the right distribution. The disadvantage of this method is that we may need to generate a very long Markov chain to make sure that it has converged to a stationary solution. Alternatives to MCMC include maximum likelihood or greedy search strategies; however these will not sample from the posterior distribution of networks, and hence will only give an incomplete picture of the true distribution.

Structure learning for Bayesian networks is explained in more detail in Chapters 2, 3 and 4.

Dynamic Bayesian Networks (DBNs) A dynamic Bayesian network is a standard Bayesian network that unfolds in time. Each time step contains the same number of nodes. Usually, the parent of a node is either a node from the same time-step, or a node from the time-step immediately preceding it[2]. Thus each time step creates a new layer, with connections only to the previous layer. It is easy to see that if the network at a given time-step satisfies the properties of a Bayesian network, so does the unfolded network over all time steps. Hence dynamic Bayesian networks are just a special case of standard Bayesian networks. Figure 1.1 shows an illustration.

Learning the structure of dynamic Bayesian networks is similar to learning a static Bayesian network, but we can take advantage of the special form that the dynamic networks take. As Friedman et al. (1998) point out, a dynamic Bayesian network can be decomposed into an initial network for time-step 0 (what they call, somewhat misleadingly, the prior network), and a transition network that shows the links between

[2] This assumption can be relaxed to include nodes that are more than one time-step away. However, crucially for the DAG assumption in Bayesian networks, there are no backward links.

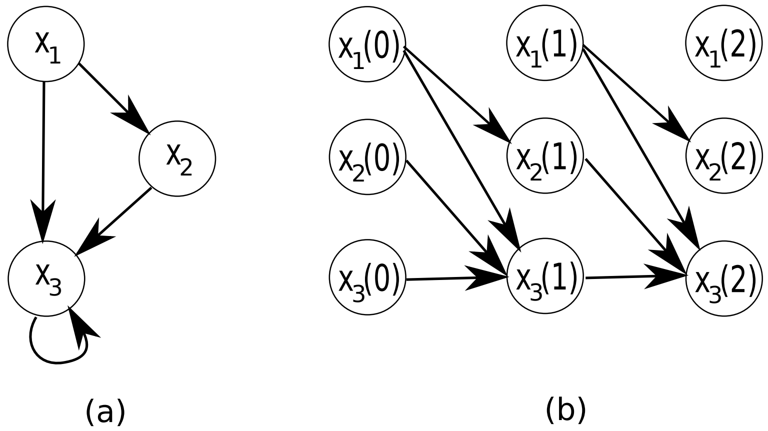


Figure 1.1: Graphical representation of a Dynamic Bayesian Network. (a) shows the static network. Note that this is not a Bayesian network due to the self-loop on x_3. (b) shows the corresponding dynamic Bayesian network for the first three time steps.

step i and step $i+1$[3]. We can then score each of these networks independently, and search for the network with the highest score. They show how to do this with complete data using the BIC score (Schwarz, 1978) and the BDe (Bayesian Dirichlet equivalent) score (Heckerman et al., 1995a), and with incomplete data using the Structural EM algorithm (Friedman et al., 1998). For some formulations of DBNs, exact sampling of network structures and parameters from the posterior via MCMC is also possible; for more in this, see Chapters 3 and 4.

Dynamic Bayesian networks have the advantage that they can be cyclical over time[4], as Figure 1.1 shows, making it possible to model feedback loops and other features that the standard Bayesian network cannot easily capture. For a detailed overview of DBN approaches for inferring gene regulatory networks, see Murphy and Mian (1999).

It turns out that DBNs are a very general class of graphical model, and as such, have received a lot of attention of the past decade. Apart from extensions to heterogeneous networks that can change in time (e.g. Robinson and Hartemink, 2009, 2010; Grzegorczyk and Husmeier, 2009, 2011; Lèbre, 2007), which I will describe in more detail in Chapters 3 and 4, there have also been extensions to continuous-time DBNs (Nodelman et al., 2002), on-line learning of DBNs (Cho et al., 2009) and even non-parametric

[3] This assumes that the network is homogeneous, i.e. it has the same structure at each time step. I will talk about the heterogeneous case in Chapters 3 and 4.

[4] This is not the same as having a cycle in the network, since two nodes representing the same entity at different time steps are considered different nodes.

approaches such as the infinite DBN (Doshi-Velez et al., 2011). State space models (a special case of DBN), have also been applied for network inference (e.g. Beal et al., 2005; Rau et al., 2010); here the relationships between observed variables and hidden states of the system are modelled over time using the state and dynamic equations. Even inference of DBNs from steady-state data has been considered (Lähdesmäki and Shmulevich, 2008). Apart from heterogeneous DBNs, what is perhaps most relevant for the inference of biological networks is the ongoing effort to model interventions and perturbations within the DBN framework, exemplified by Fröhlich et al. (2011). The dynamic nested effects models (dynoNEMs) described in Fröhlich et al. (2011) make the link between DBNs and NEMs, by describing a model that is split into signalling (S) and effect-reporting (E) genes, as in a NEM, but models observations over time like a DBN. Interventions are integral to this model due to the requirement of introducing perturbations that affect the E-genes. The obvious drawback of this approach is that it requires one perturbation per S-gene, while only producing a network that includes the S-genes. While the work presented in this thesis focuses on non-interventional data, I will sketch some ideas for including interventions in a DBN model in Chapter 6.

1.3.3 Mechanistic Models

By mechanistic models, I refer to methods that model the mechanisms and behaviour of a dynamical system using differential equations. Mechanistic should not be equated with deterministic, as these equations could be stochastic in nature. Systems of differential equations are very useful because they can describe the dynamics of a system in great detail. Once we know the dynamics, we can usually deduce the relationships between entities in the system, which is why these models are relevant for network reconstruction.

Ordinary Differential Equations A system of ordinary differential equations (ODEs) models the change in the variables, $\mathbf{x}$, via the derivatives $\dot{\mathbf{x}}$ with respect to time:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}, t|\boldsymbol{\theta}) \tag{1.10}$$

where θ contains the parameters of the system. As long as f is continuously differentiable, we can solve the system (1.10) given an initial value $\mathbf{x}(0)$ (this is called the initial value problem).

One way to do parameter estimation in ODE systems is by fitting the data to the output of the system. This requires that we solve the initial value problem. Most ODE systems cannot be solved analytically and have to be integrated numerically, using for example the popular Runge-Kutta algorithm. However, these numerical integrations are just approximations, and also slow down the model fitting process since they have to be recalculated each time the parameters change.

An alternative is given by collocation methods which approximate $\mathbf{x}$ by a basis function expansion $\hat{\mathbf{x}}$ such that:

$$\hat{x}_i(t) = \sum_k^{K_i} c_{ik}\phi_{ik}(t) = \mathbf{c}_i'\boldsymbol{\phi}_i(t) \tag{1.11}$$

where K_i is the number of basis functions in vector $\boldsymbol{\phi}$ and $\mathbf{c}_i$ is the vector of weights given to the basis functions. Varah (1982) used the collocation method for parameter estimation in a two-step procedure, by first estimating $\hat{\mathbf{x}}$ using data smoothing, and then measuring the fit of the gradient $\frac{d\hat{\mathbf{x}}}{dt}$ to $f(\mathbf{x},t|\boldsymbol{\theta})$. This means that you can avoid the numerical integration, making this a very fast procedure. Others have used different smoothers, including nonparametric estimators (Brunel, 2008) and a local polynomial smoother (Liang and Wu, 2008). However, two-step methods generally require full observation of the system, and the solution is very dependent on how well the smoothing step performs, because the smoothing does not take the ODE dynamics into account.

Ramsay et al. (2007) proposed an improved estimation technique that uses a parameter cascade in which nuisance parameters $\mathbf{c}$ are implicit functions of the structural parameters $\boldsymbol{\theta}$, $\boldsymbol{\sigma}$, and the structural parameters are a function of the smoothing parameters $\boldsymbol{\lambda}$. They then use three fitting criteria, the outer fitting criterion $F(\boldsymbol{\lambda})$, an inner fitting criterion $J(\mathbf{c}|\boldsymbol{\theta},\boldsymbol{\sigma},\boldsymbol{\lambda})$ that is optimised with respect to $\mathbf{c}$ alone, and a data fitting criterion $H(\boldsymbol{\theta},\boldsymbol{\sigma}|\boldsymbol{\lambda})$ that is optimised with respect to the structural parameters. The smoothness parameter $\boldsymbol{\lambda}$ is increased iteratively until it becomes too large. This model effectively couples the smoothing step and the ODE parameter fitting step, making sure that smoothing takes the ODE dynamics into account and vice-versa.

Ramsay et al. use spline bases as smoothers. A recent NIPS paper (Calderhead et al., 2008) uses Gaussian processes to model the data, which has the advantage that the gradients $\dot{\mathbf{x}}$ can be integrated out. However, I will show in Chapter 5 that the inference procedure as described in Calderhead et al. (2008) amounts to a return to the two-step procedure, in a situation where an adaptive approach that couples smoothing and parameter inference would be more methodologically consistent.

Another recent approach is the functional tempering of Campbell and Steele (2012).

This again uses splines for the smoothing, in an adaptive procedure that has a vector of regularization parameters which penalize the mismatch between the gradients of the smoother and of the ODE system. They then use population MCMC, where tempering is done towards the data rather than towards the prior, by using a sequence of gradually increasing regularisation parameters and thus forcing the gradients to agree. However, this approach has the potential drawback that the flexibility of the smoother is unchanged by the ODE dynamics and is only defined by the spline basis.

So far I have only presented methods for parameter inference in existing ODE models. It is not unreasonable to wonder about the case where the structure of the ODEs is unknown. Is it possible to do the equivalent of network inference for mechanistic models? Recent research has shown that model selection in ODEs is feasible (Vyshemirsky and Girolami, 2008; Toni et al., 2009), but this requires the models to be enumerated in advance, and the number of models must be reasonably small make the comparison feasible. Inferring general ODE models from data is still outside the capability of our current toolset. Some progress has been made by using simple (usually linear) parametric forms of the ODE system and learning the network structure by learning the parameters (Gardner et al., 2003; Bansal et al., 2006; Bonneau et al., 2006).

Another promising research direction is the use of ODEs as a tool for guiding more general network inference. In Li et al. (2011), the authors employ a gradient approximation as substitute for the gene expression level in a dynamic Bayesian network framework for learning gene regulatory networks. Unfortunately, it is not clear that this use of a gradient approximation actually improves network inference (see Oates et al., 2012b). More promising are the methods described in Äijö and Lähdesmäki (2009) and Oates et al. (2012a). In Äijö and Lähdesmäki (2009), a linear ODE model for gene transcription and regulation is combined with a Gaussian process for capturing the unknown, possibly non-linear regulation function. In Oates et al. (2012a), ODEs of the Michaelis-Menten type are used to model non-linear interactions in protein signalling networks. This formulation allows for network inference from steady-state data using RJMCMC.

Stochastic Differential Equations Ordinary differential equations are completely deterministic, which means that they assume that you can completely specify the behaviour of the system. Stochastic differential equations (SDEs) relax this assumption by introducing a stochastic process (or several) into the differential equation. A simple example of an SDE system is a diffusion process:

$$\dot{\mathbf{x}}(t) = F\mathbf{x}(t) + B\mathbf{z}(t) \tag{1.12}$$

where $\mathbf{z}(t)$ is a stochastic process (usually a white noise process). F encodes the drift and B is the diffusion matrix. SDEs are harder to solve numerically than ODEs, but it is still possible to do parameter inference (see e.g. Hurn and Lindsay, 1999).

Golightly and Wilkinson (2005) have used SDEs to infer the rate constants for biochemical reactions, using a Gibbs sampler to do the inference. Archambeau et al. (2007) developed a Gaussian process approximation for the posterior of an SDE. While I have not dealt explicitly with SDE models in this thesis, the ODE parameter inference method described in Chapter 5 could be adapted for use with SDE systems. I will sketch an idea for this in Chapter 6. To my knowledge there has been no exploration as of yet into the use of SDEs for network inference, although this would no doubt be a fascinating research area.

1.4 Thesis Structure

The remainder of the thesis is organised as follows:

- Chapter 2 describes an application of established network reconstruction methods from systems biology to simulated and real-world presence/absence data from ecology, which includes a comparison among methods. This chapter is based on our published work in Faisal et al. (2010).

- Chapters 3 and 4 deal with inference of gene regulatory networks in DBNs with time-varying structure using time series gene expression data. In Chapter 3 I describe two types of priors for information sharing among network segments with structure changes: a global information sharing prior and a sequential information sharing prior, and I compare the two on simulated and real-world gene expression data. This work has been published in Dondelinger et al. (2010) and in Dondelinger et al. (2012a). In Chapter 4, I revisit the sequential information sharing model from the previous chapter, and describe different functional forms and models for the priors. I then present an in-depth comparison and analysis of the properties of these priors and their hyperparameters on simulated data, as well as an application to real-world gene expression data from systems and synthetic biology. This work has been published in Husmeier et al. (2010) and Dondelinger et al. (2012b).

- Chapter 5 describes adaptive gradient matching, a reformulation of the ODE parameter inference model in Calderhead et al. (2008) that allows for adaptive feedback between smoothing and parameter fitting. Using a number of model ODE systems, I compare adaptive gradient matching to the method in Calderhead et al. (2008) and to parameter inference using the explicit solution of the ODEs, both in terms of accuracy and in terms of computational cost. I then describe the application of the new method to a realistic model of the JAK/STAT pathway.

- Finally, Chapter 6 sums up the main results of this thesis and describes promising directions for future work.

Chapter 2

Inferring species interaction networks from species abundance data

Note: This chapter describes a collaborative ecology project that I was involved in, and it is largely based on my co-authored paper "Inferring species interaction networks from species abundance data: A comparative evaluation of various statistical and machine learning methods", published in Ecological Informatics (Faisal et al., 2010). I have included major parts of this paper as verbatim copies in the present chapter. The results using the stochastic food web simulation model were obtained by myself, and the results using the European Bird Atlas data were obtained by my co-author Ali Faisal. We jointly contributed to the discussions on how to perform the real-world data analysis, including discussing the most appropriate choice of method and interpreting the findings. I have explicitly pointed out Ali Faisal's contributions in the text.

2.1 Motivation

Darwin's description of a tangled bank describes the everyday complexity of ecology that we overlook at our peril. Tampering with the population of one species can cause surprising and dramatic changes in the populations of others (Cohen et al., 1994; Henneman and Memmott, 2001). Altering pressures to which ecosystems are exposed can drive them to alternative states (Beisner et al., 2003) or catastrophic failure (Sinclair and Byrom, 2006). Understanding and predicting how ecosystems will respond to change requires untangling the tangled bank and is of enormous importance during a period of rapid global change. Yet such a task can seem impossible given the enormous complexity of ecological systems and the excruciating fieldwork needed to quantify even the simplest of foodwebs (Memmott et al., 2000; Ings et al., 2009).

Currently, most work on ecological networks has focused on quantifying food webs and pollination networks by direct observation of interactions among individuals. This approach has provided important insight into the structure and stability of some types of ecological networks, and has also had some limited success in predicting the consequences of anthropogenic changes in managed ecosystems. However, the predictive ability of these types of networks is limited by their assumption that other types of interaction, such as competition or mutuality relationships, are unimportant, when these have recently been identified as perhaps overwhelming (Werner and Peacor, 2003; Schmitz et al., 2004). Recognising this importance, some recent attempts have been made to include such non-trophic interactions within food web models (e.g. van Veen et al., 2009) but traditional field observations are unable to quantify the strength of these interactions and new methods are required to allow ecological interaction networks to expand beyond the current food web paradigm.

There has recently been a surge of interest in elucidating and modelling the structure of biological networks. A variety of summary statistics for characterising the global properties of networks have been derived, like the degree distribution (Albert and Barabási, 2002), clustering coefficient (Watts and Strogatz, 1998) and average path length (Valiente, 2002). This has been augmented by local characterisations in terms of over-represented network motifs (Milo et al., 2002), and measures of specialisation based on information theory (Blüthgen et al., 2006).

The formation and evolution of a network can then be simulated from a mathematical model, like the simple preferential attachment model of Barabási et al. (1999), or more realistic models of basic biological processes (de Silva and Stumpf, 2005). The summary statistics obtained from the ensemble of simulated networks can then be compared with those obtained from the real networks, and the discrepancy provides a measure of how accurately the mathematical model captures the true network formation processes.

A critical assumption of the approach delineated above is that the true network is known. In molecular systems biology, the structure of protein interaction networks is commonly obtained from yeast two-hybrid assays. It is well known that these experiments are noisy, that they are susceptible to large proportions of both false positives and false negatives, and that the networks extracted from different assays can differ substantially (e.g. Tong et al. 2002). In ecology, establishing the structure of a species interaction network typically requires minute observations and detailed field work. For instance, the information theoretic summary statistics proposed in Blüthgen et al. (2006) were applied to the plant-pollinator interaction networks obtained in the studies of Memmott (1999) and Vázquez and Simberloff (2002). These studies entailed detailed observations of how often a particular plant was visited by a particular pollinator, for all pollinators and plants in turn. This process is laborious and error-prone. More importantly, it is restricted to specific kinds of interactions. The interactions between pollinators and their host plants are amenable to direct observation. However, other types of species interactions, like competition for resources, are not, and might not even be clearly defined from the outset. Our work therefore aims to adapt a novel type of methodology that has recently been explored in molecular systems biology: to infer the network structure directly from the data. To reword this: rather than taking an "existing" network structure and analysing it in terms of summary statistics, we assume that the interaction network is unknown, and we aim to reconstruct it *in silico* from the species abundance counts.

Information about ecological interactions should be evident in a range of ecological data that are currently available. For example, time-series of the populations of multiple species present in a study site should allow identification of important interactions, and similarly the spatial patterns of coincidence of species should contain information about the interactions among these species, potentially at a range of scales. What is needed is a statistical tool capable of recovering networks structure from these types of data sets. Recently, the challenge of identifying regulation networks and signalling pathways from post-genomic data has resulted in the development of a number of statistical and machine learning methods for the recovery of network structure. Examples are the reconstruction of transcriptional regulatory networks from gene expression data (Friedman et al., 2000), the inference of signal transduction pathways from protein concentrations (Sachs et al., 2005), and the identification of neural information flow operating in the brains of songbirds (Smith et al., 2006). This development has potentially given ecologists a new set of tools for network recovery, if the methods can be applied to typical ecological data sets.

Our aim is to compare different models for recovering ecological interaction networks, similarly to the approach of Tirelli et al. (2009) for modelling presence/absence data of *Salmo marmoratus*. Here, we introduce and seek to test the suitability of four statistical / machine learning methods for the identification of network structure on ecological data: Graphical Gaussian models (GGMs), L1-regularised linear regression with the least absolute shrinkage and selection operator (LASSO), the relevance vector machine (or sparse Bayesian regression, SBR), and Bayesian networks. We extend these methods by including explanatory variables to model the effect of spatial autocorrelation and the impact of bio-climate variables. We first test the success of these methods for recovering the structure of simulated food webs, where the true structure is known precisely. We then use the methods to identify the large-scale interactions among 39 species of European warblers (families Phylloscopidae, Cettiidae, Acrocephalidae and Sylviidae), a subset of the European breeding bird data set (Hagemeijer and Blair, 1997) covering Europe west of 30°E and including all probable and confirmed breeding records. These data have been augmented by two bio-climate covariates, related to temperature and water availability. Our work has been motivated by preliminary explorations described in the MSc dissertations Faisal (2008) and Dondelinger (2008). However, for the present work, the methodology has been considerably expanded, new methodological concepts have been included, different ways of result and network integration have been explored, and all simulations have been rerun.

2.2 Material and Methods

2.2.1 Statistical and Machine Learning Methods for Network Reconstruction

In the most general case, our aim in describing an ecological network is to model all the interactions between and among species and their environment. It is convenient to think of this network as a 'graph' (e.g. Fig. 2.7), describing species as the 'nodes' within the graph, and interactions as the links or 'edges' that join the nodes. To identify and infer these graphs we selected four widely used methods for network recovery in postgenomic data analysis: Graphical Gaussian Models (Schäfer and Strimmer, 2005a,b), LASSO regression (Tibshirani, 1996; van Someren et al., 2006), the relevance vector machine (Tipping and Faul, 2003; Rogers and Girolami, 2005) and Bayesian Networks (Friedman et al., 2000; Werhli and Husmeier, 2007). All four methods have previously been used to recover gene regulation network structures and there is no *a priori* assumption that any method will perform best on ecological data where other statistical issues such as spatial autocorrelation (Lennon, 2000), small sample sizes, or the influence of other, unmeasured covariates may be important. Each method differs in the mechanism it uses to recover networks from data and we provide, in Section 2.3, a description of the important features of the methods we trial, along with the full details of the mathematical implementation. All methods were implemented in MATLAB © (The MathWorks, Inc.) or R (`http://www.R-project.org`) (see Table 2.2).

2.2.2 Simulation study

In order to have an objective measure of network recovery, we first tested the ability of the models to recover the true network structure from test data generated by an ecological simulation model. This model combines a niche model (Williams and Martinez, 2000) with a stochastic population model (Lande et al., 2003, chap. 8) in a 2-dimensional lattice. The niche model defines the structure of the network and has two parameters (the number of species and the connectance, or network density, defined as L/S^2 where L is the number of links and S the number of species in the network).

More precisely, to generate a food web consisting of N species, we start off by assigning to each species i a niche value n_i, drawn uniformly from [0, 1]. This gives us an ordering of the species by niche value, where higher niche values mean that species are higher up in the food chain. For each species we then draw a niche range r_i from a beta distribution with expected value 2C (where C is the desired connectance) to determine the size of the niche that that species preys upon. Then we uniformly draw a centre c_i for the niche from $[\frac{r_i}{2}, n_i]$. This allows us to construct a network of predator-prey interactions, where species i preys on species j if n_j falls within r_i of c_i. The generated networks share many characteristics with real food webs, such as the fraction of species with no prey, no predators or both prey and predators, and the amount of cannibalism and looping in the network.

The population model is defined by a stochastic differential equation where the dynamics of the log abundance X_i of species i can be expressed as:

$$\frac{dX_i}{dt} = \rho_i + \frac{\sigma_d}{\sqrt{N_i}}\frac{dA_i(t)}{dt} + \sigma_e\frac{dB_i(t)}{dt} - \gamma X_i - \Omega(\mathbf{X}) + \sigma_E\frac{dE(t)}{dt} \tag{2.1}$$

where ρ_i is the growth rate of species i, σ_d is the standard deviation of the demographic effect, N_i is the abundance of species i (e^{X_i}), $A_i(t)$ is the species-specific demographic effect (random variations in individual fitness), σ_e is the standard deviation of the species-specific environmental effect, $B_i(t)$ is the species-specific environmental effect (effect of the random variations in the environment on individual species), γ is the intra-specific density dependence, Ω is the effect of competition for common resources, σ_E is the standard deviation of the general environmental effect and $E(t)$

is the general community environmental effect (effect of the random variations in the environment on all species). Here, $A_i(t)$, $B_i(t)$ and $E(t)$ are standard Wiener processes (Brownian motion). In order to incorporate the niche model, the simulation modifies the term Ω to include predator-prey interactions in the Lotka-Volterra form.

In order to extend this model to a 2D arena, the simulation incorporates an exponential dispersal model, where the probability of a species moving from location A to location B is determined by the Euclidean distance between A and B. Locations are arranged on a rectangular grid. Each location has its own growth rates. The spatial pattern of growth rates for a single species is generated by noise with spectral density f^{β} (with $\beta < 0$, and f the frequency at which the noise is measured), and a normal error distribution.

We simulated the dynamics of this model for 3000 steps (until the system had reached equilibrium), with 10 different network structures to generate 10 independent data sets. The final 'gold-standard' network against which the recovered networks were assessed was the structure of the niche model linking among species present in the data set (as some species went extinct during the initial runs to equilibrium). We recovered networks from these data using all methods first without consideration of spatial autocorrelation, then with the inclusion of spatial autocorrelation for methods where this was possible.

The simulation described above produces continuous values for the population densities. For some of the experiments below, we needed to transform these into presence/absence data similar to the Bird Atlas data. For this, we implement an observation process as follows. We assume that there is a random variable X which functions as a threshold for observing a population; if the population density x_g is below or equal to this threshold, then that species is not observed. Let X be modelled by a Gaussian $N(\mu, \sigma^2)$. Then the probability of observing a species with population density x_g is $P(X < x_g)$, i.e. the cumulative distribution function:

$$P(X < x_g) = \frac{1}{2}\left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) \tag{2.2}$$

We can then sample a discrete value for x_g from a binomial distribution, using $P(X < x_g)$ as the parameter. Mean and variance of the Gaussian distribution are fitted so that the distribution of ones and zeros over all locations and species is the same as in the real data set.

2.2.3 Application to the European bird atlas data

Note: This subsection describes the data used by Ali Faisal, and was written by Ali Faisal and Colin Beale. The data extraction described was performed by Colin Beale.

Until relatively recently, climate was considered to be the main factor affecting large-scale (continental) distribution patterns and global climate change is already having measurable effects on the distribution of many species (Gaston, 2003). Lately, however, theoretical models have suggested that biotic interactions may also be important in shaping range limits (Holt and Barfield, 2009), and recent empirical research has suggested that the distribution of many European bird species may not be as strongly related to climate as previously thought (Beale et al., 2008). This weaker than expected association with abiotic climate variables may be explained if biotic interactions are more important than previously thought. If biotic interactions play an important role in large-scale species distributions, developing a method to identify and predict their influence must be considered a priority. If successful, therefore, application of network recovery methods to mapped data used in ecological analysis would be valuable.

To test the utility of the available methods for network recovery in this large context, we use a subset of the European breeding bird data set (Hagemeijer and Blair, 1997) covering Europe west of 30°E and including all probable and confirmed breeding records. From this data set we extracted the distributions of all 39 old world warbler species breeding in this area (families Phylloscopidae, Cettiidae, Acrocephalidae and Sylviidae). These species are all small insectivores occupying a range of habitat types from boreal forest to Mediterranean reedbeds, several of which are likely to interact at a range of spatial scales (e.g. Murray Jr, 1988). As covariates we include the mean temperature of the coldest month and the water availability for plant growth, two climate variables that had strongest influence on avian distribution (Beale et al., 2008). Climate data were available at 0.5° (data set CRU CL 1.0, New et al., 1999), and because soil types differ in their ability to retain moisture (e.g. sandy soils drain very quickly, whilst clay retains water longer) were combined with soil data (data set WISE.AWC, Batjes 1996) using a bucket model (following Prentice et al. 1992) and interpolated to 50km resolution. These or similar variables are typically used in distribution modelling exercises (e.g. Thomas et al. 2004; Thuiller et al. 2005; Araujo et al. 2005; Beale et al. 2008; Huntley et al. 2008) not because they are always expected to directly impact bird distributions, but they are perceived to have strong indirect effects on birds and other taxa through effects on food availability or habitat type (Araujo et al., 2005; Beale et al., 2008; Huntley et al., 2008). Note that although some of the warbler species we studied are migratory, and thus may be absent during the coldest month, the use of the temperature during this month is nevertheless justified, both because it correlates strongly with temperature during other months, and because of its impact on vegetation growth during the year, which will affect food source abundance. Other biologically relevant climate variables could also be used but are usually strongly correlated with one of these and have little effect on the strength of associations realised (Beale et al., 2008). As these are real valued variables, we discretise them by maximising the mutual information. For this pre-processing step, we perform a standard quantile discretisation into 20 levels and then use the information bottleneck algorithm, proposed by Hartemink (2001), to get a binary variable minimising the expected information loss.

As the simulation studies suggested that the relevance vector machine (SBR) consistently underperformed the other methods (see Section 2.6), and as the Gaussian assumption underlying graphical Gaussian models (GGMs) is violated by the binary nature of the data, we only applied L1-regularised regression (LASSO) and Bayesian networks to recover network structures from the real data sets. We used three different data sets that increased in complexity from the simple warbler dataset alone, through inclusion of spatial autocorrelation, to inclusion of the bio-climate covariates. We generated consensus networks for each data set, which should represent successively better models of true network structure. We also attempted to build a latent variable model (see Section 2.3.2.4) but the Markov chain Monte Carlo (MCMC) chains did not converge and we do not consider this further for the European Bird Atlas Data.

In the absence of complete ecological knowledge of the true network of interactions among these species, success of the modelling methods can only be assessed against known or likely relationships. To validate our methods on these real datasets we therefore determined four tests: firstly, for each pairwise interaction we sought to give an *a priori* interaction score, identifying any published studies and, when these were unavailable, using expert judgement to categorise interactions into likely, unknown or unlikely (we provide this network and relevant literature in Section A.1 in the appendix). We tested similarity between the recovered network and the *a priori* network using the area under the receiver operator characteristic (ROC) curve and the true positive rate at 5% false positives (TPFP5).

Secondly, as ecological niches are often conserved in evolutionary time (Losos, 2008) we expected there to be a relationship between phylogenetic distance and inferred interaction score (details of the phylogeny used are provided in Section A.2 in the appendix). Thirdly, we expected that ecologically similar species were most likely to interact, so for each species we identified the preferred habitat, migrant status (resident, short or long-distance migrant), wing length, body mass and body length and clutch size (all data from http://www.bto.org/birdfacts/ or Snow and Perrins (1998)) and summarised these variables to generate a measure of ecological distance (see Section A.3 in the appendix for details). Significance of both these tests with phylogenetic and ecological distance was assessed by correlation. Finally, as well as expecting these measures to be related to the final networks identified, we predicted that the simpler (and less biologically plausible) network models lacking spatial autocorrelation and bio-climate covariates would show weaker associations with the ecological datasets than the full models, and the number of significant interactions among bird species in the network would decline as complexity increases (and spurious interactions are accounted for by the additional complexity of the model).

To characterise the networks recovered using these methods and put them in the context of other ecological networks, we counted the non-zero links with each species in turn, and measured the frequency distribution of these (i.e. we measured the degree distribution: Proulx et al., 2005). We also measured the mean shortest path between all species in the network (Dunne et al., 2002) and a measure of how clustered the network is (related to the proportion of species linked to the neighbours of a focal species that are themselves linked to the focal species. We measured the clustering coefficient: Luce and Perry, 1949). As our recovered networks are not binary but identify continuous probabilities of linkage between two species, we calculated all three values across a range of threshold levels and identified network characteristics that are consistent across all thresholds.

2.2.4 Units

Table 2.1 gives an overview of the units for different quantities in our paper, along with the equations where these quantities were used.

Symbol/Quantity	Equation	Unit
$\mathbf{x}_r$, x_i, X_i	2.4, 2.18, 2.21, 2.22	Discrete presence/absence value of species over 50 km^2 area
$\hat{\mathbf{y}}_g$, $\mathbf{y}_g$	2.4, 2.5, 2.9, 2.22	Discrete presence/absence value of species over 50 km^2 area
$\hat{\mathbf{w}}_g$, $\mathbf{w}_g$, $\mathbf{w}$, v	2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 2.14, 2.15, 2.16, 2.17, 2.21, 2.22	Dimensionless weight parameters
a	2.21, 2.22	Spatial Autocorrelation: Discrete presence/absence value of species from averaging over four 50 km^2 areas
Temperature Covariate	None	Discrete warm/cold value over 50 km^2 area
Water Covariate	None	Discrete presence/absence value over 50 km^2 area

Table 2.1: Units for the different quantities in the paper, along with the equations where they are used (if any).

2.3 Theory

2.3.1 Statistical and Machine Learning Methods for Network Reconstruction

2.3.1.1 Graphical Gaussian models (GGMs)

Graphical Gaussian models (GGMs) are undirected probabilistic graphical models that allow the identification of conditional independence relations among the nodes under the assumption of a multivariate Gaussian distribution of the data. The inference of GGMs is based on a (stable) estimation of the covariance matrix of this distribution. The element C_{ik} of the covariance matrix $\mathbf{C}$ is proportional to the correlation coefficient between nodes X_i and X_k. A high correlation coefficient between two nodes may indicate a direct interaction, an indirect interaction, or a joint regulation by a common (possibly unknown) factor.

However, only the direct interactions are of interest to the construction of a species interaction network. The strengths of these direct interactions are measured by the partial correlation coefficient ρ_{ik}, which describes the correlation between nodes X_i and X_k conditional on all the other nodes in the network. From the theory of normal distributions it is known that the matrix of partial correlation coefficients ρ_{ik} is related to the inverse of the covariance matrix $\mathbf{C}$, $\mathbf{C}^{-1}$ (with elements C_{ik}^{-1}) (Edwards, 2000):

$$\rho_{ik} = -\frac{C_{ik}^{-1}}{\sqrt{C_{ii}^{-1} C_{kk}^{-1}}} \tag{2.3}$$

To infer a GGM, one typically employs the following procedure. From the given data, the empirical covariance matrix is computed, inverted, and the partial correlations ρ_{ik} are computed from (2.3). The distribution of $|\rho_{ik}|$ is inspected, and edges (i,k) corresponding to significantly small values of $|\rho_{ik}|$ are removed from the graph. The critical step in the application of this procedure is the stable estimation of the covariance matrix and its inverse. Note that the covariance matrix is only non-singular if the number of observations exceeds the number of nodes in the network. This condition might not always be satisfied in a survey study. In order to learn a GGM from a data set in such a scenario, Schäfer and Strimmer (2005b) explored various stabilisation methods, based on the Moore-Penrose pseudo inverse and bagging.

In the present work, we apply an alternative regularisation approach based on shrinkage, which Schäfer and Strimmer (2005b) found to be superior to their earlier

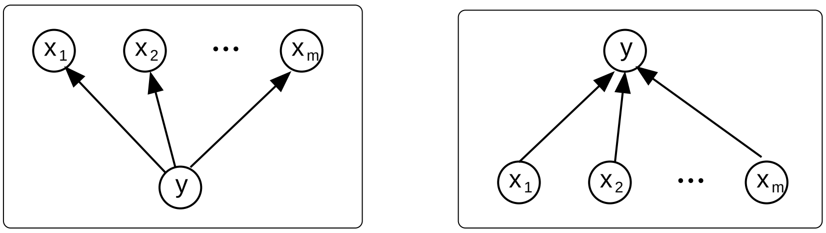


Figure 2.1: Schematic of the approach of partial correlation (left) and sparse regression (right). Left: Conditional on y, the species abundance profiles $x_1, x_2, \ldots, x_m$ are independent, and the partial correlation coefficients will be small. Right: The approach of sparse regression aims to find a minimal set of predictors $x_1, x_2, \ldots, x_m$ to explain species abundance profile y.

schemes. The idea is to add a weighted non-singular regularisation matrix, e.g. the unity matrix, to the covariance matrix so as to guarantee its non-singularity. The optimal weight parameter is estimated based on the Ledoit Wolf lemma from statistical decision theory so as to minimise the expected deviation of the regularised covariance matrix from the (unknown) true covariance matrix. The method of GGMs, which are undirected graphs, can be extended to infer putative directions of causal interactions, as proposed in Opgen-Rhein and Strimmer (2007). This scheme is based on the computation of the standardised partial variance, which is the proportion of the variance that remains if the influence of all other variables is taken into account. All significant edges in the GGM network are directed in such a fashion that the direction of the arrow points from the node with the larger standardised partial variance (the more *exogenous node*) to the node with the smaller standardised partial variance (the more *endogenous* node), provided the ratio of the two partial variances is significantly different from 1. For further details, see Opgen-Rhein and Strimmer (2007).

2.3.1.2 Linear Regression and the LASSO

The approach discussed in the previous subsection aims to predict interactions between species based on the partial correlations between their abundance profiles. In the present subsection, we review an alternative paradigm, which pursues a regression approach: given the species abundance profile $\mathbf{y}_g$ of some target species g, we aim to find a set of regulators $\{r\}$ (i.e. other species or exogenous variables related e.g. to the habitat, climate etc.), whose abundance profiles $\{\mathbf{x}_r\}$ are good predictors of abundance profile $\mathbf{y}_g$:

$$\hat{\mathbf{y}}_g = \sum_r w_{gr}\mathbf{x}_r \tag{2.4}$$

where $\hat{\mathbf{y}}_g$ is a predictor of $\mathbf{y}_g$, and the regression parameters w_{gr} represent interaction strengths between the target species g and the putative regulators r.

The different concepts are illustrated in Figure 2.1. We denote the vector of interaction strengths as $\mathbf{w}_g$, which has w_{gr} as its rth component. The mismatch between the predicted and measured expression profile of target species g is typically measured by the L2 norm:

$$E(\mathbf{w}_g) \; = \; ||\mathbf{y}_g - \hat{\mathbf{y}}_g(\mathbf{w}_g)||^2 \tag{2.5}$$

Obtaining the optimal interaction parameters $\hat{\mathbf{w}}_g$ by minimising $E(\mathbf{w}_g)$ corresponds to a maximum likelihood estimator under the assumption of isotropic Gaussian noise. In practice, this approach is usually susceptible to over-fitting, which calls for the application of some regularisation scheme. The standard method of ridge regression is given by:

$$\hat{\mathbf{w}}_g \; = \; \arg\min_{\mathbf{w}_g} \left(E(\mathbf{w}_g) + \lambda \sum_r w_{gr}^2 \right) \tag{2.6}$$

This can be interpreted in three different ways:

1. Maximising the penalised likelihood with an L2-norm penalty term and regularisation parameter λ.

2. Constrained maximisation of the likelihood under the L2-norm constraint $\sum_r w_{gr}^2 < C$, where λ is a Lagrange parameter.

3. Bayesian *maximum a posteriori* estimate under a zero-mean Gaussian prior on $\mathbf{w}_g$ with diagonal isotropic covariance matrix $\lambda^{-1}\mathbf{I}$:
   $P(\mathbf{w}_g) = \mathcal{N}(0, \lambda^{-1}\mathbf{I})$.

A disadvantage of ridge regression is that the set of interaction parameters $\{w_{gr}\}$ does usually not tend to be sparse. This is a consequence of the fact that the derivative of the regularisation term with respect to w_{gr} approaches zero as $w_{gr} \to 0$. Consequently, there is no “force” pulling the parameters to zero when they are small. According to our current knowledge, species interaction networks are usually sparse, and a stronger regularisation term is therefore desirable. This can be achieved with an L1-norm instead of the L2-norm regularisation term:

$$\hat{\mathbf{w}}_g \; = \; \arg\min_{\mathbf{w}_g} \left(E(\mathbf{w}_g) + \lambda \sum_r |w_{gr}| \right) \tag{2.7}$$

which can be interpreted as a Bayesian *maximum a posteriori* estimate under a Laplacian prior on $\mathbf{w}_g$, as first proposed by Williams (1995). The derivative of the regularisation term with respect to the parameters is now constant, which provides a stronger "force" driving small parameters to zero. The discontinuity of the derivative at $w_{gr} \to 0$ can be exploited to implement an effective pruning scheme for discarding interactions, as discussed in Williams (1995). The L1-norm regularisation term was introduced to the statistics community by Tibshirani (1996), where it was termed the LASSO (least absolute shrinkage and selection operator). One of the first applications to the somewhat related problem of reconstructing gene regulatory networks is reported in van Someren et al. (2006). Grandvalet and Canu (1999) showed that the LASSO estimate of the interaction strengths is equivalent to ridge regression with r-dependent regularisation hyperparameters:

$$\hat{\mathbf{w}}_g = \arg\min_{\mathbf{w}_g} \left(E(\mathbf{w}_g) + \sum_r \lambda_r w_{gr}^2 \right) \tag{2.8}$$

subject to the constraint $\sum_{r=1}^{R} 1/\lambda_r = R/\lambda$, for some predefined constant λ.

The regulatory network between the target species g and the regulators $\{r\}$ is defined by the set of interactions with nonzero interaction strengths w_{gr}. The degree of sparsity is determined by the regularisation hyperparameter λ, with larger values of λ resulting in sparser networks. The question, then, is how to set λ. Williams (1995) suggested integrating λ out; this approach has been subject to some controversy, though (MacKay, 1996). A standard non-Bayesian approach is to estimate λ with k-fold cross-validation. This is the approach that was implemented in the software we applied in the present study, with $k = 10$. An alternative Bayesian approach would be to estimate λ by maximising the evidence, as discussed in the next subsection.

Note that the generalisation of the sparse regression approach to more target species g is straightforward: $E(\mathbf{w}_g)$ in equation (2.5) just needs to be replaced by:

$$E(\mathbf{W}) = \sum_g ||\mathbf{y}_g - \hat{\mathbf{y}}_g(\mathbf{w}_g)||^2 \tag{2.9}$$

where $\mathbf{W}$ is a matrix with column vectors $\mathbf{w}_g$. If there is no clear separation between the set of target and regulatory species, the effect of species g needs to be excluded when forming the predictor $\hat{\mathbf{y}}_g(\mathbf{w}_g)$. Again, this requirement is straightforward to implement. To avoid notational opacity, we have not described this approach in its full generality, though.

2.3.1.3 Relevance Vector Machine

As mentioned in the previous subsection, the minimisation of $E(\mathbf{w}_g)$ in equation (2.5) corresponds to maximising the likelihood $P(\mathbf{D}|\mathbf{w}_g)$ under the assumption of isotropic Gaussian noise, where $\mathbf{D} = \{\mathbf{y}_g, \{\mathbf{x}_r\}\}$ is used to denote the data. The estimates $\hat{\mathbf{w}}_g$ in equations (2.6) and (2.8) are equivalent to the *maximum a posteriori* estimates:

$$\hat{\mathbf{w}}_g \;=\; \arg\max_{\mathbf{w}_g} P(\mathbf{w}_g|\mathbf{D},\lambda) \;=\; \arg\max_{\mathbf{w}_g} \Big[\log P(\mathbf{D}|\mathbf{w}_g) + \log P(\mathbf{w}_g|\lambda)\Big] \tag{2.10}$$

under the assumption of an isotropic Gaussian or Laplacian prior $P(\mathbf{w}_g|\lambda)$ on the interaction strengths $\mathbf{w}_g$. If we now want to do this within the Bayesian framework, the hyperparameter λ is optimised by maximising the marginal likelihood or evidence:

$$P(\mathbf{D}|\lambda) \;=\; \int P(\mathbf{D}|\mathbf{w},\lambda)P(\mathbf{w}|\lambda)d\lambda \tag{2.11}$$

as discussed in MacKay (1992). In the present study, we applied the relevance vector machine, also known as “sparse Bayesian regression” (SBR)[1], of Rogers and Girolami (2005), which is based on the work of Tipping and Faul (2003). Here, the prior on the interaction parameters is chosen to be a product of zero-mean Gaussian distributions:

$$P(\mathbf{w}_g|\boldsymbol{\lambda}) \;=\; \prod_r \mathcal{N}(w_{gr}|0,\lambda_r^{-1}) \tag{2.12}$$

with separate hyperparameters for species r. This scheme is similar to equation (2.8), except that the constraint: $\sum_{r=1}^{R} 1/\lambda_r = R/\lambda$ is missing. We can think of this as ARD (Automatic Relevance determination) in the sense used by MacKay (1992)

The hyperparameters λ_r are optimised with the evidence scheme described above[2]. Tipping and Faul (2003) showed that the marginal likelihood can be decomposed into separate contributions from the individual regulatory species $\{r\}$. This leads to a fast, iterative maximisation algorithm not only for the hyperparameters λ_r, but also for the network structure: interactions between the target species g and the putative regulatory species $\{r\}$ are progressively added and removed until a local maximum of the marginal likelihood is reached. Specific details of the algorithm can be found in Tipping and Faul (2003).

[1] This is not to be confused with other Bayesian approaches to sparse regression, most notably the spike-and-slab prior (see, e.g. Titsias and Lázaro-Gredilla (2011)), which uses a mixture of a uniform distribution and a delta spike at zero as a prior for the regression weights.

[2] In statistics this is called a type-II maximum likelihood estimation.

The reason for the sparsity of the relevance vector machine may not be immediately apparent. In fact, as Tipping (2001) points out, it comes from the hierarchical nature of the prior on weights in equation 2.12. Each hyperparameter λ_r has a prior from the Gamma family of distributions. In the algorithm, we assume an uninformative Gamma prior with the shape and inverse scale parameters set to zero, which leads to an improper prior for the weights if we integrate the hyperparameter out:

$$P(\mathbf{w}_g) = \int P(\mathbf{w}_g|\boldsymbol{\lambda})P(\boldsymbol{\lambda})d\boldsymbol{\lambda} \tag{2.13}$$

Which for an individual weight gives:

$$P(w_{gr}) \propto \frac{1}{w_{gr}} \tag{2.14}$$

This is clearly a sparse prior. In fact, we can make an analogy to LASSO here. If one takes a Bayesian view of the LASSO, as in Park and Casella (2008), then each weight in the LASSO estimate has an independent Laplace prior, so that:

$$P(w_{gr}) \propto exp(-|w_{gr}|) \tag{2.15}$$

Both the LASSO and the relevance vector machine prior are sparse. However, they differ in the amount of regularisation that they apply, as can be seen by taking the derivative of the negative log likelihood for both priors:

$$\text{RVM: } \frac{d}{dw_{gr}}\left\{-logP(w_{gr})\right\} \propto \frac{1}{w_{gr}} \tag{2.16}$$

$$\text{LASSO: } \frac{d}{dw_{gr}}\left\{-logP(w_{gr})\right\} \propto const \tag{2.17}$$

The regularisation term for LASSO is constant, while the regularisation term for the relevance vector machine tends to infinity as the weight tends to zero. Therefore, the relevance vector machine applies much stronger regularisation than the LASSO penalty. Note that it is the use of an improper prior for the weights which leads to this situation, and it could be resolved by using a proper prior. However, for the purpose of this work we were interested in comparing the existing relevance vector machine, which uses the improper prior, with an existing implementation of the LASSO.

2.3.1.4 Bayesian networks (BNs)

Bayesian networks (BNs) have received substantial attention from the computational biology community as models of regulatory and interaction networks (Friedman et al., 2000; Losos, 2008; Needham et al., 2007). Formally, a BN is defined by a graphical structure $\mathcal{M}$, a family of (conditional) probability distributions F, and their parameters $\mathbf{q}$, which together specify a joint distribution over a set of random variables of interest. The structure $\mathcal{M}$ of a BN consists of a set of nodes and a set of directed edges. The nodes represent random variables, e.g. species and their abundance values, while the edges indicate conditional dependence relations. The structure $\mathcal{M}$ of a BN is a directed acyclic graph (DAG), which defines a unique rule for expanding the joint probability in terms of simpler conditional probabilities. Let $X_1, X_2, ..., X_n$ be a set of random variables represented by the nodes $i \in \{1, ..., n\}$ in the graph, define $pa[i]$ to be the set of nodes with a directed edge feeding into node i (the "parents"), and let $X_{pa[i]}$ represent the set of random variables associated with $pa[i]$. Then

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i | X_{pa[i]}) \tag{2.18}$$

The objective of learning is to find network structures with high posterior probabilities, i.e. to sample network structures $\mathcal{M}$ from the posterior distribution

$$P(\mathcal{M}|\mathbf{D}) \propto P(\mathbf{D}|\mathcal{M})P(\mathcal{M}) \tag{2.19}$$

where $\mathbf{D}$ denotes the training data. This requires a marginalisation over the parameters $\boldsymbol{\theta}$:

$$P(\mathbf{D}|\mathcal{M}) = \int P(\mathbf{D}|\boldsymbol{\theta}, \mathcal{M})P(\boldsymbol{\theta}|\mathcal{M})d\boldsymbol{\theta} \tag{2.20}$$

If certain regulatory conditions, discussed in Heckerman (1999), are satisfied and the data are complete, then the integral in (2.20) is analytically tractable. Two function families F that satisfy these conditions are the multinomial distribution with a Dirichlet prior (Heckerman et al., 1995b) and the linear Gaussian distribution with a normal-Wishart prior (Geiger and Heckerman, 1994). The resulting scores $P(\mathbf{D}|\mathcal{M})$ are usually referred to as the BDe (discretised data, multinomial distribution) or the BGe (continuous data, linear Gaussian distribution) score. Direct sampling from the posterior distribution (2.19) is analytically intractable and is therefore approximated with Markov Chain Monte Carlo (MCMC) (Madigan and York, 1995; Friedman and Koller, 2003; Grzegorczyk et al., 2008a). To restrict the size of the configuration space, we restrict the fan-in to a node, i.e. we keep the number of incoming edges from other nodes below a pre-specified threshold (3 in our study). This approach, which is commonly adopted in other studies, e.g. Friedman and Koller (2003), incorporates our prior knowledge that interaction networks are usually sparse.

The ultimate objective is to infer causal relations among the interacting nodes. While such a causal network forms a valid Bayesian network, the inverse relation does not always hold. One reason for this discrepancy is the existence of unobserved nodes. Even under the assumption of complete observation, the inference of causal interaction networks can be impeded by symmetries within so-called equivalence classes, which consist of networks that define the same conditional independence relations. Each Bayesian network corresponds to a whole equivalence class, represented by a complete partially directed acyclic graph (CPDAG); see Chickering (1995). Under the assumption of complete observation, directed edges in a CPDAG can be taken as indications of putative causal interactions (Friedman et al., 2000).

Several tutorials on Bayesian networks have been published; see for instance Heckerman (1999), Husmeier et al. (2005) and Grzegorczyk et al. (2008b) for further details.

2.3.2 Extension

2.3.2.1 Spatial autocorrelation

Spatial autocorrelation, the phenomenon that observations at nearby locations are more similar than observations at more distant locations, is nearly ubiquitous in ecology and can have a strong impact on statistical inference (Legendre, 1993; Lennon, 2000; Dale and Fortin, 2002). In our case, spatial autocorrelation could lead to the identification of spurious interactions as a mere consequence of two species co-occurring in similar geographical regions. Where possible, we applied an autoregressive approach similar to that of Augustin et al. (1996) to incorporate potential spatial autocorrelation into the models. To this end, we computed the average population at neighbouring cells, weighted inversely proportional to the distance of the neighbours, which we will call the autocorrelation variable:

$$a = \frac{\sum_{i=1}^{N} \omega_i x_i}{\sum_{i=1}^{N} \omega_i} \tag{2.21}$$

where N is the number of neighbours that we are considering (usually $N = 4$), x_i is the population density at neighbour i, and ω_i is the weight given to that neighbour, which is inversely proportional to the Euclidean distance of the neighbour. A slight subtlety when working with real world data that is not distributed in a regular grid is to work out which neighbouring locations to consider. In this work, we have opted for the closest neighbours by Euclidean distance. The extension to the discrete case is straightforward; we simply discretise the autocorrelation variable using a threshold.

The regression then becomes:

$$\hat{\mathbf{y}}_g = \sum_r w_{gr}\mathbf{x}_r + va \tag{2.22}$$

where w_{gr} denotes the weights associated with each species r, and v is the additional weight assigned to the autocorrelation variable. The weight v will catch the effects of the spatial autocorrelation, leaving the other weights to determine the effects of other species on species g.

For Bayesian networks, we connect each node to a parent node whose value is given by (2.21), i.e., a representation of the spatial neighbourhood. The incoming edge from the parent node is enforced and excluded from the fan-in count. In this way the observation status at a node is, in the first instance, predicted by the spatial neighbourhood. Only if the explanatory power of the latter is not sufficient will there be an incentive for the inference scheme to include further edges related to species interactions.

Introducing spatial autocorrelations into GGMs is less straightforward. Since we did not apply GGMs to the real data (owing to their binary nature), we did not further pursue this issue in our work.

2.3.2.2 Bio-climate Covariates

We include the bio-climate covariates (discretised temperature and water availability) as extra variables, in the same way as we included the spatial autocorrelation variable. In particular, in the Bayesian networks, we introduce fixed connections between the bio-climate covariates and the other nodes. We modify the fan-in limit so that it does not take these extra variables into account (i.e. if the fan-in limit is three, then that means that a species can have up to six parent nodes: three other species, the covariates, and the spatial autocorrelation node).

2.3.2.3 Consensus networks

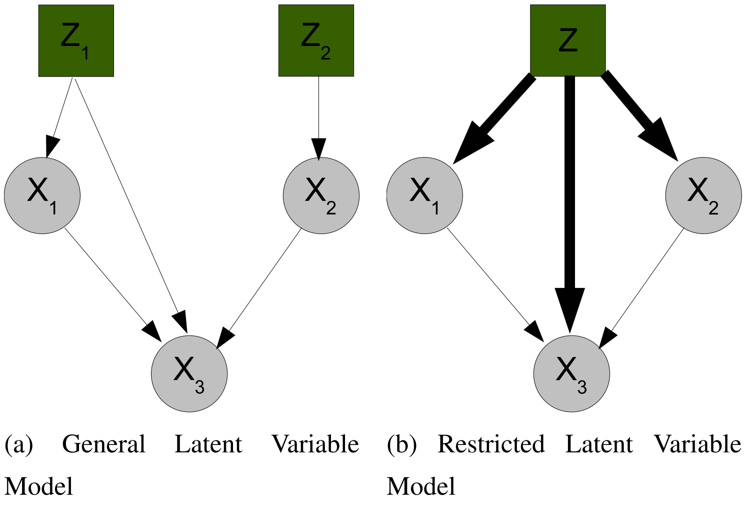
As each of the network reconstruction methods has advantages and disadvantages, it may be useful to combine outputs of different methods into one single recovered network. Such a network would capture the consensus between the various methods, whilst simultaneously allowing the strengths of the different methods to be combined (e.g. interaction size and sign inferred with regression-based methods could inform the marginal posterior probabilities obtained for Bayesian networks). Simulation experiments showed that the expected accuracy of the consensus network is higher than the expected average accuracy of the individual networks (Section 2.6.2). In the present project, we generated consensus networks by normalising the estimated interaction probabilities and absolute strengths (where available) from each method to the range [0, 1], then taking the arithmetic mean across all methods included within the consensus graph. (For a comparison with other combination methods, e.g. based on the harmonic mean, see Section 2.6.2) This potentially confuses statistical significance (probabilities) with biological significance (strengths). However, for methods where both significance measures were available we found a very strong correlation between the two ($\rho = 0.92$), as discussed in more detail in Section 2.4.

2.3.2.4 Latent variable model allowing for unobserved factors

We want to extend the Bayesian network approach to allow for unobserved factors in the environment, e.g. related to climate change or the availability of natural resources. This can be achieved by including additional so-called latent variables in the model. Ideally, the interactions between the latent variables and the species would be treated as flexible (Fig. 2.2 a). To reduce the computational complexity, we keep them fixed, i.e. they were enforced to be connected to all species. It is easy to prove that for discrete values, this is equivalent to a model with a single latent variable and a flexible number of discretisation levels (Fig. 2.2 b); this is the mixture model described in Grzegorczyk et al. (2008a).

Inference can then be carried out with the allocation sampler described in Nobile and Fearnside (2007) and Grzegorczyk et al. (2008a). The idea of the allocation sampler is to sample from a Gaussian mixture with K mixture components, and an allocation vector V that determines which mixture component each datapoint belongs to. The mixture weights are integrated out; see Nobile and Fearnside (2007) for more details.

We can think of V as a binary latent variable. Sampling is based on the following iterative procedure: given the network structure, a new value for the latent variable is sampled using the MCMC moves described in Nobile and Fearnside (2007) (imputation step). Then, given the complete data (real data, and imputed value for the latent variable), the network structure is modified with a standard structure MCMC



(a) General Latent Variable Model (b) Restricted Latent Variable Model

Figure 2.2: (a) Unrestricted latent variable model, here with two latent variables and three observed ones. (b) Alternative model with a single completely connected latent variable; this is effectively a mixture model. Zs are latent variables, Xs are observed variables. Thin edges are learnt, thick edges are fixed.

step (Madigan and York, 1995). This procedure is iterated, and leads to a Markov chain which (on convergence) samples both the network structure and the allocation of the latent variable from the posterior distribution.

While the application of this scheme to the simulated data led to encouraging results (Section 2.6.3), the MCMC simulations did not properly converge for the warbler data. The reason is that a straightforward adaptation of the method proposed in Grzegorczyk et al. (2008a) introduces a separate latent variable for each spatial location, leading to a model that is significantly more complex than explored in the original application. Our future work therefore aims to simplify the model complexity and explore alternative inference schemes based on variational learning.

2.3.3 Performance evaluation

Each network reconstruction method infers a matrix of interaction strengths among all species (nodes) in the network (graph). The nature of interaction strengths varies among the methods (GGMs: partial correlation coefficients, LASSO and SBR: regularised regression coefficients, Bayesian networks: marginal posterior probabilities). However, all three scores define a ranking of the edges. If the true interaction network is known, this ranking defines a receiver operator characteristics (ROC) curve, where the relative number of real interactions (i.e. the true positive or TP rate) is plotted against the relative number of spurious interactions (the false positive or FP rate) for all possible thresholds on the rank. To assess the network reconstruction accuracy, we follow the procedure outlined in Werhli et al. (2006) and apply two complementary performance measures. The first measure is the area under the receiver operator characteristics curve (AUC), which is a widely used global measure of reconstruction accuracy. The expectation value for a random predictor is AUC=0.5, a perfect predictor gives AUC=1.0, and larger values indicate a better reconstruction accuracy overall. As we are particularly interested in the performance of the network recovery methods when setting the threshold to a value that generates few false positives, we also identified the threshold that leads to an FP rate of 5% and counted the proportion of true interactions that were recovered at this threshold. We call this second measure the TP rate at 5% FP rate (the TPFP5 score). A good network reconstruction method is characterised by both a high AUC score and a high TPFP5 score.

2.3.4 Implementations

Table 2.2 shows which software we used for the different network reconstruction methods described in Section 2.3.1, as well as where to get the MATLAB code for our own implementations of the extensions in Section 2.3.2.

2.4 Investigation into LASSO Weights versus Confidence Values for Edges

2.4.1 Motivation

When using LASSO linear regression to reconstruct an interaction network, we have two options. One is to use the weights found during the regression and interpret them as edge strengths between the target variable and the other variables in the network (we will refer to this as "the weight method"). The other is to obtain confidence values for the presence of an edge ("the confidence value method"). Obtaining the weights is straightforward, and only requires one regression per variable. However, it is potentially biased towards edges that have a strong effect, and may ignore edges with a small (but consistent) effect.

To obtain confidence values, we use a method that is essentially an approximation

Method	**Software**	**Package**	**Description**
GGM	R	GeneNet	The software implementing Graphical Gaussian models is described in Schäfer and Strimmer (2005b) and can be found at: `http://strimmerlab.org/software/genenet/`
LASSO (Linear)	MATLAB	Genelab	For LASSO regression with continuous data, we used software from the Genlab package referenced in van Someren et al. (2006).
LASSO (Logistic)	C	BBR	For LASSO regression with discrete data, we used the BBR package (for Bayesian Binary Regression), which implements logistic LASSO regression. The package can be found at: `http://www.stat.rutgers.edu/~madigan/BBR/`
SBR	MATLAB	RegNets	We used the relevance vector machine software referenced in Rogers and Girolami (2005) and available here: `http://www.dcs.gla.ac.uk/~srogers/reg_nets.htm`
Structure MCMC	MATLAB	None	The implementations for Structure MCMC and Structure MCMC with latent variables were developed from code by Marco Grzegorczyk and can be found at: `http://www.bioss.ac.uk/students/frankd.html`
Population Simulation	MATLAB	None	The simulation code was developed by Jonathan Yearsley and slightly modified for this project. It can be found at: `http://www.bioss.ac.uk/students/frankd.html`

Table 2.2: Network reconstruction software used

of a full Bayesian approach to regression. Rather than obtaining the probability that an edge is zero from a posterior distribution of the weights, we follow Friedman et al. (2000) and approximate this value by 'sampling' data from the original dataset[3] using bootstrapping and subsampling. In bootstrap sampling, we sample data points with replacement until the sample size is the same as the size of the original dataset. In subsampling, we sample without replacement until we have obtained a dataset that is half the size of the original dataset.

For each dataset sampled in this way, we run a LASSO regression. Then we record the non-zero weights. After we have done this for a large number of samples, we average over the results. This gives the confidence value for the occurrence of each edge, independent of the strength of that edge. The drawback is that it requires many more runs of the regression algorithm than just calculating the weights once.

We wanted to find out if the difference between using confidence values and using the weights is substantial enough to warrant the extra computational cost. For that reason, we used two synthetic datasets: A simple network model without cycles (in other words, a DAG) from which we generated data using a linear regression model, and a more complex ecological simulation based on Lotka-Volterra interactions between species in a food web (see Section 2.2.2).

2.4.2 Simple Network Model

To simulate data from the simple network model based on linear regression, we first sample a network from the niche model described in Section 2.2.2. If the model is not a DAG, we remove edges until acyclicity has been restored. For each remaining edge, we draw an interaction strength from the Gaussian distribution $N(0,1)$.

Then we identify species without any parents in the network and draw their population numbers from the Gaussian distribution $N(0,1)$[4]. For each of the remaining species, we do a standard regression:

$$\hat{y_g} = \sum_r w_{gr} x_r + \varepsilon \tag{2.23}$$

where r ranges over all species x_r that are parents of species y_g, and w_{gr} is the weight of the edge linking x_r and y_g, and $\varepsilon \sim N(0,0.1)$. The factor ε adds a small

[3] This should not be confused with sampling from a posterior distribution.

[4] This allows for negative population numbers, but this is not a problem since LASSO regression does not assume that population numbers have to be positive.

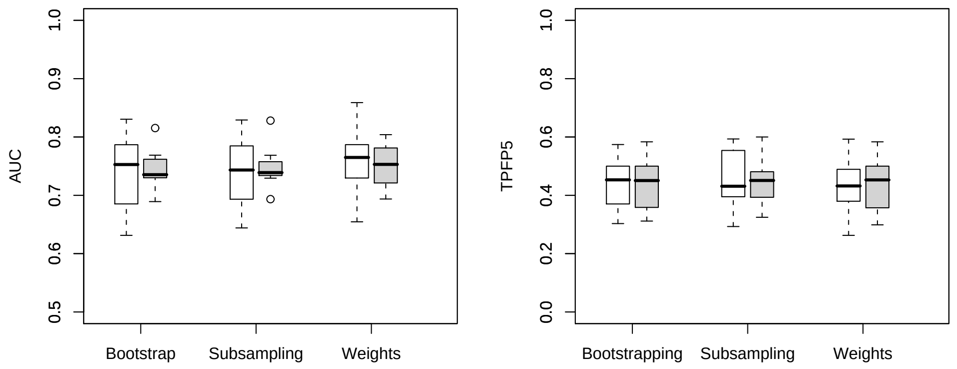


Figure 2.3: AUC and TPFP5 performance measures for the LASSO reconstruction of the simple network model. Shaded boxes show the result when thresholding is applied.

amount of observational noise. We repeat this process, drawing new population numbers each time to generate different data points.

2.4.3 Results

Simple Network Model We generated data from 10 random networks using the simple linear regression model, and for each network we generated 100 bootstrap/subset replica. Figure 2.3 shows the results. We started off by computing the confidence values straightforwardly: For each sampled dataset, every weight that was not set to 0 by the LASSO regression was counted as detecting an edge. The results of this basic approach are shown in the unshaded boxes in Figure 2.3.

Using a two-sided paired t-test, we determined that while the difference in TPFP5 values was not significant, the difference in AUC values between the two sampling methods and the weight method was significant ($p < 0.01$). It is surprising to see the weight method outperform the confidence value methods, as we would expect confidence values to produce equally good if not better results.

The reason for this discrepancy becomes apparent once we change the procedure for estimating confidence values slightly. Instead of treating all non-zero weight values in each sampled dataset as evidence of an edge, we only keep those above a certain threshold (arbitrarily set at 0.1). To be fair in our comparison, we also apply the threshold to the weight method. When we do this, we notice that the AUC values of the confidence value methods and the weight method are no longer significantly different ($p > 0.3$).

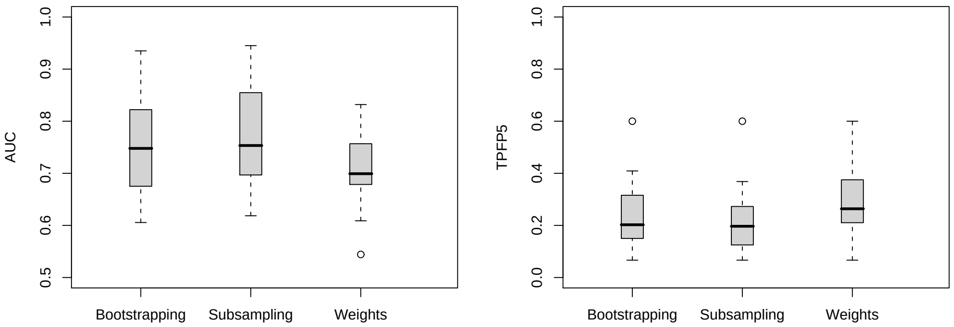


Figure 2.4: AUC and TPFP5 performance measures for the LASSO reconstruction of the ecological network simulation model.

The problem is that the selection process which sets some weights to zero is not a very conservative process. This means that some weights may never or rarely get set to zero, despite having a very low value. A threshold artificially removes those weights, and thus reduces the variance in the performance. This evens out the difference between the weight method and the confidence value methods.

Ecological Network Simulation Model We also want to compare the different methods using the simulation model described in Section 2.2.2. We use the same datasets that were used in the rest of this study.

Since we have already established that thresholding is needed to remove the variance due to small but persistent weights in the confidence value methods, we also use this method here. Figure 2.4 shows the results on the ecological simulation data. A two-sided paired t-test shows that all differences in AUC values are significant ($p < 0.01$), but none of the differences in TPFP5 values are ($p > 0.08$).

Interestingly, the significant difference in AUC now shows an increased performance for the confidence value methods. However, one must remember that the model does not include any spatial autocorrelation (cf. Section 2.3.2.1), which is by necessity, as sampling destroys the spatial structure. But this also means that sampling reduces the spatial autocorrelation, because we only sample a subset of the total number of nodes, so some of the neighbours of a selected location are left out. This explains why we see a slight increase in performance in AUC. It is reasonable that it would not be mirrored in the TPFP5 score, because this score relies on edges with high edge weights, which will be found in any case.

2.5 Number of Neighbouring Locations in the Spatial Autocorrelation Model

As described in Section 2.3.2.1, our model for spatial autocorrelation calculates the average population at neighbouring locations. One open question is how many neighbouring locations to consider. If we assume that locations are distributed on a grid, then two natural choices are to either consider 4 direct neighbours, or all 8 surrounding neighbours.

We have compared the effect of calculating spatial autocorrelation using 4 direct neighbours versus using 8 neighbours for the LASSO network reconstruction method on simulated data. There was no significant difference between the two approaches ($p > 0.2$ for AUC and TPFP5 scores). Figure 2.5 shows scatterplots comparing the edge weights, AUC and TPFP5 scores for simulated data. We have also investigated the effect of using 8 neighbours for the Warbler dataset (Figure 2.6), and found that the edge weights inferred with 4 neighbours correlate very well with the edge weights inferred for 8 neighbours. These findings lead us to conclude that 4 direct neighbours are sufficient to accurately model the spatial autocorrelation.

2.6 Network Inference Results on Simulated Data

2.6.1 Method Comparison

All four network recovery methods succeeded in recovering some of the true network structure, even when spatial autocorrelation was not incorporated (Fig. 2.7), though the methods varied in their performance (Fig. 2.8). The relevance vector machine recovered networks that were significantly worse than those recovered by the other methods, having significantly lower AUC and TPFP5 scores ($t_9 > 3$, $p < 0.01$) except for the comparison with BN using the AUC score ($t_9 = 2.19$, $p = 0.06$). All t statistics and p-values have been calculated using a two-sided paired t-test, and the significance level was set at $p = 0.05$. Tables 2.3-2.5 give a full overview of all p-values. Analysis of the inferred interaction strengths indicates that poor performance of SBR is a consequence of recovering networks that have too few links (i.e. are too sparse). This is the result of SBR being over-regularised (see Section 2.3.1.3 for a discussion of this phenomenon), which could be remedied by using a proper prior. Note that the effect would probably be less pronounced for larger networks, as these can benefit from a higher degree of

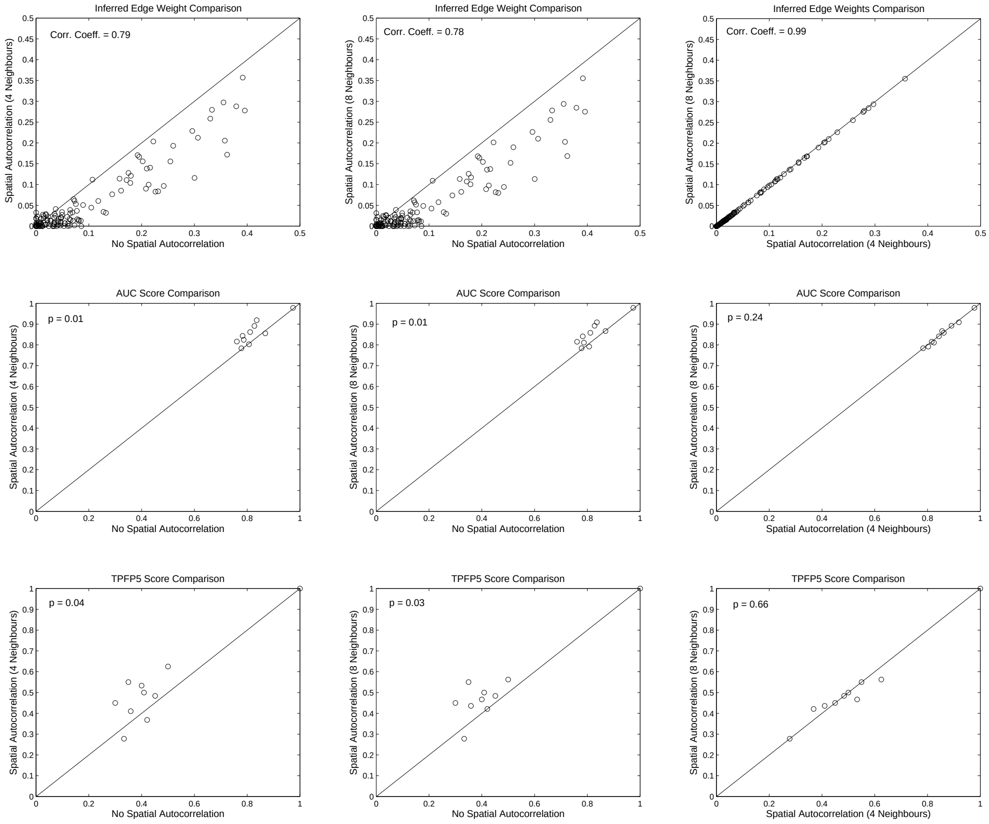


Figure 2.5: Comparison between no spatial autocorrelation modelled (0 neighbours), spatial autocorrelation with 4 neighbours and spatial autocorrelation with 8 neighbours, for networks reconstructed from simulated data with LASSO. From left to right, we compare 0 neighbours with 4 neighbours, 0 neighbours with 8 neighbours and 4 neighbours with 8 neighbours. The top row compares the inferred edge weights. The middle row compares the AUC network reconstruction scores for each of the 10 simulated networks. The bottom row compares the TPFP5 reconstruction scores for each of the 10 simulated networks. For edge weights, we show the Spearman rank correlation coefficient, while for network reconstruction scores, we show the p value of a two-sided paired t-test.

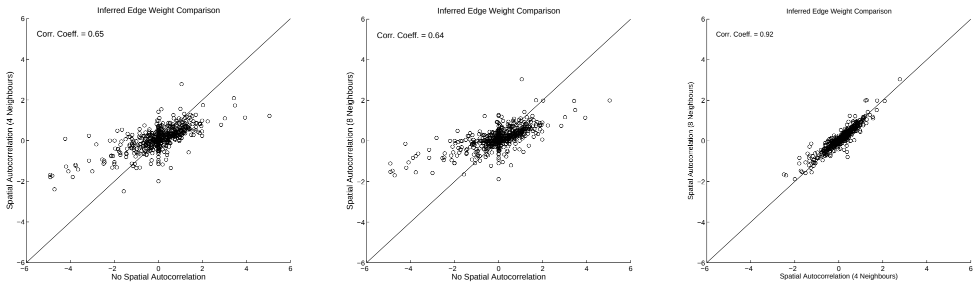


Figure 2.6: Comparison between no spatial autocorrelation modelled (0 neighbours), spatial autocorrelation with 4 neighbours and spatial autocorrelation with 8 neighbours, for networks reconstructed from the Warbler dataset with LASSO. From left to right, we compare 0 neighbours with 4 neighbours, 0 neighbours with 8 neighbours and 4 neighbours with 8 neighbours. In each plot, the inferred edge weights are compared.

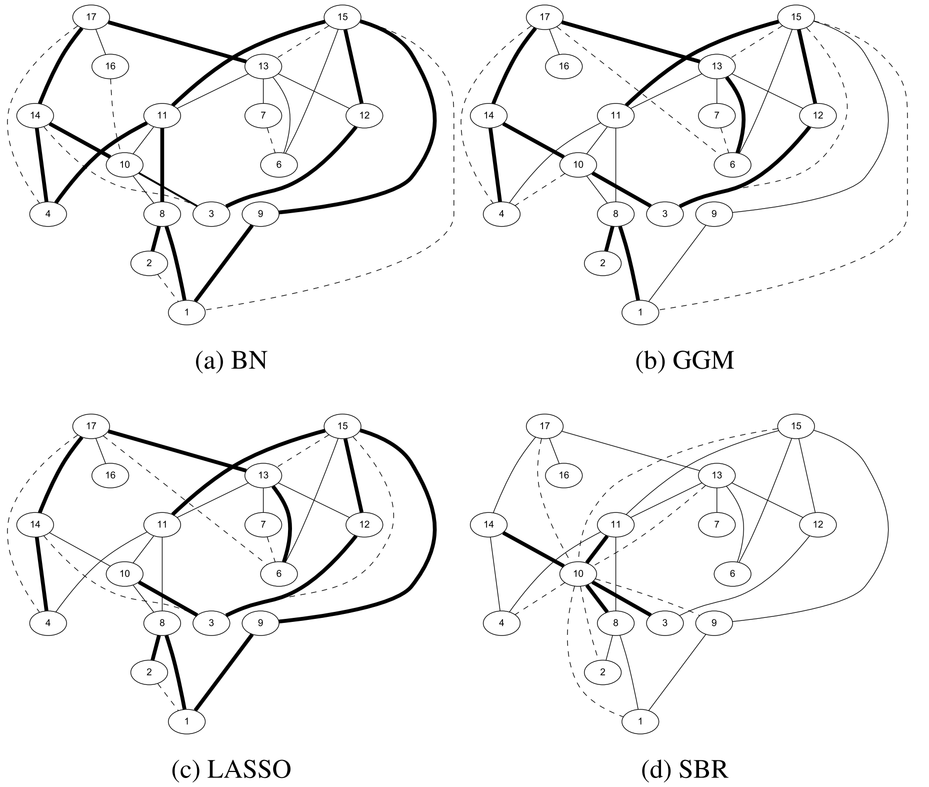
sparsity, as opposed to the small-to-medium sized networks in this study.

For the three network recovery methods where this was applied (LASSO, BN, SBR), incorporating spatial autocorrelation resulted in improved performance, especially for those methods that performed less well in the simple model (Fig. 2.8). In particular, although incorporating spatial autocorrelation improved the performance of SBR, it was still significantly worse than the other two methods ($t_9 > 3$, $p < 0.01$) except in the case of BN with the AUC score again ($t_9 = 0.68$, $p = 0.52$).

Adding an observation process to discretise the simulation datasets (described in Section 2.2.2) gave qualitatively similar results (beyond an expected drop in AUC and TPFP5 scores). The results of applying our network reconstruction methods on the discrete data can be seen in Figure 2.9.

As expected, the performance decreased when compared to the continuous data, due to the information loss inherent in the discretisation process. The AUC scores dropped around 0.1 for all methods, and the TPFP5 scores showed a similar drop, except in the case of SBR, which stayed about the same. This is because discretisation mostly hinders the identification of the more subtle interactions, which SBR had not even detected in the continuous case. Apart from SBR, there is no significant difference in the scores between methods for discrete data.

To finish our investigation, we looked at the effect of including spatial autocorrelation for the discretised data. The results are shown in Figure 2.9 (shaded boxes). Unfortunately, none of the scores improved significantly when including spatial au-



(a) BN

(b) GGM

(c) LASSO

(d) SBR

Figure 2.7: An example of a network recovered by GGM, BN, LASSO and SBR. Thick edges represent edges that were identified correctly (true positives), thin edges represent edges that were not found (false negatives) and dashed edges are spurious edges (false positives). The threshold was chosen so that the false positive rate was constant at 5%, resulting in 7 false positive edges.

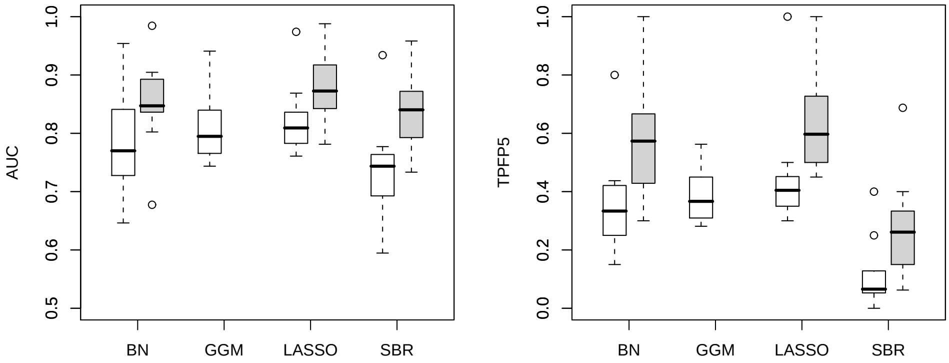


Figure 2.8: AUC and TPFP5 performance measures for continuous simulation data. Shaded boxes represent models which include spatial autocorrelation. The expected random performance scores are AUC=0.5 and TPFP5=0.05. See Section 2.3.1 for an explanation of the abbreviations BN, GGM, LASSO, and SBR.

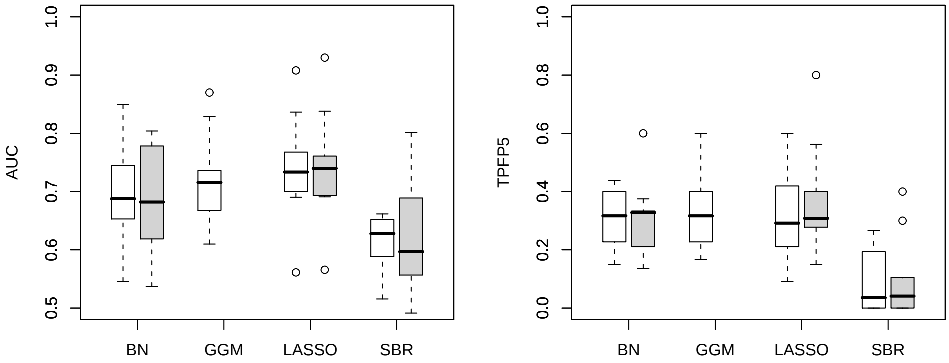


Figure 2.9: AUC and TPFP5 performance measures for discretised simulation data. Shaded boxes show the result when spatial autocorrelation is included in the model.

	BN	GGM	LASSO	SBR
BN	1	0.06	**0.02**	0.06
GGM		1	0.28	**0.00**
LASSO			1	**0.00**
SBR				1

(a) Continuous, No Spat. Autocorr. Model

	BN	GGM	LASSO	SBR
BN	1	0.09	0.06	**0.00**
GGM		1	0.08	**0.00**
LASSO			1	**0.00**
SBR				1

(b) Discrete, No Spat. Autocorr. Model

	BN	LASSO	SBR
BN	1	0.21	0.52
LASSO		1	**0.00**
SBR			1

(c) Continuous, With Spat. Autocorr. Model

	BN	LASSO	SBR
BN	1	0.08	0.16
LASSO		1	**0.01**
SBR			1

(d) Discrete, With Spat. Autocorr. Model

Table 2.3: Significance values obtained using a two-sided paired t-test when comparing different methods based on the AUC scores of the reconstructed networks. Significant results (with threshold $p = 0.05$) are marked in bold.

tocorrelation in the discrete case. This is likely due to the information loss in the observation process, which makes it harder to estimate spatial autocorrelation effects reliably. Our future work aims to reduce the information loss by applying more complex spatial-temporal models, e.g. along the lines of the Markov random field model proposed in Wei and Li (2007).

2.6.2 Consensus Networks

It is useful to combine outputs of different network reconstruction methods into one single recovered network. We call this a consensus network, because it captures the consensus between the various methods, whilst simultaneously allowing the strengths of the different methods to be combined. There are several different ways in which we can combine these methods:

- Arithmetic Mean: Edge strengths produced by regression methods are scaled to the range $[0, 1]$ (posterior probabilities obtained by Bayesian nets are left unchanged), then we take the arithmetic mean of the scaled strengths and proba-

	BN	GGM	LASSO	SBR
BN	1	0.55	**0.02**	**0.00**
GGM		1	0.38	**0.00**
LASSO			1	**0.00**
SBR				1

(a) Continuous, No Spat. Autocorr. Model

	BN	GGM	LASSO	SBR
BN	1	0.22	0.58	**0.00**
GGM		1	0.71	**0.00**
LASSO			1	**0.01**
SBR				1

(b) Discrete, No Spat. Autocorr. Model

	BN	LASSO	SBR
BN	1	0.17	**0.00**
LASSO		1	**0.00**
SBR			1

(c) Continuous, With Spat. Autocorr. Model

	BN	LASSO	SBR
BN	1	0.06	**0.01**
LASSO		1	**0.01**
SBR			1

(d) Discrete, With Spat. Autocorr. Model

Table 2.4: Significance values obtained using a two-sided paired t-test when comparing different methods based on the TPFP5 scores of the reconstructed networks. Significant results (with threshold $p = 0.05$) are marked in bold.

	AUC	TPFP5
BN	**0.01**	**0.00**
LASSO	**0.00**	**0.00**
SBR	**0.00**	**0.04**

(a) Continuous Data

	AUC	TPFP5
BN	0.58	0.80
LASSO	0.51	0.07
SBR	0.65	0.84

(b) Discrete Data

Table 2.5: Significance values obtained using a two-sided paired t-test when comparing network reconstruction methods with spatial autocorrelation model to those without. Significant results (with threshold $p = 0.05$) are marked in bold.

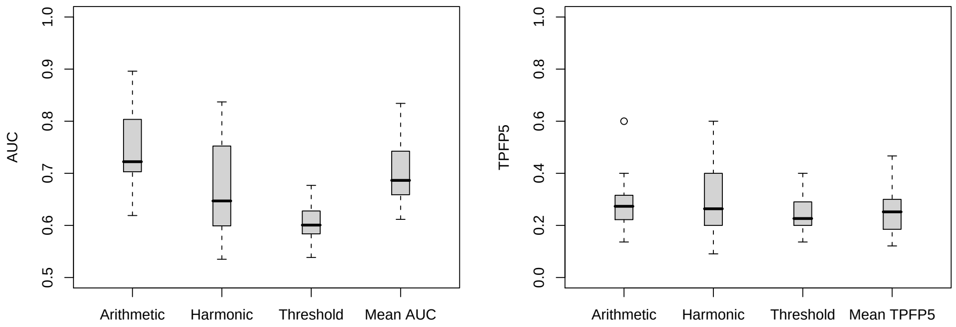


Figure 2.10: AUC and TPFP5 performance measures for different types of consensus networks. This figure only shows the results for discrete data with a spatial autocorrelation model. Results for other datasets were similar.

bilities obtained by all methods and use this as indication of the confidence we have in each edge.

- Harmonic Mean: This is the same as the previous method, but instead of using the arithmetic mean, we calculate the harmonic mean, which is generally more appropriate for rates.

- Thresholded: In this method, we use the posterior probabilities obtained by Bayesian nets as a threshold. All edges with probability less than 0.1 are removed. Then the remaining edges are evaluated based on the interaction strengths found in regression.

Note that some of these methods potentially confuse confidence values (probabilities) with interaction strengths, but for methods where both were available we found a very strong Spearman rank correlation between the two ($\rho = 0.92$), so this is not problematic. As a base line, we used the mean of the AUC or TPFP5 scores obtained from the different network reconstruction methods in isolation. A consensus method works if it produces a better score than the mean score of the individual methods.

Figure 2.10 shows the results using the discretised dataset with spatial autocorrelation modelled. This most closely mirrors the experiments on the bird data; however, results using continuous data and data without modelling the spatial autocorrelation were similar. As can be seen, the only method performing better than our baseline is the arithmetic mean. For AUC the difference is significant (using a two-sided paired t-test, $p = 0.03$) while the harmonic mean does not perform significantly different (though only barely, $p = 0.05$) and the thresholded approach performs significantly worse ($p = 10^{-3}$). For the TPFP5 score, none of the three consensus methods performs significantly different from the baseline of taking the mean of the scores, although the arithmetic mean comes closest ($p = 0.06$ versus $p = 0.25$ and $p = 0.58$ for harmonic mean and thresholded approach, respectively).

These results show that the arithmetic mean performs best when it comes to combining different network reconstruction methods. On the basis of this investigation, we have used the arithmetic mean to construct consensus networks for the bird atlas data.

2.6.3 Allowing for Unobserved Effects

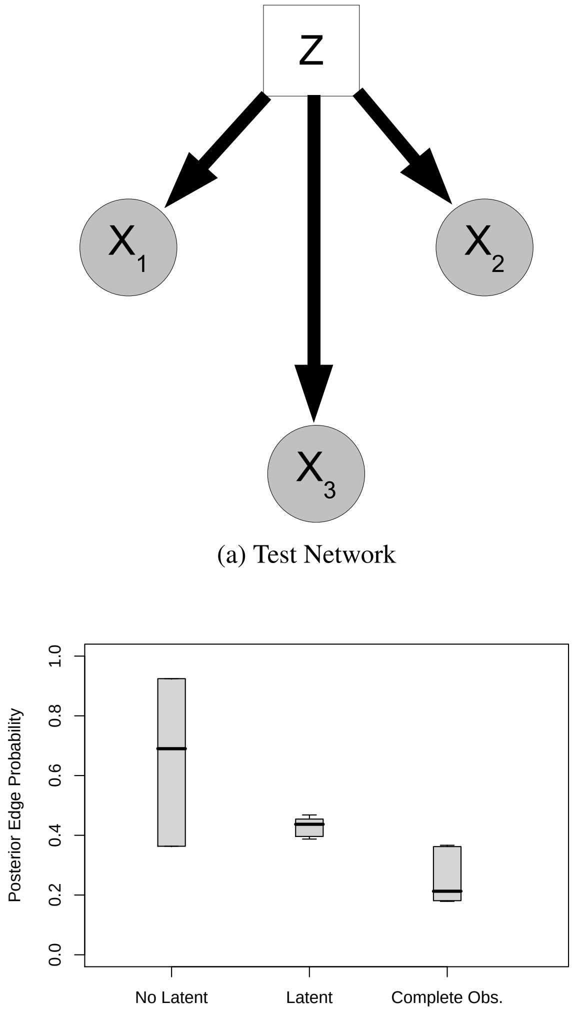
As explained in Section 2.3.2.4, we may want to take account of unobserved effects that act on the different species. While there are no explicit environmental factors (other than noise) in the simulation model, it is easy to model an unobserved effect by adding a species that acts directly on all other species, and removing the presence/absence data for that species when reconstructing the network. To assess the helpfulness of this approach, we tested it on a small network consisting of three observed nodes and one unobserved node, with no interactions between the observed nodes (Fig. 2.11a). Under these circumstances, the latent variable model should produce fewer spurious interactions than a model without latent variables. In the Bayesian network model, this means that the posterior probability of edges between observed nodes should be lower when using the latent variable model.

Figure 2.11b shows the performance of the Latent Variable Model, compared with the baseline of using simple Structure MCMC with a missing species and the optimal scenario of having complete data. As can be seen, the Latent Variable Model succeeds in reducing the median probability of spurious edges, although not quite to the level of having complete knowledge of the data.

2.7 Real-World Network Inference Results

Note: The results presented in this sections were obtained by Ali Faisal, as described in the preamble to this chapter.

We recovered three consensus networks for the warbler data: for data sets with birds only, with birds and spatial autocorrelation and with birds, spatial autocorrelation and bio-climate covariates. The first two can be found in the appendix (Figs.



(a) Test Network

(b) Results

Figure 2.11: (a) The network used to test the performance of the latent variable model, consisting of one fully connected species Z, and three unconnected species X_1, X_2, X_3. (b) Boxplot showing the posterior probabilities of spurious edges found using Structure MCMC with one fully-connected missing species, Structure MCMC with a latent variable, and Structure MCMC with a complete dataset (no missing species).

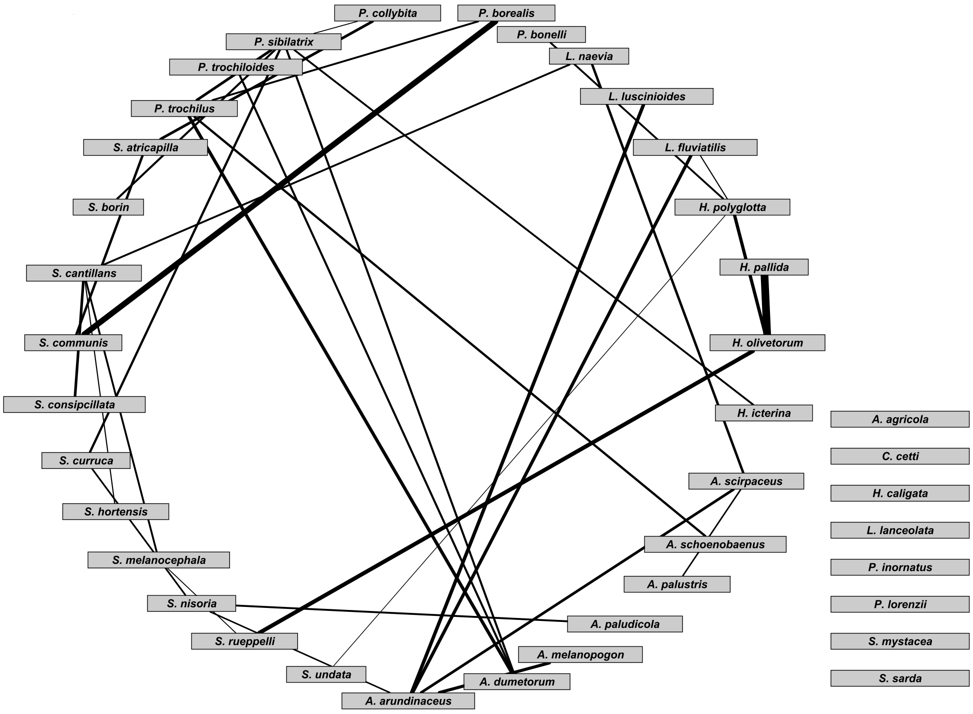


Figure 2.12: An example consensus network for the warbler data, with spatial autocorrelation and bio-climate covariates. The edges are pruned by placing a threshold value of 0.5 on the original consensus network, which corresponds to a p-value of 0.01. See Section A.5 in the appendix for a description of how these p-values were calculated. The thickness of an edge represents the strength of the interaction. The boxes on the right represent unconnected species. Equivalent plots of consensus networks for the other datasets are also available (Figs. A.3 and A.4 in the appendix).

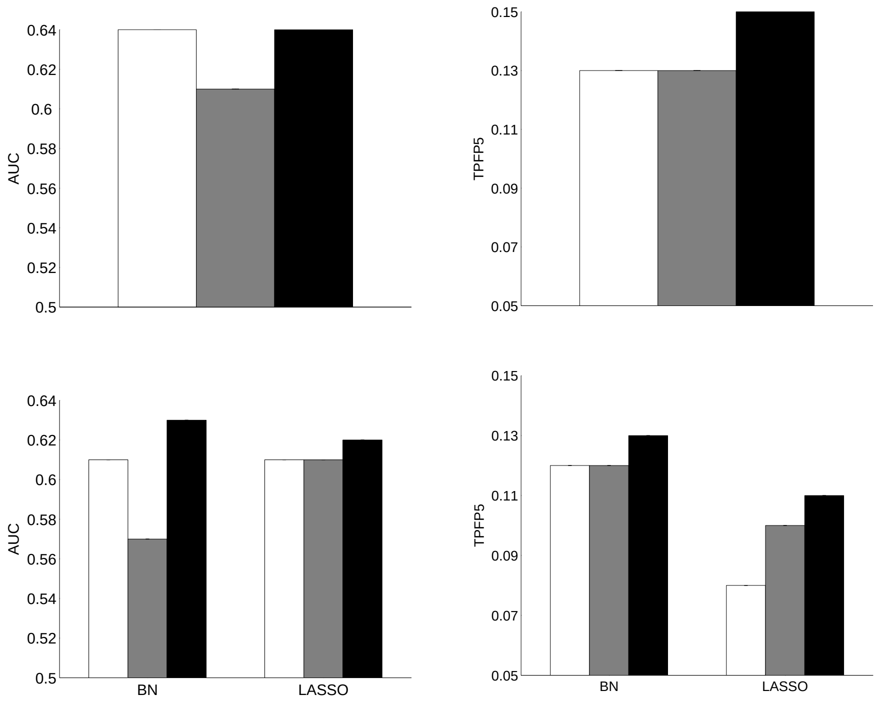


Figure 2.13: Comparison of recovered consensus networks with the *a priori* interaction network: AUC scores on the left and TPFP5 scores on the right. White bars show the birds only dataset, grey bars the birds and spatial autocorrelation, black bars the birds, spatial autocorrelation and bio-climate covariate dataset. The top row shows the results for consensus networks, while the bottom row shows the results for BN and LASSO individually. Note that the AUC and TPFP5 scores tend to increase as the model complexity increases. The vertical position of the horizontal axis indicates the expected performance of a random predictor.

A.3 and A.4); here we just present the third (Fig. 2.12). Comparison of the recovered consensus networks with the *a priori* network predicted from the literature and expert judgement revealed small but statistically significant relationships (Fig. 2.13). We also identified small but significant relationships between the interaction score for the recovered consensus networks and both the phylogenetic and ecological distances (Table 2.6). Increasing model complexity (i.e. sequentially adding autocorrelation and bio-climate covariates) generally led to both stronger correlations with the predicted network structure and sparser networks (Fig. 2.14). Our predictions in Section 2.2.3 were therefore corroborated.

Network characterisation identified that the degree distribution of the consensus

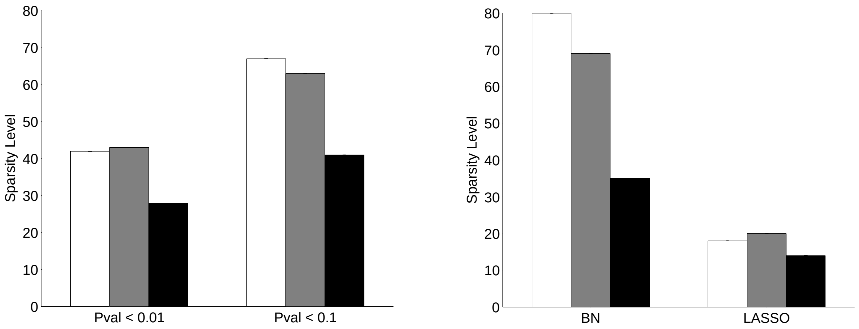


Figure 2.14: Sparsity of the recovered networks. White bars show the birds only dataset, grey bars the birds and spatial autocorrelation, black bars the birds, spatial autocorrelation and bio-climate covariate dataset. The left figure shows the results for consensus networks at two different thresholds, while the right figure shows the results for BN and LASSO individually at a threshold with p-value < 0.01.

Recovered Network	*A priori* net	Phylogenetic Dist.	Ecological Dist.
Basic Dataset	-0.98	-0.11	-0.13
	(0.32, -2.28)	(-0.18, -0.04)	(-0.20, -0.06)
Spatial Autocorrelation	-1.40	-0.12	-0.15
	(-0.03, -3.16)	(-0.19, -0.05)	(-0.22, -0.08)
Spatial Autocorrelation	-1.60	-0.14	-0.14
and Bio-climate Covariates	(-0.03, -3.16)	(-0.21, -0.07)	(-0.22, -0.07)

Table 2.6: Results of comparison between recovered consensus networks with the *a priori* interaction network, phylogenetic distance and ecological distance. For comparisons with the *a priori* network (second column), we show the regression coefficient of a logistic regression, other results (third and fourth column) are Pearson's correlation coefficients, all with 95% confidence intervals shown in brackets. Confidence intervals that do not include zero indicate that the correlation is significant.

networks was consistent across all threshold values, with all networks showing an exponential distribution. Both the clustering coefficient and the mean shortest path length varied greatly as the threshold level changed and are therefore not considered a useful description of these networks. Further details on the network characterisation can be found in Section A.7 in the appendix.

2.8 Discussion

As expected, we found that warblers in Europe form a well connected network, with most well known interactions (e.g. several *Acrocephalus* warblers: *A. arundianceus/ A. melanopogon /A. schoenobaenus/A. scirpaceus* (Schäfer et al., 2006; Rolando and Palestrini, 1991), and a triangle of interacting *Sylvia* warblers: *S. borin/S. atricapilla/ S. communis* (Elle, 2003; Garcia, 1983)) accurately described by the better consensus network structures.

Given the general expectation that climate alone shapes distributions at large scales, it might seem surprising that the chosen bioclimate variables were not more strongly connected to species distributions. We believe there are two primary reasons for the relatively low effect of climate variables: firstly, our discretised climate data is likely to be too crude to capture all the meaningful climate variation, reducing the association with these parameters. Secondly, there is growing evidence to suggest that the importance of climate and abiotic variables has previously been overstated (e.g. Watts and Worner 2008) largely because processes like the biotic interactions included in our models have previously been neglected (Davis et al., 1998; Beale et al., 2008; Holt and Barfield, 2009; La Sorte et al., 2009). It would clearly be valuable to develop the methods further to include both continuous variables and binary variables in the same analyses. Defining appropriate probability distributions is rather straightforward. However, these distributions depend on parameters, and integrating them out in the likelihood is analytically intractable. To address this difficulty, one can either seek approximate solutions based on variational calculus, or resort to an extended sampling scheme with MCMC. A development of these ideas and a comparative evaluation study provides an interesting and challenging project for future work.

To quantify the network reconstruction accuracy, we have applied various evaluation criteria (described in Section 2.2.3). We found that the correlations between the interaction scores obtained from the network reconstruction methods and those used for evaluation – phylogenetic distances and ecological similarities – were significant

(Table 2.6). Likewise, the reconstruction assessment scores obtained on the basis of an overall *a priori* network structure elicited from expert judgement – AUC and TPFP5 (Fig. 2.13) – were significantly better than random. We note that the correlations are weak (Table 2.6) and the AUC and TPFP5 scores (Fig. 2.13) are significantly below the score of a perfect reconstruction (AUC = TPFP5 = 1.0). This is over-pessimistic in that the scores are based on evaluation criteria which themselves are noisy and distorted characterisations of the unknown true species interaction network: Tables A.3 and A.4 in the appendix demonstrate that the correlation coefficients and network reconstruction scores for these criteria are also weak. This is a general problem when trying to assess the network reconstruction on real data, for which the true interaction network is unknown. The fact that the reconstructed networks show weak yet consistently significant agreement with the various evaluation criteria indicates that the machine learning methods investigated in our study have reconstructed genuine patterns of the (unknown) species interaction network.

To compensate for the lack of gold standard for the warbler data, we have extended our study by applying the network reconstruction methods to simulated data, for which the underlying network is known. Our results are consistent with related studies in molecular systems biology (Werhli et al., 2006). The global network reconstruction in terms of AUC scores typically lies in the range between 0.75 and 0.9, which is considerably better than random (0.5), but not perfect (1.0). In terms of TPFP5 scores, we can expect to reconstruct about 60% of the true species interactions at a false prediction rate of 5%. Aiming for a perfect reconstruction would be an unrealistic target, given the noise in the data, the limited data set size, and the fact that all reconstruction models investigated in our study are simplifications of the complex ecological processes.

Our comparative evaluation of different network reconstruction methods has found that SBR performed significantly worse than the other methods (Fig. 2.8) and discovered a much smaller proportion of edges than the other methods (illustrated e.g. in Fig. 2.7). We provide a mathematical explanation in Section 2.3.1.3. We have also shown that including spatial autocorrelation effects leads to a clear and significant improvement in the network reconstruction accuracy on simulated data (Fig. 2.8). The evaluation on the warbler data was more difficult due to the lack of a gold standard. In general, more complex models, which included spatial autocorrelations and bio-climate covariates, resulted in stronger matches between the predicted species interactions and the prior network derived from expert judgement (Fig. 2.13). We also found that the absolute value of the correlations between predicted species interaction strengths and both phylogenetic and ecological distance scores increased as a consequence of including spatial autocorrelations and bio-climate covariates (Table 2.6). This suggests that accounting for additional sources of variation removed spurious interactions and led to a more plausible network structure.

The reconstructed warbler interaction networks have shown an exponential rather than a power law degree distribution (Figs. A.6 and A.7 in the appendix). This finding is consistent with Dunne et al. (2002) and contributes to the ongoing discussion about the global characteristics of species interaction networks. The networks inferred in our study suggest a number of novel strong interactions that may exist among the warblers. This leads to the formulation of new hypotheses: do *S. currucca* and *S. nisoria* interact, and is the relationship between *H. icterina* and *P. sibilatrix* real? Investigation of the mechanisms behind these interactions may prove valuable.

2.9 Conclusion

We have carried out one of the first studies to address the problem of reconstructing species interaction networks from species abundance data. To this end, we have applied and adapted four machine learning methods recently developed in the field of computational molecular systems biology. We have applied these models and their adaptations to a subset of the European bird atlas data (warblers), and have discovered both interactions that are known from the literature, and significant correlations with interaction scores based on phylogenetic distances and ecological similarities.

It should be noted that finding an interaction between two species in a reconstructed network does not reveal the mechanism that underlies the interaction. If it is found that species A interacts with species B, this could be due to factors as disparate as a predator-prey relationship, competition for common resources, or a symbiotic relationship. Adding further covariates to the data can often elucidate the possible cause; if for example adding information about availability of food sources causes the interaction to disappear, then it is likely that the two species compete for the same resources. However, in the absence of perfect and complete observations, it is not always possible to say with certainty what caused an interaction to be inferred.

We have complemented our study with an evaluation of the network reconstruction on simulated data, for which a proper gold-standard is known. The reconstruction performance was considerably better than random, but we note that perfect reconstruction is unlikely given limited data and the complexity of the ecological processes involved.

The machine learning methods investigated in our study therefore do not provide a mechanism for hypothesis validation. However, our findings suggest that they do offer a useful tool for hypothesis generation, which can enrich and complement traditional methods based on fieldwork and experimental analysis.

The comparative evaluation of different network reconstruction methods has deepened our insight into their relative performance. However, we have found that for a successful application in ecology, the network reconstruction methods currently applied in molecular systems biology need to be modified and improved. We have incorporated a mechanism for taking spatial autocorrelations into account, and we have expanded the models so as to include exogenous bio-climate variables.

Future model improvement should focus on the explicit inclusion of ecological prior knowledge, along the lines of Werhli and Husmeier (2007), and the inclusion of latent variables to allow for unobserved effects (see Sections 2.3.2.4 and 2.6.3 for a preliminary exploration). We have investigated the adaptation of the model proposed in Grzegorczyk et al. (2008a) to include latent variables in Bayesian networks. While our preliminary results on the simulated data were encouraging, as shown in Figure 2.11, the application of this scheme to the warbler data suffered from convergence and mixing problems of the MCMC simulations, which calls for further methodological improvements.

The true value of our study lies in demonstrating that even using large-scale spatial datasets, relevant patterns in ecological networks can be identified using the machine learning methods described here. This suggests that these methods have the potential to contribute novel important tools for gaining deeper insight into the structure and stability of ecosystems, managing biodiversity, and predicting the impact of climate change.

Chapter 3

Time-varying networks: global vs sequential information sharing

Note: This chapter is largely based on the paper "Heterogeneous Continuous Dynamic Bayesian Networks with Flexible Structure and Inter-Time Segment Information Sharing" (Dondelinger et al., 2010), which I presented at "The 27th International Conference on Machine Learning (ICML 2010)". Some of the sections from this paper have been reproduced verbatim. The results on the circadian clock genes in Arabidopsis in Section 3.6.3 have been reproduced from the journal paper "Dynamic Bayesian networks in molecular plant science: Inferring gene regulatory networks from multiple gene expression time series" (Dondelinger et al., 2012a). In addition, some of the text for the introduction and methodology has been adapted from the Machine Learning paper (Dondelinger et al., 2012b).

3.1 Introduction

As pointed out in Chapter 1, one of the challenging problems in the field of systems biology is the inference of gene regulatory networks from high-throughput transcriptomic profiles. While protein interactions can be measured directly with various high-throughput assays, gene regulatory interactions involve several intermediate steps related to the formation, activation and complex formation of transcription factors (e.g. via phosphorylation or dimerization). These processes are generally not directly observable, especially in a high-throughput fashion. For that reason the inference of interactions has to be based on indirect noisy measurements of mRNA concentrations (a proxy for gene activity), rendering the problem of regulatory network reconstruction more difficult than for proteins. Various statistical techniques that I have described in Chapter 1 aim to perform network inference on this data, and the reconstructed regulation networks can reveal how the genes and the proteins they code for interact.

However, many of the regulatory interactions in the cell vary in time. During the development and growth of an organism, some genes and pathways are more active during the early stages, but show practically no activity during the later stages, or vice-versa. *Drosophila melanogaster*, for instance, goes through several developmental stages, from embryo to larva to pupa to adult. Genes involved in wing muscle development would naturally fulfil different roles during the embryonal phase, when no wings are present, than they do in the adult fly, when the wings have fully developed. Gene regulatory and signalling networks can also vary in time in reaction to an environmental trigger, such as the type of growth substrate, or the type of drug treatment applied. Such a trigger can enhance or prevent the interactions of certain proteins, which in turn can have repercussions for the whole network. A recent pertinent example comes from cancer biology, where it has been shown that treatment with one kind of anticancer drug can lead to a rewiring of the signalling network, which affects the response to subsequent treatment with a different drug (Lee et al., 2012).

We are therefore presented with the problem of inferring a regulatory network from a series of discrete measurements or observations in time, where the structure of the network is subject to potential change. Moreover, we may not always know at which stage structural changes are likely to occur, as the underlying processes may be time-delayed, or dependent on unobservable external factors. To extend conventional reverse engineering methods, which only aim to infer a single immutable regulatory network, this work builds on recent research in combining dynamic Bayesian networks (DBNs) with multiple changepoint processes (Robinson and Hartemink, 2009, 2010; Grzegorczyk and Husmeier, 2009, 2011; Lèbre, 2007; Lèbre et al., 2010; Kolar et al., 2009). Below, I will briefly review the state of the art, and the shortcomings of existing methods that we aim to address.

The standard assumption underlying DBNs is that time-series have been generated from a homogeneous Markov process. This assumption is too restrictive, as discussed above, and can potentially lead to erroneous conclusions. While there have been various efforts to relax the homogeneity assumption for undirected graphical models (Talih and Hengartner, 2005; Xuan and Murphy, 2007), relaxing this restriction in DBNs is a more recent research topic (Robinson and Hartemink, 2009, 2010; Grzegorczyk and Husmeier, 2009, 2011; Ahmed and Xing, 2009; Lèbre, 2007; Lèbre et al., 2010; Kolar et al., 2009). At present, none of the proposed methods is without its limitations, leaving room for further methodological innovation. The method proposed in Ahmed and Xing (2009) and Kolar et al. (2009) is non-Bayesian. This requires certain regularization parameters to be optimized "externally", by applying information criteria (like AIC or BIC), cross-validation or bootstrapping. The first approach is suboptimal, the latter approaches are computationally expensive[1]. In this chapter, we therefore follow the Bayesian paradigm, as in Robinson and Hartemink (2009, 2010); Grzegorczyk and Husmeier (2009, 2011); Lèbre (2007) and Lèbre et al. (2010). These approaches also have their limitations. The method proposed in Grzegorczyk and Husmeier (2009, 2011) assumes a fixed network structure and only allows the interaction parameters to vary with time. This assumption is too rigid when looking at processes where changes in the overall regulatory network structure are expected, e.g. in morphogenesis or embryogenesis. The method proposed in Robinson and Hartemink (2009, 2010) requires a discretization of the data, which incurs an inevitable information loss. These limitations are addressed in Lèbre (2007) and Lèbre et al. (2010), where the authors propose a method for continuous data that allows network structures associated with different nodes to change with time in different ways. However, this high flexibility causes potential problems when applied to time series with a low number of measurements, as typically available from systems biology, leading to overfitting or inflated inference uncertainty.

The objective of this chapter is to present a novel model that addresses the methodological shortcomings of the three Bayesian methods mentioned above, and to demonstrate its viability by application to gene expression time series from *Drosophila melanogaster* and *Arabidopsis thaliana*. Unlike Robinson and Hartemink (2009, 2010), the model is continuous and therefore avoids the information loss inherent in a discretization of the data. We further improve on the model in Robinson and Hartemink (2009, 2010) by allowing different nodes in the networks to have different penalty terms. Unlike Grzegorczyk and Husmeier (2009, 2011), our model allows the network structure to change among segments, leading to greater model flexibility. As an improvement on Lèbre (2007) and Lèbre et al. (2010), our model introduces information sharing among time series segments, which provides an essential regularization effect. In this chapter, we have applied the model to reconstruct two regulatory networks: a network of genes involved in wing muscle development during the life cycle

[1] See Larget and Simon (1999) for a demonstration of the higher computational costs of bootstrapping over Bayesian approaches based on MCMC.

of *Drosophila melanogaster*, based on data from Arbeitman et al. (2002), and a network of circadian clock genes in *Arabidopsis thaliana*, based on data from Edwards et al. (2006), Mockler et al. (2007) and Grzegorczyk et al. (2008a).

I compare two different information coupling paradigms in this chapter: global information coupling and sequential information coupling. Global information coupling is appropriate when there is no natural sequential order of the time series segments, such as for segments derived from different experimental conditions. This is the case for the *Arabidopsis thaliana* dataset. Sequential information sharing, which I will also investigate in more detail in Chapter 4, is appropriate for modelling a temporal developmental process, such as those related to morphogenesis, where changes to the network structure happen sequentially. I will present a comparison between the two approaches based on simulation data, before applying them to real-world datasets.

This chapter is organized as follows. Section 3.2 reviews the time-varying, non-homogeneous DBN on which our work is based. Section 3.3 describes the methodological innovation of Bayesian regularization via information coupling. Section 3.4 describes the implementation of our method and the setup of the simulation studies. Section 3.5 gives a description of the synthetic data, and the two real-world datasets from *Drosophila melanogaster* and *Arabidopsis thaliana* that were used in this chapter. Section 3.6 presents and discusses the results of applying our network inference method to this data. The chapter concludes in Section 3.7 with a general discussion and summary.

3.2 Background: Non-homogeneous DBNs

This section summarizes the auto regressive time-varying DBN proposed in Lèbre (2007) and Lèbre et al. (2010). A similar model was proposed in Punskaya et al. (2002). The idea is to combine the Bayesian regression model of Andrieu and Doucet (1999) with multiple changepoint processes and pursue Bayesian inference with reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995). We call this method TVDBN (Time-Varying Dynamic Bayesian Network).

The model is based on the first-order Markov assumption. This assumption is not critical, though, and a generalization to higher orders, as pursued in Punskaya et al. (2002), is straightforward. The value that a node in the graph takes on at time t is determined by the values that the node's parents (i.e. potential regulators, see below) take on at the previous time point, $t-1$. More specifically, the conditional probability of the observation associated with a node at a given time point is a conditional Gaussian distribution, where the conditional mean is a linear weighted sum of the parent values at the previous time point, and the interaction parameters and parent sets depend on the time series segment. The latter dependence adds extra flexibility to the model and thereby relaxes the homogeneity assumption. The interaction parameters, the variance parameters, the number of potential parents, the location of changepoints demarcating the time series segments, and the number of changepoints are given (conjugate) prior distributions in a hierarchical Bayesian model. For inference, all these quantities are sampled from the posterior distribution with RJMCMC. Note that a complete specification of all node-parent configurations determines the structure of a regulatory network: each node receives incoming directed edges from each node in its parent set.

In what follows, we will refer to nodes as genes and to the network as a gene regulatory network. The method is not restricted to molecular systems biology, though.

3.2.1 Graph

Let p be the number of observed genes, and let $\boldsymbol{x} = (x_i(t))_{1 \leq i \leq p, 1 \leq t \leq N}$ be the expression values measured at N time points. $\mathcal{M}^h$ represents a directed graph, i.e. the network defined by a set of directed edges among the p genes. $\mathcal{M}_i^h$ is the subnetwork associated with target gene i, determined by the set of its parents, i.e. the nodes with a directed edge feeding into gene i; these are the potential regulators of the target gene. The meaning of the superscript h is explained in the next section.

3.2.2 Multiple changepoint process

The set of regulatory relationships among the genes, defined by $\mathcal{M}^h$, may vary across time, which we model with a multiple changepoint process. For each target gene i, an unknown number k_i of changepoints define $k_i + 1$ non-overlapping segments. Segment $h = 1,..,k_i + 1$ starts at changepoint ξ_i^{h-1} and stops before ξ_i^h, where $\boldsymbol{\xi}_i = (\xi_i^0,...,\xi_i^{h-1},\xi_i^h,...,\xi_i^{k_i+1})$ with $\xi_i^{h-1} < \xi_i^h$. To delimit the bounds, two pseudo-changepoints are introduced: $\xi_i^0 = 2$ and $\xi_i^{k_i+1} = N+1$. Thus vector $\boldsymbol{\xi}_i$ has length $|\boldsymbol{\xi}_i| = k_i + 2$. The set of changepoints is denoted by $\boldsymbol{\xi} = (\boldsymbol{\xi}_i)_{1 \leq i \leq p}$. This changepoint process induces a partition of the time series, $\boldsymbol{x}_i^h = (x_i(t))_{\xi_i^{h-1} \leq t < \xi_i^h}$, with different network structures $\mathcal{M}_i^h$ associated with the different segments $h \in \{1,\ldots,k_i+1\}$. Identifiability is satisfied by ordering the changepoints based on their position in the time series. We define $\mathcal{M}_i = \{\mathcal{M}_i^h\}_{1 \leq h \leq k_i+1}$ and $\mathcal{M} = \{\mathcal{M}_i\}_{1 \leq i \leq p}$.

3.2.3 Regression model

For each gene i, the random variable $X_i(t)$ refers to the expression of gene i at time t. Within any segment h, the expression of gene i depends on the p gene expression values measured at the previous time point through a regression model defined by (a) a set of s_i^h parents denoted by $\mathcal{M}_i^h = \{j_1,...,j_{s_i^h}\} \subseteq \{1,\ldots,p\}$, $|\mathcal{M}_i^h| = s_i^h$, and (b) a set of parameters $(\boldsymbol{a}_i^h, \sigma_i^h)$ where $\boldsymbol{a}_i^h = (a_{ij}^h)_{0\leq j\leq p}$, $a_{ij}^h \in \mathbb{R}$ and $\sigma_i^h > 0$. For all $j \neq 0$, $a_{ij}^h = 0$ if $j \notin \mathcal{M}_i^h$. For each gene i, for each time point t in segment h ($\xi_i^{h-1} \leq t < \xi_i^h$), the random variable $X_i(t)$ depends on the p variables $\{X_j(t-1)\}_{1\leq j\leq p}$ according to

$$X_i(t) \;=\; a_{i0}^h + \textstyle\sum_{j\in\mathcal{M}_i^h} a_{ij}^h\, X_j(t-1) \;+ \varepsilon_i^h(t) \tag{3.1}$$

where the noise $\varepsilon_i^h(t)$ is assumed to be Gaussian with mean 0 and variance $(\sigma_i^h)^2$, $\varepsilon_i^h(t) \sim N(0,(\sigma_i^h)^2)$. We define $\boldsymbol{a}_i = (\boldsymbol{a}_i^h)_{1\leq h\leq k_i+1}$, $\boldsymbol{a} = (\boldsymbol{a}_i)_{0\leq i\leq p}$, $\boldsymbol{\sigma}_i^2 = (\sigma_i^h)^2_{1\leq h\leq k_i+1}$ and $\boldsymbol{\sigma}^2 = (\boldsymbol{\sigma}_i^2)_{0\leq i\leq p}$.

3.2.4 Prior

The $k_i + 1$ segments are delimited by k_i changepoints, where k_i is distributed a priori as a truncated Poisson random variable with mean λ and maximum $\overline{k} = N-2$:

$$P(k_i|\lambda) \;\propto\; \frac{\lambda^{k_i}}{k_i!}\mathbb{1}_{\{k_i\leq\bar{k}\}}; \qquad P(\boldsymbol{k}|\lambda) \;=\; \prod_{i=1}^{p} P(k_i|\lambda) \tag{3.2}$$

where $\boldsymbol{k} = (k_1,\ldots,k_p)$. Conditional on k_i changepoints, the changepoint position vector $\boldsymbol{\xi}_i = (\xi_i^0,\xi_i^1,...,\xi_i^{k_i+1})$ takes non-overlapping integer values, which we take to be uniformly distributed a priori. There are $(N-2)$ possible positions for the k_i changepoints, thus vector $\boldsymbol{\xi}_i$ has prior density:

$$P(\boldsymbol{\xi}_i|k_i) = 1/\binom{N-2}{k_i} = \frac{k_i!(N-2-k_i)!}{(N-2)!} \tag{3.3}$$

For each gene i, for each segment h, the number s_i^h of parents for node i follows a truncated Poisson distribution with mean Λ and maximum $\bar{s} = 5$:

$$P(s_i^h|\Lambda) \;\propto\; \frac{\Lambda^{s_i^h}}{s_i^h!}\mathbb{1}_{\{s_i^h\leq\bar{s}\}} \tag{3.4}$$

Conditional on s_i^h, the prior for the parent set $\mathcal{M}_i^h$ is a uniform distribution over all parent sets with cardinality s_i^h,

$$P(\mathcal{M}_i^h|s_i^h) = 1/\binom{p}{s_i^h} = \frac{s_i^h!(p-s_i^h)!}{p!} \tag{3.5}$$

The overall prior on the network structures is given by marginalization:

$$P(\mathcal{M}_i^h|\Lambda) = \sum_{s_i^h=0}^{\overline{s}} P(\mathcal{M}_i^h|s_i^h)P(s_i^h|\Lambda) \tag{3.6}$$

Conditional on the parent set $\mathcal{M}_i^h$ of size s_i^h, the s_i^h+1 regression coefficients form a subset of $\boldsymbol{a}_i^h$ denoted by $\boldsymbol{a}_{\mathcal{M}_i^h} = (a_{i0}^h,(a_{ij}^h)_{j\in\mathcal{M}_i^h})$. They are assumed zero-mean multivariate Gaussian with covariance matrix $(\sigma_i^h)^2\boldsymbol{\Sigma}_{\mathcal{M}_i^h}$,

$$P(\boldsymbol{a}_i^h|\mathcal{M}_i^h,\sigma_i^h)=|2\pi(\sigma_i^h)^2\boldsymbol{\Sigma}_{\mathcal{M}_i^h}|^{-\frac{1}{2}}\exp\left(-\frac{\boldsymbol{a}_{\mathcal{M}_i^h}^{\dagger}\boldsymbol{\Sigma}_{\mathcal{M}_i^h}^{-1}\boldsymbol{a}_{\mathcal{M}_i^h}}{2(\sigma_i^h)^2}\right) \tag{3.7}$$

where $|.|$ denotes the determinant of a matrix, the symbol † denotes matrix transposition, $\boldsymbol{\Sigma}_{\mathcal{M}_i^h}^{-1} = \delta^{-2}\boldsymbol{D}_{\mathcal{M}_i^h}^{\dagger}\boldsymbol{D}_{\mathcal{M}_i^h}$ and $\boldsymbol{D}_{\mathcal{M}_i^h}$ is the $(\xi_i^h - \xi_i^{h-1}) \times (s_i^h+1)$ matrix whose first column is a vector of 1's (for the constant in model (3.1)) and each $(j+1)^{th}$ column contains the observed values $(x_j(t))_{\xi_i^{h-1}-1\leq t<\xi_i^h-1}$ for each factor gene j in $\mathcal{M}_i^h$. This so-called g-prior was also used in Andrieu and Doucet (1999) and is motivated in Zellner (1986). Note that although the g-prior can be considered data-dependent via the design matrix $\boldsymbol{D}$, which seems to violate the Likelihood Principle, this violation is less critical than one may suspect due to fact that $\boldsymbol{D}$ is only used to obtain the Fisher information matrix for the covariates of the regression, and not for obtaining information about the response. For more details on the g-prior, see Liang et al. (2008). Finally, the conjugate prior for the variance $(\sigma_i^h)^2$ is the inverse gamma distribution, $P((\sigma_i^h)^2) = \mathcal{IG}(\upsilon_0,\gamma_0)$. Following Lèbre (2007) and Lèbre et al. (2010), we set the hyper-hyperparameters for shape, $\upsilon_0 = 0.5$, and scale, $\gamma_0 = 0.05$, to fixed values that give a vague distribution. The terms λ and Λ can be interpreted as the expected number of changepoints and parents, respectively, and δ^2 is the expected signal-to-noise ratio. These hyperparameters are drawn from vague conjugate hyperpriors, which are in the (inverse) gamma distribution family:

$$P(\Lambda) = P(\lambda) = \mathcal{G}a(0.5,1) = \Lambda^{-0.5}\frac{\exp(-\Lambda)}{\Gamma(0.5)} \tag{3.8}$$

and

$$P(\delta^2) = \mathcal{IG}(2,0.2) = \delta^{-6}\frac{0.04\exp(-\frac{0.2}{\delta^2})}{\Gamma(2)} \tag{3.9}$$

3.2.5 Posterior

Equation (3.1) implies that

$$P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\boldsymbol{a}_i^h,\sigma_i^h) = \left(\sqrt{2\pi}\sigma_i^h\right)^{-\mathrm{length}(\boldsymbol{x}_i^h)} \exp\left(-\frac{(\boldsymbol{x}_i^h-\boldsymbol{D}_{\mathcal{M}_i^h}\boldsymbol{a}_{\mathcal{M}_i^h})^\dagger\,(\boldsymbol{x}_i^h-\boldsymbol{D}_{\mathcal{M}_i^h}\boldsymbol{a}_{\mathcal{M}_i^h})}{2(\sigma_i^h)^2}\right) \quad (3.10)$$

where $\mathrm{length}(\boldsymbol{x}_i^h)$ is the length of the time series segment h. From Bayes' theorem, the posterior is given by the following equation, where all prior distributions have been defined above:

$$\begin{aligned} P(\boldsymbol{k},\boldsymbol{\xi},\mathcal{M},\boldsymbol{a},\boldsymbol{\sigma^2},\lambda,\Lambda,\delta^2|\boldsymbol{x}) \propto & P(\delta^2)P(\lambda)P(\Lambda)\prod_{i=1}^{p}P(k_i|\lambda)P(\boldsymbol{\xi}_i|k_i)\prod_{h=1}^{k_i}P(\mathcal{M}_i^h|\Lambda) \\ & P([\sigma_i^h]^2)P(\boldsymbol{a}_i^h|\mathcal{M}_i^h,[\sigma_i^h]^2,\delta^2)P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\boldsymbol{a}_i^h,[\sigma_i^h]^2) \end{aligned} \quad (3.11)$$

An attractive feature of the chosen model is that the integration over the parameters $\boldsymbol{a}$ and $\boldsymbol{\sigma^2}$ in the posterior distribution of equation (3.11) is analytically tractable:

$$\begin{aligned} P(\boldsymbol{k},\boldsymbol{\xi},\mathcal{M},\lambda,\Lambda,\delta^2|\boldsymbol{x}) &= \int\int P(\boldsymbol{k},\boldsymbol{\xi},\mathcal{M},\boldsymbol{a},\boldsymbol{\sigma^2},\lambda,\Lambda,\delta^2|\boldsymbol{x})d\boldsymbol{a}d\boldsymbol{\sigma^2} && (3.12) \\ &\propto P(\delta^2)P(\lambda)P(\Lambda)\prod_{i=1}^{p}\int\int P(k_i,\boldsymbol{\xi}_i,\mathcal{M}_i,\boldsymbol{a}_i,\boldsymbol{\sigma}_i^2,\boldsymbol{x}_i|\lambda,\Lambda,\delta^2)d\boldsymbol{a}_i d\boldsymbol{\sigma}_i^2 \end{aligned}$$

For each gene i, the joint distribution for k_i, $\boldsymbol{\xi}_i$, $\mathcal{M}_i$, $\boldsymbol{a}_i$, $\boldsymbol{\sigma}_i^2$, $\boldsymbol{x}_i$ conditional on hyperparameters λ, Λ, δ^2, is integrated over the parameters $\boldsymbol{a}_i$ (normal distribution) and $\boldsymbol{\sigma}_i^2$ (inverse gamma distribution). Solving this integral (for details see Lèbre et al., 2010), the following expression is obtained:

$$\begin{aligned} &\int\int P(k_i,\boldsymbol{\xi}_i,\mathcal{M}_i,\boldsymbol{a}_i,\boldsymbol{\sigma}_i^2,\boldsymbol{x}_i|\lambda,\Lambda,\delta^2)d\boldsymbol{a}_i d\boldsymbol{\sigma}_i^2 = \\ &\quad C_\lambda\,\lambda^{k_i}\,\frac{(N-2-k_i)!}{(N-2)!}\prod_{h=1}^{k_i+1}\left\{\frac{(p-s_i^h)!}{p!}\,C_\Lambda\,\Lambda^{s_i^h}\,P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)\right\} \end{aligned} \quad (3.13)$$

where C_λ, C_Λ are the normalization constants required by the truncation of the Poisson distribution (3.2) and (3.4) and where

$$\begin{aligned} P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2) &= (\delta^2+1)^{-\frac{s_i^h+1}{2}}\,\frac{\left(\frac{\gamma_0}{2}\right)^{\upsilon_0/2}}{\Gamma(\frac{\upsilon_0}{2})} \\ &\quad \Gamma\left(\frac{\upsilon_0+\mathrm{length}(\boldsymbol{x}_i^h)}{2}\right)\left(\frac{\gamma_0+(\boldsymbol{x}_i^h)^\dagger\boldsymbol{P}_i^h\boldsymbol{x}_i^h}{2}\right)^{-\frac{\upsilon_0+\mathrm{length}(\boldsymbol{x}_i^h)}{2}} \end{aligned} \quad (3.14)$$

where the matrices $\boldsymbol{P}_i^h$ and $\boldsymbol{M}_i^h$ are defined as follows, with $\boldsymbol{I}$ referring to the identity matrix of size length$(\boldsymbol{x}_i^h)$:

$$\boldsymbol{P}_i^h = \boldsymbol{I} - \boldsymbol{D}_{\mathcal{M}_i^h} \boldsymbol{M}_i^h \boldsymbol{D}^{\dagger}_{\mathcal{M}_i^h}, \tag{3.15}$$

$$\boldsymbol{M}_i^h = \frac{\delta^2}{\delta^2+1} \left(\boldsymbol{D}^{\dagger}_{\mathcal{M}_i^h} \boldsymbol{D}_{\mathcal{M}_i^h} \right)^{-1}. \tag{3.16}$$

The number of changepoints $\boldsymbol{k}$ and their location, $\boldsymbol{\xi}$, the network structure $\mathcal{M}$ and the hyperparameters λ, Λ and δ^2 can be sampled from the posterior $P(\boldsymbol{k}, \boldsymbol{\xi}, \mathcal{M}, \lambda, \Lambda, \delta^2 | \boldsymbol{x})$ with a reversible jump MCMC (Green, 1995) scheme detailed in the next subsection.

3.2.6 RJMCMC scheme

Four different update moves are proposed: birth of a new changepoint (B); death (removal) of an existing changepoint (D); shift of a changepoint to a different time-point (S); and update of the network topology within the segments (N). These moves occur with probabilities b_{k_i} for B, d_{k_i} for D, u_{k_i} for S and v_{k_i} for N, depending only on the current number of changepoints k_i and satisfying $b_{k_i} + d_{k_i} + u_{k_i} + v_{k_i} = 1$. The changepoint birth and death moves represent changes from, respectively, k_i to $k_i + 1$ segments and k_i to $k_i - 1$ segments. In order to preserve the restriction on the number of changepoints, some probabilities are set to 0: $d_0 = u_0 = 0$ and $b_{\overline{k}} = 0$. Otherwise, following Green (1995), these probabilities are chosen as follows,

$$b_{k_i} = c \ \min\left\{1, \frac{P(k_i+1|\lambda)}{P(k_i|\lambda)}\right\}, \quad d_{k_i+1} = c \ \min\left\{1, \frac{P(k_i|\lambda)}{P(k_i+1|\lambda)}\right\} \tag{3.17}$$

where $P(k_i|\lambda)$ is the prior distribution for the number of changepoints defined in equation (3.2) and the constant c is chosen to be smaller than $1/4$ so that network structure updates and changepoint position shifts are proposed more frequently than births and deaths of changepoints. This improves mixing and convergence with respect to changepoint positions and network structures within the different segments. Shifting of a changepoint is proposed with probability $u_{k_i} = (1 - b_{k_i} - d_{k_i+1})/3$, and updating of the network structure within each segment is proposed with probability $v_{k_i} = 1 - (b_{k_i} + d_{k_i} + u_{k_i})$.

Following Green (1995), the RJMCMC acceptance probability of a changepoint birth is equal to $\min\{1, R\}$, where the acceptance ratio R reads as follows:

$$R = (\text{likelihood ratio}) \times (\text{prior ratio}) \times (\text{proposal ratio}) \times (\text{Jacobian}). \tag{3.18}$$

The product of the likelihood and the prior ratio is the posterior ratio which is derived from equation (3.12). The computation of the proposal ratio and the Jacobian depends on the choice for the various moves designed to sample the time-varying network distribution. We briefly describe below the chosen moves and their associated acceptance ratio. A complete description of the computation of the acceptance ratio for each move can be found in Lèbre et al. (2010).

Let $\boldsymbol{\xi}_i$ be the current changepoint vector containing k_i changepoints. For a changepoint birth move, a new changepoint position $\xi^\star$ is sampled uniformly from the available positions. The new changepoint is within an existing segment $h^\star$ of the target gene i, $\xi_i^{h^\star-1} < \xi^\star < \xi_i^{h^\star}$. Let us denote by $h_L^\star$ and $h_R^\star$ the segments to the left and to the right of the new changepoint respectively and by $\boldsymbol{x}_i^{h^\star} = (\boldsymbol{x}_i^{h_L^\star}, \boldsymbol{x}_i^{h_R^\star})$ the observed values for gene i in those segments. One of $h_L^\star$ and $h_R^\star$ is chosen with equal probability. That segment retains the current network topology $\mathcal{M}_i^{h^\star}$ of segment $h^\star$, and an entirely new topology is sampled from the prior defined in equation (3.6) for the other segment. Let us denote by $s^\star$ the number of edges of the new topology. The Jacobian is equal to 1 and the prior ratio is computed from the probability of choosing a new changepoint position and a new network structure for the new segment. Then the birth of the proposed changepoint is accepted with probability $A(\boldsymbol{\xi}_i^+|\boldsymbol{\xi}_i) = \min\{1, R(\boldsymbol{\xi}_i^+|\boldsymbol{\xi}_i)\}$, with

$$R(\boldsymbol{\xi}_i^+|\boldsymbol{\xi}_i) = \frac{1}{(\delta^2+1)^{(s^\star+1)/2}} \frac{\left(\frac{\gamma_0}{2}\right)^{\upsilon_0/2}}{\Gamma\left(\frac{\upsilon_0}{2}\right)} \frac{\Gamma_{h_L^\star}\Gamma_{h_R^\star}}{\Gamma_{h^\star}} \left(\frac{\upsilon_0 + (\boldsymbol{x}_i^{h^\star})^\dagger \boldsymbol{P}_i^{h^\star} \boldsymbol{x}_i^{h^\star}}{2}\right)^{\frac{1}{2}(\upsilon_0+\xi_i^{h^\star}-\xi_i^{h^\star-1})}$$
$$\left(\frac{\upsilon_0 + (\boldsymbol{x}_i^{h_L^\star})^\dagger \boldsymbol{P}_i^{h_L^\star} \boldsymbol{x}_i^{h_L^\star}}{2}\right)^{-\frac{1}{2}(\upsilon_0+\xi_i^{h_L^\star}-\xi_i^{h_L^\star-1})} \left(\frac{\upsilon_0 + (\boldsymbol{x}_i^{h_R^\star})^\dagger \boldsymbol{P}_i^{h_R^\star} \boldsymbol{x}_i^{h_R^\star}}{2}\right)^{-\frac{1}{2}(\upsilon_0+\xi_i^{h_R^\star}-\xi_i^{h_R^\star-1})} \tag{3.19}$$

For details see Lèbre et al. (2010). Here $\boldsymbol{\xi}_i^+$ refers to the proposed changepoint vector after adding the new changepoint $\xi^\star$ to the current vector $\boldsymbol{\xi}_i$ and for all h in $\{1,..,k_{i+1}\}$, $\Gamma_h = \Gamma\left(\frac{\upsilon_0+\xi_i^h-\xi_i^{h-1}}{2}\right)$, and all other quantities are defined in Section 3.2.5.

For a changepoint death move, an existing changepoint in the current configuration is selected uniformly at random. The two segments adjacent to this changepoint are proposed to be merged into one segment, which will conserve the network structure of one of the two segments (selected with equal probability). Let us denote by $\boldsymbol{\xi}_i^-$ the proposed changepoint vector after removing the selected changepoint from the current vector $\boldsymbol{\xi}_i$. The acceptance ratio of the changepoint death move is equal to the inverse of the changepoint birth acceptance ratio $R(\boldsymbol{\xi}_i|\boldsymbol{\xi}_i^-)$ for proposing a change from $\boldsymbol{\xi}_i^-$ to $\boldsymbol{\xi}_i$, given in equation (3.19). Therefore the acceptance probability of a changepoint death move is,

$$A(\boldsymbol{\xi}_i^-|\boldsymbol{\xi}_i) = \min\{1, \left(R(\boldsymbol{\xi}_i|\boldsymbol{\xi}_i^-)\right)^{-1}\}. \tag{3.20}$$

Proposed shifts in changepoint positions are accepted using a standard Metropolis-Hastings step (Hastings, 1970) where a change is accepted with probability $\min\{1, R\}$ where $R =$ (posterior ratio) × (proposal ratio). The new changepoint vector $\tilde{\boldsymbol{\xi}}_i$ is obtained by replacing ξ_i^h with $\tilde{\xi}_i^h$ such that the absolute value $|\xi_i^h - \tilde{\xi}_i^h| = 1$. The posterior ratio is obtained from equation (3.12). Let us denote by $Q(\tilde{\boldsymbol{\xi}}_i|\boldsymbol{\xi}_i)$ the probability of shifting changepoint ξ_i^h to $\tilde{\xi}_i^h$ in the current changepoint vector $\boldsymbol{\xi}^i$ (and reciprocally for $Q(\boldsymbol{\xi}_i|\tilde{\boldsymbol{\xi}}_i)$), then the changepoint shift is accepted with probability $A(\tilde{\boldsymbol{\xi}}_i|\boldsymbol{\xi}_i) = \min\{1, R(\tilde{\boldsymbol{\xi}}_i|\boldsymbol{\xi}_i)\}$ where,

$$R(\tilde{\boldsymbol{\xi}}_i|\boldsymbol{\xi}_i) = \left(\frac{\left(\gamma_0 + (\tilde{\boldsymbol{x}}_i^h)^\dagger \tilde{\boldsymbol{P}}_i^h \tilde{\boldsymbol{x}}_i^h\right)^{(\upsilon_0 + \tilde{\xi}_i^h - \xi_i^{h-1})} \left(\gamma_0 + (\tilde{\boldsymbol{x}}_i^{h+1})^\dagger \tilde{\boldsymbol{P}}_i^{h+1} \tilde{\boldsymbol{x}}_i^{h+1}\right)^{(\upsilon_0 + \xi_i^{h+1} - \tilde{\xi}_i^h)}}{\left(\gamma_0 + (\boldsymbol{x}_i^h)^\dagger \boldsymbol{P}_i^h \boldsymbol{x}_i^h\right)^{(\upsilon_0 + \xi_i^h - \xi_i^{h-1})} \left(\gamma_0 + (\boldsymbol{x}_i^{h+1})^\dagger \boldsymbol{P}_i^{h+1} \boldsymbol{x}_i^{h+1}\right)^{(\upsilon_0 + \xi_i^{h+1} - \xi_i^h)}} \right)^{1/2}$$
$$\frac{\Gamma\left(\frac{\upsilon_0 + \tilde{\xi}_i^h - \xi_i^{h-1}}{2}\right) \Gamma\left(\frac{\upsilon_0 + \xi_i^{h+1} - \tilde{\xi}_i^h}{2}\right)}{\Gamma\left(\frac{\upsilon_0 + \xi_i^h - \xi_i^{h-1}}{2}\right) \Gamma\left(\frac{\upsilon_0 + \xi_i^{h+1} - \xi_i^h}{2}\right)} \; \frac{Q(\boldsymbol{\xi}_i|\tilde{\boldsymbol{\xi}}_i)}{Q(\tilde{\boldsymbol{\xi}}_i|\boldsymbol{\xi}_i)}, \tag{3.21}$$

where $\tilde{\boldsymbol{x}}_i^h$ and $\tilde{\boldsymbol{x}}_i^{h+1}$ refer to the expression levels for gene i observed in phase h and $h+1$ of the new changepoint vector $\tilde{\boldsymbol{\xi}}_i$, and $\tilde{\boldsymbol{P}}_i^h$ and $\tilde{\boldsymbol{P}}_i^{h+1}$ are the projection matrices built from $\tilde{\boldsymbol{x}}_i^h$ and $\tilde{\boldsymbol{x}}_i^{h+1}$ as defined in equation (3.15), and all other quantities are as defined in Section 3.2.5. See Lèbre et al. (2010) for the derivation of this equation.

Finally, network structure updates within segments invoke a second RJMCMC scheme, which was adapted from the model selection approach of Andrieu and Doucet (1999). When such a move is chosen, for each segment successively, we consider either the birth or death of an edge. For an edge birth move, a new edge is selected uniformly at random from the set of possible edges. For an edge death move, an edge to be removed is selected uniformly at random from the set of existing edges. The edge birth and death moves represent changes from s_i^h to $s_i^h + 1$ or $s_i^h - 1$ parents in the regression model. The probabilities of choosing these moves, $b_{s_i^h}$ and $d_{s_i^h}$ respectively, are defined as follows,

$$b_{s_i^h} = C_{s_i^h} \min\left\{1, \frac{P_{\bar{s}}(s_i^h + 1)}{P_{\bar{s}}(s_i^h)}\right\} \quad \text{and} \quad d_{s_i^h} = C_{s_i^h} \min\left\{1, \frac{P_{\bar{s}}(s_i^h - 1)}{P_{\bar{s}}(s_i^h)}\right\} \tag{3.22}$$

where $C_{s_i^h}$ is a normalization constant dependent on s_i^h, and set to ensure that $b_{s_i^h} + d_{s_i^h} = 1$. Additionally, we define $b_0 = 1$, $d_0 = 0$, $b_{\bar{s}} = 0$ and $d_{\bar{s}} = 1$. The acceptance ratio $R(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)$ for the new set of $\tilde{s}_i^h$ parents $\tilde{\mathcal{M}}_i^h$ (which corresponds to $\mathcal{M}_i^h$ with a parent added or removed) is computed according to equation (3.18). Using equations (3.4) and (3.5), the edge birth prior ratio becomes

$$R_{prior} = \frac{P(\tilde{\mathcal{M}}_i^h|\tilde{s}_i^h)}{P(\mathcal{M}_i^h|s_i^h)} \frac{P(\tilde{s}_i^h|\Lambda)}{P(s_i^h|\Lambda)} \tag{3.23}$$

and the proposal ratio becomes

$$R_{proposal} = \frac{Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)} \tag{3.24}$$

where $Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)$ is the proposal probability of parent set $\tilde{\mathcal{M}}_i^h$ given parent set $\mathcal{M}_i^h$, which is defined as follows:

$$\begin{aligned} Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = b_{|\mathcal{M}_i^h|}\delta(|\tilde{\mathcal{M}}_i^h|, |\mathcal{M}_i^h|+1) Q^+(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) + \\ d_{|\mathcal{M}_i^h|}\delta(|\tilde{\mathcal{M}}_i^h|, |\mathcal{M}_i^h|-1) Q^-(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) \end{aligned} \tag{3.25}$$

with $\delta(x,y)$ being the Kronecker delta function. $Q^+(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = 1/(p - |\tilde{\mathcal{M}}_i^h|)$ is the proposal probability of an edge birth move, and $Q^-(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = 1/|\tilde{\mathcal{M}}_i^h|$ is the proposal probability of an edge death move. The Jacobian equals 1. Then using equation (3.14) for the likelihood ratio, the Metropolis-Hastings acceptance ratio for an edge move becomes

$$R(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = \frac{Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)} \frac{P(\tilde{s}_i^h|\Lambda)}{P(s_i^h|\Lambda)} \frac{P(\tilde{\mathcal{M}}_i^h|\tilde{s}_i^h)}{P(\mathcal{M}_i^h|s_i^h)} \frac{P(\boldsymbol{x}_i^h|\tilde{\mathcal{M}}_i^h,\delta^2)}{P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)} \tag{3.26}$$

Note that the prior ratio and the proposal ratio cancel out, and hence the edge move acceptance ratio is equal to the likelihood ratio, that is,

$$R(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = \frac{P(\boldsymbol{x}_i^h|\tilde{\mathcal{M}}_i^h,\delta^2)}{P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)} \tag{3.27}$$

Finally, the probability of accepting an edge move is,

$$A(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = \min\{1, R(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)\} \tag{3.28}$$

The sampling scheme for updating the hyperparameters δ^2, λ and Λ is described in Lèbre (2007) and Lèbre et al. (2010). Together the four moves B, D, S and N allow the generation of samples from probability distributions defined on unions of spaces of different dimensions for both the number of changepoints k_i and the number of parents s_i^h within each segment h for gene i.

3.3 Model Improvement

Allowing the network structure to change between segments leads to a highly flexible model. However, this approach faces a conceptual and a practical problem. The *practical* problem is potential model over-flexibility. If subsequent changepoints are close together, network structures have to be inferred from short time series segments. This will almost inevitably lead to overfitting (in a maximum likelihood context) or inflated inference uncertainty (in a Bayesian context). The *conceptual* problem is the underlying assumption that structures associated with different segments are a priori independent. This is not realistic. For instance, for the evolution of a gene regulatory network during embryogenesis, we would assume that the network evolves gradually and that networks associated with adjacent time intervals are a priori similar.

To address these problems, we propose two methods of information sharing among time series segments. The first method is based on the hierarchical Bayesian model of Werhli and Husmeier (2008). However, rather than sharing information hierarchically – comparing all network structures to a central latent structure – we share information sequentially: a network structure is a priori assumed to be similar to the adjacent ones. The second method uses information from all the other segments to define a prior distribution on the edges for a given segment. We will investigate the relative merits of these two information sharing schemes below.

3.3.1 Sequential information sharing

Denote by $K_i := k_i + 1$ the number of partitions associated with node i, and recall that each time series segment y^h is associated with a separate subnetwork $\mathcal{M}_i^h$, $1 \leq h \leq K_i$. We impose a prior distribution $P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i)$ on the structures, and the joint probability distribution factorizes according to a Markovian dependence:

$$P(y^1,\dots,y^{K_i},\mathcal{M}_i^1,\dots,\mathcal{M}_i^{K_i},\beta_i) = \prod_{h=1}^{K_i} P(y^h|\mathcal{M}_i^h)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i)P(\beta_i) \qquad (3.29)$$

This leads to a graphical structure for our model as represented in Figure 3.1.

Similar to Werhli and Husmeier (2008) we define

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i) = \frac{\exp(-\beta_i|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|)}{Z_i(\beta_i,\mathcal{M}_i^{h-1})} \qquad (3.30)$$

for $h \geq 2$, where β_i is a hyperparameter that defines the strength of the coupling between $\mathcal{M}_i^h$ and $\mathcal{M}_i^{h-1}$. For $h = 1$, $P(\mathcal{M}_i^h)$ is given by (3.6). The denominator

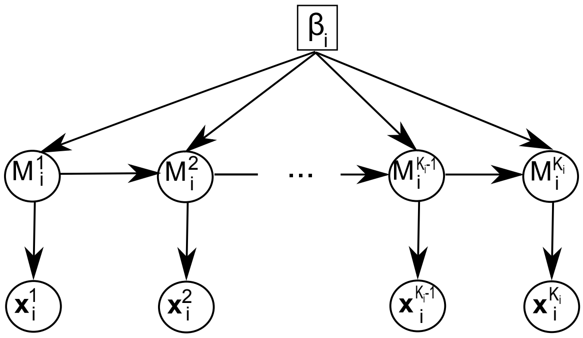


Figure 3.1: Sequential information sharing scheme, whereby each sub-network $\mathcal{M}_i^h$ for node i depends on the previous sub-network $\mathcal{M}_i^{h-1}$, through an exponential prior on the number of differences between the two networks, regularized by a hyperparameter β_i.

$Z_i(\beta_i, \mathcal{M}_i^{h-1})$ in (3.30) is a normalizing constant, also known as the partition function:

$$Z_i(\beta_i) = \sum\nolimits_{\mathcal{M}_i^h \in \mathbb{M}_i} e^{-\beta_i|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|} \tag{3.31}$$

where $\mathbb{M}_i$ is the set of all valid subnetwork structures. If we ignore any fan-in restriction that might have been imposed a priori (via $\bar{s}$), then the expression for the partition function can be simplified: $Z_i(\beta_i) \approx \prod_i \prod_j Z_{ij}(\beta_i)$ where

$$Z_{ij}(\beta_i) = \sum\nolimits_{e_{ij}^h=0}^{1} e^{-\beta_i|e_{ij}^h - e_{ij}^{h-1}|} = 1 + e^{-\beta_i} \tag{3.32}$$

and hence

$$Z_i = \left(1 + e^{-\beta_i}\right)^p \tag{3.33}$$

Inserting (3.33) into (3.30) gives:

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1}, \beta_i) = \frac{\exp(-\beta_i|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|)}{\left(1 + e^{-\beta_i}\right)^p} \tag{3.34}$$

It is straightforward to integrate the proposed model into the RJMCMC scheme of Lèbre (2007). When proposing a new network structure $\mathcal{M}_i^h \to \tilde{\mathcal{M}}_i^h$ for segment h, the prior probability ratio has to be replaced by the following one:

$$\frac{P(\mathcal{M}_i^{h+1}|\tilde{\mathcal{M}}_i^h, \beta_i)P(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^{h-1}, \beta_i)}{P(\mathcal{M}_i^{h+1}|\mathcal{M}_i^h, \beta_i)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1}, \beta_i)} = \frac{\exp[-\beta_i(|\mathcal{M}_i^{h+1} - \tilde{\mathcal{M}}_i^h| + |\tilde{\mathcal{M}}_i^h - \mathcal{M}_i^{h-1}|)]}{\exp[-\beta_i(|\mathcal{M}_i^{h+1} - \mathcal{M}_i^h| + |\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|)]} \tag{3.35}$$

An additional MCMC step is introduced for sampling the hyperparameters β_i from the posterior distribution. For a proposal move $\beta_i \to \tilde{\beta}_i$ with symmetric proposal probability $Q(\tilde{\beta}_i|\beta_i) = Q(\beta_i|\tilde{\beta}_i)$ we get the following acceptance probability:

$$A(\tilde{\beta}_i|\beta_i) = \min\left\{ \prod_{h=2}^{K_i} \frac{\exp(-\tilde{\beta}_i|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|)}{\exp(-\beta_i|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|)} \frac{\left(1+e^{-\beta_i}\right)^p P(\tilde{\beta}_i)}{\left(1+e^{-\tilde{\beta}_i}\right)^p P(\beta_i)}, 1 \right\} \tag{3.36}$$

where in our study the hyperprior $P(\beta_i)$ was chosen as the uniform distribution on the interval $[0,5]$. Note that the scheme proposed in Robinson and Hartemink (2009) can be regarded as a special case of the one we propose. However, Robinson and Hartemink (2009) use two simplifications that are not present in our method: (1) Changepoints are not allowed to vary between nodes. (2) The common hyperparameter $\beta_i = \beta\ \forall i$ has to be chosen by the user in advance and is not inferred from the data.

3.3.2 Global information sharing

We investigate an alternative scheme, based on ideas presented in Ferrazzi et al. (2008). Let $e_{ij}^h \in \{0,1\}$ denote the indicator variable for a directed edge from node i to node j in the hth network (i.e. the network corresponding to the hth section of the time series), and let $\theta_{ij} \in [0,1]$ denote the probability that the node pair (i,j) is connected by a directed edge. We assume that for a given node pair (i,j), the edge indicator variables $\{e_{ij}^h\}$ are iid distributed,

$$P(e_{ij}^h|\theta_{ij}) = (\theta_{ij})^{e_{ij}^h}(1-\theta_{ij})^{1-e_{ij}^h} \tag{3.37}$$

with a conjugate beta prior on the parameters θ_{ij}:

$$P(\theta_{ij}) = \frac{\Gamma(\alpha_{ij}+\overline{\alpha_{ij}})}{\Gamma(\alpha_{ij})\Gamma(\overline{\alpha_{ij}})} \theta_{ij}^{\alpha_{ij}-1}(1-\theta_{ij})^{\overline{\alpha_{ij}}-1} \tag{3.38}$$

where α_{ij} and $\overline{\alpha_{ij}}$ are hyperparameters. Given the subnetworks $\mathcal{M}_i^{\tilde{h}}$ in all segments $\tilde{h}$ different from the current segment h, the prior probability of the subnetwork in the current segment, $\mathcal{M}_i^h$, is

$$\begin{aligned} P(\mathcal{M}_i^h|\{\mathcal{M}_i^{\tilde{h}}\}_{\tilde{h}\neq h}) &= \prod\nolimits_j P(e_{ij}^h|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}) \\ P(e_{ij}^h|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}) &= \int P(e_{ij}^h|\theta_{ij})P(\theta_{ij}|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h})d\theta_{ij} \end{aligned} \tag{3.39}$$

where

$$P(\theta_{ij}|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}) \propto P(\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}|\theta_{ij})P(\theta_{ij}) \tag{3.40}$$

We introduce the following sufficient statistics: B_{ij}^h is the number of networks in segments different from the current segment h in which the node pair (i,j) is connected by a directed edge. Conversely, $\overline{B_{ij}^h}$ is the size of the complement set, i.e. the number of networks in segments different from the current segment h without an edge from node i to node j. Obviously, $B_{ij}^h + \overline{B_{ij}^h} = K_i - 1$, and

$$P(\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}|\theta_{ij}) \;=\; \theta_{ij}^{B_{ij}^h}(1-\theta_{ij})^{\overline{B_{ij}^h}} \tag{3.41}$$

Inserting (3.41) and (3.38) into (3.40) leads to:

$$P(\theta_{ij}|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}) = \frac{\Gamma(\alpha_{ij}+B_{ij}^h+\overline{\alpha_{ij}}+\overline{B_{ij}^h})}{\Gamma(B_{ij}^h+\alpha_{ij})\Gamma(\overline{B_{ij}^h}+\overline{\alpha_{ij}})}\theta_{ij}^{B_{ij}^h+\alpha_{ij}-1}(1-\theta_{ij})^{\overline{B_{ij}^h}+\overline{\alpha_{ij}}-1} \tag{3.42}$$

Inserting (3.37) and (3.42) into (3.39) yields:

$$\begin{aligned} P(e_{ij}^h|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}) &= \frac{\Gamma(\alpha_{ij}+B_{ij}^h+\overline{\alpha_{ij}}+\overline{B_{ij}^h})}{\Gamma(B_{ij}^h+\alpha_{ij})\Gamma(\overline{B_{ij}^h}+\overline{\alpha_{ij}})}\int(\theta_{ij})^{B_{ij}^h+e_{ij}^h+\alpha_{ij}-1}(1-\theta_{ij})^{\overline{B_{ij}^h}+\overline{e_{ij}^h}+\overline{\alpha_{ij}}-1}d\theta_{ij} \\ &= \frac{\Gamma(\alpha_{ij}+B_{ij}^h+\overline{\alpha_{ij}}+\overline{B_{ij}^h})}{\Gamma(B_{ij}^h+\alpha_{ij})\Gamma(\overline{B_{ij}^h}+\overline{\alpha_{ij}})}\frac{\Gamma(B_{ij}^h+\alpha_{ij}+e_{ij}^h)\Gamma(\overline{B_{ij}^h}+\overline{\alpha_{ij}}+\overline{e_{ij}^h})}{\Gamma(\alpha_{ij}+B_{ij}^h+\overline{\alpha_{ij}}+\overline{B_{ij}^h}+1)} \end{aligned} \tag{3.43}$$

where we have defined $\overline{e_{ij}^h} = 1 - e_{ij}^h$. Using $\Gamma(x+1) = x\Gamma(x)$, this expression can be simplified:

$$P(e_{ij}^h = 1|\{e_{ij}^{\tilde{h}}\}_{\tilde{h}\neq h}) \;=\; \frac{\alpha_{ij}+B_{ij}^h}{\alpha_{ij}+B_{ij}^h+\overline{\alpha_{ij}}+\overline{B_{ij}^h}} \tag{3.44}$$

The MCMC scheme is identical to the one described in Section 3.2.6, except that $P(\mathcal{M}_i^h|\{\mathcal{M}_i^{\tilde{h}}\}_{\tilde{h}\neq h})$ has to be used as the prior on $\mathcal{M}_i^h$, which is obtained by inserting (3.44) into (3.39). In our study, we have set $\alpha_{ij} = \overline{\alpha_{ij}} = 1$, in which case $P(\theta_{ij})$ in (3.38) reduces to the uniform distribution over the unit interval. One can extend this scheme by imposing a hyperprior on α_{ij} and $\overline{\alpha_{ij}}$, and sampling these hyperparameters from the posterior distribution with MCMC – this is the subject of future work.

3.4 Simulation Study

The methods described in this chapter have been implemented in R, based on code from Lèbre et al. (2010). Our program sets up an RJMCMC simulation to sample the network structure, the changepoints and the hyperparameters from the posterior distribution. As a convergence diagnostic, we monitor the potential scale reduction factor (PSRF) (Gelman and Rubin, 1992), computed from the within-chain and between-chain variances of marginal edge posterior probabilities. Values of PSRF ≤ 1.1 are usually taken as indication of sufficient convergence. In our simulations, we extended the burn-in phase until a value of PSRF ≤ 1.05 was reached, and then sampled 1000 network and changepoint configurations in intervals of 200 RJMCMC steps. From these samples we compute the marginal posterior probabilities of all potential interactions, which defines a ranking of the edges in the recovered network. For the synthetic simulation study (see below), the gold standard (i.e. the true interaction network) is known. Therefore, by varying the threshold on the rank, we can construct the Receiver Operating Characteristic, or ROC curve (plotting the sensitivity or recall against the complementary specificity), and the precision-recall or PR curve (plotting the precision against the recall). To assess and succinctly score the network reconstruction accuracy, we follow a three-prong approach and compute three figures of merit that have been widely applied in the literature: the area under the ROC curve (AUROC), the area under the PR-curve (AUPRC), and the true positive rate at a fixed false positive rate of 5% (TPFP5).

3.5 Data

3.5.1 Synthetic data

We generated synthetic time series, each consisting of $K = 10$ segments of length 50, as follows. Random networks $\mathcal{M}^h$, $1 \leq h \leq K$, are generated stochastically, with the number of edges drawn from a Poisson distribution. Each directed edge from node j (the parent) to node i (the child) has a weight a_{ij}^h that determines the interaction strength, drawn from a Normal distribution. The signal associated with node i at time t, $y_i(t-1)$, evolves according to the non-homogeneous first-order Markov process of equation (3.1). Denote by $\mathbf{A}^h$ the matrix of all interaction strengths a_{ij}^h. To ensure stationarity of the time series, we tested if all eigenvalues of $\mathbf{A}^h$ had a modulus ≤ 1, and removed edges randomly until this condition was met.

The networks $\mathcal{M}^h$ that generated the time series consisted of 10 nodes, with on average 3 parents per node. To simulate a sequence of networks separated by changepoints, we sampled Δn_h from a Poisson distribution and then randomly changed Δn_h edges between $\mathcal{M}^h$ and $\mathcal{M}^{h+1}$, leaving the total number of edges unchanged. The parameter of the Poisson distribution, which determines the average number of changes between adjacent structures, $\mathcal{M}^h$ and $\mathcal{M}^{h+1}$, was varied, as described in more detail in Section 3.6.1.

3.5.2 Gene expression times course during morphogenesis in Drosophila

We also applied our method to the developmental gene expression time series for *Drosophila melanogaster* (fruit fly), obtained by Arbeitman et al. (2002). Expression values of 4028 genes were measured with microarrays at 67 time points during the Drosophila life cycle, which contains the four distinct phases of embryo, larva, pupa and adult. In our study we concentrated on a subset of 11 genes that regulate muscle development. This dataset has also been used in Guo et al. (2007), Zhao et al. (2006) and Robinson and Hartemink (2009).

3.5.3 Gene expression time courses measuring circadian rhythms in Arabidopsis

Plants assimilate carbon via photosynthesis during the day, but have a negative carbon balance at night. They buffer these daily alternations in their carbon budget by storing some of the assimilated carbon as starch in their leaves in the light, and utilising it as a carbon supply during the night. In order to synchronize these processes with the external 24 hour photo period, plants possess a circadian clock that can potentially provide predictive, temporal regulation of metabolic processes over the day/night cycle. The proper working of this circadian regulation is paramount to biomass production and growth, and considerable research efforts are therefore underway to elucidate its underlying molecular mechanism. We aim to reconstruct the regulatory network of nine circadian genes in the model plant *Arabidopsis thaliana*.

Our analysis is based on four independent gene expression profiling experiments described in Mockler et al. (2007), Edwards et al. (2006) and Grzegorczyk et al. (2008a). In these studies, wild-type Col-0 seedlings of *Arabidopsis thaliana* were grown for 7 days under artificially controlled light-dark cycles. On the 8th day the seedlings were placed in constant light. From these seedlings, RNA was extracted and assayed on Affymetrix GeneChip oligonucleotide arrays at regular time intervals. The data were background-corrected and normalised according to standard procedures, using the GeneSpring software (Agilent Technologies). The experiments were carried out at different laboratories and under different pre-experiment entrainment conditions and for different time intervals of measurements. An overview is provided in Table 3.1.

	Mockler et al.(2007)	Edwards et al. (2006)	Grzegorcyk et al. (2008) Data 1	Grzegorcyk et al. (2008) Data 2
Time points	12	13	13	13
Time point interval	4h	4h	2h	2h
Pretreatment entrainment	12h-light 12h-dark cycle	12h-light 12h-dark cycle	10h-light 10h-dark cycle	14h-light 14h-dark cycle
Measurement conditions	Constant light	Constant light	Constant light	Constant light
Laboratory	Kay Lab	Millar Lab	Millar Lab	Millar Lab

Table 3.1: Overview of the gene expression profiling experiments for *Arabidopsis thaliana*. Measurements were started after 7 days of growth of the seedlings and were repeated every 2 or 4 hours, depending on the dataset, for up to two days. Pretreatment entrainment specifies the light conditions before measurements were taken.

3.6 Results and Discussion

3.6.1 Experiments on simulated data

We compared the network reconstruction accuracy of three models: the time-varying DBN based on Lèbre et al. (2010) as described in Section 3.2 (TVDBN-0), the time-varying DBN with the sequential information sharing scheme proposed in Section 3.3.1 (TVDBN-SI), and the time-varying DBN with the global information sharing scheme proposed in Section 3.3.2 (TVDBN-GI). The methods were applied to the synthetic data described in Section 3.5.1. We repeated the simulations for each experimental setting on 10 independent data instantiations, and scored the network reconstruction accuracy with three separate measures, as discussed in Section 3.4. We investigated how the average number of changes in network structure between adjacent segments affects the performance.

Figure 3.2 shows boxplots of the score distributions. To test for significance of the discerned trends, we carried out paired t-tests. Table 3.2 shows paired t-tests between the AUROC scores of the different information sharing methods. We see that when varying the number of changes between segments, all differences in AUROC scores

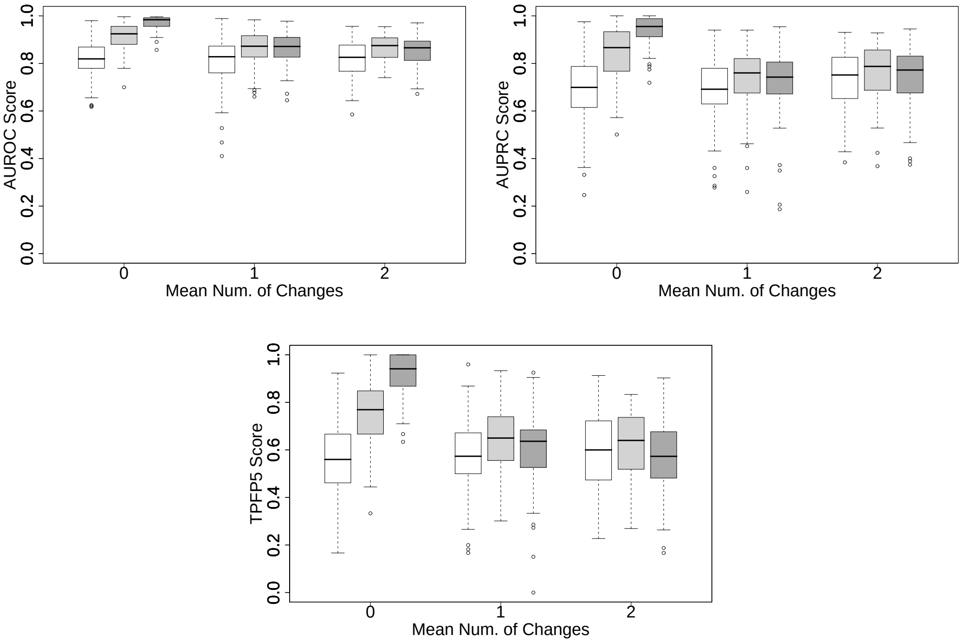


Figure 3.2: Network reconstruction accuracy, measured with three scoring schemes, as discussed in Section 3.4. Top left panel: AUROC; top right panel: AUPRC; bottom panel: TPFP5. The boxplots show the distributions of these scores, where the horizontal bar shows the median, the box margins show the 25th and 75th percentiles, the whiskers indicate data within 2 times the interquartile range, and circles are outliers. The grey shading indicates the method. Unshaded boxes: TVDBN-0. Light shading: TVDBN-SI. Dark shading: TVDBN-GI. The numbers on the horizontal axes indicate the average number of network structure changes per node between adjacent time series segments. A paired t-test showed that all differences are significant at the 5% level except for the following. AUROC: TVDBN-GI versus TVDBN-SI, 1 change; AUPRC: TVDBN-0 versus TVDBN-GI, 2 changes; TPFP5: TVDBN-0 versus TVDBN-GI, 1 and 2 changes.

are significant except for the difference between TVDBN-SI and TVDBN-GI when the number of changes is 1. For the AUPRC score, the results are similar, except that when the number of changes is 2, the difference between TVDBN-0 and TVDBN-GI is no longer significant. For the TPFP5 score, the difference between TVDBN-0 and TVDBN-GI is never significant when the number of changes is > 0. Note however, that a significant effect does not always denote an effect that is large.

Table 3.2: P-values from paired t-tests for AUROC scores of TVDBN-0, TVDBN-SI and TVDBN-GI for full DBN model when varying the mean number of changes between segments (TOP) AUROC Scores (MIDDLE) AUPRC Scores (BOTTOM) TPFP5 Scores.

(A)

CHANGE NUM	0	1	2
TVDBN-0 VS -SI	$< 1e-5$	$< 1e-5$	$< 1e-5$
TVDBN-SI VS -GI	$< 1e-5$	0.92	$< 1e-2$
TVDBN-GI VS -0	$< 1e-5$	$< 1e-5$	$< 1e-4$

(B)

CHANGE NUM	0	1	2
TVDBN-0 VS -SI	$< 1e-5$	$< 1e-4$	$< 1e-3$
TVDBN-SI VS -GI	$< 1e-5$	0.04	$< 1e-3$
TVDBN-GI VS -0	$< 1e-5$	$< 1e-3$	0.26

(C)

CHANGE NUM	0	1	2
TVDBN-0 VS -SI	$< 1e-5$	$< 1e-3$	0.04
TVDBN-SI VS -GI	$< 1e-5$	$< 1e-3$	$< 1e-3$
TVDBN-GI VS -0	$< 1e-5$	0.19	0.26

When there are no changes in the network structure, information sharing results in a considerable performance improvement, and TVDBN-GI outperforms TVDBN-SI. The latter finding is plausible, as TVDBN-GI utilizes information from all the segments, whereas TVDBN-SI only utilizes information from the adjacent segments. When the number of edge changes between segments increases, information sharing achieves a less substantial, yet still significant improvement over TVDBN-0. Also, the performance between the two approaches is inverted, with TVDBN-SI slightly yet significantly outperforming TVDBN-GI. Again, this result is plausible. Larger differences among network structures imply that, per se, less is gained from information sharing. Also, given a segment, a network associated with a remote segment will on average have accumulated a larger number of structure differences than a network associated with a close segment; this explains the superiority of the sequential (TVDBN-SI) over the global (TVDBN-GI) information sharing scheme.

To investigate the trend more thoroughly, we reduced the computational costs of the MCMC simulations by reducing the network complexity to 1 target node and 20 potential parents, and keeping the hyperparameters fixed. We then carried out simulations over an extended range of average structure differences. The resulting AUROC scores are shown in Figure 3.3. For small numbers of differences among the network structures associated with different segments, information sharing results in a considerable performance improvement over TVDBN-0. The amount of improvement degrades as the differences among structures increase. For small differences, TVDBN-GI tends to outperform TVDBN-SI. This trend is inverted when the difference among network structures increases. These results thus confirm the patterns found in Figure 3.2, which have been discussed above.

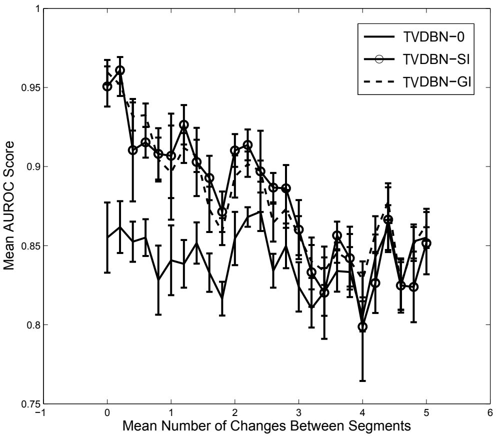


Figure 3.3: Network reconstruction accuracy for different methods. The plots show mean AUROC scores (vertical axis) plotted against the average number of network structure changes per node between adjacent time series segments (horizontal axis). Mean values and standard errors were obtained from 10 independent time series.

Some additional results for TVDBN-GI on simulation data generated with a slightly different simulation model can be found in Appendix B. These results are qualitatively

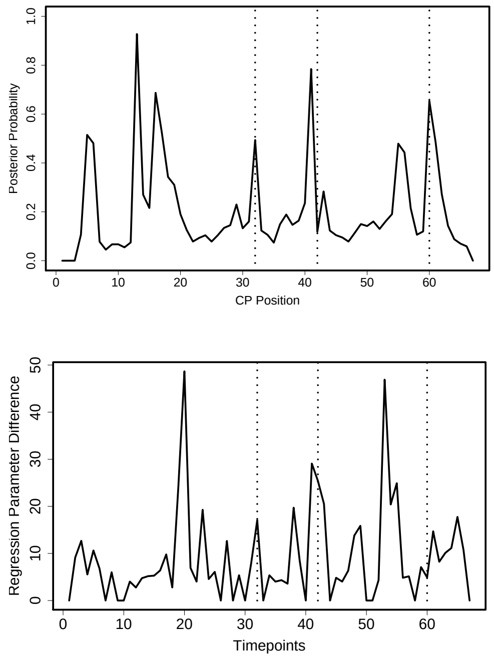


Figure 3.4: Changepoints during morphogenesis in *Drosophila melanogaster*. Top panel: TVDBN-SI, posterior probability of a changepoint occurring for any node at a given time (vertical axis) plotted against time (horizontal axis). Bottom panel: TESLA, L1-norm of the difference of the regression parameter vectors associated with two adjacent time points (vertical axis) plotted against time (horizontal axis). The vertical dotted lines indicate the three morphogenic transitions.

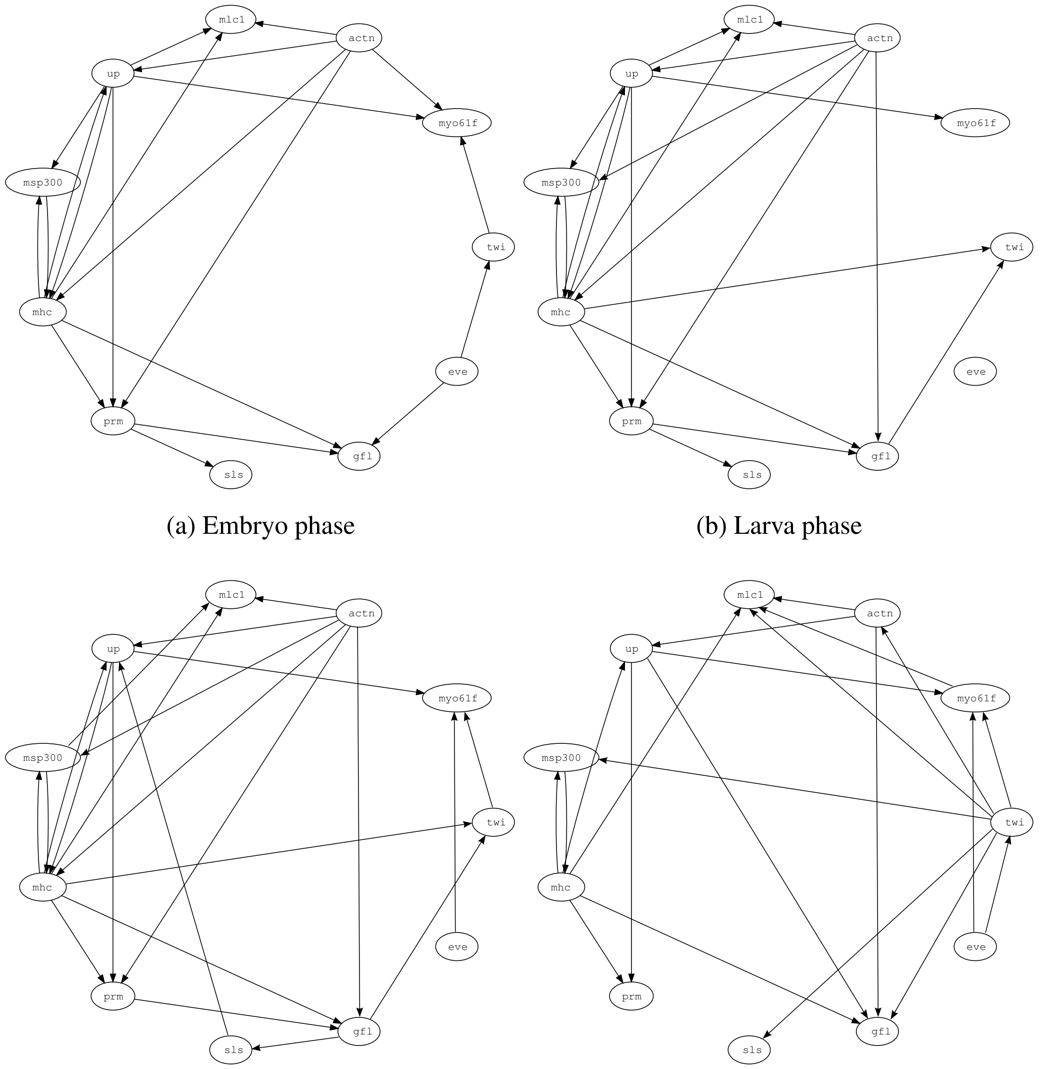
similar in that they demonstrate that using information sharing outperforms TVDBN-0.

3.6.2 Gene networks related to morphogenesis in the Drosophila life cycle

The top panel in Figure 3.4 shows the marginal posterior probability of changepoints during the life cycle of *Drosophila melanogaster*, inferred with TVDBN-SI from the gene expression time series described in Section 3.5.2. Since it is clear that developmental changes will happen sequentially, we have only considered TVDBN-SI in this section. For a comparison, we applied the method proposed in Ahmed and Xing (2009), using the authors' software package TESLA. Note that this model depends on various regularization parameters, which were optimized by maximizing the BIC score, as in Ahmed and Xing (2009). The results are shown in the bottom panel of Figure 3.4, where the graph shows the L1-norm of the difference of the regression parameter vectors associated with adjacent time points. Robinson and Hartemink (2009) applied their discrete time-varying DBN to the same data set, and a plot corresponding to the top panel of Figure 3.4 can be found in their paper. A comparison of these plots suggests that our method is the only one that clearly detects all three morphogenic transitions: embryo $\rightarrow$ larva, larva $\rightarrow$ pupa, and pupa $\rightarrow$ adult. The bottom panel of Figure 3.4 indicates that the last transition, pupa $\rightarrow$ adult, is less clearly detected with TESLA, and it is completely missing in Robinson and Hartemink (2009). Both our method, TVDBN-SI, as well as TESLA detect additional transitions during the embryo stage, which are missing in Robinson and Hartemink (2009). We would argue that a complex gene regulatory network is unlikely to transit into a new morphogenic phase all at once, and some pathways might have to undergo activational changes earlier in preparation for the morphogenic transition. As such, it is not implausible that additional transitions at the gene regulatory network level occur. However, a failure to detect known morphogenic transitions can clearly be seen as a shortcoming of a method, and on these grounds our model appears to outperform the two alternative ones.

In addition to the changepoints, we have inferred network structures for the morphogenic stages of embryo, larva, pupa and adult. We present a graphical representation of the networks recovered using the time-varying DBN model with sequential information sharing (TVDBN-SI). These networks have been constructed by discarding all edges with marginal posterior probability < 0.25. Due to the enforced sparsity introduced by our prior on the number of parents, a higher threshold would have led to overly sparse networks, which might miss the subtle distinctions between the phases. The recovered networks are presented in Figure 3.5.

An objective assessment of the reconstruction accuracy is not feasible due to the limited existing biological knowledge and the absence of a gold standard. Even if we had perfect knowledge of the regulatory network, the difficulties of detecting post-translational effects such as phosphorylation or ubiquitination of proteins would make a perfect reconstruction using only gene expression data very unlikely; we would require additional information about protein concentrations and state (e.g. phosphorylation). However, our reconstructed networks show many similarities with the networks discovered by Robinson and Hartemink (2009), Guo et al. (2007) and Zhao et al. (2006). For instance, we recover the interaction between two genes, *eve* and *twi*. This interaction is also reported in Guo et al. (2007) and Zhao et al. (2006), while Robinson and Hartemink (2009) seem to have missed it. We also recover a cluster



(a) Embryo phase

(b) Larva phase

(c) Pupa phase

(d) Adult phase

Figure 3.5: Recovered networks for each of the morphological phases in the development of *Drosophila melanogaster*, using the TVDBN-SI method.

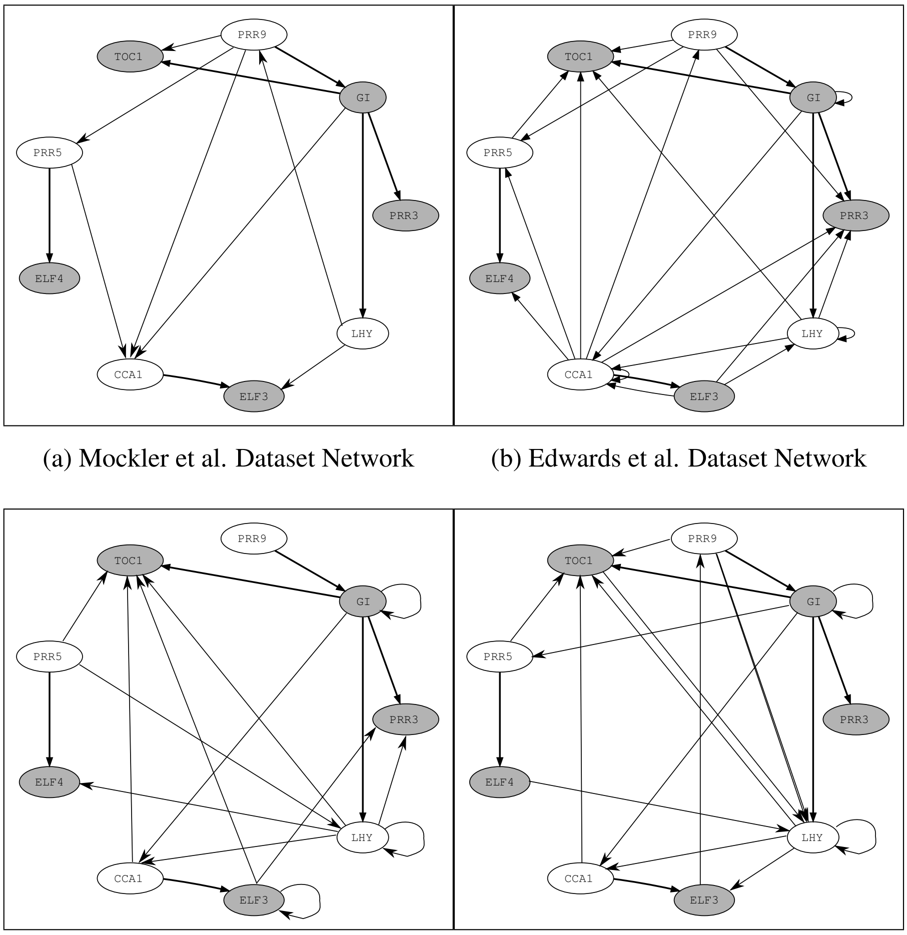
of interactions among the genes *myo61f*, *msp300*, *mhc*, *prm*, *mlc1* and *up* during all morphogenic phases. This result is not implausible, as all genes (except *up*) belong to the myosin family. However, unlike Robinson and Hartemink (2009), we find that *actn* also participates as a regulator in this cluster. There is some indication of this in Zhao et al. (2006), where *actn* is found to regulate *prm*. As far as changes between the different stages are concerned, we found an important change in the role of *twi*. This gene does not have an important role as a regulator during the early phases, but functions as a regulator of five other genes during the adult phase: *mlc1, gfl, actn, msp300* and *sls*. The absence of a regulatory role for *twi* during the earlier phases is consistent with Elgar et al. (2008), who found that another regulator, *mef2* (not included in the dataset) controls the expression of *mlc1*, *actn* and *msp300* during early development.

3.6.3 Circadian clock gene regulation network in *Arabidopsis thaliana*

We applied DBN network inference with our global information sharing method (TVDBN-GI) to the Arabidopsis data described in Section 3.5.3. In this setting, we treat each dataset as a network segment, which means that the changepoints are fixed and correspond to the boundaries between datasets. This is motivated by the observation that different experimental or growth conditions can often lead to the activation of different pathways, which introduces changes in the inferred networks. In this situation, global information sharing is appropriate because there is assumed to be a global network underlying the inferred networks. We compared TVDBN-GI to network inference without information sharing (TVDBN-0), where each network is only inferred from one dataset. Fig. 3.6 shows the results for TVDBN-0. To determine which interactions were relevant, we put a threshold on the marginal posterior probability of the interaction (the fraction of this interaction being present in the sampled networks).

We can observe a couple of effects of neglecting to use information sharing. First of all, the connectivity of the inferred networks varied widely between the four datasets. In fact, the network reconstructed from the Mockler et al. dataset had so few interactions with high posterior probability that we had to lower the threshold to obtain a network with a similar number of interactions compared to the other networks. Another effect is that some genes, such as *LHY*, vary from being regulated by just one or two genes (Fig. 3.6a-3.6c) to being regulated by no less than 5 genes (Fig. 3.6d).

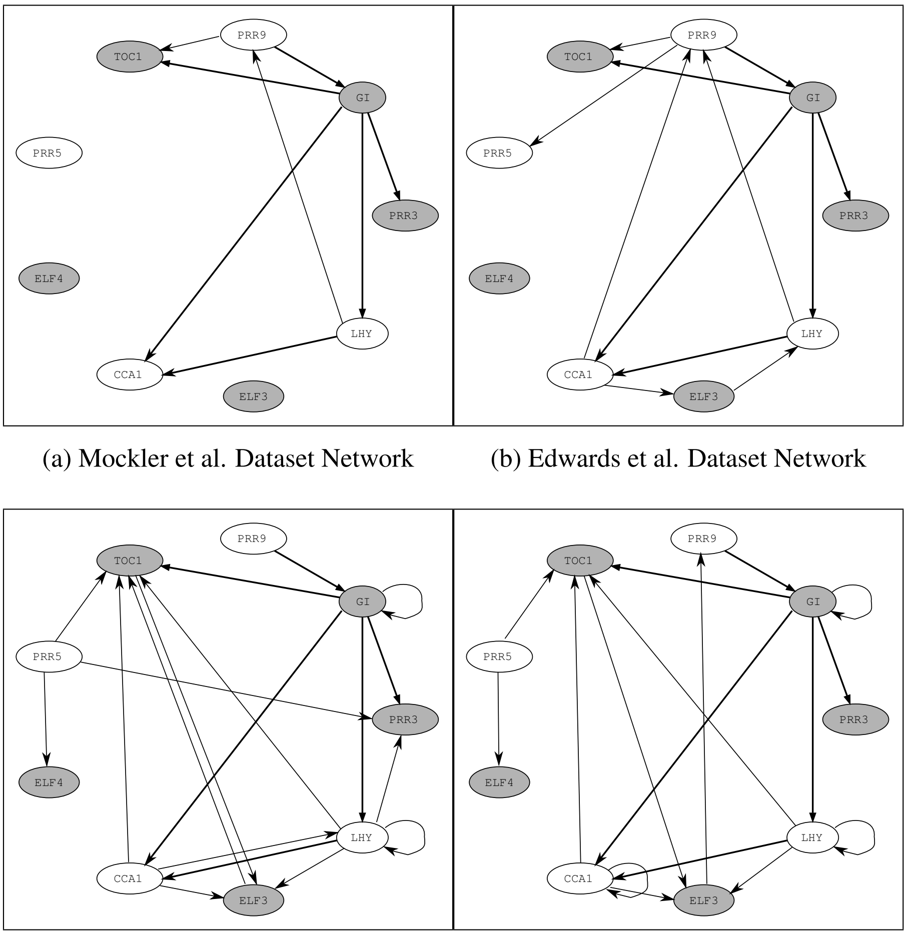
Fig. 3.7 shows the networks obtained when using the global information sharing



(a) Mockler et al. Dataset Network (b) Edwards et al. Dataset Network

(c) Grzegorczyk et al. Dataset 1 Network (d) Grzegorczyk et al. Dataset 2 Network

Figure 3.6: Networks reconstructed from the four datasets, without information sharing (TVDBN-0), though with common hyperparameters and initialisation. Only interactions that were present in more than 35% of the sampled networks have been selected (except for the Mockler et al. dataset, where the sampled networks were sparse, and we lowered the threshold to 25% of the sampled networks). Interactions that were found in all datasets are marked in bold.



(a) Mockler et al. Dataset Network (b) Edwards et al. Dataset Network

(c) Grzegorczyk et al. Dataset 1 Network (d) Grzegorczyk et al. Dataset 2 Network

Figure 3.7: Networks reconstructed from the four datasets, using the information sharing method described in Section 3.3.2 (TVDBN-GI). In all networks, only interactions that were present in more than 15% of the sampled networks have been selected. Interactions that were found in all datasets are marked in bold. The lower threshold compared to Fig. 3.6 can be explained by considering the information sharing as a penalisation factor. Even very strong edges will be penalised if they only occur in one or two of the four segments. This makes the procedure more selective and allows us to point out a restricted subset of interactions, i.e. interactions which are strong enough to be found in the data after penalisation.

approach (TVDBN-GI). The first thing to note is that the information sharing has a regularising effect on the network density, which allowed us to apply the same threshold to all four inferred networks. Overall, the sampled networks have fewer interactions, due to the penalising effect of the information sharing prior described in Section 3.3.2. This made it much easier to find an appropriate threshold on the posterior probability of the interactions.

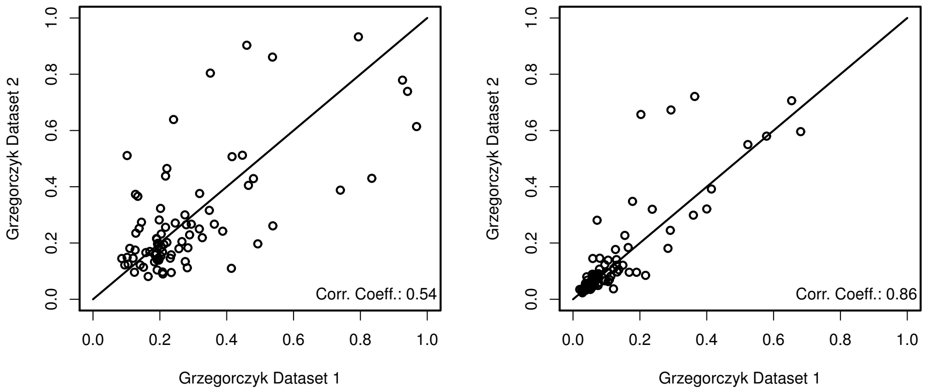
Compared to the networks in Fig. 3.6, we notice that there is less variation in the connectivity, although the first network is still sparser. Also, the variation in the number of regulators for *LHY* is no longer as drastic as in Fig. 3.6.

These networks reveal several gene interactions that can be found in the literature. For instance McClung (2006) shows *CCA1* and *LHY*, two genes that are active in the morning, as central regulators of genes that are active in the evening, such as *PRR9*, *TOC1* and *ELF3*. We recover these interactions in most (though not all) datasets. In addition, it seems that *LHY* regulates *CCA1*; this interaction was discovered consistently in all datasets using our information sharing method.

Conversely, some of the evening genes are known or suspected to activate the morning genes. We discovered consistent interactions which identified *GI* (an evening gene) as a regulator of *CCA1* and *LHY*. In addition, we also found that *GI* regulates *TOC1*, an interaction which seems likely given results in Locke et al. (2005). One interesting interaction that we found consistently was the regulation of *GI* by *PRR9*; McClung (2006) depicts *PRR9* as regulating *CCA1* and *LHY* directly, while our model seems to favour an indirect regulation via *GI*. Using the information sharing model helps us to identify these interactions more consistently, as comparing Fig. 3.6 and Fig. 3.7 shows: Although one can find all of the interactions listed above in at least one of the networks in Fig. 3.6, they are found much more consistently across networks in Fig. 3.7.

We can also investigate the effect of information sharing more directly, by comparing whether the similarity of the marginal posterior probabilities of the gene interactions inferred from different datasets increases when we introduce information sharing. Fig. 3.8 shows scatterplots comparing the posterior probabilities obtained from Grzegorczyk et al. Dataset 1 and Grzegorczyk et al. Dataset 2. Originally, the posterior probabilities are quite scattered, with a Spearman rank correlation[2] of only 0.54. Using information sharing, the rank correlation increases to 0.86. For comparisons between other pairs of datasets, the increase in rank correlation was even bigger, as Table 3.3

[2] Spearman rank correlation measures whether the order is similar, with values closer to 1 indicating probabilities that would produce the same ranking.



(a) No Information Sharing

(b) Information Sharing

Figure 3.8: Comparison of the marginal posterior probabilities of the gene interactions inferred from Grzegorczyk et al. Dataset 1 and Grzegorczyk et al. Dataset 2. (a) Without information sharing (TVDBN-0), (b) with global information sharing (TVDBN-GI). Correlation coefficients were calculated using Spearman rank correlation.

Table 3.3: Spearman rank correlations between the posterior probabilities for the gene interactions that were inferred for each dataset.

(A) No Information Sharing

Dataset	Mockler	Edwards	Grzegorczyk 1	Grzegorczyk 2
Mockler	1	0.42	0.39	0.39
Edwards	0.42	1	0.33	0.40
Grzegorczyk 1	0.39	0.33	1	0.54
Grzegorczyk 2	0.39	0.40	0.54	1

(B) Information Sharing

Dataset	Mockler	Edwards	Grzegorczyk 1	Grzegorczyk 2
Mockler	1	0.79	0.72	0.69
Edwards	0.79	1	0.74	0.73
Grzegorczyk 1	0.72	0.74	1	0.86
Grzegorczyk 2	0.69	0.73	0.86	1

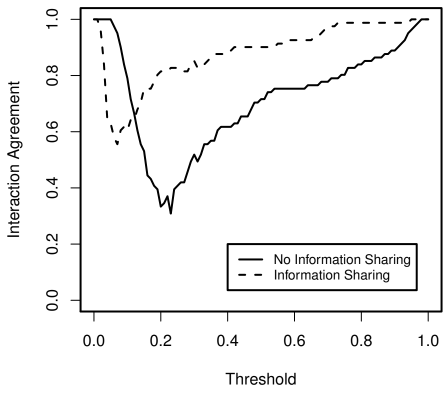


Figure 3.9: Agreement between the networks inferred from the four datasets, plotted as the fraction of coinciding interactions (including coinciding non-interactions) in all four networks as the threshold on the posterior probability of the edges increases from 0 to 1. The solid line shows the agreement without information sharing (TVDBN-0), and the dotted line shows agreement with global information sharing (TVDBN-GI).

shows.

Fig. 3.9 shows how the fraction of interactions and non-interactions that coincide in all four inferred networks changes as we increase the threshold on the posterior probabilities of the interactions from 0 to 1. For very low thresholds, all possible interactions will be included, so that the networks all coincide, while for high thresholds, no interactions will be included, again resulting in perfect agreement. The interesting part of the plot is the middle area, where we see that the agreement with information sharing increases much faster than the agreement without information sharing.

3.7 Conclusions

We have proposed a novel time-varying DBN, which has various advantages over existing schemes: it does not require the data to be discretized (as opposed to Robinson and Hartemink (2009)); it allows the network structure to change with time (as opposed to Grzegorczyk and Husmeier (2009)); it includes a regularization scheme based on inter-time segment information sharing (as opposed to Lèbre (2007) and Lèbre et al. (2010)); and it allows all hyperparameters to be inferred from the data via a consistent Bayesian inference scheme (as opposed to Ahmed and Xing (2009)).

An evaluation on synthetic data has demonstrated an improved performance over

Lèbre (2007) and Lèbre et al. (2010). We have carried out a comparison between two alternative paradigms of information sharing, global versus sequential. Our investigation has revealed that global information sharing will win out if there is a lot of commonality between all the network segments, but sequential information sharing will win out if the number of sequential changes is such that distant networks have little in common.

The application of our sequential information sharing method to gene expression time series taken during the life cycle of *Drosophila melanogaster* has revealed better agreement with known morphogenic transitions than the methods of Robinson and Hartemink (2009) and Ahmed and Xing (2009), and we have detected changes in gene regulatory interactions that are consistent with independent biological findings.

To further test our global information sharing method, we applied it to the problem of inferring a regulatory network of circadian clock genes in *Arabidopsis thaliana*. We showed that information sharing leads to greater agreement between networks inferred from datasets obtained under different experimental and growth conditions, and that the networks we obtained have good agreement with known facts about circadian clock gene regulation.

In the next chapter, I will investigate sequential information sharing in more detail, and present several alternative information sharing priors, with different treatment of inter-node coupling and allowing for different penalties for changing edges and non-edges in the network. I will also introduce a new RJMCMC sampling scheme, and show that it leads to improved sampling in the presence of information sharing.

Chapter 4

Time-varying networks with sequential information sharing

4.1 Introduction

In Chapter 3, I have described two paradigms for introducing information sharing when inferring networks with structure changes. The first is global information sharing, where differences between network segments are assumed to be the result of changes that have been applied to some unchanging underlying network, for example as a result of different experimental conditions or of using different cell lines. The second is sequential information sharing, where differences arise as a result of changes applied sequentially to the initial network, so that network segments that are temporally distant from each other are more dissimilar than those that are temporally close. This situation can arise during the development and growth of an organism, where we can observe that some genes and pathways show different levels of activity during different morphological or developmental stages. Another example where one would use sequential information sharing is to capture the reaction of gene regulatory networks to the application of a drug, or a change in growth environment.

In this chapter, I will concentrate on sequential information sharing, and will apply it to reconstruct two real-world gene regulatory networks: the network of genes involved in wing muscle development during the life cycle of *Drosophila melanogaster* described in Section 3.5.2, and an engineered network from synthetic biology, consisting of five genes in *Saccharomyces cerevisiae*.

In Chapter 3, I only looked at one form of sequential information sharing, the exponential prior (Section 3.3.1). I also assumed that there was no inter-segment coupling among nodes in the network; i.e. the hyperparameters controlling the strength of the regularising effect of information sharing were independent for each segment. In this chapter, I will present different functional forms of the prior (exponential versus binomial) and different versions of information coupling (hard versus soft), and compare their performance on simulated data. I will also present an improved sampling scheme. In the previous chapter, a standard Metropolis-Hastings-Green (RJMCMC) sampler was employed. In this chapter, I have identified several scenarios where this sampler is bound to fail, and I describe a new type of MCMC proposal move. I show that these moves avoid the convergence problems encountered with the original sampler, leading to a substantial improvement in mixing.

The Bayesian hierarchical models that I propose here depend on various hyperparameters. I have investigated the influence of the higher level hyperparameters by carrying out a set of simulation studies for the proposed models. To substantiate the findings, I have additionally presented a semi-analytical investigation for a simplified scenario, in which the computation of the marginal likelihood is tractable (see Section 4.5.2).

Shortly before the submission of the paper on which this chapter is based (Dondelinger et al. (2012b)), a somewhat related paper was published: Wang et al. (2011). While methodologically similar, there is an important difference in the application and inference, though. The objective of Wang et al. (2011) is online parameter estimation via particle filtering, with applications e.g. in tracking. This is a different scenario from most systems biology applications, where an interaction structure is typically learnt off-line after completion of a series of high-throughput experiments. Unlike Wang et al. (2011), our work thus follows other applications of DBNs in systems biology (Robinson and Hartemink, 2009, 2010; Grzegorczyk and Husmeier, 2009, 2011; Lèbre, 2007; Lèbre et al., 2012; Kolar et al., 2009) and aims to infer the model structure by marginalising out the parameters in closed form. Inference in Wang et al. (2011) is based on a filter, while inference in our work is based on a smoother.

The chapter is organised as follows. Section 4.2 describes the different information sharing priors and node-coupling strategies for our Bayesian regularisation scheme. Section 4.3 introduces the improved RJMCMC scheme based on multi-segment moves. Note that I will not recap the underlying time-varying dynamic Bayesian network model or the original RJMCMC scheme here. For these, see Chapter 3, Section 3.2. Section 4.5 discusses results obtained on synthetic data, with an investigation of the influence of the hyperparameters. Section 4.6 describes and interprets the two real-world

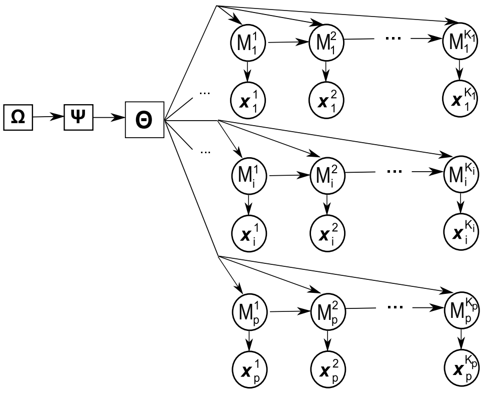


Figure 4.1: Hierarchical Bayesian model for inter-segment and hard inter-node information coupling. Hard coupling among nodes i is achieved by a common hyperparameter $\mathbf{\Theta}$ regulating the strength of the coupling between structures associated with adjacent segments, $\mathcal{M}_i^h$ and $\mathcal{M}_i^{h+1}$. This corresponds to the models in Section 4.2.2, with $\mathbf{\Theta} = \{\beta\}$, $\mathbf{\Psi} = [0,20]$, and no $\mathbf{\Omega}$, and Section 4.2.4, with $\mathbf{\Theta} = \{a,b\}$, $\mathbf{\Psi} = \{\alpha,\overline{\alpha},\gamma,\overline{\gamma}\}$, and $\mathbf{\Omega} = \{1,2,...,100\}$.

applications, related to morphogenesis in *Drosophila melanogaster* and synthetic biology in *Saccharomyces cerevisiae*. The chapter concludes in Section 4.7 with a general discussion and summary.

4.2 Sequential information coupling methods

Sequential information sharing over network structures makes sense when changes to the network take place gradually over the time period of the measurements. For instance, for the evolution of a gene regulatory network during embryogenesis, we would assume that the network evolves gradually and that networks associated with adjacent time intervals are a priori similar. In these kinds of situations, sequential information sharing provides a regularisation that reduces the potential over-flexibility of the model, and thus reduces inference uncertainty.

We propose four methods of information sharing among time series segments: two functional forms, and two types of information coupling among nodes. The first method is based on hard information coupling between the nodes, using the exponential distribution proposed in Werhli and Husmeier (2008) that has already been described in Section 3.3.1. The second scheme uses the same exponential distribution, but replaces the hard coupling by a soft information coupling scheme via a hy-

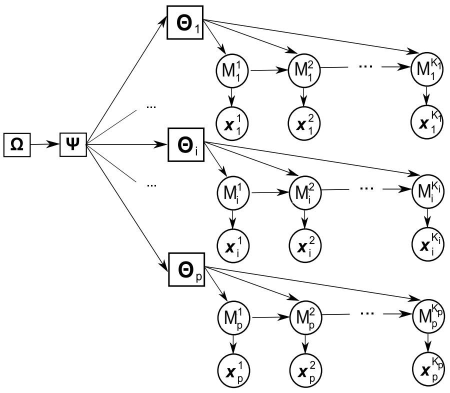


Figure 4.2: Hierarchical Bayesian model for inter-segment and soft inter-node information coupling. Soft coupling among nodes i is achieved by node-specific hyperparameters $\mathbf{\Theta}_i$ regulating the strength of the coupling between structures associated with adjacent segments, $\mathcal{M}_i^h$ and $\mathcal{M}_i^{h+1}$, coupled via level2-hyperparameters $\mathbf{\Psi}$. This corresponds to the model in Section 4.2.3, with $\mathbf{\Theta}_i = \{\beta_i\}$, $\mathbf{\Psi} = \kappa$, and $\mathbf{\Omega} = \lambda_\kappa = 10$, and Section 4.2.5, with $\mathbf{\Theta}_i = \{a_i, b_i\}$, $\mathbf{\Psi} = \{\alpha, \overline{\alpha}, \gamma, \overline{\gamma}\}$, and $\mathbf{\Omega} = \{1, 2, ..., 100\}$.

perprior.[1]. The third and fourth scheme are also based on hard and soft information coupling, respectively, but use a binomial distribution with a conjugate beta prior. The difference between hard and soft information coupling is illustrated in Figures 4.1 and 4.2, and is explained in more detail in the following subsection.

4.2.1 Hard versus soft information coupling of nodes

As noted above, we propose to share information about the network structure among the different time series segments that result from the changepoint process. The strength of these couplings is governed by the hyperparameters associated with the information sharing prior. We represent these hyperparameters collectively by $\mathbf{\Theta}$. However, another level of coupling is possible, coupling genes (nodes in the network) rather than time series segments.

Recall from Section 3.2 that each node in the network is associated with a random variable $X_i(t)$ that represents the gene expression level of gene i at time t. Under the regression model in equation (3.1), the regulators for gene i are independent of the structure of the rest of the network. Once we bring in information sharing, however, there is a set of hyperparameters that could conceivably be shared among different

[1] Note that these two schemes are very similar to the sequential information sharing scheme from Chapter 3. The difference lies in the fact that there was no information coupling among nodes in Chapter 3; the β_i were inferred independently for each node i.

nodes; namely $\mathbf{\Theta}$. We address this by proposing two different ways of sharing $\mathbf{\Theta}$: Hard coupling, where the information sharing prior has the same hyperparameters $\mathbf{\Theta}$ for all nodes (with hyperprior having level-2 hyperparameters $\mathbf{\Psi}$); and soft coupling, where the information sharing prior has node-specific hyperparameters $\mathbf{\Theta}_i$, with common level-2 hyperparameters $\mathbf{\Psi}$. In both cases we have a prior on $\mathbf{\Psi}$ with level-3 hyperparameters $\mathbf{\Omega}$. See Figures 4.1 and 4.2 for an illustration of hard versus soft information coupling of nodes.

In the following sub-sections, I will describe the different information sharing schemes in more detail.

4.2.2 Hard information coupling based on an exponential prior

Denote by $K_i := k_i + 1$ the total number of partitions in the time series associated with node i, and recall that each time series segment $\mathbf{D}_i^h$ is associated with a separate subnetwork $\mathcal{M}_i^h$, $1 \leq h \leq K_i$. We modify the prior from equation (3.6) by imposing a prior distribution $P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1}, \beta)$ on the structures, and the joint probability distribution factorizes according to a Markovian dependence:

$$\begin{aligned} P(\mathbf{D}_i^1, \ldots, \mathbf{D}_i^{K_i}, \mathcal{M}_i^1, \ldots, \mathcal{M}_i^{K_i}, \beta) \ = \ & P(\mathbf{D}_i^1|\mathcal{M}_i^1)P(\mathcal{M}_i^1)P(\beta) \\ & \prod_{h=2}^{K_i} P(\mathbf{D}_i^h|\mathcal{M}_i^h)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1}, \beta) \end{aligned} \tag{4.1}$$

Similar to Werhli and Husmeier (2008) we define

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1}, \beta) \ = \ \frac{\exp(-\beta|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|)}{Z(\beta, \mathcal{M}_i^{h-1})} \tag{4.2}$$

for $h \geq 2$, where β is a hyperparameter that defines the strength of the coupling between $\mathcal{M}_i^h$ and $\mathcal{M}_i^{h-1}$, and $|.|$ denotes the Hamming distance. For $h = 1$, $P(\mathcal{M}_i^h)$ is given by (3.6). The denominator $Z(\beta, \mathcal{M}_i^{h-1})$ in (4.2) is a normalizing constant, also known as the partition function: $Z(\beta, \mathcal{M}_i^{h-1}) = \sum_{\mathcal{M}_i^h \in \mathbb{M}} e^{-\beta|\mathcal{M}_i^h - \mathcal{M}_i^{h-1}|}$ where $\mathbb{M}$ is the set of all valid subnetwork structures. If we ignore any fan-in restriction that might have been imposed a priori (via $\bar{s}$ in equation (3.4)), then the expression for the partition function can be simplified: $Z(\beta, \mathcal{M}_i^{h-1}) \ \approx \ \prod_{j=1}^p Z_j(\beta, e_{ij}^{h-1})$, where e_{ij}^h is a binary variable indicating the presence or absence of a directed edge from node j to node i in time series segment h, and $Z_j(\beta, e_{ij}^{h-1}) = \sum_{e_{ij}^h=0}^1 e^{-\beta|e_{ij}^h - e_{ij}^{h-1}|} = 1 + e^{-\beta}$. Note that this expression no longer depends on $\mathcal{M}_i^{h-1}$, and hence

$$Z(\beta, \mathcal{M}_i^{h-1}) = Z(\beta) = \left(1 + e^{-\beta}\right)^p \tag{4.3}$$

Inserting this expression into (4.2) gives:

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta) \;=\; \frac{\exp(-\beta|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}{\left(1+e^{-\beta}\right)^p} \tag{4.4}$$

It is straightforward to integrate the proposed model into the RJMCMC scheme of Lèbre (2007) and Lèbre et al. (2010), which we have summarized in Section 3.2.6. When proposing a new network structure $\mathcal{M}_i^h \to \tilde{\mathcal{M}}_i^h$ for segment h, the prior probability ratio in equation (3.23) has to be replaced by $\frac{P(\mathcal{M}_i^{h+1}|\tilde{\mathcal{M}}_i^h,\beta)P(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^{h-1},\beta)}{P(\mathcal{M}_i^{h+1}|\mathcal{M}_i^h,\beta)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta)}$, leading to the acceptance probability

$$A(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) \;=\; \min\left\{\frac{P(\mathbf{D}_i^h|\tilde{\mathcal{M}}_i^h)P(\mathcal{M}_i^{h+1}|\tilde{\mathcal{M}}_i^h,\beta)P(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^{h-1},\beta)Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{P(\mathbf{D}_i^h|\mathcal{M}_i^h)P(\mathcal{M}_i^{h+1}|\mathcal{M}_i^h,\beta)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta)Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)},1\right\} \tag{4.5}$$

This equation is equivalent to equation (3.28), with the prior probabilities in equation (3.23) replaced by those in equation (4.4). Note that $P(\mathbf{D}_i^h|\mathcal{M}_i^h)$ is short for $P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)$ which is defined in equation (3.14) and the proposal ratio $\frac{Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)}$ is defined in equations (3.24-3.25). An additional MCMC step is introduced for sampling the hyperparameter β from the posterior distribution. For a proposal move $\beta \to \tilde{\beta}$ with symmetric proposal probability $Q(\tilde{\beta}|\beta) = Q(\beta|\tilde{\beta})$ we get the following acceptance probability:

$$A(\tilde{\beta}|\beta) \;=\; \min\left\{\frac{P(\tilde{\beta})}{P(\beta)}\prod_{i=1}^{p}\prod_{h=2}^{K_i}\frac{\exp(-\tilde{\beta}|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}{\exp(-\beta|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}\frac{\left(1+e^{-\beta}\right)^p}{\left(1+e^{-\tilde{\beta}}\right)^p},1\right\} \tag{4.6}$$

where in our study the hyperprior $P(\beta)$ was chosen as the uniform distribution on the interval $[0,20]$.

4.2.3 Soft information coupling based on an exponential prior

We modify the model defined in (4.1) by making the hyperparameter β, which defines the prior coupling strength between structures associated with adjacent segments, node-dependent: $\beta \to \beta_i$, and

$$P(\mathbf{D}_i^1,\dots,\mathbf{D}_i^{K_i},\mathcal{M}_i^1,\dots,\mathcal{M}_i^K,\beta_i) \;=\; P(\mathbf{D}_i^1|\mathcal{M}_i^1)P(\mathcal{M}_i^1)\prod_{h=2}^{K_i}P(\mathbf{D}_i^h|\mathcal{M}_i^h)$$
$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i)P(\beta_i) \tag{4.7}$$

with

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i) \;=\; \frac{\exp(-\beta_i|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}{Z(\beta_i,\mathcal{M}_i^{h-1})} \;=\; \frac{\exp(-\beta_i|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}{(1+e^{-\beta_i})^p} \tag{4.8}$$

where by analogy with the previous section, $Z(\beta_i, \mathcal{M}_i^{h-1}) \approx (1+e^{-\beta_i})^p$. To introduce soft information coupling between the subnetworks, we choose a hierarchical structure for the prior distribution on the hyperparameters β_i. At the first level, the hyperparameters are given a common gamma prior:

$$P(\beta_i) = P(\beta_i|\kappa,\rho) \;=\; \beta_i^{\kappa-1}\frac{\exp(-\beta_i/\rho)}{\rho^{\kappa}\Gamma(\kappa)} \tag{4.9}$$

with shape parameter $\kappa > 0$ and scale parameter $\rho > 0$. Recall that the gamma distribution has mean $\mu = \kappa\rho$ and variance $\sigma^2 = \kappa\rho^2$. We elect to set the scale parameter $\rho = 0.1$ fixed. The shape parameter κ is given a vague exponential prior:

$$P(\kappa|\lambda_\kappa) \;=\; \lambda_\kappa \exp(-\kappa/\lambda_\kappa) \tag{4.10}$$

with $\lambda_\kappa = 10$ to reflect our prior ignorance. This choice of prior has the following motivation. The coupling strength between the substructures is defined by the coefficient of variation $\sigma/\mu = 1/\sqrt{\kappa}$, with smaller coefficients corresponding to stronger coupling strengths, and a zero coefficient ($\kappa \to \infty$) reducing to the hard coupling scheme discussed in the previous section. By inferring the shape parameter κ from the data, starting from a vague yet proper prior distribution, we determine if the coupling strength should be strong or weak.

It is straightforward to adapt the RJMCMC scheme of the previous section. When proposing a new network structure $\mathcal{M}_i^h \to \tilde{\mathcal{M}}_i^h$ for segment h, the prior probability ratio in equation (3.23) has to be replaced by the ratio $\frac{P(\mathcal{M}_i^{h+1}|\tilde{\mathcal{M}}_i^h,\beta_i)P(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^{h-1},\beta_i)}{P(\mathcal{M}_i^{h+1}|\mathcal{M}_i^h,\beta_i)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i)}$, leading to the equivalent of the acceptance probability in equation (3.28):

$$A(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) \;=\; \min\left\{\frac{P(\mathbf{D}_i^h|\tilde{\mathcal{M}}_i^h)P(\mathcal{M}_i^{h+1}|\tilde{\mathcal{M}}_i^h,\beta_i)P(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^{h-1},\beta_i)Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{P(\mathbf{D}_i^h|\mathcal{M}_i^h)P(\mathcal{M}_i^{h+1}|\mathcal{M}_i^h,\beta_i)P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},\beta_i)Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)},1\right\} \tag{4.11}$$

Note that $P(\mathbf{D}_i^h|\mathcal{M}_i^h)$ is short for $P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)$ which is defined in equation (3.14) and the proposal ratio $\frac{Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)}$ defined in equations (3.24-3.25). When proposing new hyperparameters $\tilde{\beta}_i$ from a symmetric proposal distribution $Q(\tilde{\beta}_i|\beta_i) = Q(\beta_i|\tilde{\beta}_i)$ we get the following acceptance probability:

$$A(\tilde{\beta}_i|\beta_i) \;=\; \min\left\{\frac{P(\tilde{\beta}_i|\rho,\kappa)}{P(\beta_i|\rho,\kappa)}\prod_{h=2}^{K_i}\frac{\exp(-\tilde{\beta}_i|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}{\exp(-\beta_i|\mathcal{M}_i^h-\mathcal{M}_i^{h-1}|)}\left(\frac{1+e^{-\beta_i}}{1+e^{-\tilde{\beta}_i}}\right)^p,1\right\} \tag{4.12}$$

An additional sampling step is needed for the shape parameter κ of the level-2 hyperprior. Drawing a new shape parameter $\tilde{\kappa}$ from a symmetric proposal distribution

$Q(\tilde{\kappa}|\kappa)$, the acceptance probability is given by

$$A(\tilde{\kappa}|\kappa) \; = \; \min\left\{\frac{\exp(-\tilde{\kappa}/\lambda_\kappa)}{\exp(-\kappa/\lambda_\kappa)}\prod_{i=1}^{p}\frac{P(\beta_i|\tilde{\kappa},\rho)}{P(\beta_i|\kappa,\rho)},1\right\} \tag{4.13}$$

4.2.4 Hard information coupling based on a binomial prior

An alternative way of information sharing among segments and nodes is by using a binomial prior:

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},a,b) \; = \; a^{N_1^1[h,i]}(1-a)^{N_1^0[h,i]}b^{N_0^0[h,i]}(1-b)^{N_0^1[h,i]} \tag{4.14}$$

where we have defined the following sufficient statistics: $N_1^1[h,i]$ is the number of edges in $\mathcal{M}_i^{h-1}$ that are matched by an edge in $\mathcal{M}_i^h$, $N_1^0[h,i]$ is the number of edges in $\mathcal{M}_i^{h-1}$ for which there is no edge in $\mathcal{M}_i^h$, $N_0^1[h,i]$ is the number of edges in $\mathcal{M}_i^h$ for which there is no edge in $\mathcal{M}_i^{h-1}$, and $N_0^0[h,i]$ is the number of coinciding non-edges in $\mathcal{M}_i^{h-1}$ and $\mathcal{M}_i^h$. Since the hyperparameters are shared, the joint distribution can be expressed as:

$$\begin{aligned} P(\{\mathcal{M}_i^h\}|a,b) \; &= \; \prod_{i=1}^{p}P(\mathcal{M}_i^1)\prod_{h=2}^{K_i}P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},a,b) \\ &= \; a^{N_1^1}(1-a)^{N_1^0}b^{N_0^0}(1-b)^{N_0^1}\prod_{i=1}^{p}P(\mathcal{M}_i^1) \end{aligned} \tag{4.15}$$

where we have defined $N_k^l = \sum_{i=1}^{p}\sum_{h=2}^{K_i}N_k^l[h,i]$, and the right-hand side follows from Eq. (4.14). The conjugate prior for the hyperparameters a,b is a beta distribution,

$$P(a,b|\alpha,\bar{\alpha},\gamma,\bar{\gamma}) \propto a^{(\alpha-1)}(1-a)^{(\bar{\alpha}-1)}b^{(\gamma-1)}(1-b)^{(\bar{\gamma}-1)} \tag{4.16}$$

which using Bayes' rule leads to the (beta) posterior distribution:

$$P(a,b|\alpha,\bar{\alpha},\gamma,\bar{\gamma},\{\mathcal{M}_i^h\}) \propto a^{(\alpha+N_1^1-1)}(1-a)^{(\bar{\alpha}+N_1^0-1)}b^{(\gamma+N_0^0-1)}(1-b)^{(\bar{\gamma}+N_0^1-1)} \tag{4.17}$$

This allows the hyperparameters to be integrated out in closed form:

$$\begin{aligned} P(\{\mathcal{M}_i^h\}|\alpha,\bar{\alpha},\gamma,\bar{\gamma}) &= \int\int P(\{\mathcal{M}_i^h\}|a,b)P(a,b|\alpha,\bar{\alpha},\gamma,\bar{\gamma})dadb \\ &\propto \frac{\Gamma(\alpha+\bar{\alpha})}{\Gamma(\alpha)\Gamma(\bar{\alpha})}\frac{\Gamma(N_1^1+\alpha)\Gamma(N_1^0+\bar{\alpha})}{\Gamma(N_1^1+\alpha+N_1^0+\bar{\alpha})}\frac{\Gamma(\gamma+\bar{\gamma})}{\Gamma(\gamma)\Gamma(\bar{\gamma})}\frac{\Gamma(N_0^0+\gamma)\Gamma(N_0^1+\bar{\gamma})}{\Gamma(N_0^0+\gamma+N_0^1+\bar{\gamma})} \end{aligned} \tag{4.18}$$

The level-2 hyperparameters $\alpha,\bar{\alpha},\gamma,\bar{\gamma}$, which can be interpreted as fictitious prior observations due to the conjugacy of the prior, are given a discrete uniform hyperprior over $\{1,2,...,100\}$. The MCMC scheme of Section 3.2.6 has to be modified as follows. When proposing a new network structure for node i and segment h, $\mathcal{M}_i^h \to \tilde{\mathcal{M}}_i^h$, the structures $\mathcal{M}_i^h$ and $\tilde{\mathcal{M}}_i^h$ enter the prior probability ratio in equation (3.23) via the expression $P(\{\mathcal{M}_i^h\}|\alpha,\bar{\alpha},\gamma,\bar{\gamma})$. The prior probability ratio becomes:

$$\frac{P(\{\mathcal{M}_i^1,\ldots,\tilde{\mathcal{M}}_i^h,\ldots,\mathcal{M}_i^{K_i}\}_{i=1}^p|\alpha,\bar{\alpha},\gamma,\bar{\gamma})}{P(\{\mathcal{M}_i^1,\ldots,\mathcal{M}_i^h,\ldots,\mathcal{M}_i^{K_i}\}_{i=1}^p|\alpha,\bar{\alpha},\gamma,\bar{\gamma})}$$

leading to the acceptance probability:

$$\begin{aligned} A(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) &= \\ &\min\left\{\frac{P(\mathbf{D}_i^h|\tilde{\mathcal{M}}_i^h)P(\{\mathcal{M}_i^1,\ldots,\tilde{\mathcal{M}}_i^h,\ldots,\mathcal{M}_i^{K_i}\}_{i=1}^p|\alpha,\bar{\alpha},\gamma,\bar{\gamma})Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{P(\mathbf{D}_i^h|\mathcal{M}_i^h)P(\{\mathcal{M}_i^1,\ldots,\mathcal{M}_i^h,\ldots,\mathcal{M}_i^{K_i}\}_{i=1}^p|\alpha,\bar{\alpha},\gamma,\bar{\gamma})Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)},1\right\} \end{aligned} \tag{4.19}$$

This equation is equivalent to equation (3.28), with the prior probabilities in equation (3.23) replaced by those in equation (4.18). Note that $P(\mathbf{D}_i^h|\mathcal{M}_i^h)$ is short for $P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)$ which is defined in equation (3.14) and the proposal ratio $\frac{Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)}$ defined in equations (3.24-3.25). From Figure 4.1, it becomes clear that as a consequence of integrating out the hyperparameters, all network structures become interdependent, and information about the structures is contained in the sufficient statistics N_1^1,N_1^0,N_0^1,N_0^0. A new proposal move for the level-2 hyperparameters is added to the existing RJMCMC scheme of Section 3.2.6. New values for the level-2 hyperparameters α are proposed from a uniform distribution over the support of $P(\alpha)$. For a move $\alpha \to \tilde{\alpha}$, the acceptance probability is:

$$A(\tilde{\alpha}|\alpha) = \min\left\{\frac{P(\{\mathcal{M}_i^1,\ldots,\mathcal{M}_i^{K_i}\}_{i=1}^p|\tilde{\alpha},\bar{\alpha},\gamma,\bar{\gamma})}{P(\{\mathcal{M}_i^1,\ldots,\mathcal{M}_i^{K_i}\}_{i=1}^p|\alpha,\bar{\alpha},\gamma,\bar{\gamma})},1\right\} \tag{4.20}$$

and similarly for $\bar{\alpha}$, γ and $\bar{\gamma}$.

4.2.5 Soft information coupling based on a binomial prior

We can relax the information sharing scheme from a hard to a soft coupling by introducing node-specific hyperparameters a_i,b_i that are softly coupled via a common level-2 hyperprior, $P(a_i,b_i|\alpha,\bar{\alpha},\gamma,\bar{\gamma}) \propto a_i^{(\alpha-1)}(1-a_i)^{(\bar{\alpha}-1)}b_i^{(\gamma-1)}(1-b_i)^{(\bar{\gamma}-1)}$ as illustrated in Figure 4.2:

$$P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1},a_i,b_i) = (a_i)^{N_1^1[h,i]}(1-a_i)^{N_1^0[h,i]}(b_i)^{N_0^0[h,i]}(1-b_i)^{N_0^1[h,i]} \tag{4.21}$$

This leads to a straightforward modification of equation (4.15) – replacing a,b by a_i,b_i – from which we get as an equivalent to (4.18), using the definition $N_k^l[i] =$

$\sum_{h=2}^{K_i} N_k^l[h,i]$:

$$\begin{aligned} P(\mathcal{M}_i^1,\ldots,\mathcal{M}_i^{K_i}|\alpha,\overline{\alpha},\gamma,\overline{\gamma}) \quad \propto \quad & \frac{\Gamma(\alpha+\overline{\alpha})}{\Gamma(\alpha)\Gamma(\overline{\alpha})} \frac{\Gamma(N_1^1[i]+\alpha)\Gamma(N_1^0[i]+\overline{\alpha})}{\Gamma(N_1^1[i]+\alpha+N_1^0[i]+\overline{\alpha})} \\ & \frac{\Gamma(\gamma+\overline{\gamma})}{\Gamma(\gamma)\Gamma(\overline{\gamma})} \frac{\Gamma(N_0^0[i]+\gamma)\Gamma(N_0^1[i]+\overline{\gamma})}{\Gamma(N_0^0[i]+\gamma+N_0^1[i]+\overline{\gamma})} \end{aligned} \tag{4.22}$$

As in Section 4.2.4, we extend the RJMCMC scheme from Section 3.2.6 so that when proposing a new network structure, $\mathcal{M}_i^h \to \tilde{\mathcal{M}}_i^h$, the prior probability ratio in equation (3.23) has to be replaced by: $\frac{P(\mathcal{M}_i^1,\ldots,\tilde{\mathcal{M}}_i^h,\ldots,\mathcal{M}_i^{K_i}|\alpha,\overline{\alpha},\gamma,\overline{\gamma})}{P(\mathcal{M}_i^1,\ldots,\mathcal{M}_i^h,\ldots,\mathcal{M}_i^{K_i}|\alpha,\overline{\alpha},\gamma,\overline{\gamma})}$, leading to the equivalent of the acceptance probability in equation (3.28):

$$A(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) \;=\; \min\left\{\frac{P(\mathbf{D}_i^h|\tilde{\mathcal{M}}_i^h)P(\mathcal{M}_i^1,\ldots,\tilde{\mathcal{M}}_i^h,\ldots,\mathcal{M}_i^{K_i}|\alpha,\overline{\alpha},\gamma,\overline{\gamma})\,Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{P(\mathbf{D}_i^h|\mathcal{M}_i^h)P(\mathcal{M}_i^1,\ldots,\mathcal{M}_i^h,\ldots,\mathcal{M}_i^{K_i}|\alpha,\overline{\alpha},\gamma,\overline{\gamma})\,Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)},1\right\} \tag{4.23}$$

Note that $P(\mathbf{D}_i^h|\mathcal{M}_i^h)$ is short for $P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)$ which is defined in equation (3.14) and the proposal ratio $\frac{Q(\mathcal{M}_i^h|\tilde{\mathcal{M}}_i^h)}{Q(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h)}$ defined in equations (3.24-3.25). In addition, we have to add a new level-2 hyperparameter update move: when proposing a level-2 hyperparameter $\alpha \to \tilde{\alpha}$, where the prior and proposal probabilities are the same as in Section 4.2.4, the acceptance probability becomes:

$$A(\tilde{\alpha}|\alpha) = \min\left\{\prod_{i=1}^{p}\frac{P(\mathcal{M}_i^1,\ldots,\mathcal{M}_i^{K_i}|\tilde{\alpha},\overline{\alpha},\gamma,\overline{\gamma})}{P(\mathcal{M}_i^1,\ldots,\mathcal{M}_i^{K_i}|\alpha,\overline{\alpha},\gamma,\overline{\gamma})},1\right\} \tag{4.24}$$

and similarly for $\overline{\alpha}$, γ and $\overline{\gamma}$.

4.3 Improved MCMC scheme

The various information sharing priors that I have introduced in the previous section (4.2.2, 4.2.3, 4.2.4, 4.2.5) share the characteristic that they encourage the networks of all segments to be similar to each other[2]. When applying the MCMC scheme from Lèbre et al. (2010), summarized in Section 3.2.6 and adapted to our prior as discussed above, this can lead to the following curious effect. On simulated data where the network structure is the same for all segments, we found that the network reconstruction accuracy deteriorated when we increased the coupling strength between the structures. The results will be presented below, in Section 4.5 and Figure 4.4. These findings appear counter-intuitive, given that increasing the coupling strength brings the prior more

[2] Note that the binomial information sharing prior (Section 4.2.4 and 4.2.5) can in principle encourage either similarity or dissimilarity depending on the hyperparameters a and b. As discussed in Section 4.5, we had originally envisaged setting the level-2 hyperparameters $\overline{\alpha}$ and $\overline{\gamma}$ equal to 1 to enforce similarity, but Figure 4.8 demonstrates that this constraint is too restrictive.

in line with the truth (the perfect prior would have infinitely strong coupling). However, it is easily seen that increasing the coupling strength adversely affects the mixing of the Markov chains. Consider a set of identical network structures which, at an initial stage of the MCMC simulations, are all poor at explaining the data. We now visit a segment and propose a modification of the network structure associated with it. This modification introduces a mismatch between the structures and is, hence, discouraged by the prior. For strong coupling this discouragement might outweigh the gain in the likelihood that would result from a better structure. The structures thus remain identical, which in turn will tend to increase the coupling strength. The MCMC simulation thus gets trapped in a suboptimal state of the configuration space (local optimum).

To deal with this problem, we have implemented an alternative MCMC scheme where changes are applied to multiple segments. The new moves will propose changes to the network structure in more than one segment, and we will hence refer to them as multi-segment moves. Note that the moves for proposing new changepoint configurations are unaffected by these modifications. The multi-segment moves are presented as target-node specific (i.e. they presuppose a choice of target node i). However, they can be generalized for inference over the whole network by simply picking a target node at random.

Given a node, the proposal move consists of two steps: (1) Pick one of p possible parents for the target node i. (2) For each segment h of the K_i segments, flip the edge status (changing an edge to a non-edge or vice-versa) between the parent node and the target node with probability q. In our simulations, we set $q = \frac{1}{2}$ so that flipping the edge status and conserving it are equally likely outcomes. It is straightforward to adapt this parameter during the burn-in phase. This means that the probability of proposing a new set of structures $\tilde{\mathcal{M}}_i$ given the set of network structures $\mathcal{M}_i$ using the multi-segment move is:

$$Q(\tilde{\mathcal{M}}_i|\mathcal{M}_i) = \frac{1}{p2^{K_i}} \tag{4.25}$$

where $\mathcal{M}_i = \{\mathcal{M}_i^h\}_{1\leq h\leq K_i}$ as before.

We now derive the acceptance ratio for multi-segment moves. We define the quantity $R_{prior}(\tilde{\mathcal{M}}_i|\mathcal{M}_i)$ to be the ratio of the prior probabilities of the original set $\mathcal{M}_i$ and the proposed set $\tilde{\mathcal{M}}_i$. Let $R_{likelihood}(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) = \frac{P(\boldsymbol{x}_i^h|\tilde{\mathcal{M}}_i^h,\delta^2)}{P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)}$ be the likelihood ratio of the original and proposed network structures for segment h and target node i, where the likelihood $P(\boldsymbol{x}_i^h|\mathcal{M}_i^h,\delta^2)$ is defined in equation (3.14) of Section 3.2.6. Note that the changes introduced by multi-segment moves are equivalent to a sequence of add and remove edge moves applied to individual segments, so that this ratio remains unchanged. Then the acceptance ratio for multi-segment moves can be expressed as:

$$R(\tilde{\mathcal{M}}_i|\mathcal{M}_i) = R_{prior}(\tilde{\mathcal{M}}_i|\mathcal{M}_i)R_{proposal}(\tilde{\mathcal{M}}_i|\mathcal{M}_i)\prod_{h=1}^{K_i} R_{likelihood}(\tilde{\mathcal{M}}_i^h|\mathcal{M}_i^h) \tag{4.26}$$

where $R_{prior}(\tilde{\mathcal{M}}_i|\mathcal{M}_i) = \frac{P(\tilde{\mathcal{M}}_i)}{P(\mathcal{M}_i)}$. The form of $P(\mathcal{M}_i)$ depends on our choice of prior. If segments are independent, then $P(\mathcal{M}_i) = \prod_{h=1}^{K_i} P(\mathcal{M}_i^h)$, where $P(\mathcal{M}_i^h)$ is the prior from equation (3.6), with a Poisson distribution on the number of parents. If we want to use information sharing between segments, then the prior for segment h depends on segment $h-1$, so that $P(\mathcal{M}_i) = P(\mathcal{M}_i^1)\prod_{h=2}^{K_i} P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1})$, where $P(\mathcal{M}_i^h|\mathcal{M}_i^{h-1})$ could be any of the information sharing priors introduced in Section 4.2. Finally, $R_{proposal}(\tilde{\mathcal{M}}_i|\mathcal{M}_i)$ is the Hastings ratio:

$$R_{proposal}(\tilde{\mathcal{M}}_i|\mathcal{M}_i) = \frac{Q(\mathcal{M}_i|\tilde{\mathcal{M}}_i)}{Q(\tilde{\mathcal{M}}_i|\mathcal{M}_i)} \tag{4.27}$$

where $Q(\tilde{\mathcal{M}}_i|\mathcal{M}_i)$ is defined in equation (4.25). Since the proposal probability $Q(\tilde{\mathcal{M}}_i|\mathcal{M}_i)$ is independent of the set of network structures $\mathcal{M}_i$, the multi-segment moves are symmetric, and we obtain that $R_{proposal}(\tilde{\mathcal{M}}_i|\mathcal{M}_i) = 1$.

We have explored an alternative proposal scheme consisting of two moves: (1) a move proposing network structures where an edge has been set identical in all segments, and (2) the move described above, which corresponds to a random perturbation of an edge. However, we found that including the first kind of proposal move adversely affected mixing and convergence in simulations where the true network structure presented differences among segments. These network structures are less likely to be proposed when both moves are included.

4.4 Implementation and Simulations

We have implemented our model in R, based on code from Lèbre (2007) and Lèbre et al. (2010). The network structure, the changepoints and the hyperparameters are sampled from the posterior distribution using RJMCMC as described in Sections 3.2.6 and 4.3. We ran the MCMC chains until we were satisfied that convergence was reached. Then we sampled 1000 network and changepoint configurations in intervals of 200 RJMCMC steps. By marginalization and under the assumption of convergence, this represents a sample from the posterior distribution in equation (3.12). By further marginalization, we get the posterior probabilities of all gene regulatory interactions, which defines a ranking of the interactions in terms of posterior confidence. We use the potential scale reduction factor (PSRF) (Gelman and Rubin, 1992), computed from the within-chain and between-chain variances of marginal edge posterior probabilities, as a convergence diagnostic. The usual threshold for sufficient convergence lies at PSRF ≤ 1.1. In our simulations, we extended the burn-in phase until a value of PSRF ≤ 1.05 was reached.

Subsequent to the publication of the paper on which this chapter is based, I further developed the R code and released it on the Comprehensive R Archive Network (CRAN). The resulting software package EDISON (Estimation of Directed Interactions from Sequences Of Nonhomogeneous gene expression) is described in Appendix F. Software and documentation can be found at `http://cran.r-project.org/web/packages/EDISON/`.

For the study on simulated data, and the synthetic biology data, the true interaction network is known. Therefore, varying the threshold on this ranking allows us to construct the Receiver Operating Characteristic (ROC) curve (plotting the sensitivity or recall[3] against the complementary specificity[4]) and the precision-recall (PR) curve (plotting the precision[5] against the recall), and to assess the network reconstruction accuracy in terms of the areas under these graphs (AUROC and AUPRC, respectively); see Davis and Goadrich (2006). These two measures are widely used in the systems biology literature to quantify the overall network reconstruction accuracy (Prill et al., 2010), with larger values indicating a better prediction performance overall.

4.5 Evaluation on simulated data

4.5.1 Comparative evaluation of network reconstruction and hyperparameter inference

The purpose of the simulation study is two-fold. Firstly, we want to carry out a comparative evaluation of the proposed Bayesian regularization schemes for a controlled scenario in which the true network structure is known. Secondly, we want to assess the Bayesian inference scheme and test the viability of the proposed MCMC samplers. To focus on the task of network reconstruction, we keep the changepoints fixed at their true values. The inference of the changepoints will be investigated later, on the real gene expression time series (see Figure 4.12).

[3] The *sensitivity* or *recall* denotes the fraction of true interaction that have been recovered.

[4] The *specificity* denotes the fraction of spurious interactions that have been successfully avoided.

[5] The *precision* is the fraction of predicted interactions that are correct.

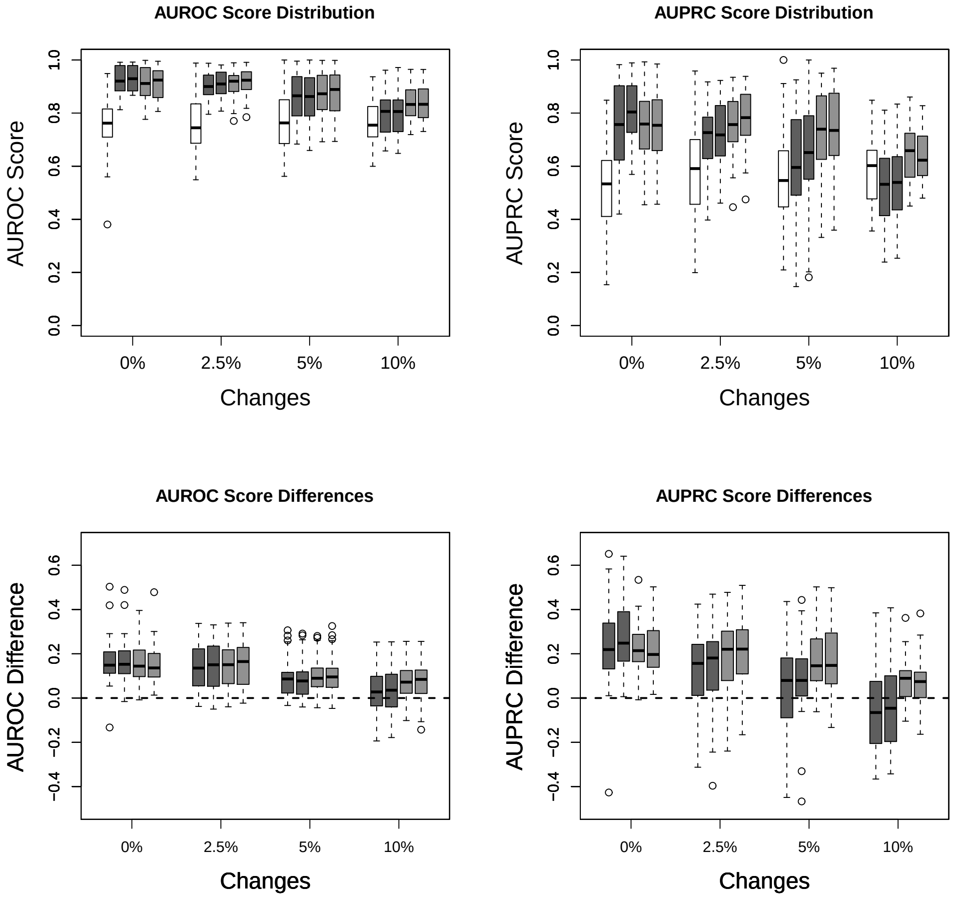
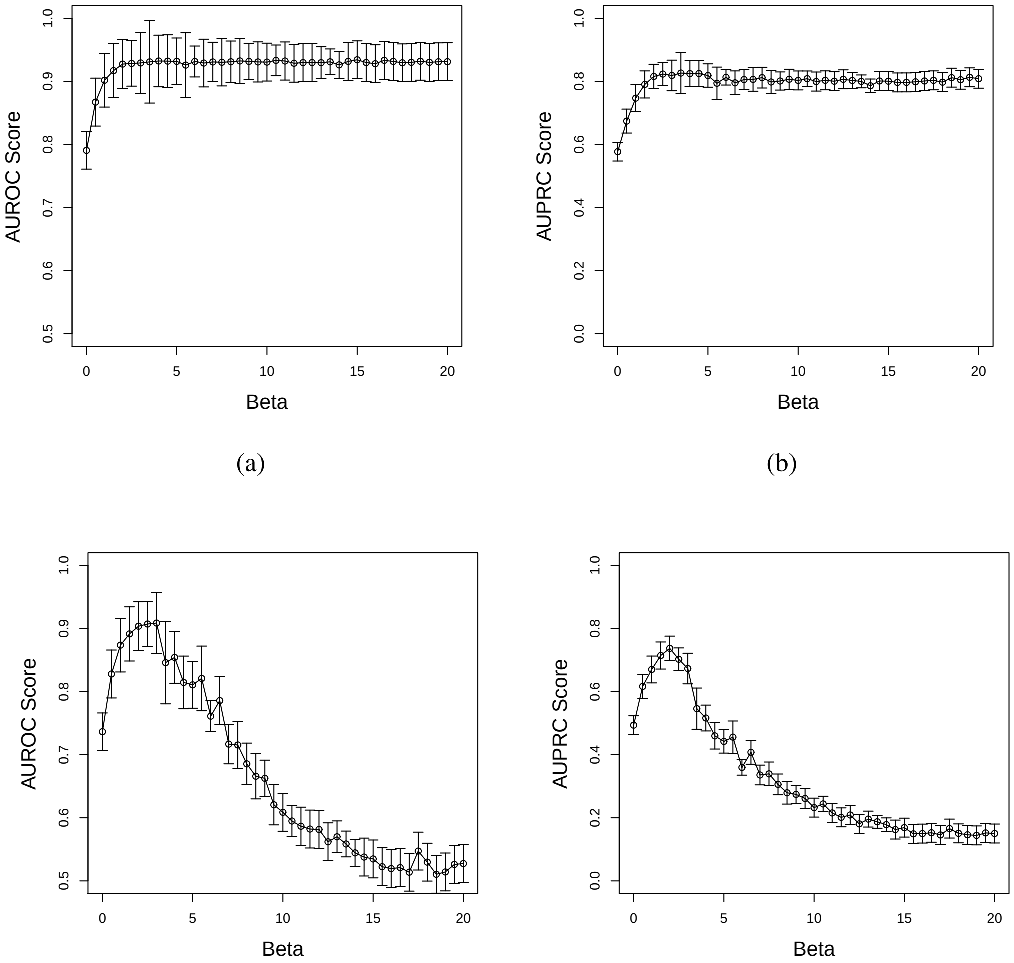


Figure 4.3: Evaluation of AUROC and AUPRC network reconstruction scores for the five methods, TVDBN-0 (white), TVDBN-Exp-hard (dark grey, left), TVDBN-Exp-soft (dark grey, right), TVDBN-Bino-hard (light grey, left), TVDBN-Bino-soft (light grey, right). Top Row: The boxplots show the distributions of the reconstruction scores. Bottom Row: The boxplots show the difference of the AUROC and AUPRC reconstruction scores to TVDBN-0; larger differences indicate better performance with information sharing. All simulations were repeated for 10 independent data sets with 4 network segments each. Structure changes were applied to the segments sequentially, changing between 0-10% of the edges with each new segment. A paired t-test shows that for 0% changes, the difference to TVDBN-0 was significant for all methods ($p < 0.05$). For $>$ 0% changes, the difference to TVDBN-0 was significant ($p < 0.05$) except for the difference in AUPRC scores for TVDBN-Exp-hard for 5% changes ($p = 0.08$) and TVDBN-Exp-hard and TVDBN-Exp-soft for 10% changes ($p = \{0.07, 0.18\}$). In all plots, the horizontal bar of the boxplot shows the median, the box margins show the 25th and 75th percentiles, the whiskers indicate data within 2 times the interquartile range, and circles are outliers. See Table 4.1 for hyperparameter settings.



(a)

(b)

(c)

(d)

Figure 4.4: **Results for the exponential prior with hard coupling on the simulated data without mismatch among the structures.** Panel (a) shows the AUROC scores for different values of the hyperparameter β. Panel (b) shows a corresponding plot for the AUPRC scores. The simulations were repeated on 10 independent data instantiations of time series length 60. The error bars show the standard error. The results were obtained with the novel MCMC sampler, described in Section 4.3. Panels (c) and (d) show the results from corresponding simulations with the old MCMC sampler adapted from Lèbre et al. (2010) and described in Section 3.2.6. The reconstruction performance deteriorates with larger values of the hyperparameter, as a consequence of poor MCMC mixing and convergence.

Table 4.1: List of different information sharing (IS) priors for the TVDBN (Time-Varying Dynamic Bayesian Network), the equation where they were defined, and the most common hyperparameter settings that were used, or hyperparameter ranges if they are inferred. Only the highest level hyperparameters in the Bayesian hierarchy are shown.

Name	Prior	Section	Equation	Hyperparameters
TVDBN-0	Poisson (No IS)	3.2.4	3.4, 3.6	$\Lambda = 3$
TVDBN-Exp-hard	Exponential Hard IS	4.2.2	4.2	$\beta \in [0, 20]$
TVDBN-Exp-soft	Exponential Soft IS	4.2.3	4.10	$\lambda_\kappa = 10$
TVDBN-Bino-hard	Binomial Hard IS	4.2.4	4.17, 4.18	$\alpha, \overline{\alpha}, \gamma, \overline{\gamma} \in \{1, 2, ..., 100\}$
TVDBN-Bino-soft	Binomial Soft IS	4.2.5	4.22	$\alpha, \overline{\alpha}, \gamma, \overline{\gamma} \in \{1, 2, ..., 100\}$

The simulation model is the same as the one used in Chapter 3, which I will now recap briefly. We randomly generated 10 networks with 10 nodes each. A Poisson distribution with mean $\lambda_{parents} = 3$ was used to determine the number of parents for each node. We simulated changes in the network structure by producing 4 different network segments, where a Poisson distribution with mean $\lambda_{changes} \in \{0.25, 0.5, 1\}$ was used to determine the number of changes per node. The changes were then applied uniformly at random to edges and non-edges in the previous segment. For each segment h, we generated a time series of length 15 using a linear regression model:

$$\mathbf{D}(t) = \boldsymbol{W}^h \mathbf{D}(t-1) + \boldsymbol{\varepsilon} \tag{4.28}$$

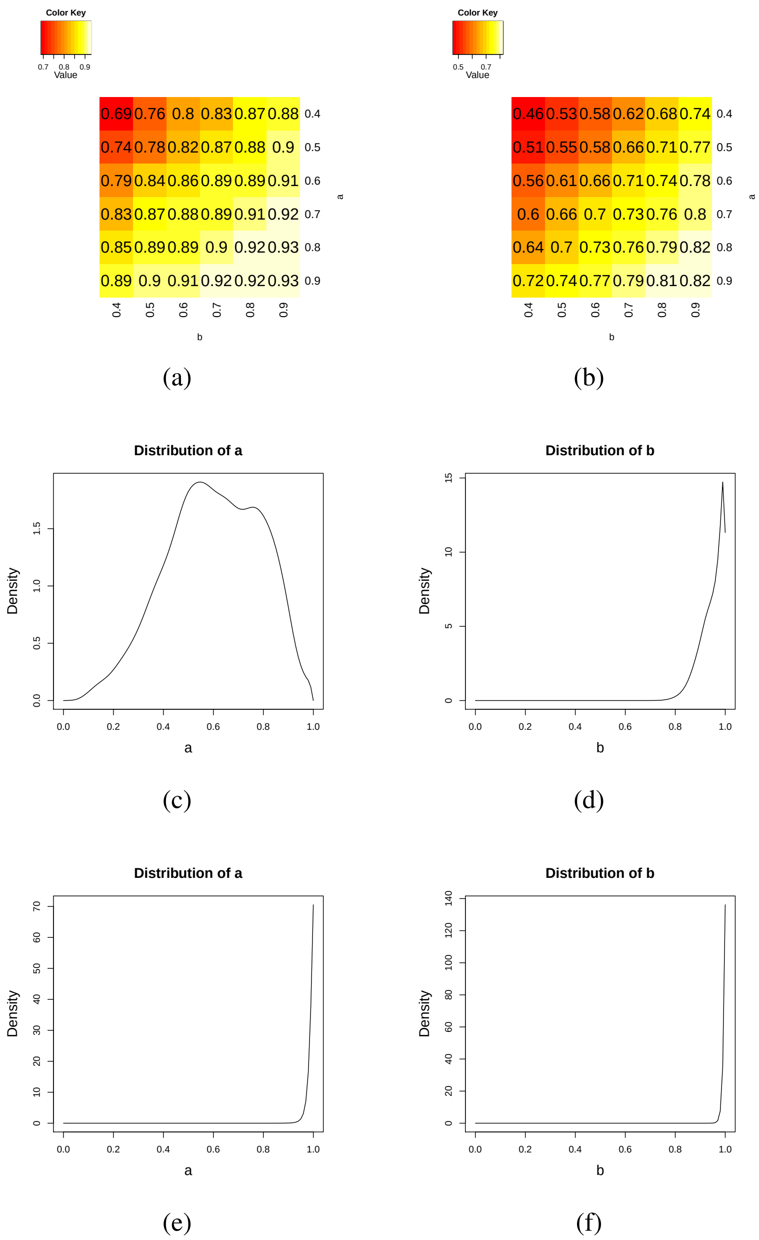
where $\mathbf{D}(t)$ is the 10×1 vector of observations at time t and $\boldsymbol{W}^h = \{w_{ij}^h\}$ is the 10×10 matrix of segment-specific regression weights for each edge. We chose the regression weights such that $w_{ij}^h = 0$ if there is no edge between node i and node j in the network structure for segment h, and $w_{ij}^h \sim N(0, 1)$ otherwise. We added Gaussian observation noise $\varepsilon_i \sim N(0, 1)$ independently for each observation of node i.

First, we consider the scenario of homogeneous time series in which the regulatory network structure does not change (although the regression coefficients associated with each edge may change between segments). This is the situation in which the proposed Bayesian regularization scheme should achieve the strongest boost in the network reconstruction accuracy. We indeed found this conjecture confirmed in our simulations, as demonstrated in Figure 4.3 (0% changes). We would also assume that high values of the hyperparameter β should lead to the best network reconstruction accuracy, as this corresponds to the tightest tying between adjacent structures. However, repeating the MCMC simulations initially did not confirm this conjecture; see Figures 4.4c and 4.4d. As discussed in Section 4.3, the observed mismatch was a consequence of poor mixing and convergence for large hyperparameter values, which is endemic to the naive extension of the MCMC sampler from Lèbre et al. (2010). Repeating the simulations with the novel MCMC scheme proposed in Section 4.3 leads to the graphs of Figure 4.4a and 4.4b. Here, the network reconstruction accuracy no longer deteriorates with increasing hyperparameters, indicating that the mixing and convergence problems have been averted.

Another question we investigated is whether the sampled values of the hyperparameters concur with those that optimize the network reconstruction accuracy. While the hyperparameter β of the exponential prior does indeed tend to higher values, the situation is different for the hyperparameters a and b of the binomial prior. The top panels in Figure 4.5 show the network reconstruction accuracy in terms of AUROC and AUPRC scores for several fixed values of the hyperparameters a and b. As expected, the peak performance is reached for the highest values, as no mismatch between the structures implies that tight coupling is consistent with the data. The centre panels of Figure 4.5 show the posterior distribution of the hyperparameters that was obtained with the conventional MCMC proposal scheme adapted from Lèbre et al. (2010) and described in Section 3.2.6. There is an obvious mismatch between the high-posterior probability region and the region of hyperparameters that optimize the network reconstruction. This provides more evidence that the sampler adapted for segment coupling from Lèbre et al. (2010) suffers from mixing and convergence problems. The bottom panels of Figure 4.5 show the marginal posterior distributions of the hyperparameters inferred in the MCMC simulations with the novel multi-segment proposal move introduced in Section 4.3. It is seen that, unlike the centre panels in Figure 4.5, and as a consequence of the different proposal scheme, the high posterior probability region now concurs with the region of maximum network reconstruction accuracy. This agreement suggests that the novel MCMC sampler leads to a significant improvement in mixing and convergence, in corroboration of our conjecture in Section 4.3.

Next, we turn our attention to varying network structures. We varied the percentage of edges that change from segment to segment between 2.5% to 10%[6]. A significant

[6] Because our simulation was set up so that we had on average 3 regulatory interactions per node, this corresponds to a change of between 8.25% and 33% of the original interactions.



(a)

(b)

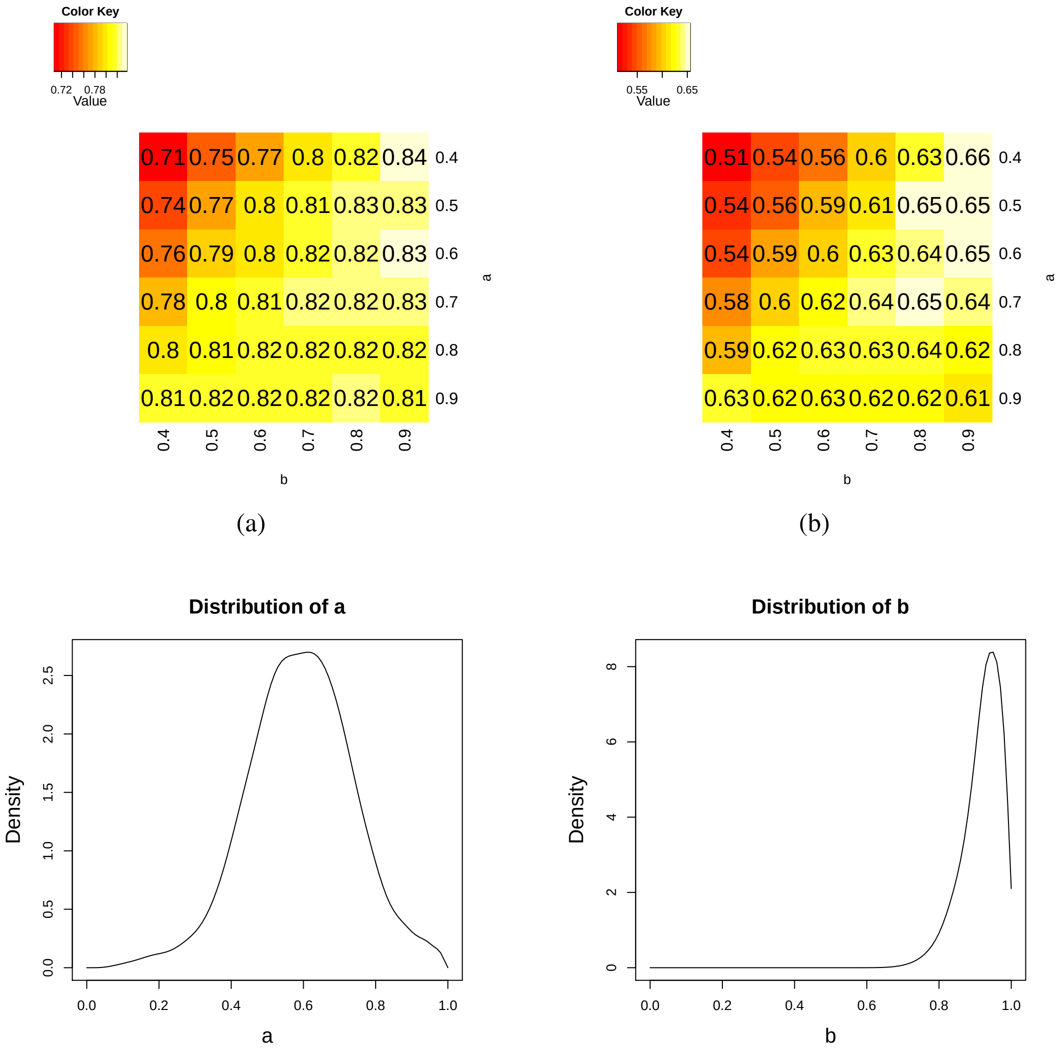
(c)

(d)

(e)

(f)

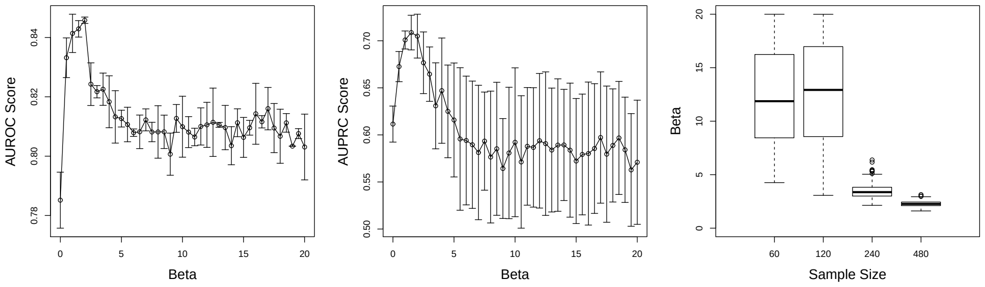
Figure 4.5: **Results for the binomial prior with hard coupling on the simulated data without mismatch among the structures.** Panel (a) shows the AUROC scores for different values of the hyperparameters a and b. Panel (b) shows a corresponding plot for the AUPRC scores. Panels (c) and (d) show the marginal posterior distribution of the hyperparameters a and b, as obtained with the MCMC sampler adapted from Lèbre et al. (2010) and described in Section 3.2.6. Panels (e) and (f) show the marginal posterior distribution of the hyperparameters a and b, as obtained with the new MCMC sampler proposed in Section 4.3. The marginal distributions of a and b are obtained from the sampled values of the level-2 hyperparameters α, $\bar{\alpha}$, γ, $\bar{\gamma}$ and from the sampled networks using a kernel density estimator with the beta distribution from equation (4.17). The level-2 hyperparameters were given a uniform prior over the discrete set $\{1, 2, ..., 100\}$.



(a) (b)

(c) (d)

Figure 4.6: **Results for the binomial prior with hard coupling on the simulated data with mismatch among the structures.** Panel (a) shows the AUROC scores for different values of the hyperparameters a and b. Panel (b) shows a corresponding plot for the AUPRC scores. Panels (c) and (d) show the marginal posterior distribution of the hyperparameters a and b, as obtained with the novel MCMC sampler proposed in Section 4.3. The marginal distributions of a and b were obtained from the sampled values of the level-2 hyperparameters α, $\overline{\alpha}$, γ, $\overline{\gamma}$ and from the sampled networks using a kernel density estimator with the beta distribution from equation (4.17). The level-2 hyperparameters were given a uniform prior over the discrete set $\{1,2,...,100\}$.



(a) (b) (c)

Figure 4.7: **Results for the exponential prior with hard coupling on the simulated data with mismatch among the structures.** Panel (a) shows the AUROC scores and their standard deviations for different values of the hyperparameter β. Panel (b) shows a corresponding plot for the AUPRC scores. Panel (c) shows box plot representations of the inferred posterior distribution of β, for different sample sizes, using the MCMC scheme from Section 4.3. The horizontal bar shows the median, the box margins show the 25th and 75th percentiles, the whiskers indicate data within 2 times the interquartile range, and circles are outliers. The simulations were repeated on 10 independent data instantiations of time series length $n = 60$.

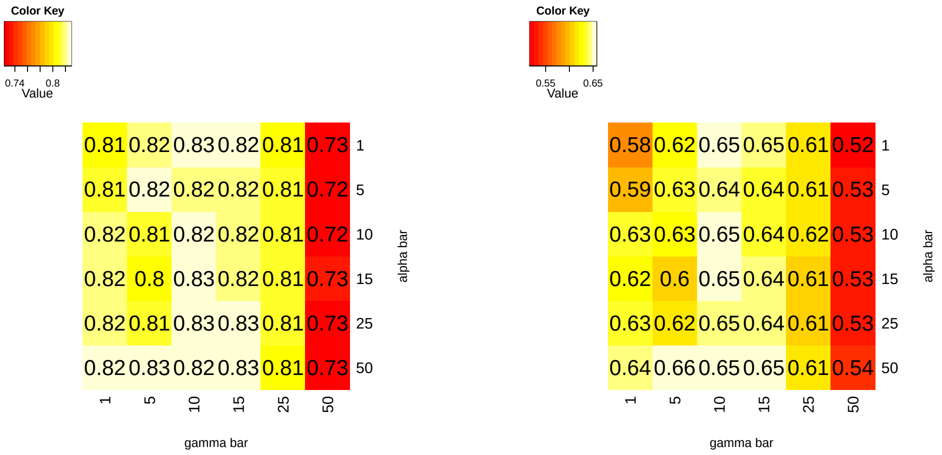
improvement in the network reconstruction accuracy can be achieved over the unregularized method, as shown in the bottom panels of Figure 4.3. However, the magnitude of the improvement in the scores decreases as the number of changes between adjacent segments increases. This is plausible: as we introduce more structural changes between adjacent networks, we would expect to gain less benefit from information sharing. We note that the degradation in performance seems to be stronger for the exponential prior than for the binomial prior.

We investigated whether the inferred hyperparameters coincide with the optimal reconstruction performance for the case where 10% of the edges in the network change between adjacent segments. There are two effects to be traded off. Hyperparameter values that are too low will not bring about any improvement over the uncoupled unregularized scenario. Hyperparameter values that are too high will not allow the network structure to change with time. We would therefore expect to find some optimal finite range of hyperparameter values, $0 < \beta < \infty$ and $0 < a, b < 1$. This has in fact been borne out in our simulations. Figure 4.6 shows the network reconstruction accuracy in terms of AUROC and AUPRC scores for different values of the hyperparameters a, b. The best network reconstruction accuracy is obtained when b, which governs consistency among non-interactions, is high (≥ 0.9), while a, which controls agreement among interactions, is reduced to a range around its uninformative setting $a \approx 0.5$.

The bottom panel of Figure 4.6 shows that the inferred posterior distribution is consistent with these ranges, and that the Bayesian inference scheme thus optimizes the network reconstruction accuracy. A slightly different picture emerges for the exponential prior, though. Figures 4.7a-4.7b show the AUROC and AUPRC scores for different values of β, indicating a clear peak in the network reconstruction accuracy for finite $0 < \beta < \infty$. This peak does not coincide with the high posterior probability range of β, as shown in Figure 4.7c. Only when increasing the data set size by a factor of 4 does the Bayesian inference scheme succeed in optimizing the network reconstruction accuracy in the sense that the high posterior probability region now coincides with the range of the highest AUROC/AUPRC scores. The obvious question to ask is whether this trend is another artifact of poor MCMC convergence/mixing. To this end we have devised a simplified model for which the posterior distribution can be computed in closed form. Our analysis, which we present in Section 4.5.2, reproduces the results from this simulation study, suggesting that the suboptimal performance of the Bayesian inference scheme is intrinsic to the chosen form of the prior.[7] In principle this problem can be addressed by choosing a sparse prior for the hyperparameter β. However, this requires us to set the desired level of sparsity in advance (if we have prior knowledge about the similarity between adjacent network segments). Inferring the level of sparsity from the data would be infeasible, since Section 4.5.2 shows that the problem only arises when the data are not sufficiently informative. In the absence of prior knowledge, we therefore recommend using an alternative form of information sharing prior, such as the binomial prior.

Returning to the binomial prior, we finally investigated the influence of the level-2 hyperparameters $\alpha, \overline{\alpha}, \gamma$, and $\overline{\gamma}$. Recall that owing to the conjugacy of the prior, these values can be interpreted as fictitious prior observation counts. Our initial idea was to keep the mismatch hyperparameters fixed at $\overline{\alpha} = \overline{\gamma} = 1$, while putting a vague uniform distribution over the set $\{1, 2, , \ldots, 100\}$ as a prior on the match hyperparameters α and γ. The rationale behind this choice is that the regularization scheme is intended to encourage similarity rather than dissimilarity between adjacent network structures.

[7] We may note that the results for the exponential prior seem to be at odds with those reported in Chapter 3. The reason is that in the previous chapter, I had selected, by a fluke, a more restrictive prior on the hyperparameter: $\beta_i \in [0, 5]$. As the discussion in Section 4.5.2 shows, this setting boosts the network reconstruction performance.



alpha bar \ gamma bar	1	5	10	15	25	50
1	0.81	0.82	0.83	0.82	0.81	0.73
5	0.81	0.82	0.82	0.82	0.81	0.72
10	0.82	0.81	0.82	0.82	0.81	0.72
15	0.82	0.8	0.83	0.82	0.81	0.73
25	0.82	0.81	0.83	0.83	0.81	0.73
50	0.82	0.83	0.82	0.83	0.81	0.73

(a)

alpha bar \ gamma bar	1	5	10	15	25	50
1	0.58	0.62	0.65	0.65	0.61	0.52
5	0.59	0.63	0.64	0.64	0.61	0.53
10	0.63	0.63	0.65	0.64	0.62	0.53
15	0.62	0.6	0.65	0.64	0.61	0.53
25	0.63	0.62	0.65	0.64	0.61	0.53
50	0.64	0.66	0.65	0.65	0.61	0.54

(b)

Figure 4.8: **Results for the binomial prior with hard coupling on the simulated data with mismatch among the structures: dependence of the reconstruction accuracy on the higher-level hyperparameters.** Panel (a) shows the AUROC scores for different values of the level-2 hyperparameters $\bar{\alpha}$ and $\bar{\gamma}$. Panel (b) shows a corresponding plot for the AUPRC scores. The results indicate that setting $\bar{\alpha} = \bar{\gamma} = 1$ is over-restrictive and that the reconstruction accuracy improves as a consequence of employing a non-informative prior.

However, repeating the MCMC simulations for different values of the level-2 hyperparameters revealed that the setting $\bar{\alpha} = \bar{\gamma} = 1$ is too restrictive and that the network reconstruction accuracy can be improved by relaxing this constraint (see Figure 4.8).

The findings of our simulation study can be summarized as follows. A naive extension of the MCMC sampler of Lèbre et al. (2010), as described in Section 3.2.6, leads to a poor network reconstruction accuracy for high values of the hyperparameters; this problem can be resolved with the novel proposal scheme introduced in Section 4.3. With this new proposal scheme, information sharing with the binomial prior leads to a significant improvement in the network reconstruction accuracy in all cases, while information sharing with the exponential prior leads to a significant improvement when the true network structures are sufficiently similar. A detailed analysis of hyperparameter inference shows that the Bayesian inference scheme is consistent for the binomial prior in the sense that the high posterior probability region of the hyperparameters concurs with the one that optimizes the network reconstruction accuracy. For the exponential prior, this consistency is only given when the data set size is sufficiently large; otherwise a more restrictive hyperprior (i.e. prior on β) is needed. On

Table 4.2: Likelihood and prior scores for the edges contained in the sets defined in Figure 4.9. The product of the prior and the likelihood defines the rank of the edge; the truth indicator is shown in the second column.

Set	True edge	Supported by the data	Supported by the prior	Likelihood	Prior	Number of edges
L	yes	yes	no	A	$e^{-\beta}$	N_L
LB	yes	yes	yes	A	1	N_{LB}
LB^*	yes	no	yes	1	1	N_{LB^*}
L^*	yes	no	no	1	$e^{-\beta}$	N_{L^*}
B	no	no	yes	1	1	N_B
B^*	no	yes	yes	A^*	1	N_{B^*}
F^*	no	yes	no	A^*	$e^{-\beta}$	N_{F^*}
F	no	no	no	1	$e^{-\beta}$	N_F

the other hand, a restrictive setting for the level-2 hyperparameters of the binomial prior is counter-productive, and better network reconstruction scores are obtained with a non-informative hyperprior.

4.5.2 Closed-form inference for the exponential prior

The results in Figure 4.7 indicated that for the exponential prior, the Bayesian inference scheme might fail to find the hyperparameters that optimize the network reconstruction accuracy. Our conjecture is that this is not a consequence of poor mixing and convergence of the MCMC sampler, but intrinsic to the Bayesian inference scheme *per se*. As a demonstration, we reproduce the observation from Figure 4.7 with a simpler model for which a closed-form expression of the posterior distribution of the hyperparameter can be derived.

We consider the scenario depicted in Figure 4.9, where edges of a hypothetical network can be divided into different categories, depending on whether or not they are true, supported by the data, or included in the prior network. An overview of the notation is presented in Table 4.2. We make the simplifying assumption that the edges are a posteriori independent, leading to a multiplicative contribution of each edge to the likelihood. While this is clearly an unrealistic assumption, our aim is to demonstrate that even this simplified model will produce the same behaviour that the

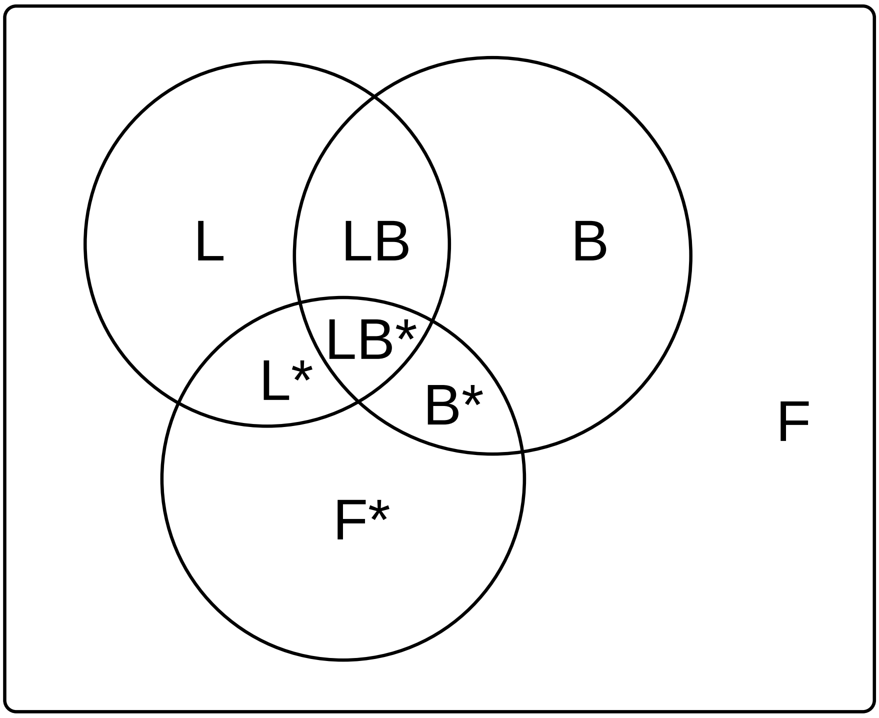


Figure 4.9: Illustration of a hypothetical network scenario, where edges fall into several categories. Edges in sets L, LB, L^* and LB^* are true edges, which means they are included in the network corresponding to the current time series segment. Edges in sets L and LB are 'true positives' in that they contribute a score $A > 1$ to the likelihood. Edges in sets L^* and LB^* are 'false negatives', which contribute the neutral score of 1 to the likelihood. The edges in sets F^* and B^* are 'false positives', which contribute a score $A^* > A > 1$ to the likelihood. The edges in sets LB, LB^*, B^* and B are consistent with the prior network, all those in the complementary sets are not found in the prior network. Edges in set F are neither included in the network associated with the current segment, nor can they be found in the prior network. They also don't contribute any score to the log likelihood (i.e. they contribute a neutral score of 1 to the likelihood). An overview can be found in Table 4.2.

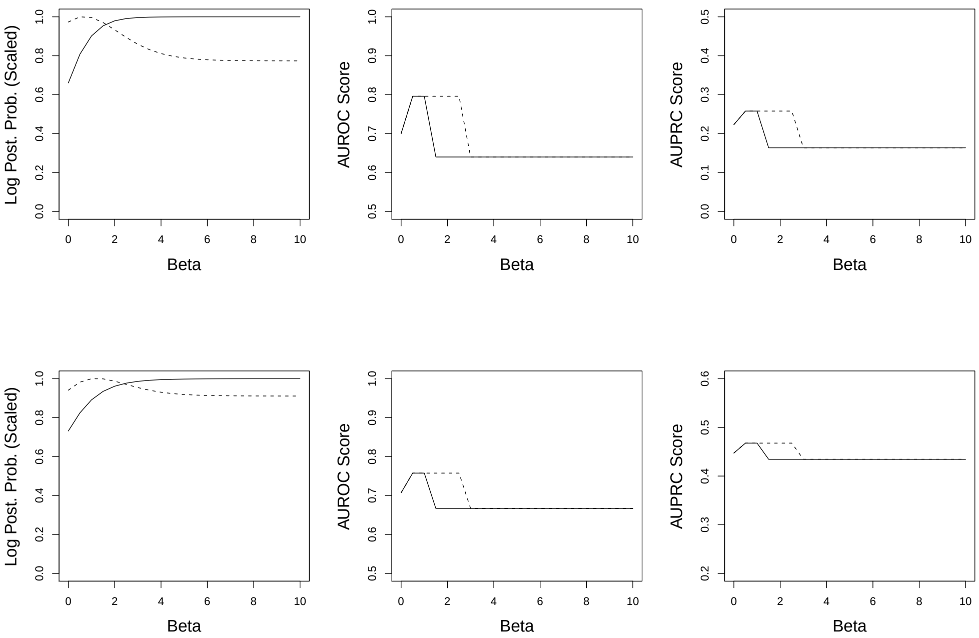


Figure 4.10: **Results for the simplified model with exponential prior.** The leftmost column shows the marginal posterior distribution of β, computed from equation (4.31). The middle column shows the AUROC score as β varies. The rightmost column shows the AUPRC score as β varies. Solid line: $A = 2$, $A^* = 4$, dashed line: $A = 12$, $A^* = 14$. The top and bottom rows correspond to two different settings of the set sizes. Top Row: $\{L : 15, LB : 0, B : 40, F : 60, L^* : 0, LB^* : 10, B^* : 25, F^* : 0\}$. Bottom Row: $\{L : 15, LB : 20, B : 10, F : 25, L^* : 0, LB^* : 10, B^* : 20, F^* : 0\}$.

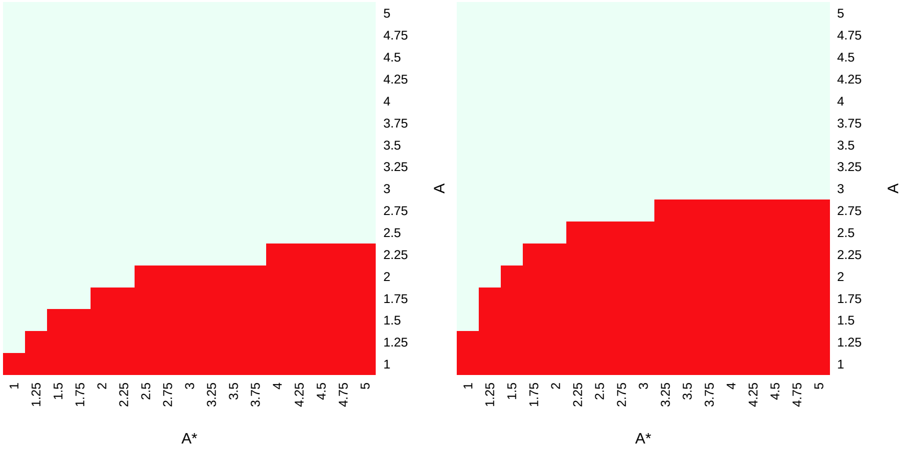


Figure 4.11: **Existence of a peak in the posterior distribution of β for the simplified model with exponential prior.** The two plots show values of A and A^* for which the marginal posterior probability of β monotonically increases as β increases (red tiles), and those where the posterior probability decreases for high β (white tiles), indicating the existence of a peak in the distribution. We used the same settings of the set sizes as in Figure 4.10. Left: $\{L:15, LB:0, B:40, F:60, L^*:0, LB^*:10, B^*:25, F^*:0\}$. Right: $\{L:15, LB:20, B:10, F:25, L^*:0, LB^*:10, B^*:20, F^*:0\}$.

more sophisticated DBN model exhibits. The likelihood is given by

$$P(\mathbf{D}|\mathcal{M}) = A^{(n_L+n_{LB})}A^{*(n_{B^*}+n_{F^*})} \tag{4.29}$$

where n_S counts the number of elements in set S for network $\mathcal{M}$, and the symbols denoting the sets have been defined in Table 4.2. Assuming a uniform prior on β, the posterior distribution of the hyperparameter becomes:

$$\begin{aligned} P(\beta|\mathbf{D}) \propto P(\mathbf{D},\beta) &= \sum_{\mathcal{M}} P(\mathbf{D}|\mathcal{M})P(\mathcal{M}|\beta)P(\beta) \\ &\propto \frac{1}{Z(\beta)}\sum_{\mathcal{M}} P(\mathbf{D}|\mathcal{M})\exp(-\beta|\mathcal{M}-\mathcal{M}^0|) \end{aligned} \tag{4.30}$$

where $\mathcal{M}^0$ represents our prior knowledge. Inserting (4.29) into (4.30) we get, with equation (4.3) for $Z(\beta)$ and under the assumption of a uniform prior on β:

$$\begin{aligned} P(\beta|\mathbf{D}) \propto & \frac{1}{\left(1+e^{-\beta}\right)^N} \sum_{n_L=0}^{N_L} \sum_{n_{LB}=0}^{N_{LB}} \sum_{n_B=0}^{N_B} \sum_{n_F=0}^{N_F} \sum_{n_{L^*}=0}^{N_{L^*}} \sum_{n_{LB^*}=0}^{N_{LB^*}} \sum_{n_{B^*}=0}^{N_{B^*}} \sum_{n_{F^*}=0}^{N_{F^*}} \\ & \binom{N_L}{n_L} \binom{N_{LB}}{n_{LB}} \binom{N_B}{n_B} \binom{N_F}{n_F} \binom{N_{L^*}}{n_{L^*}} \binom{N_{LB^*}}{n_{LB^*}} \binom{N_{B^*}}{n_{B^*}} \binom{N_{F^*}}{n_{F^*}} \\ & A^{(n_L+n_{LB})} A^{*(n_{B^*}+n_{F^*})} \\ & \exp(-\beta[n_L+n_F+N_{LB}-n_{LB}+N_B-n_B+ \\ & n_{L^*}+n_{F^*}+N_{LB^*}-n_{LB^*}+N_{B^*}-n_{B^*}]) \end{aligned} \tag{4.31}$$

A plot of (4.31) is shown in Figure 4.10. The optimal network reconstruction in terms of AUROC and AUPRC scores is achieved for a finite value of $\beta \approx 1$. The effect of the data set size is emulated by varying the settings of the parameters entering the likelihood. For small values of A and A^*, corresponding to small data sets, the posterior probability increases monotonically in β, and the Bayesian inference scheme intrinsically fails to find the range of hyperparameters that optimizes the network reconstruction accuracy. When we increase the data set size, this mismatch disappears, and the two regions concur. These findings are consistent with those presented in Figure 4.7 and suggest that the observed mismatch is a genuine inference feature rather than an MCMC artifact.

To further analyse this effect, we have investigated the values of A and A^* for which the posterior distribution shows a peak for a finite value of β. Analytically, this corresponds to finding values for A and A^* such that the equation $\frac{dP(\beta|\mathbf{D})}{d\beta} = 0$ has a solution. Unfortunately, it is non-trivial to determine the existence of a solution analytically; we have therefore resorted to numerically calculating $\frac{dP(\beta|\mathbf{D})}{d\beta}$ for $\beta = 20$. At $\beta = 0$, we have $\frac{dP(\beta|\mathbf{D})}{d\beta} > 0$; therefore, if $\frac{dP(\beta|\mathbf{D})}{d\beta} < 0$ at $\beta = 20$, this indicates that the distribution has a peak on the interval $[\beta, 20]$. On the other hand, under the assumption of unimodality, $\frac{dP(\beta|\mathbf{D})}{d\beta} > 0$ at $\beta = 20$ indicates that the marginal posterior probability of β increases monotonically with β. The results of this analysis are shown in Figure 4.11, which shows a clear phase shift towards distributions with a peak as A and A^* increase.

What does this analysis entail for the general applicability of the exponential prior? It is clear that when the data set size is too small, then the marginal posterior distribution of β will be biased towards high values. The exact definition of ”too small” will crucially depend on the nature of the dataset. Given that we have shown in Section 4.5.1 that the binomial prior avoids this weakness and outperforms the exponential prior in terms of network reconstruction accuracy, we would recommend that this form of information sharing prior be used in preference of the exponential prior.

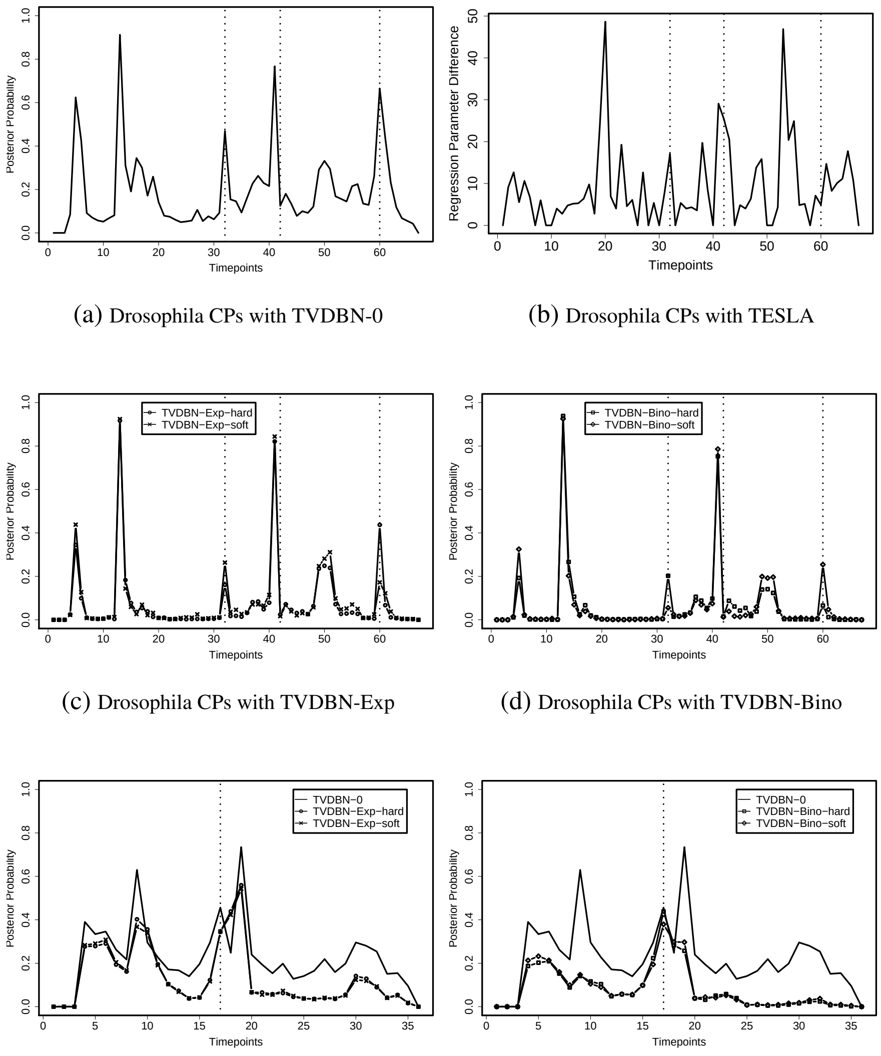
4.6 Real-world applications

4.6.1 Morphogenesis in *Drosophila melanogaster*

This subsection describes the application of our sequential information sharing methods to gene expression data from the life-cycle of Drosophila. This is analogous to the application described in Section 3.6.2. However, the results presented in Chapter 3 have been extended in two important regards: By applying an additional functional forms of the prior (binomial) and different node coupling schemes (soft and hard), and by presenting the networks obtained with the objectively best method (TVDBN-Bino-hard) and validating a number of interactions using the FLIGHT database (Sims et al., 2006).

During its life-cycle, *Drosophila melanogaster* undergoes four major stages of morphogenesis: embryo, larva, pupa and adult. Arbeitman et al. (2002) obtained a gene expression time series covering all four stages. We have applied our methods to a subset of this gene expression time series consisting of eleven genes involved in wing muscle development. First, we investigated whether the changepoints inferred by our methods correspond to the known transitions between stages. Figure 4.12a shows the posterior probabilities of inferred changepoints for any gene using TVDBN-0 (unregularized by information sharing, see Table 4.1), while Figures 4.12c-4.12d show the posterior probabilities for the information sharing methods. We compared this performance to the method proposed in Ahmed and Xing (2009), using the authors' software package TESLA (Figure 4.12b). In addition, Robinson and Hartemink (2009) used a discrete non-homogeneous DBN to analyse the same data set, and a plot corresponding to Figure 4.12b can be found in their paper.

An analysis of the results suggests that our non-homogeneous DBN methods are generally more successful than TESLA. We recover changepoints for all three transitions (embryo → larva, larva → pupa, and pupa → adult). As shown in Figure 4.12b, the last transition, pupa → adult, is less clearly detected with TESLA, and it is completely absent in Robinson and Hartemink (2009). Furthermore, TESLA and our method both detect additional changepoints during the embryo stage, which are missing in Robinson and Hartemink (2009). It is not implausible that additional transitions



(a) Drosophila CPs with TVDBN-0

(b) Drosophila CPs with TESLA

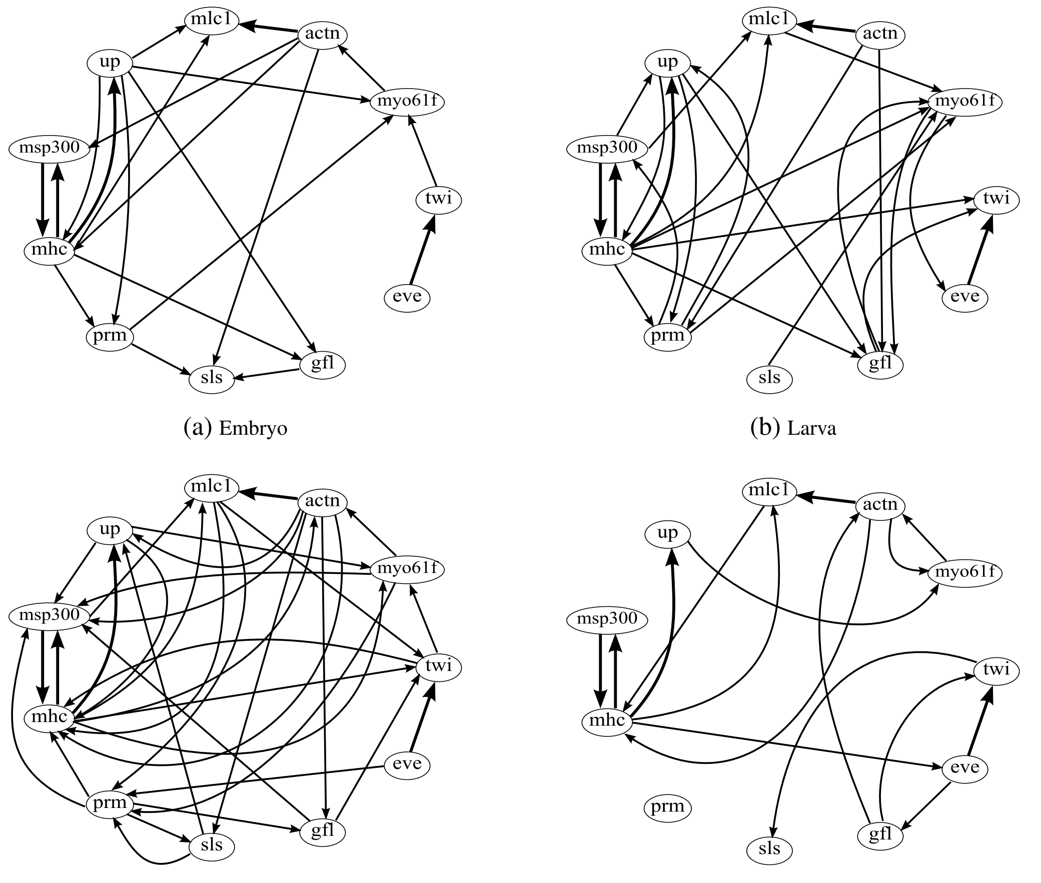
(c) Drosophila CPs with TVDBN-Exp

(d) Drosophila CPs with TVDBN-Bino

(e) Synthetic Network CPs with TVDBN-Exp

(f) Synthetic Network CPs with TVDBN-Bino

Figure 4.12: Changepoints inferred from gene expression time series related to morphogenesis in *Drosophila melanogaster*, and synthetic biology in *Saccharomyces cerevisiae* (yeast). 4.12a: TVDBN-0 changepoints for Drosophila (no information sharing). 4.12b: TESLA, L1-norm of the difference of the regression parameter vectors associated with two adjacent time points plotted against time. 4.12c and 4.12d: TVDBN changepoints for Drosophila with information sharing; the method is indicated by the legend. 4.12e and 4.12f: TVDBN changepoints for the synthetic gene regulatory network in yeast. All figures using TVDBN plot the posterior probability of a changepoint occurring for any node at a given time (ordinate) against time (abscissa). In 4.12a-4.12d, the vertical dotted lines indicate the three morphogenic transitions, while in 4.12e and 4.12f the line indicates the boundary between the “switch on” (galactose) and “switch off” (glucose) phases.



(a) Embryo

(b) Larva

(c) Pupa

(d) Adult

Figure 4.13: Gene regulatory networks inferred from gene expression time series related to morphogenesis in *Drosophila melanogaster*, using TVDBN-Bino-hard. The networks were obtained by applying a threshold of 0.25 to the marginal posterior probabilities of the gene interactions. We have reconstructed a network for each morphological phase; interactions that were consistent across all four phases are marked in bold.

at the gene regulatory network level should occur within one morphogenic phase. One would expect that a complex gene regulatory network is unlikely to transition into a new phase all at once, and some pathways might have to undergo activational changes earlier in preparation for the morphogenic transition. However, a failure to detect a known transition represents a shortcoming of a method, and so we can say that in this aspect, our model appears to outperform the two alternative approaches.

In addition to the changepoints, we have inferred network structures for the morphogenic stages of embryo, larva, pupa and adult (see Figure 4.13). An objective assessment of the reconstruction accuracy is not feasible due to the limited existing biological knowledge and the absence of a gold standard. However, our reconstructed networks show many similarities with the networks discovered by Robinson and Hartemink (2009), Guo et al. (2007) and Zhao et al. (2006). For instance, we recover the interaction between two genes, *eve* and *twi*. This interaction is also reported in Guo et al. (2007) and Zhao et al. (2006), while Robinson and Hartemink (2009) seem to have missed it. We also recover a cluster of interactions among the genes *myo61f*, *msp300*, *mhc*, *prm*, *mlc1* and *up* during all morphogenic phases. This result is not implausible, as all genes (except *up*) belong to the myosin family. However, unlike Robinson and Hartemink (2009), we find that *actn* also participates as a regulator in this cluster. There is some indication of this in Zhao et al. (2006), where *actn* is found to regulate *prm*.

We have further validated our reconstructed networks using genetic and protein interactions recorded in the FLIGHT database (Sims et al., 2006). We found that a number of the inferred interactions over all segments correspond to interactions that have been reported in the literature. Some of these result from indirect interactions, where the intermediate gene is missing in the data. Table 4.3 gives an overview of the identified interactions with references to the biological literature.

4.6.2 Synthetic biology in *Saccharomyces cerevisiae*

Synthetic biology is a rapidly developing and highly topical discipline that aims to combine the biological sciences and engineering (Andrianantoandro et al., 2006). One of its aims is to design new gene regulatory networks in living cells. We make use of these endeavours by using gene expression time series obtained *in vivo* from cells with a known gene regulatory network structure to objectively assess the network reconstruction accuracy. Our work is based on Cantone et al. (2009), where the authors constructed a synthetic regulatory network with 5 genes in *Saccharomyces cerevisiae* (yeast). Then they measured gene expression time series with RT-PCR for 16 and 21 time points under two experimental conditions, related to the carbon source: galactose (“switch on”), and glucose (“switch off”). The authors applied two established state-of-the-art methods from computational systems biology to reconstruct the known underlying network from these time series. One is based on ODEs: ordinary differential equations (TSNI), the other is based on conventional DBNs (Banjo); see Cantone et al. (2009) for details. Both methods are optimization-based and only output a single network. By comparison with the known network, the authors calculated the precision (proportion of predicted regulatory interactions in the network that are correct) and recall (proportion of predicted true interactions) scores. Figure 4.14 shows the

Table 4.3: Reconstructed interactions in the *Drosophila melanogaster* wing muscle development network, validated using the FLIGHT database (Sims et al., 2006).

Interaction	References	Interaction	Notes
actn ↔ *mhc*	Homyk Jr and Emerson Jr (1988); Nongthomba et al. (2003); Montana and Littleton (2004)	Protein	Via missing gene *wupA*
actn → *up*	Homyk Jr and Emerson Jr (1988); Nongthomba et al. (2003)	Protein	Via missing gene *wupA*
eve → *twi*	Parkhurst and Ish-Horowicz (1991)	Protein	Via missing gene *RpIIl40*
up ↔ *mhc*	Homyk Jr and Emerson Jr (1988); Nongthomba et al. (2003); Montana and Littleton (2004)	Protein	Direct interaction
actn → *msp300*	Formstecher et al. (2005)	Gene	Via missing gene *TSG101* or missing gene *Hrs*
actn → *sls*	Sanchez et al. (1999)	Gene	Direct Interaction
actn → *prm*	Formstecher et al. (2005)	Gene	Via missing gene *exo70*
prm ↔ *sls*	Sanchez et al. (1999); Formstecher et al. (2005)	Gene	Via missing gene *exo70* and present gene *actn*
sls → *up*	Sanchez et al. (1999); Formstecher et al. (2005)	Protein and Gene	Via missing gene *Act88F*

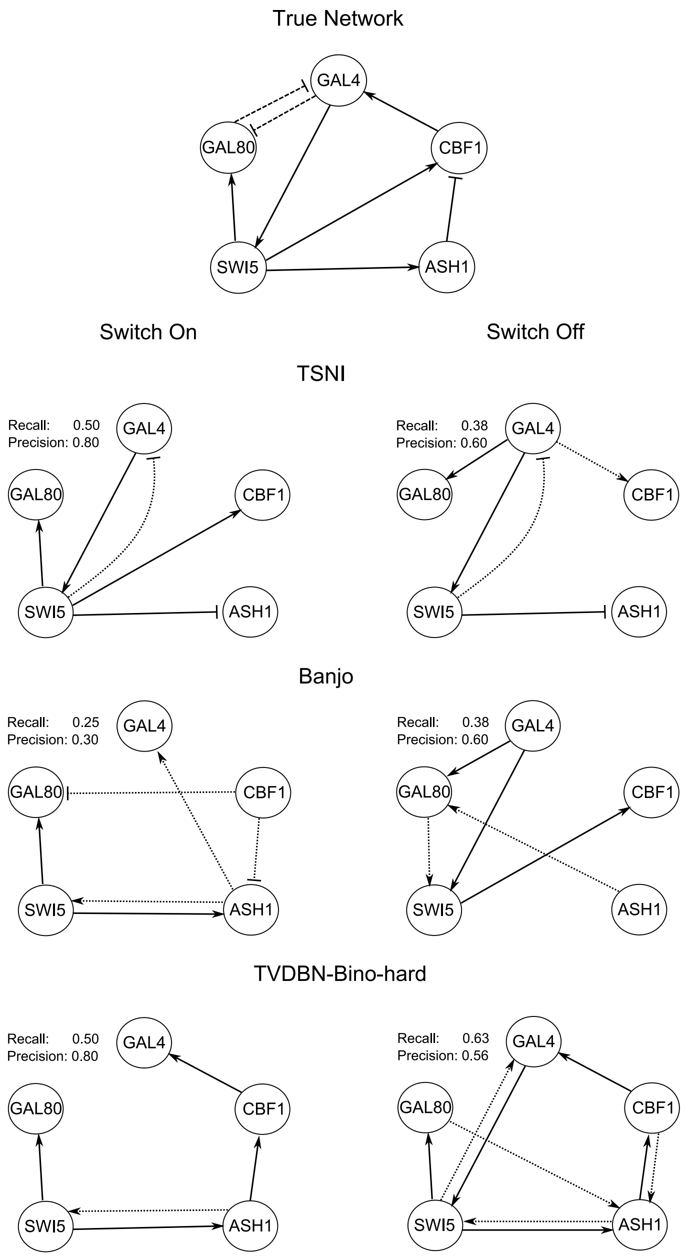


Figure 4.14: True and reconstructed networks for a synthetic biology gene regulatory network in *Saccharomyces cerevisiae* (yeast). Top Row: True network as described in Cantone et al. (2009). 2nd Row: Networks reconstructed using TSNI, a method based on ODEs. 3rd Row: Networks reconstructed using Banjo, a conventional DBN. Bottom Row: Networks reconstructed using TVDBN-Bino-hard, applying a threshold of 0.75 on the marginal posterior probabilities of gene interactions to obtain an absence/presence value for each edge. All reconstructed networks were reconstructed from two gene expression time series obtained with RT-PCR in two experimental conditions, reflecting the switch in the carbon source from galactose (“switch on”) to glucose (“switch off”). The dashed lines in the true network indicate protein-protein regulation. The dotted lines in the reconstructed networks indicate false positive gene interactions. The networks found by Banjo and TSNI are reproduced from Cantone et al. (2009).

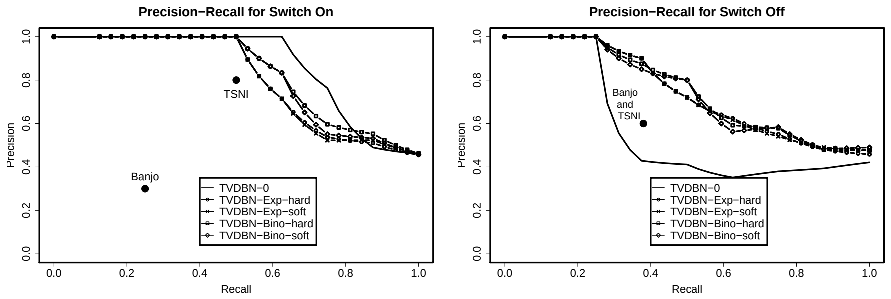


Figure 4.15: Reconstruction of a gene regulatory network designed with synthetic biology in *Saccharomyces cerevisiae*. The network was reconstructed from two gene expression time series obtained with RT-PCR in two experimental conditions, reflecting the switch in the carbon source from galactose ("switch on") to glucose ("switch off"). The reconstruction accuracy of the methods proposed in Section 4.2 and Table 4.1, where the legend is explained, is shown in terms of precision (vertical axis) - recall (horizontal axis) curves. Results were averaged over 10 independent MCMC simulations. For comparison, fixed precision/recall scores are shown for two state-of-the-art methods, as reported in Cantone et al. (2009): Banjo, a conventional DBN, and TSNI, a method based on ordinary differential equations (ODEs).

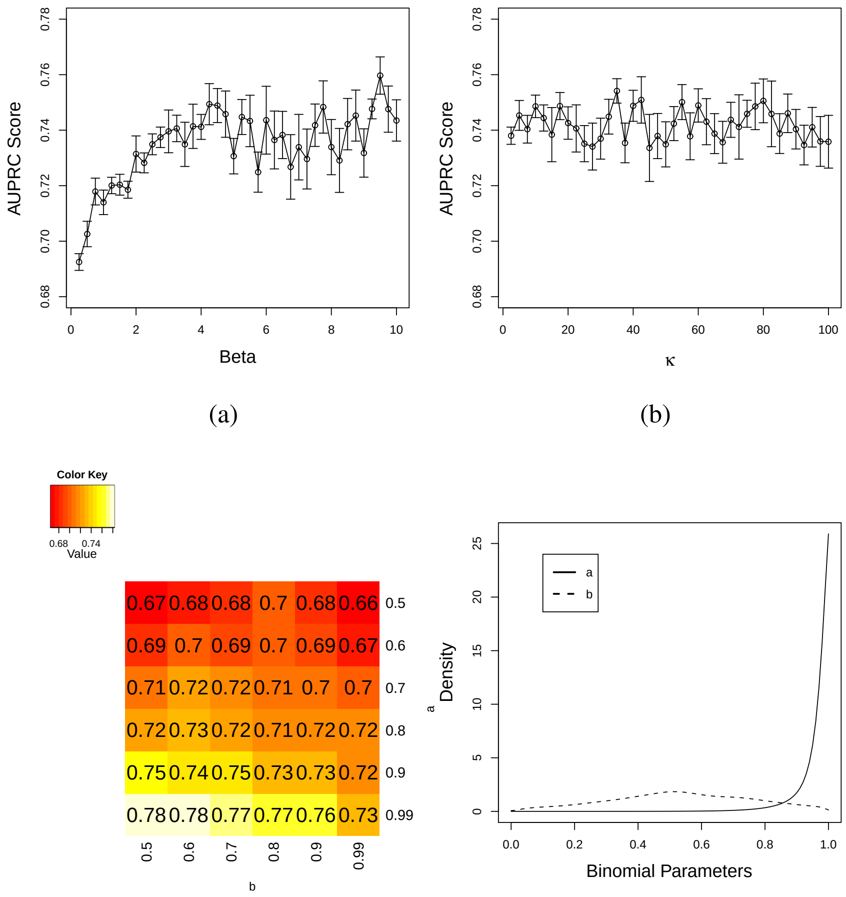
true networks, the reconstructed networks for TSNI and Banjo, as well as the reconstructed networks using TVDBN-Bino-hard, where we have applied a threshold of 0.75 to the inferred marginal posterior probabilities of the gene interactions to obtain absence/presence values for the edges.[8]

In our study, we merged the time series from the two experimental conditions under exclusion of the boundary point[9], and applied the non-homogeneous DBNs from Table 4.1. Figures 4.12e and 4.12f show the inferred marginal posterior probabilities of potential changepoints. The salient changepoint is at the boundary between the "switch on" (galactose) and "switch off" (glucose) phases, confirming that the true changepoint is consistently identified. However, in the absence of information sharing, we observe additional spurious changepoints. These changepoints are successfully suppressed with the proposed Bayesian information-coupling schemes, with the binomial prior having a slightly stronger regularizing effect than the exponential one.

As described in Section 4.4, the Bayesian inference scheme provides a ranking

[8] Note that while our TVDBN methods are in principle capable of inferring the type of interaction (activation or inhibition) by sampling regression weights, we have not investigated this for the purpose of this work. Therefore in Figure 4.14, the arrows in the networks reconstructed using TVDBN-Bino-hard only record the presence or absence of an interaction, and not its type.

[9] When merging two time series $(x_1, \ldots, x_m)$ and $(y_1, \ldots, y_n)$, only the pairs $x_i \to x_j$ and $y_i \to y_j$ are presented to the DBN, while the pair $x_m \to y_1$ is excluded due to the obvious discontinuity.



(a)

(b)

0.5	0.6	0.7	0.8	0.9	0.99	
0.67	0.68	0.68	0.7	0.68	0.66	0.5
0.69	0.7	0.69	0.7	0.69	0.67	0.6
0.71	0.72	0.72	0.71	0.7	0.7	0.7
0.72	0.73	0.72	0.71	0.72	0.72	0.8
0.75	0.74	0.75	0.73	0.73	0.72	0.9
0.78	0.78	0.77	0.77	0.76	0.73	0.99

(c)

(d)

Figure 4.16: Effect of the hyperparameters on the reconstruction of a known gene regulatory network from synthetic biology in yeast. The reconstruction accuracy is measured in terms of the average area under the precision - recall curve (AUPRC). Results were averaged over 10 independent MCMC simulations. 4.16a: Variation of the hyperparameter β for the exponential information sharing prior with hard coupling. 4.16b: Variation of the level-2 hyperparameter κ for the exponential prior with soft coupling, where the mean of the gamma distribution is kept fixed at $\mu = 5$. 4.16c: Variation of hyperparameters a and b for the binomial prior. 4.16d: Sampled distributions of hyperparameters a and b for the binomial prior with hard coupling. These distributions were obtained from the sampled values of the level-2 hyperparameters α, $\overline{\alpha}$, γ, $\overline{\gamma}$ using a kernel density estimator with the beta distribution from equation (4.17).

of the potential gene interactions in terms of their marginal posterior probabilities. From this ranking we computed the precision-recall curves (Davis and Goadrich, 2006) shown in Figure 4.15. By using information sharing, our non-homogeneous DBN outperforms Banjo and TSNI both in the "switch on" and the "switch off" phase. The information sharing methods also perform better than TVDBN-0 on the "switch off" data, but are slightly worse on the "switch on" data. Cantone et al. (2009) showed that in general, the reconstruction accuracy on the "switch off" data is poorer than on the "switch on" data . This lends credence to our results, suggesting that the proposed Bayesian regularization and information sharing schemes substantially improve the gene network reconstruction accuracy on the poorer time series segment, at the cost of a slightly degraded performance on the stronger one. Overall, the effect of information sharing is a performance improvement, as shown by the average areas under the PR curves, averaged over both phases ("switch on and off"): TVDBN-0= 0.68, TVDBN-Exp-hard= 0.74, TVDBN-Exp-soft= 0.74, TVDBN-Bino-hard= 0.76, TVDBN-Bino-soft= 0.75.

We complete our investigation of the yeast network by providing an analysis of the network reconstruction performance (in terms of average area under the PR curve) as the hyperparameters vary. This is analogous to the evaluation we performed in Section 4.5.1 on simulated data. The results are shown in Figure 4.16. As expected, higher values of the hyperparameter β, which correspond to stronger coupling, result in a better performance (Figure 4.16a). Figure 4.16b shows the effect of different values for κ in Equation (4.9). There is no discernible trend, which suggests that the strength of the coupling scheme does not matter much for this application, and that when moving closer to the hard coupling scheme (higher κ while keeping the mean μ of the gamma distribution fixed), the network reconstruction performance does not change significantly. The results obtained with the binomial prior demonstrate that, for this application, encouraging agreement related to the presence of interactions is more important than agreement related to the absence of interactions (Figure 4.16c). Figure 4.16d confirms that our sampled hyperparameters a and b are in the correct range for optimal network reconstruction.

4.7 Discussion

In this chapter, I have looked at sequential information sharing in detail, and investigated four different formulations for the information sharing prior: an exponential prior with hard or soft information coupling among nodes, and a binomial prior with hard or soft information coupling among nodes.

Note that the model of Robinson and Hartemink (2009, 2010) is conceptually similar to the exponential information sharing prior with hard coupling described in Section 4.2.2. By including three alternative information sharing schemes, we have extended the model of Robinson and Hartemink (2009, 2010) in two further respects:

1) We allow for different penalties between edges and non-edges. The method in Robinson and Hartemink (2009, 2010) simply penalizes the number of different edges, i.e. the Hamming distance, between two adjacent structures. This corresponds to the approach taken for the exponential prior in Sections 4.2.2 and 4.2.3. The inclusion of an extra edge leads to the same penalty as the deletion of an existing edge. This might not always be appropriate. Removing a rate-limiting reaction step of a critical signalling pathway is a more substantial change than including some redundant bypass pathway. Our two models based on the binomial prior (Sections 4.2.4 and 4.2.5) allow for that by introducing different prior penalties for the deviation between edges and for the deviation between non-edges. In Section 5 we have experimentally shown that an information sharing approach based on different penalties for edges and non-edges can outperform the simpler approach when the number of changes among segments is small, but non-zero.

2) We allow for different nodes of the network to have different penalty terms. The model in Robinson and Hartemink (2009, 2010) has a single hyperparameter for penalizing differences between structures: λ_s. This might not be appropriate if different subnetworks are conserved to a different degree. For instance, we would assume that molecular network substructures related to generic functionality, e.g. to maintain an essential baseline metabolism, are conserved to a greater extent than more peripheral pathways. By introducing node-dependent hyperparameters, the priors described in Sections 4.2.3 and 4.2.5 generalize the approach in Robinson and Hartemink (2009, 2010) by allowing different parts of the network to be conserved during the temporal process to a different extent.

A further difference to Robinson and Hartemink (2009, 2010) merits some additional discussion. In our model, the changepoints are node-dependent. This gives us extra model flexibility, which is biologically motivated: on infection of an organism by a pathogen, genes involved in defence pathways are likely to be up-regulated, while others are not. Hence, it is plausible that different genes respond to changes in the environment differently, and this is directly incorporated in our model. In Robinson and Hartemink (2010), node-specific changepoints can be obtained indirectly: the calculation of the sufficient statistics for computing the marginal likelihood depends on the intervals during which each parent set is active. The marginal likelihood is recomputed for epochs, where an epoch is the union of consecutive time intervals during which a node-dependent substructure does not change. Since these unions of sets can be different for different nodes, the model does allow different changepoint sets to be associated with different nodes. However, there is a considerable price to pay for that: a changepoint in Robinson and Hartemink (2010) is intrinsically associated with a structure change, whereas in our model, a changepoint can be related to either a structure or a parameter change, or both. This gives us extra model flexibility, which is important for systems biology: when adapting to environmental change, several molecular interactions in signalling pathways may be up- or down-regulated, rather than switched on or off altogether.

An evaluation on simulated data has demonstrated that the proposed Bayesian regularization and sequential information sharing schemes lead to an improved performance over Lèbre (2007) and Lèbre et al. (2010). We have carried out a comparative evaluation of four different information coupling schemes: a binomial versus an exponential prior, and hard versus soft information coupling. This comparison has revealed that the binomial prior allows for more consistent inference of the right level of information sharing, while the exponential prior tends to enforce overly-strong information sharing. The difference between hard and soft information coupling seems negligible in the scenarios we investigated. A detailed investigation of the hyperparameter inference has allowed us to improve the MCMC sampler for better convergence, and to explore the limitations of the exponential information sharing prior.

The application of our method to gene expression time series taken during the life cycle of *Drosophila melanogaster* has revealed better agreement with known morphogenic transitions than the methods of Robinson and Hartemink (2009, 2010) and Ahmed and Xing (2009), and we have been able to identify several gene and protein interactions that are known from the literature. In an application to data from a topical study in synthetic biology (Cantone et al., 2009), our methods have outperformed two established network reconstruction methods from computational systems biology, and information sharing has allowed us to reconstruct the true underlying gene network with higher overall precision and recall than would have been possible without it.

We have investigated the performance of our methods on datasets which arise from gene regulatory networks with temporal changes in the structure of the network. There are several special cases of this situation which merit further discussion. The simplest case occurs when the changes of the underlying process are limited to parameter changes, and the true structure of the network remains constant. We have shown in Section 4.5.1 that our methods can deal with this situation effectively thanks to information sharing among segments. A more complicated case could involve a reoccurring event that causes certain gene interactions to switch on or off, leading to repeated network structures. For example, in a circadian clock system such as Locke et al. (2006); Pokhilko et al. (2010), the absence of sunlight might deactivate the interaction between two genes in the network, causing its structure to change from A to B[10]. If gene expression levels are measured both during the day and at night for three days, then we will observe a sequence like ABABAB. While our methods can in principle represent repeated segments, the multiple changepoint process was not designed with this in mind. A better model for repeated segments might be a Hidden Markov Model (HMM), where each hidden state corresponds to a network structure, and transitions between states correspond to changes in the structure, in the same vein as applied to changing tree structures in phylogeny (Husmeier and McGuire, 2003). The disadvantage of using HMMs is that they impose a geometric distribution on the segment lengths, and in that respect our changepoint process is more flexible. To have the same flexibility with HMMs, model extensions along the lines of hierarchical HMMs or HMMs with weighting times could be pursued, as known from speech processing, but this would come at significantly increased computational costs. Hence, this approach only appears to make sense if there is strong prior indication that repetitions occur.

An interesting topic for future work is to investigate other functional forms of the information sharing mechanism. In our work, we have investigated four different models, based on an exponential versus binomial distribution, with or without gene-specific hyperparameters. It has recently come to our attention that Wang et al. (2011) have experimented with a different approach, which effectively combines our exponential prior with an additional factor that encourages network sparsity. Sparsity in our model is encouraged by the truncated Poisson prior of equation (3.4), as explained in the paragraph under equation (3.30). It would be interesting to explore the effect of the additional factor used in equation (7) of Wang et al. (2011) in the context of gene network reconstruction.

Reconstructing gene regulatory networks from transcriptional profiles remains a challenging problem, which a flurry of ongoing methodological developments in the

[10] Note that our definition of a deactivated gene interaction includes interactions that no longer occur because one of the interacting genes is no longer expressed.

computational systems biology community are trying to address. I believe that our work adds a valuable contribution to this field, by presenting a consistent and flexible Bayesian model for the case where the network structures change over time.

Chapter 5

ODE network parameter inference using adaptive gradient matching

Note: Section 5.5 is partly based on the extended conference abstract "Parameter Inference in Mechanistic Models of Cellular Regulation and Signalling Pathways using Gradient Matching" (Dondelinger et al., 2012c), although the simulation results have been updated since the publication of that abstract.

5.1 Introduction

The previous chapters of this thesis have mainly been concerned with structure inference in biological networks. The interactions among entities in the networks have been the main focus of study, with the parameters either integrated out, or treated as informative by-products of the network inference. I have shown that this approach leads to robust models for determining the network structure from data. Once a network structure has been inferred, systems biologists may use the information gleaned from it to build a mechanistic model that allows them to make predictions about the dynamics of the system. These models will have many unknown parameters, which need to be determined before any predictions can be made. For this reason, I will now shift the focus from probabilistic models for network structure inference, to mechanistic models for network parameter inference. In particular, I want to look at systems of ordinary differential equations (ODEs), and how we can efficiently infer the parameters that govern their behaviour from data.

In many domains of applications, ordinary differential equations (ODEs) are a useful tool for modelling the behaviour of a system. Systems where they have been applied range from physics and engineering to ecology (Lotka, 1932), and recently, systems biology (see e.g. De Jong, 2002). In systems biology, ODEs have been used to describe the dynamics of pathways and gene regulatory interactions in the cell (Pokhilko et al., 2010). Frequently, molecular biologists will have sufficient knowledge about a system to define the equations that govern its behaviour, but there will be uncertainty about the kinetic or thermodynamic parameters. A common way to resolve this uncertainty is to use some form of parameter inference based on the available experimental data (Ashyraliyev et al., 2009). Previous approaches to parameter inference in ODEs have ranged from maximum likelihood over variational approximations to Markov Chain Monte Carlo (MCMC). Generally, all of these approaches involve explicitly solving the ODE system at each inference step to evaluate how well the inferred parameter values match the data. As this incurs a computational cost at each step, which grows linearly with the dataset size and size of the system, alternatives have been developed that avoid explicitly solving the system of differential equations (Varah, 1982; Poyton et al., 2006; Ramsay et al., 2007; Calderhead et al., 2008). These alternatives work by interpolating the signal from the observed experimental data and calculating the gradients, to which the ODE system can then be fitted directly.

One recent approach is described in Calderhead et al. (2008). This approach uses Gaussian Processes (GPs) to model the experimental data, which has the advantage that all the parameters can be inferred from the data. A disadvantage of the method proposed in Calderhead et al. (2008) is that the hyperparameters of the Gaussian process are inferred based on the data alone, without any rectifying feedback mechanism from the ODE system. This falls short of related previous approaches, like Ramsay et al. (2007). While the approach in Calderhead et al. (2008) generally works well for the limiting case of zero noise, we have observed that it tends to lead to rather poor parameter estimation from data subject to noise. In the present chapter, I propose an improved inference scheme, which I call adaptive gradient matching (AGM). In this scheme, both the hyperparameters of the Gaussian process as well as the ODE parameters are jointly and consistently inferred from the posterior distribution, leading to an essential information coupling between both, by taking account of their correlation. The scheme is adaptive, in that unlike in Calderhead et al. (2008), the GP is adapted during the inference based on information from the ODE system. I demonstrate that this leads to a significant improvement in the robustness with respect to noise.

I will describe the original scheme from Calderhead et al. (2008) in more detail in Section 5.2.1, and present our adaptive gradient matching approach to solving the problems with this scheme in Section 5.2.2. Section 5.3 describes the sampling pro-

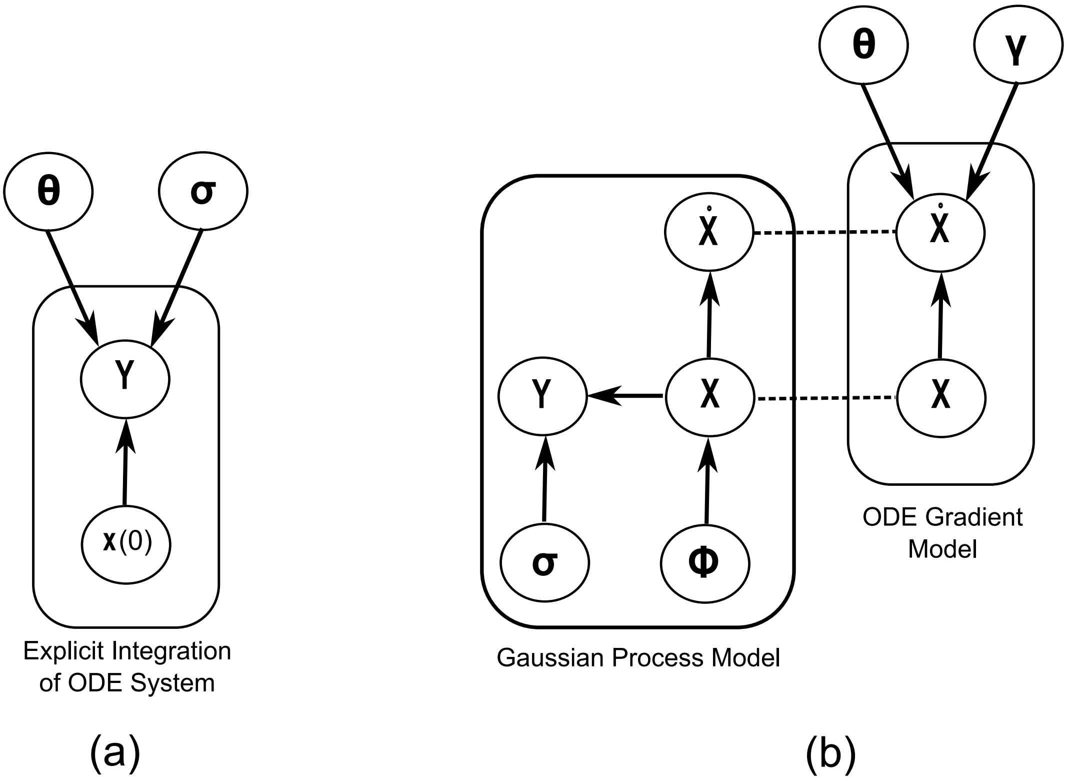


Figure 5.1: Graphical representation of the explicit ODE integration model (a), and the Calderhead et al. (2008) and adaptive gradient matching models (b). This figure has been adapted from Calderhead et al. (2008). Note that graphically, the models in Sections 5.2.1 and 5.2.2 are identical; the difference is in the inference scheme. Dashed lines denote the product of experts combination of gradient models, as in Calderhead et al. (2008).

cedure, and Section 5.4.1 describes three models that we used for benchmarking the performance of adaptive gradient matching. In Section 5.4.2, I describe the effect of different covariance functions for the GP. Our experiments and results for parameter inference on the benchmark models are described in Section 5.4.3, while results for speed and computational complexity are presented in Section 5.4.4. Finally, Section 5.5 describes an application to a detailed model of the JAK/STAT pathway. The chapter concludes with a discussion in Section 5.6.

5.2 Method

5.2.1 Original Proposal by Calderhead et al. (2008)

Consider a set of T arbitrary time points $t_1 < \ldots < t_T$, and a sequence of noisy observations $\mathbf{Y} = (\mathbf{y}(t_1), ..., \mathbf{y}(t_T))$,

$$\mathbf{y}(t) = \mathbf{x}(t) + \varepsilon(\mathbf{t}) \tag{5.1}$$

of a K-dimensional process $\mathbf{X} = (\mathbf{x}(t_1), ..., \mathbf{x}(t_T))$, $\dim[\mathbf{x}(t)] = \dim[\mathbf{y}(t)] = \dim[\varepsilon(\mathbf{t})] = K$, whose evolution is defined by a system of K ordinary differential equations (ODEs):

$$\dot{\mathbf{x}}(t) \;=\; \frac{d\mathbf{x}(t)}{dt} \;=\; \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}, t); \quad \mathbf{x}(t_1) \;=\; \mathbf{x}_1 \tag{5.2}$$

with parameter vector $\boldsymbol{\theta}$ of length P, and ε is a multivariate Gaussian noise process $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$, where $D_{ik} = \sigma_k^2 \delta_{ik}$, i.e. for simplicity we assume the covariance matrix $\mathbf{D}$ to be diagonal:

$$P(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}) \;=\; \prod_k \prod_t P(y_k(t)|x_k(t), \sigma_k) \;=\; \prod_k \prod_t \mathcal{N}(y_k(t)|x_k(t), \sigma_k^2) \tag{5.3}$$

The matrices $\mathbf{X}$ and $\mathbf{Y}$ are of dimension K-by-T. Let $\mathbf{x}_k$ and $\mathbf{y}_k$ denote T-dimensional column vectors that contain the kth row of the matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. Hence, $\mathbf{x}_k$ and $\mathbf{y}_k$ represent the respective time series of the kth state.

Given that any inference based on an explicit numerical integration of the differential equations, as pursued in Vyshemirsky and Girolami (2008), tends to incur high computational costs, an alternative approach based on non-parametric Bayesian modelling with Gaussian processes was proposed in Calderhead et al. (2008). The idea is to put a Gaussian process prior on $\mathbf{x}_k$,

$$p(\mathbf{x}_k|\boldsymbol{\phi}_k) \;=\; \mathcal{N}(\mathbf{x}_k|\mathbf{0}, \mathbf{C}_{\boldsymbol{\phi}_\mathbf{k}}) \tag{5.4}$$

where $\mathbf{C}_{\boldsymbol{\phi}_\mathbf{k}}$ denotes a positive definite matrix of covariance functions with hyperparameters $\boldsymbol{\phi}_k$. Assuming additive Gaussian noise with a state-specific error variance σ_k^2, we get:

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{x}_k, \sigma_k) \;&=\; \mathcal{N}(\mathbf{y}_k|\mathbf{x}_k, \sigma_k^2\mathbf{I}) && (5.5) \\ p(\mathbf{y}_k|\boldsymbol{\phi}_k, \sigma_k) \;&=\; \int p(\mathbf{y}_k|\mathbf{x}_k, \sigma_k) p(\mathbf{x}_k|\boldsymbol{\phi}_k) d\mathbf{x}_k \\ &=\; \int \mathcal{N}(\mathbf{y}_k|\mathbf{x}_k, \sigma_k^2\mathbf{I}) \mathcal{N}(\mathbf{x}_k|\mathbf{0}, \mathbf{C}_{\boldsymbol{\phi}_\mathbf{k}}) d\mathbf{x}_k \;=\; \mathcal{N}(\mathbf{y}_k|\mathbf{0}, \mathbf{C}_{\boldsymbol{\phi}_\mathbf{k}} + \sigma_k^2\mathbf{I}) && (5.6) \end{aligned}$$

The conditional distribution for the state derivatives is given by

$$p(\dot{\mathbf{x}}_k|\mathbf{x}_k,\boldsymbol{\phi}) = \mathcal{N}(\mathbf{m}_k,\mathbf{A}_k) \tag{5.7}$$

where

$$\mathbf{m}_k = {}'\mathbf{C}_{\phi_\mathbf{k}}\mathbf{C}_{\phi_\mathbf{k}}{}^{-1}\mathbf{x}_k; \quad \mathbf{A}_k = \mathbf{C}''_{\phi_\mathbf{k}} - {}'\mathbf{C}_{\phi_\mathbf{k}}\mathbf{C}_{\phi_\mathbf{k}}{}^{-1}\mathbf{C}'_{\phi_\mathbf{k}} \tag{5.8}$$

Here, the matrix $\mathbf{C}''_{\phi_\mathbf{k}}$ denotes the auto-covariance for each state derivative, and the matrices $\mathbf{C}'_{\phi_\mathbf{k}}$ and ${}'\mathbf{C}_{\phi_\mathbf{k}}$ denote the cross-covariances between the kth state and its derivative. A derivation can be found in Rasmussen and Williams (2006), from which we can derive that:

$$\mathbf{C}'_{\phi_\mathbf{k}}(i,j) = \frac{d\mathcal{K}(x_{k,t_i},x_{k,t_j})}{dx_{k,t_i}} \tag{5.9}$$

$${}'\mathbf{C}_{\phi_\mathbf{k}}(i,j) = \frac{d\mathcal{K}(x_{k,t_i},x_{k,t_j})}{dx_{k,t_j}} \tag{5.10}$$

$$\mathbf{C}''_{\phi_\mathbf{k}}(i,j) = \frac{d^2\mathcal{K}(x_{k,t_i},x_{k,t_j})}{dx_{k,t_i}dx_{k,t_j}} \tag{5.11}$$

where $x_{k,t_i} = x_k(t_i)$, $x_{k,t_j} = x_k(t_j)$ and $\mathcal{K}(x_{k,t_i},x_{k,t_j})$ is the chosen covariance function for the Gaussian process. Specific expressions for the two kernel functions considered in this work can be found in Section 5.2.3.

Assuming additive Gaussian noise with a state-specific error variance γ_k, one gets from (5.2):

$$p(\dot{\mathbf{x}}_k|\mathbf{X},\boldsymbol{\theta},\gamma_k) = \mathcal{N}(\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta}),\gamma_k\mathbf{I}) \tag{5.12}$$

where where $\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta}) = \{\mathbf{f}_k(\mathbf{x}_k(t),\boldsymbol{\theta},t)\}_{t\in 1:T}$. Next, the approach taken in Calderhead et al. (2008) is to combine (5.7) and (5.12) with a product of experts approach:

$$\begin{aligned}
p(\boldsymbol{\theta},\boldsymbol{\gamma}|\mathbf{X},\boldsymbol{\phi}) &= \int p(\dot{\mathbf{X}},\boldsymbol{\theta},\boldsymbol{\gamma}|\mathbf{X},\boldsymbol{\phi})d\dot{\mathbf{X}} \qquad (5.13)\\
&\propto p(\boldsymbol{\theta})p(\boldsymbol{\gamma})\int p(\dot{\mathbf{X}},\mathbf{X},\boldsymbol{\phi}|\boldsymbol{\theta},\boldsymbol{\gamma})d\dot{\mathbf{X}}\\
&\propto p(\boldsymbol{\theta})p(\boldsymbol{\gamma})\prod_k\int p(\dot{\mathbf{x}}_k|\mathbf{x}_k,\boldsymbol{\phi})p(\dot{\mathbf{x}}_k|\mathbf{X},\boldsymbol{\theta},\gamma_k)d\dot{\mathbf{x}}_k\\
&= p(\boldsymbol{\theta})p(\boldsymbol{\gamma})\prod_k\int \mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{m}_k,\mathbf{A}_k)\mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta}),\gamma_k\mathbf{I})d\dot{\mathbf{x}}_k\\
&\propto \frac{p(\boldsymbol{\theta})p(\boldsymbol{\gamma})}{\prod_k Z(\gamma_k)}\exp\left\{-\frac{1}{2}\sum_k(\mathbf{f}_k-\mathbf{m}_k)^\mathsf{T}(\mathbf{A}_k+\gamma_k\mathbf{I})^{-1}(\mathbf{f}_k-\mathbf{m}_k)\right\}
\end{aligned}$$

where $p(\boldsymbol{\theta})$ and $p(\boldsymbol{\gamma})$ are the prior distributions on $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, $Z(\gamma_k) = |2\pi(\mathbf{A}_k+\gamma_k\mathbf{I})|^{1/2}$ and we have defined $\boldsymbol{\gamma} = (\gamma_1,\ldots,\gamma_K)$ and $\mathbf{f}_k = \mathbf{f}_k(\mathbf{X},\boldsymbol{\theta})$. We can view γ_k as the mismatch between the ODE and Gaussian process parts of the model. In Calderhead et al. (2008) and in our work, γ_k is inferred from the data, based on the intuition that there will always be some mismatch between the specified ODE system and the Gaussian process fit to the data in realistic applications. This could potentially lead to non-identifiability issues: high values of the mismatch parameter can give more weight to the GP fit, while low values give more weight to the ODE system. Thus, if there is no clear optimal posterior mode for γ_k, the likelihood landscape could present a valley, with smooth transitions between two regimes of equal likelihood. As an alternative to inferring γ_k from the data, one can view it as a tuning parameter during inference, starting off at a larger value to allow for exploration of the model space, and then gradually being constrained towards zero to enforce agreement between the ODE system and the GP model. This approach has been pursued in Campbell and Steele (2012), as discussed in Section 5.6.

Inference is based on sampling the parameters of the ODEs $\boldsymbol{\theta}$, the hyperparameters of the Gaussian process $\boldsymbol{\phi}$, the noise variances $\boldsymbol{\gamma}, \boldsymbol{\sigma}$, and the state variables $\mathbf{X}$ from the posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\phi}, \boldsymbol{\sigma}, \mathbf{X}|\mathbf{Y})$ with the following Gibbs sampling procedure:

$$\begin{aligned} \boldsymbol{\phi}, \boldsymbol{\sigma} &\sim p^*(\boldsymbol{\phi}, \boldsymbol{\sigma}|\mathbf{Y}) && (5.14) \\ \mathbf{X} &\sim p(\mathbf{X}|\mathbf{Y}, \boldsymbol{\sigma}, \boldsymbol{\phi}) && (5.15) \\ \boldsymbol{\theta}, \boldsymbol{\gamma} &\sim p(\boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{X}, \boldsymbol{\phi}) && (5.16) \end{aligned}$$

The distribution in the last sampling step, (5.16), is given by (5.13). Note that $p(\boldsymbol{\phi}, \boldsymbol{\sigma}|\mathbf{Y}) = \int p(\dot{\mathbf{X}}, \boldsymbol{\phi}, \boldsymbol{\sigma}, \boldsymbol{\theta}, \boldsymbol{\gamma}|\mathbf{Y}) d\dot{\mathbf{X}} d\boldsymbol{\theta} d\boldsymbol{\gamma}$ is analytically intractable. Calderhead et al. (2008) therefore approximate $p(\boldsymbol{\phi}, \boldsymbol{\sigma}|\mathbf{Y})$ by a distribution derived from a standard Gaussian process that is decoupled from the rest of the model. We call this $p^*(\boldsymbol{\phi}, \boldsymbol{\sigma}|\mathbf{Y})$. This distribution does not have a standard form, and sampling from it directly is infeasible. Hence, MCMC with the Metropolis-Hastings algorithm (Hastings, 1970) is used. The sampling steps (5.14) and (5.15) are broken up into the contributions from the individual states k:

$$\begin{aligned} \boldsymbol{\phi}_k, \sigma_k \sim p^*(\boldsymbol{\phi}_k, \sigma_k|\mathbf{y}_k) &\propto p(\mathbf{y}_k|\boldsymbol{\phi}_k, \sigma_k) p(\boldsymbol{\phi}_k) p(\sigma_k) \\ &= \mathcal{N}(\mathbf{y}_k|\mathbf{0}, \sigma_k^2 \mathbf{I} + \mathbf{C}_{\phi_\mathbf{k}}) p(\boldsymbol{\phi}_k) p(\sigma_k) && (5.17) \\ \mathbf{x}_k &\sim p(\mathbf{x}_k|\mathbf{y}_k, \sigma_k, \boldsymbol{\phi}_k) = \mathcal{N}(\mathbf{x}_k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) && (5.18) \end{aligned}$$

where $\boldsymbol{\mu}_k = \mathbf{C}_{\phi_\mathbf{k}}(\mathbf{C}_{\phi_\mathbf{k}} + \sigma_k^2 \mathbf{I})^{-1} \mathbf{y}_k$ and $\boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{C}_{\phi_\mathbf{k}}(\mathbf{C}_{\phi_\mathbf{k}} + \sigma_k^2 \mathbf{I})^{-1}$. Equation (5.18) follows from $p(\mathbf{x}_k|\mathbf{y}_k, \sigma_k, \boldsymbol{\phi}_k) = p(\mathbf{y}_k|\mathbf{x}_k, \sigma_k) p(\mathbf{x}_k|\boldsymbol{\phi}_k) / p(\mathbf{y}_k|\sigma_k, \boldsymbol{\phi}_k)$, equations (5.4–5.6) are well-established results for Gaussian distributions. Sampling of the vector of latent variables $\mathbf{x}_k$ in (5.18) follows directly from a multivariate Gaussian distribution. For sampling $\boldsymbol{\phi}_k$ and σ_k in (5.17), one again has to resort to MCMC. The overall MCMC scheme then iteratively loops through the steps (5.14–5.16) until some convergence criterion has been met.[1] However, the approximation in equation (5.14) of the sampling scheme introduces a certain weakness: the parameters of the ODE systems, $\boldsymbol{\theta}, \boldsymbol{\gamma}$, which are sampled in the third step of the Gibbs sampling routine (5.16), never feed back into the first and second steps, (5.14–5.15). This implies that $\boldsymbol{\theta}, \boldsymbol{\gamma}$ have no bearing on the inference of the state variables $\mathbf{X}$; these state variables are solely inferred from the observed data via a standard Gaussian process interpolation, (5.14–5.15).

To paraphrase this: the method proposed in Calderhead et al. (2008) is a two-step procedure, in which first an interpolation problem is solved, corresponding to (5.14–5.15), and then the parameters of the ODEs are inferred by matching the derivatives of the interpolant with those predicted from the ODEs, via (5.16). This falls short of the method proposed in Ramsay et al. (2007), where the interpolation fits both the noisy data and the derivatives from the ODEs simultaneously, allowing the system of ODEs to feed back onto the interpolation.

5.2.2 Adaptive Gradient Matching

In order to address the issues with the model described in the previous section, we need to close the feedback loop between interpolation and parameter estimation of the ODEs. We will develop a new method, which we call adaptive gradient matching (AGM). The new method will have the same graphical structure (shown in Figure 5.1) as the model in the previous section. The innovation of AGM is a mathematically more consistent formulation of the inference procedure that allows us to jointly estimate all parameters of the model, and thereby close the feedback loop.

Following Calderhead et al. (2008), we combine (5.7) and (5.12) with a product of experts approach:

$$\begin{aligned} p(\dot{\mathbf{x}}_k|\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\gamma_k) &\propto p(\dot{\mathbf{x}}_k|\mathbf{x}_k,\boldsymbol{\phi})p(\dot{\mathbf{x}}_k|\mathbf{X},\boldsymbol{\theta},\gamma_k) \\ &= \mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{m}_k,\mathbf{A}_k)\mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta}),\gamma_k\mathbf{I}) \end{aligned} \tag{5.19}$$

[1] Note that the method proposed in Calderhead et al. (2008) slightly deviates from the summary given here in that the definition (5.8) is modified as follows: $\mathbf{m}_k = {}'\mathbf{C}_{\phi_\mathbf{k}}[\mathbf{C}_{\phi_\mathbf{k}} + \sigma_k^2\mathbf{I}]^{-1}\mathbf{x}_k$ and $\mathbf{A}_k = \mathbf{C}''_{\phi_\mathbf{k}} - {}'\mathbf{C}_{\phi_\mathbf{k}}[\mathbf{C}_{\phi_\mathbf{k}} + \sigma_k^2\mathbf{I}]^{-1}\mathbf{C}'_{\phi_\mathbf{k}}$, which leads to the dependence of (5.13) on $\boldsymbol{\sigma}$. However, this modification, which is motivated by including information from the data $\mathbf{Y}$, is methodologically inconsistent.

We obtain for the joint distribution:

$$\begin{aligned} p(\dot{\mathbf{X}},\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma}) &= p(\dot{\mathbf{X}}|\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma})p(\mathbf{X}|\boldsymbol{\phi})p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\gamma}) \\ &= p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\gamma})\prod_k p(\dot{\mathbf{x}}_k|\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\gamma_k)p(\mathbf{x}_k|\boldsymbol{\phi}_k) \qquad (5.20) \end{aligned}$$

Inserting (5.4) and (5.13), we get:

$$\begin{aligned} p(\dot{\mathbf{X}},\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma}) &\propto p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\gamma})\prod_k \mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{m}_k,\mathbf{A}_k) \\ &\quad \mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta}),\gamma_k\mathbf{I})\mathcal{N}(\mathbf{x}_k|\mathbf{0},\mathbf{C}_{\phi_\mathbf{k}}) \qquad (5.21) \end{aligned}$$

The marginalization over the state derivatives $\dot{\mathbf{X}}$

$$\begin{aligned} p(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma}) &= \int p(\dot{\mathbf{X}},\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma})d\dot{\mathbf{X}} \qquad (5.22) \\ &\propto p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\gamma})\prod_k \mathcal{N}(\mathbf{x}_k|\mathbf{0},\mathbf{C}_{\phi_\mathbf{k}}) \\ &\quad \int \mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{m}_k,\mathbf{A}_k)\mathcal{N}(\dot{\mathbf{x}}_k|\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta}),\gamma_k\mathbf{I})d\dot{\mathbf{x}}_k \end{aligned}$$

is analytically tractable and yields:

$$\begin{aligned} p(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma}) &\propto p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\gamma})p(\mathbf{X}|\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma}) \qquad (5.23) \\ &\propto \prod_k \mathcal{N}(\mathbf{x}_k|\mathbf{0},\mathbf{C}_{\phi_\mathbf{k}})\exp\left[-\frac{1}{2}(\mathbf{f}_k-\mathbf{m}_k)^\mathsf{T}(\mathbf{A}_k+\gamma_k\mathbf{I})^{-1}(\mathbf{f}_k-\mathbf{m}_k)\right] \\ &\propto \exp\left[-\frac{1}{2}\sum_k\left(\mathbf{x}_k^\mathsf{T}\mathbf{C}_{\phi_k}^{-1}\mathbf{x}_k+(\mathbf{f}_k-\mathbf{m}_k)^\mathsf{T}(\mathbf{A}_k+\gamma_k\mathbf{I})^{-1}(\mathbf{f}_k-\mathbf{m}_k)\right)\right] \end{aligned}$$

where, as before, we use $\mathbf{f}_k$ as shorthand for $\mathbf{f}_k(\mathbf{X},\boldsymbol{\theta},\mathbf{t})$, and $\mathbf{m}_k$ and $\mathbf{A}_k$ were defined in (5.8). Note that this distribution is a complicated function of the states $\mathbf{X}$, owing to the nonlinear dependence via $\mathbf{f}_k = \mathbf{f}_k(\mathbf{X},\boldsymbol{\theta},\mathbf{t})$. For the joint probability distribution of the whole system we obtain:

$$p(\mathbf{Y},\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma},\boldsymbol{\sigma}) = p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma})p(\mathbf{X}|\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma})p(\boldsymbol{\theta})p(\boldsymbol{\phi})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}) \qquad (5.24)$$

where the first factor, $p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma})$, was defined in (5.3), and the second factor is given by (5.23). Note that the functional form of the second term is defined up to an unknown normalization constant. To bypass the problem of normalizing the distribution (5.23), we follow a Metropolis-Hastings scheme. Denote by $q_1(\boldsymbol{\sigma})$, $q_2(\boldsymbol{\phi})$, $q_3(\mathbf{x}_k)$, $q_4(\boldsymbol{\theta})$ and $q_5(\boldsymbol{\gamma})$ the proposal distributions for the inferred parameters. We propose new values from these distributions; q_1 and q_5 are sparse exponential distributions with $\lambda = 10$ to ensure small noise values and q_2, q_3 and q_4 are uniform distributions over the intervals

$[0,100]$, $[0,10]$ and $[0,20]$, respectively. These proposal distributions correspond to the prior distributions for the parameters in our model, except for $\boldsymbol{\sigma}$ where we use a sparse gamma prior $\Gamma(1,1)$, and $\boldsymbol{\theta}$, where we have imposed a gamma distribution $\Gamma(4,0.5)$ as a prior to encode our prior belief about parameter values, which is that most parameters will be > 0 and < 5.

We then accept or reject these proposal moves according to the standard Metropolis-Hastings criterion (Hastings, 1970):

$$P_{accept} = \left\{1, \frac{p(\mathbf{Y},\tilde{\mathbf{X}},\tilde{\theta},\tilde{\phi},\tilde{\gamma},\tilde{\sigma})}{p(\mathbf{Y},\mathbf{X},\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\gamma},\boldsymbol{\sigma})} \frac{q_1(\boldsymbol{\sigma})q_2(\boldsymbol{\phi})q_4(\boldsymbol{\theta})q_5(\boldsymbol{\gamma})\prod_k q_3(\mathbf{x}_k)}{q_1(\tilde{\sigma})q_2(\tilde{\phi})q_4(\tilde{\theta})q_5(\tilde{\gamma})\prod_k q_3(\tilde{\mathbf{x}}_k)}\right\} \tag{5.25}$$

For improved mixing and convergence, it is advisable to not propose all moves simultaneously, but to apply a blocking strategy and employ a Gibbs sampling scheme. We do not make that explicit in our notation, though. The effect of (5.25) is that the parameters $\boldsymbol{\theta}$ have an influence on the acceptance probabilities for $\mathbf{X}$. This mechanism closes the feedback loop, with the system of ODEs acting back in an adaptive manner on the interpolants $\mathbf{x}_k$ via the parameters $\boldsymbol{\theta}$. In this way, we address the main shortcoming of the method proposed in Calderhead et al. (2008).

5.2.3 Covariance Functions and Derivatives

The flexibility and dynamics of a Gaussian process response model depend on the choice of covariance function. In this work, we considered two alternatives: the radial basis covariance function, and the sigmoid covariance function. For other possible choices of covariance functions and more information on this topic, see Chapter 4 in Rasmussen and Williams (2006).

The radial basis function (RBF) covariance function is defined as:

$$\mathcal{K}_{rbf}(t,t') = \sigma^2_{rbf} \exp(-0.5 * (t-t')^2/l^2) \tag{5.26}$$

with hyperparameters σ^2_{rbf} and l^2 (variance and characteristic lengthscale). This is probably the most well-known covariance functions for Gaussian processes. It is infinitely differentiable, and results in a response that is very smooth.

The sigmoid covariance function is defined as:

$$\mathcal{K}_{sig}(t,t') = \sigma^2_{sig} \arcsin\left(\frac{a+b*t*t'}{\sqrt{(a+b*t*t+1)(a+b*t'*t'+1)}}\right) \tag{5.27}$$

with hyperparameters σ^2_{sig}, a and b. This is also sometimes known as the "neural network" covariance function, because it arises from a neural network with one hidden layer. For the derivation, see Chapter 4 in Rasmussen and Williams (2006). This covariance function has the advantage that it is non-stationary, and so can model dynamics with varying length-scale. In particular, the sigmoid covariance function is well-suited for modelling signals with high variance close to zero, and low variance further away from zero. This can be advantageous for modelling biological systems that start in a perturbed state before eventually reaching a steady state with saturation.

In order to calculate the auto- and cross-covariance matrices in Equation (5.8), we require the partial and full derivatives of these functions. For the RBF covariance function, we obtain:

$$\frac{d\mathcal{K}_{rbf}(t,t')}{dt} = -\frac{(t-t')}{l^2}\mathcal{K}_{rbf}(t,t') \tag{5.28}$$

$$\frac{d\mathcal{K}_{rbf}(t,t')}{dt'} = \frac{(t-t')}{l^2}\mathcal{K}_{rbf}(t,t') \tag{5.29}$$

$$\frac{d^2\mathcal{K}_{rbf}(t,t')}{dtdt'} = \left(\frac{1}{l^2}-\frac{(t-t')^2}{l^4}\right)\mathcal{K}_{rbf}(t,t') \tag{5.30}$$

For the sigmoid covariance function, we obtain:

$$\frac{d\mathcal{K}_{sig}(t,t')}{dt} = \frac{\sigma^2_{sig}}{\sqrt{1-Z^2}}\frac{dZ}{dt} \tag{5.31}$$

$$\frac{d\mathcal{K}_{sig}(t,t')}{dt'} = \frac{\sigma^2_{sig}}{\sqrt{1-Z^2}}\frac{dZ}{dt'} \tag{5.32}$$

$$\frac{d^2\mathcal{K}_{sig}(t,t')}{dtdt'} = \frac{\sigma^2_{sig}}{\sqrt{1-Z^2}}\left(\frac{Z}{1-Z^2}\frac{dZ}{dt'}\frac{dZ}{dt}+\frac{d^2Z}{dtdt'}\right) \tag{5.33}$$

where:

$$Z = \frac{a+b*t*t'}{Z_{norm}} \tag{5.34}$$

with $Z_{norm} = \sqrt{(a+b*t*t+1)(a+b*t'*t'+1)}$, and we have:

$$\frac{dZ}{dt} = b\left(\frac{t'}{Z_{norm}}-\frac{tZ}{a+b*t*t+1}\right) \tag{5.35}$$

$$\frac{dZ}{dt'} = b\left(\frac{x}{Z_{norm}}-\frac{t'Z}{a+b*t'*t'+1}\right) \tag{5.36}$$

$$\frac{d^2Z}{dtdt'} = b\left(\frac{1}{Z_{norm}}-\frac{bt't'}{(a+b*t'*t'+1)Z_{norm}}-\frac{t}{(a+b*t*t+1)}\frac{dZ}{dt'}\right) \tag{5.37}$$

5.3 Sampling Setup

For running simulations with the model in Calderhead et al. (2008), we make use of the Matlab code provided by the authors. Our adaptive gradient matching model was implemented in R, where we followed the sampling scheme from Calderhead et al. (2008) whenever possible. Like Calderhead et al., we used population MCMC (Jasra et al., 2007) to deal with the potentially rugged likelihood landscapes of the non-linear ODE systems. For all MCMC simulations in this paper, we ran 10 chains at different temperatures, which we tuned during the burn-in phase to achieve an acceptance rate of 0.25 for exchange moves. Similarly, proposal widths for all parameters and hyperparameters were tuned to achieve an acceptance rate of 0.25. We initialised $\mathbf{X}$ and $\boldsymbol{\phi}$ using a GP regression fit to the data $\mathbf{Y}$. To obtain the regression fit, $\boldsymbol{\phi}$ was optimised for maximum likelihood, using scaled conjugate gradients. I implemented this procedure with help of the R package `gptk`, extended with my own code to include the sigmoidal covariance function. The same initial GP hyperparameters were used for the Calderhead et al. model and for our improved gradient matching model. All other parameters were initialised by drawing samples from the prior distributions defined in Section 5.2.2.

The sampling of the hyperparameters $\boldsymbol{\phi}$ and the latent variables $\mathbf{X}$ warrants further explanation. Although we could in principle propose new values for $\mathbf{X}$ and $\boldsymbol{\phi}$ by sampling them alternately from the prior, or from some other distribution, e.g. via a random walk, this is highly inefficient due to the strong coupling between them. To avoid this problem, we apply a whitening of the prior, following Murray and Adams (2010). We introduce an independent Gaussian vector ν, and update the hyperparameters $\boldsymbol{\phi}$ for fixed ν instead of fixed $\mathbf{X}$, by using the transformation $\mathbf{X} = L_{\mathbf{C}_{\phi_\mathbf{k}}}\nu$, where $L_{\mathbf{C}_{\phi_\mathbf{k}}} L^{\top}_{\mathbf{C}_{\phi_\mathbf{k}}} = \mathbf{C}_{\phi_\mathbf{k}}$. Since ν and $\boldsymbol{\phi}$ are independent, this scheme removes the problems created by strong coupling. Furthermore, these updates will change both $\mathbf{X}$ and $\boldsymbol{\phi}$; in effect, we are now treating the latent variables as ancillary to the GP hyperparameters.

For the GP methods, the choice of covariance function can be important, as the GP needs to be able to fit the dynamics of the data. For the *PIF4/5* model and the Lotka-Volterra model described in Section 5.4.1, we used the radial basis function covariance function (Equation (5.26)), which provided a good fit. However, this covariance function does not provide a good fit for data from the model for the signal transduction cascade (also described in Section 5.4.2). We therefore switched to the sigmoid covariance function (Equation (5.27)). Note that in general the sigmoid covariance function gives good regression fits for all models (see Section 5.4.2). Both of these covariance functions are defined in Section 5.2.3, along with their derivatives.

In addition to the scheme described in Section 5.2.2, we also implemented a sampler which uses the explicit integration of the ODE system. This sampler is based on the same population MCMC setup as above, but samples from the distribution: $P(\mathbf{Y},\boldsymbol{\theta}^*,\boldsymbol{\sigma}) = P(\mathbf{Y}|\boldsymbol{\theta}^*,\boldsymbol{\sigma})P(\boldsymbol{\theta}^*)P(\boldsymbol{\sigma})$, where $\boldsymbol{\theta}^*$ is the parameter vector for the ODE system, augmented with the initial concentrations for each species, and $P(\boldsymbol{\theta}^*)$ and $P(\boldsymbol{\sigma})$ are the priors defined in Section 5.2.2. Then we have $P(\mathbf{Y}|\boldsymbol{\theta}^*,\boldsymbol{\sigma}) = \prod_k \prod_t P(y_k(t)|\boldsymbol{\theta}^*,\sigma_k)$, with $P(y_k(t)|\boldsymbol{\theta}^*,\sigma_k) = \mathcal{N}(y_k(t)|x_k(t,\boldsymbol{\theta}^*),\sigma_k^2)$ where $x_k(t,\boldsymbol{\theta}^*)$ is the solution of the ODE system for species k at time t, given $\boldsymbol{\theta}^*$. Parameters corresponding to the initial concentrations are initialised using the observed concentrations at time $t = 0$ for each species.

5.4 Three benchmark ODE Systems

In this section, I present three small-to-medium-sized ODE models of biological systems, and use them to benchmark the parameter inference methods from this chapter.

5.4.1 ODE Model Description

The *PIF4/5* model. We apply our GP parameter inference method to a model for gene regulation of genes *PIF4* and *PIF5* by *TOC1* in the circadian clock gene regulatory network of *Arabidopsis thaliana*. The overall network is represented by the Locke 2-loop model (Locke et al., 2005), with fixed parameters that were originally inferred following Pokhilko et al. (2010). Only the parameters involved in regulation of *PIF4* and *PIF5* are inferred from the data using the methods described in this chapter. We simplify the model to represent genes *PIF4* and *PIF5* as a combined gene *PIF4/5*. We are interested in the promoter strength s, the rate constant K_d and Hill coefficient h of the regulation by $TOC1$, and the degradation rate d of the $PIF4/5$ mRNA. The regulation process is represented by the following ODE:

$$\frac{d[PIF4/5]}{dt} = s \cdot \frac{K_d^h}{K_d^h + [TOC1]^h} - d \cdot [PIF4/5] \tag{5.38}$$

where $[PIF4/5]$ and $[TOC1]$ represent the concentration of *PIF4/5* and *TOC1*, respectively.

For the experiments presented in this chapter, data were generated with parameters $\{s = 1, K_d = 0.46, h = 2, d = 1\}$, which generates concentrations that are close to real-life measurements of *PIF4* and *PIF5*. For each dataset, we simulated data over the interval $[0,24]$ with sampling intervals in $\{1,2,4\}$. The initial value for the *PIF4/5* concentration was taken from a measurement of Arabidopsis gene expressions at the beginning of the day, and was set to 0.386. We applied additive white Gaussian noise with standard deviation in $\{0,0.1\}$ to the time courses.

The Lotka-Volterra model. The Lotka-Volterra model is a 2-equation system that was originally developed for modelling predator-prey interaction in ecology (Lotka, 1932). There are two species, a prey species S (the 'sheep') and a predator species W (the 'wolves'). The dynamics of their interactions are described by the following system of two ODEs:

$$\begin{aligned} \frac{d[S]}{dt} &= [S] \cdot (\alpha - \beta \cdot [W]) \\ \frac{d[W]}{dt} &= -[W] \cdot (\gamma - \delta \cdot [S]) \end{aligned} \tag{5.39}$$

This system is of interest because it is periodic for most parameter settings, and there are non-linear interactions between the two species.

For the experiments presented in this chapter, data were generated with parameters $\{\alpha = 2, \beta = 1, \gamma = 4, \delta = 1\}$, which generates stable oscillations. For each dataset, we simulated data over the interval $[0,2]$ with sampling intervals of 0.25. The initial values for the prey species S and the predator species W were set at $[S] = 5$ and $[W] = 3$. We applied additive white Gaussian noise with standard deviation in $\{0,0.1,0.5\}$ to the time courses.

The signal transduction cascade. Our third and final model is a model of a signal transduction cascade that was described in Vyshemirsky and Girolami (2008) (Model 1). At the top of the cascade we have protein S, which can degrade into S_d. S activates protein R into active state Rpp by first binding to it to form RS, which is then activated to turn into Rpp. Rpp can degrade back into R, and RS can separate back into S and R. The model is described by the following system of five ODEs:

$$\begin{aligned} \frac{d[S]}{dt} &= -k_1 \cdot [S] - k_2 \cdot [S] \cdot [R] + k_3 \cdot [RS] \\ \frac{d[S_d]}{dt} &= k_1 \cdot [S] \\ \frac{d[R]}{dt} &= -k_2 \cdot [S] \cdot [R] + k_3 \cdot [RS] + \frac{V \cdot [Rpp]}{Km + [Rpp]} \\ \frac{d[RS]}{dt} &= k_2 \cdot [S] \cdot [R] - k_3 \cdot [RS] - k_4 \cdot [RS] \\ \frac{d[Rpp]}{dt} &= k_4 \cdot [RS] - \frac{V \cdot [Rpp]}{Km + [Rpp]} \end{aligned} \tag{5.40}$$

This system is of interest as it represents a realistic and commonly-used formulation of signal transduction as an ODE system, using mass action and Michaelis-Menten kinetics.

For the experiments presented in this chapter, data were generated with parameters $\{k_1 = 0.07, k_2 = 0.6, k_3 = 0.05, k_4 = 0.3, V = 0.017, Km = 0.3\}$, following Vyshemirsky and Girolami (2008). For each dataset, we simulated data over the interval $[0, 100]$ and took samples at time points $\{0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100\}$. This means that we sampled more timepoints during the earlier part of the timeseries, where the dynamics tend to be faster. We also followed Vyshemirsky and Girolami (2008) in setting the initial values for 5 species: $\{[S] = 1, [S_d] = 0, [R] = 1, [RS] = 0, [Rpp] = 0\}$. We applied additive white Gaussian noise with standard deviation in $\{0, 0.1\}$ to the time courses.

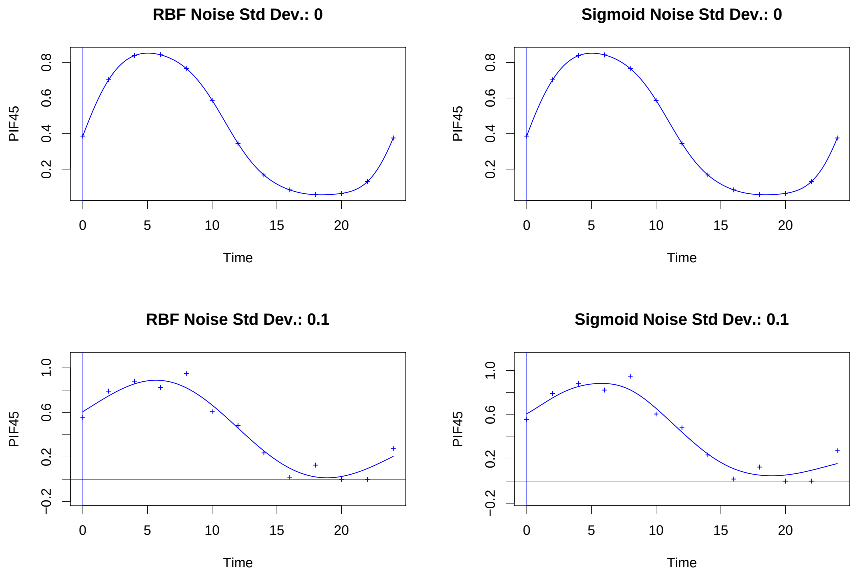


Figure 5.2: GP Regression fits to PIF4/5 expression levels, using the RBF and the sigmoidal covariance function. The crosses represent the data points, the solid line is the GP mean. Top Row: Gaussian noise with standard deviation 0. Bottom Row: Gaussian noise with standard deviation 0.1.

5.4.2 GP Covariance Function Comparison

The two covariance functions used in this work are the RBF (radial basis function) covariance function $\mathcal{K}_{rbf}(t, t')$ defined in Equation (5.26), and the sigmoid covariance function $\mathcal{K}_{sig}(t, t')$ defined in Equation (5.27). Figures 5.2 - 5.4 show a comparison of

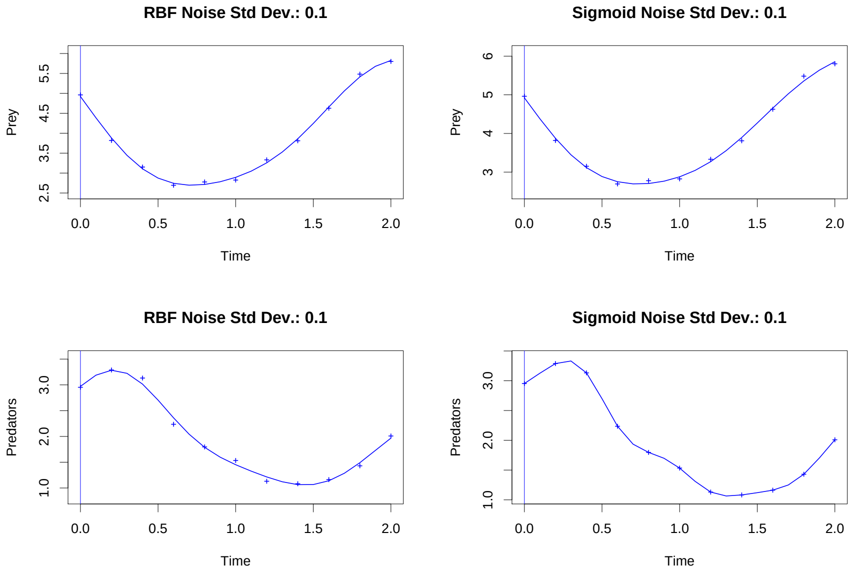


Figure 5.3: GP Regression fits to predator and prey concentrations in the Lotka-Volterra model, using the RBF and the sigmoidal covariance function. The crosses represent the data points, the solid line is the GP mean. Gaussian noise with standard deviation 0.1 was applied. Top Row: Prey species. Bottom Row: Predator Species.

the GP regression fits (using maximum likelihood, as described in Section 5.3) to data from the different model systems. We see that the sigmoid covariance function always provides a good fit, while the RBF covariance function breaks down for some of the species in the signal transduction cascade. This is due to the fact that the RBF covariance function assumes a fixed lengthscale l^2, while the sigmoid covariance function is non-stationary and can deal with varying lengthscales.

5.4.3 Parameter Inference Results

We use the three benchmark systems described in Section 5.4 to analyse the performance of our adaptive gradient matching, and to provide a thorough comparison with both the method by Calderhead et al. (2008), and the sampler which explicitly solves the ODE system, as described in Section 5.3.[2] We generated data from each system using the R package `deSolve` (Soetaert et al., 2010) for numerically integrating the systems of differential equations. See Section 5.4.1 for the parameter and initial concentration settings. We then added white Gaussian observation noise to the datasets.

[2] Note that due to the higher computational cost involved, we could only apply the explicit solver to the Lotka-Volterra model and the signal transduction cascade. Applying it to the *PIF4/5* system would have required solving the entire 14-equation system of the Locke 2-loop model at each step, which was not feasible with the time and resources at our disposal.

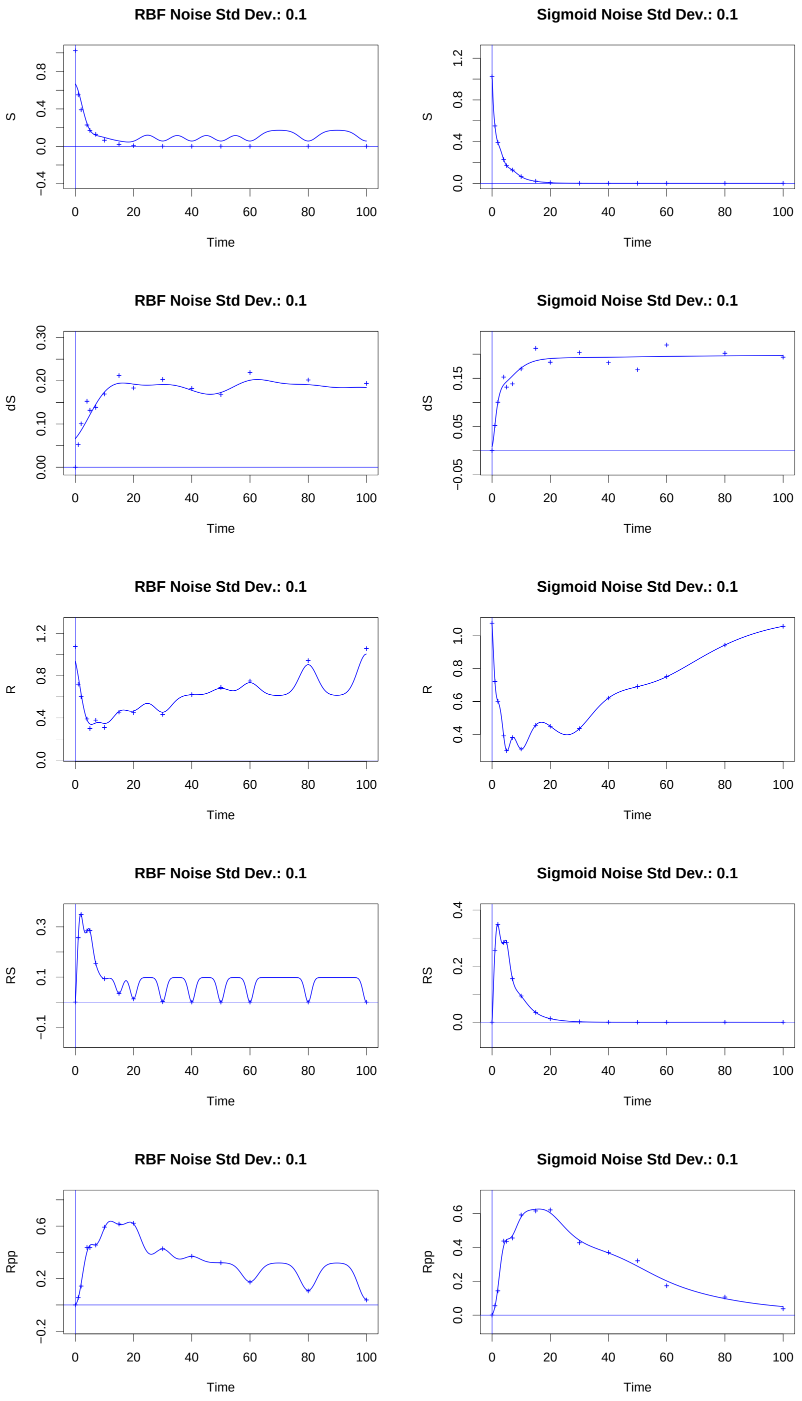


Figure 5.4: GP Regression fits to species concentrations in the signal transduction pathway, using the RBF and the sigmoidal covariance function. The crosses represent the data points, the solid line is the GP mean. Gaussian noise with standard deviation 0.1 was applied. From top to bottom, the rows show species S, dS, R, RS and Rpp.

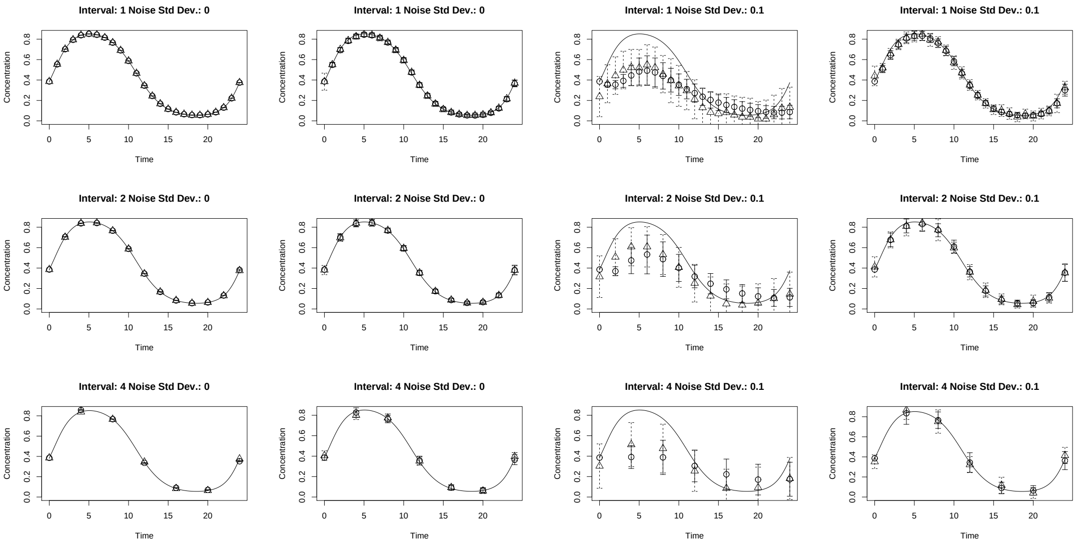


Figure 5.5: PIF4/5 expression levels with varying sampling intervals and observational noise. We show the true (noiseless) expression values, the sampled latent variables (triangles) and the expression profile simulated using the sampled $\boldsymbol{\theta}$ values (circles). Error bars show one standard deviation. First and Third Column: Calderhead et al. Model. Second and Fourth Column: Adaptive gradient matching.

For the *PIF4/5* system and the signal transduction cascade, we added noise with standard deviation $\in \{0, 0.1\}$, and for the Lotka-Volterra system we added noise with standard deviation $\in \{0, 0.1, 0.5\}$. The additional noise level for the Lotka-Volterra system reflects the higher amplitude of the signal in this system.

We generated 10 datasets for each noise level and system, before applying the parameter inference methods. Convergence was monitored via diagnostic plots and the Gelman and Rubin potential scale reduction factor (PSRF) (Gelman and Rubin, 1992). A PSRF < 1.1 was taken as an indication of sufficient convergence. We collected 1000 samples at intervals of 100 steps from the converged chains. Samples from all 10 independent datasets were pooled to obtain the final predictions. Note that we were unable to obtain a PSRF < 1.1 for the Calderhead et al. model in the presence of non-zero Gaussian observation noise; in this case, we resorted to running the MCMC chains for 200,000 steps, which corresponds to roughly twice the number of steps that it took adaptive gradient matching to reach convergence, before taking samples as described above.

Figure 5.5 shows the results for the *PIF4/5* system. The data used for the parameter inference was sampled at intervals $\in \{1, 2, 4\}$ timesteps, where 4 is a realistic sampling

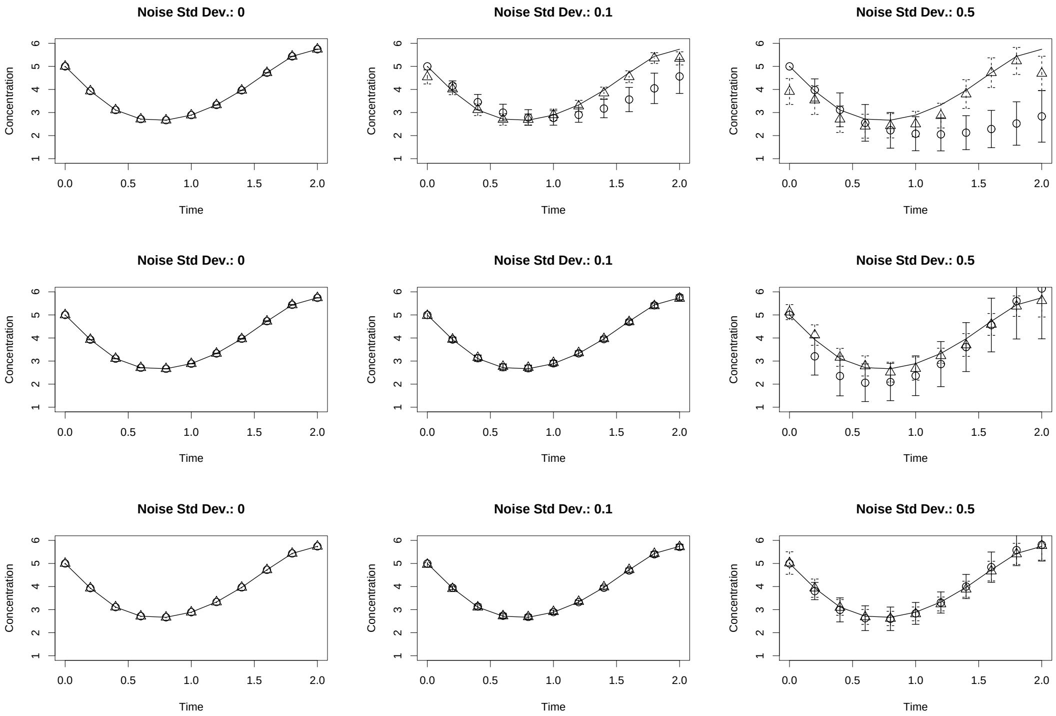


Figure 5.6: Lotka-Volterra concentrations for the prey species with varying observational noise. We show the true (noiseless) expression values, the sampled latent variables (triangles) and the expression profile simulated using the sampled $\boldsymbol{\theta}$ values (circles). Error bars show one standard deviation. Top Row: Calderhead et al. Model. Middle Row: Adaptive gradient matching. Bottom Row: Explicit ODE integration.

interval for actual measurements. We compare the method in Calderhead et al. (2008) with our adaptive gradient matching technique. We see that when there is no noise, the two methods perform equally well, but as soon as we introduce noise into the system, the predictions by the Calderhead et al. method become unreliable.

Figures 5.6 and 5.7 show the results for the Lotka-Volterra system, for the prey and predator species, respectively. The data used for the parameter inference was sampled at intervals of 0.25 timesteps. Again the method by Calderhead et al. showed a deteriorated performance in the presence of noise. For noise levels 0 and 0.1, adaptive gradient matching performed as well as the MCMC with explicit ODE integration, and for the highest noise level of 0.5, the performance of adaptive gradient matching is still competitive.

Finally, Figure 5.8 shows the results for the signal transduction cascade. Figure 5.8 only shows the predictions for *Rpp*, which represents the activated protein complex, and is arguably the central species in this system. Predictions for the other species can

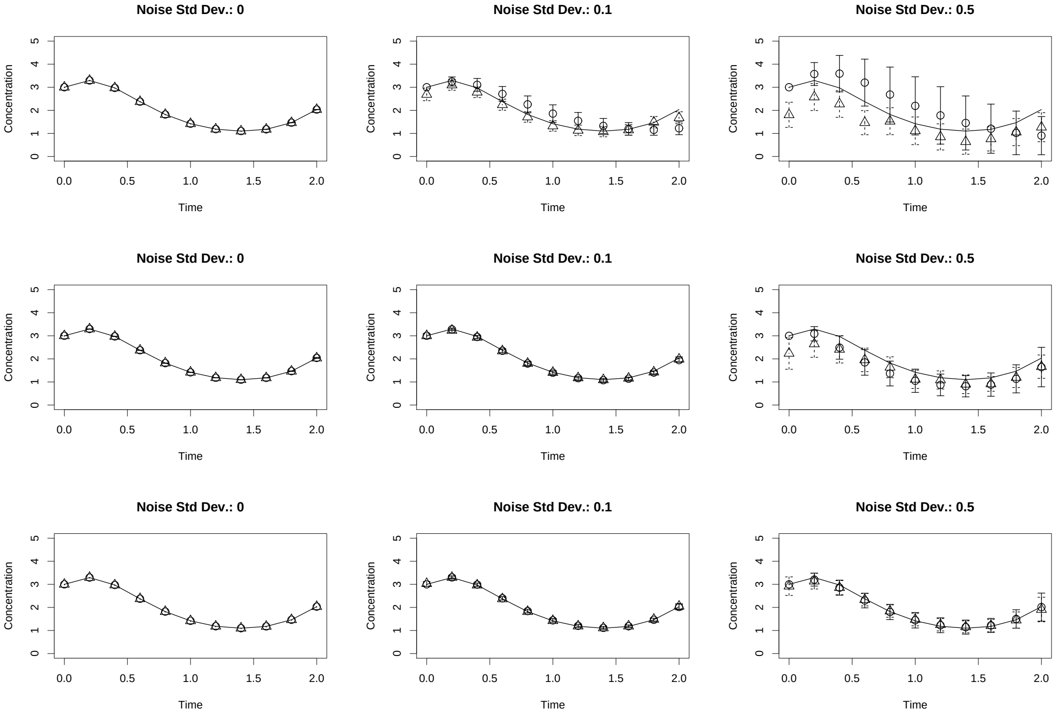


Figure 5.7: Lotka-Volterra concentrations for the predator species with varying observational noise. We show the true (noiseless) expression values, the sampled latent variables (triangles) and the expression profile simulated using the sampled $\boldsymbol{\theta}$ values (circles). Error bars show one standard deviation. Top Row: Calderhead et al. Model. Middle Row: Adaptive gradient matching. Bottom Row: Explicit ODE integration.

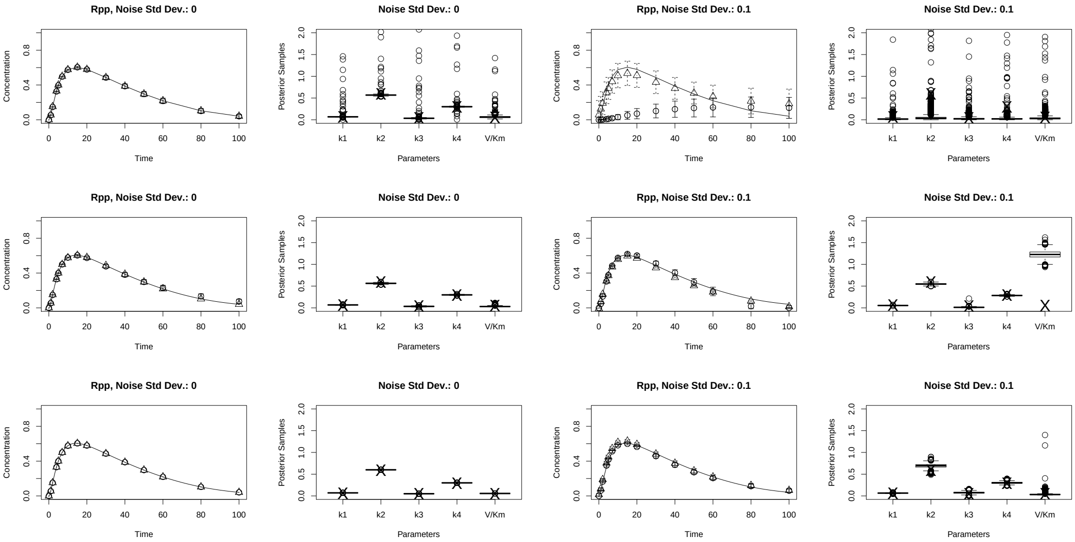


Figure 5.8: Expression levels of activated protein complex Rpp in the signal transduction pathway, with varying observational noise. Expression levels for other species in the system can be found in the supplementary material. We show the true (noiseless) expression values, the sampled latent variables (triangles) and the expression profile simulated using the sampled $\boldsymbol{\theta}$ values (circles). Error bars show one standard deviation. The boxplots give an idea of the distribution of the sampled parameters, where the true parameter value is marked with an X. The horizontal bar shows the median, the box margins show the 25th and 75th percentiles, the whiskers indicate data within 2 times the interquartile range, and circles are outliers. Top Row: Calderhead et al. Model. Middle Row: Adaptive gradient matching. Bottom Row: Explicit ODE integration.

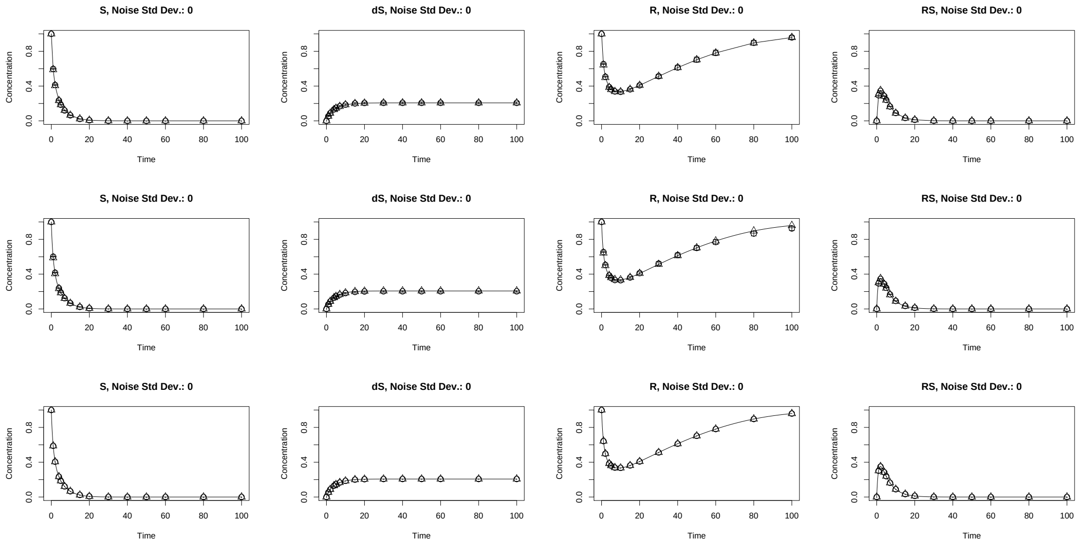


Figure 5.9: Expression levels of species other than Rpp in the signal transduction pathway, with no observational noise. We show the true (noiseless) expression values, the sampled latent variables (triangles) and the expression profile simulated using the sampled $\boldsymbol{\theta}$ values (circles). Error bars show one standard deviation. Top Row: Calderhead et al. Model. Middle Row: Adaptive gradient matching. Bottom Row: Explicit ODE integration.

be found in Figures 5.9 (no noise) and 5.10 (Gaussian noise with standard deviation 0.1). Figure 5.8 also includes boxplots for the sampled parameters. For the last two parameters, we present the ratio V/Km, as this is the crucial quantity that determines reconstruction accuracy. Once again, our adaptive gradient matching easily outperforms the method by Calderhead et al., and remains competitive with the explicit ODE integration in the presence of noise. Note that even though the ratio V/Km is overestimated by our method for noise level 0.1, the sampled parameters still result in a good fit to the observed data.

5.4.4 Speed and Computational Complexity

In Calderhead et al. (2008), the authors demonstrate that the moves of their sampler scale with O(NT^3), due to the requirement of inverting a TxT data matrix N times (where T is the length of the input time series and N is the number of species in the system). We can make a similar argument for adaptive gradient matching. The dominant computational cost for each sampling step comes from Equation (5.23), which requires inverting two TxT data matrices. Thus the complexity of each sampling step is

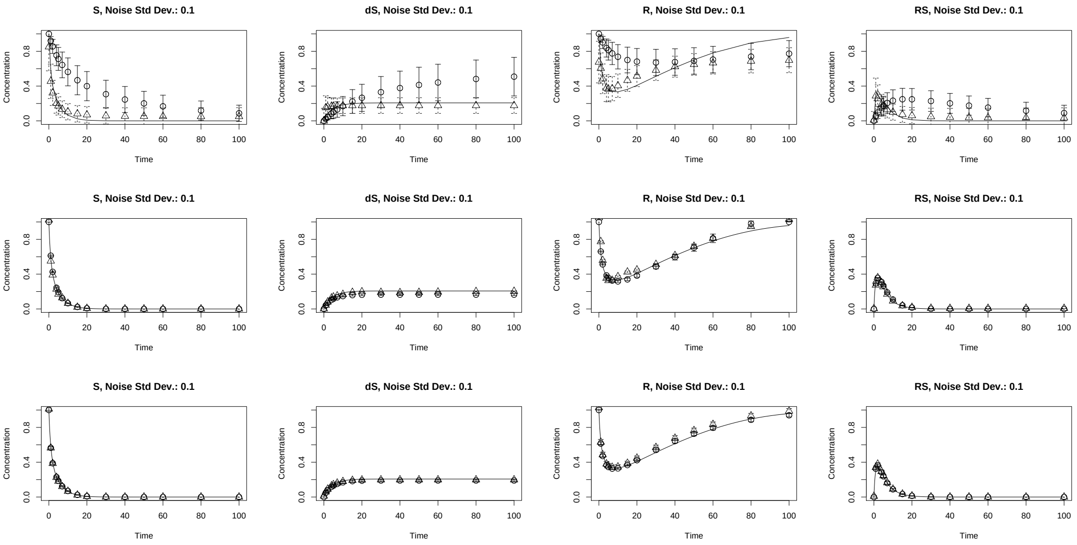


Figure 5.10: Expression levels of species other than Rpp in the signal transduction pathway, with observational noise with standard deviation 0.1. We show the true (noiseless) expression values, the sampled latent variables (triangles) and the expression profile simulated using the sampled $\boldsymbol{\theta}$ values (circles). Error bars show one standard deviation. Top Row: Calderhead et al. Model. Middle Row: Adaptive gradient matching. Bottom Row: Explicit ODE integration.

O($2NT^3$)=O(NT^3) when the sampling is done for all N species[3]. Hence each MCMC move using adaptive gradient matching has the same computational complexity as a move in Calderhead et al. (2008).

What will matter most in practice is how long each method takes to converge. Although it is difficult to prove convergence, we can get an indication by using the potential scale reduction factor (PSRF) as a convergence diagnostic, as described in Section 5.4.3. For convenience, we will refer to an MCMC run as converged if the PSRF is ≤ 1.1. Figure 5.11 compares the explicit ODE integration with the model by Calderhead et al. (2008), and with adaptive gradient matching in terms of computational time for 1e5 iterations (in seconds) and number of MCMC iterations before reaching convergence. We used the signal transduction cascade described in Section 5.4 as the test model. Each method was run 10 times using 10 different data instantiations (adding Gaussian observation noise with standard deviation 0.1). We see that, as expected, adaptive gradient matching and the method in Calderhead et al. (2008) are

[3] Note that in practice the inverted matrices can be cached, so we only have to invert both matrices for MCMC moves that change the GP hyperparameters. Therefore we should not expect the computational costs to be double those of Calderhead et al. (2008).

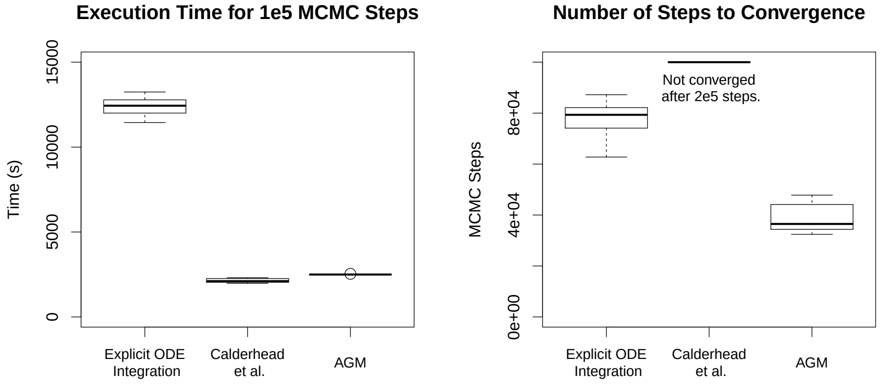


Figure 5.11: Computational efficiency of the different methods: Explicit ODE Integration, Calderhead et al. (2008) and adaptive gradient matching (AGM). We use parameter inference for the signal transduction model as a test case. Left: Time taken for 1e5 MCMC iterations. Right: Number of MCMC iterations to convergence. Note that Calderhead et al. (2008) did not achieve convergence in any of the runs. The horizontal bar of the boxplots shows the median, the box margins show the 25th and 75th percentiles, the whiskers indicate data within 2 times the interquartile range, and circles are outliers.

both faster than explicit ODE integration for a fixed number of iterations. Furthermore, adaptive gradient matching is only marginally slower than the method in Calderhead et al. (2008). We see that the method in Calderhead et al. (2008) does not converge for any of the runs, confirming our observation from Section 5.4.3. Adaptive gradient matching, on the other hand, converges in fewer iterations than explicit ODE integration. This can be explained by the difference in the dimensionality of the parameter space; as we have pointed out in Section 5.3, to integrate the ODE system, we also need to infer the initial concentrations for each species, in effect increasing the number of parameters. Adaptive gradient matching avoids having to infer the initial concentrations by effectively profiling over them, which, along with the treatment of latent variables $\mathbf{X}$ as ancillary variables (see Section 5.3), leads to fast convergence.

5.5 Application to the JAK/STAT Pathway

In this section, we apply our improved parameter inference method to a realistic model of interleukin-6 signalling in the JAK/STAT signalling pathway. This model was compiled by Roberta Cretella at the University of Glasgow, based on the current knowledge in the literature.

5.5.1 ODE Model Description

We analyse a model for interleukin-6 signalling (IL-6) in vascular endothelial cells. IL-6 binds to a receptor on the plasma membrane, activating the JAK/STAT pathway (Heinrich et al., 1998). The receptor is phosphorylated, creating docking sites for signalling molecules like *STAT3*. *STAT3* binds to the phosphorylated receptor and is phosphorylated itself. Phosphorylated *STAT3* molecules are released from the receptor, dimerize and then migrate to the nucleus to trigger mRNA transcription of target proteins like *SOCS3*. *SOCS3* acts as a feedback mechanism for the signalling pathway: it binds to active receptors to prevent *STAT3* activation and to provide a signal termination. See Figure 5.12 for a schematic representation of this pathway. The model we consider is a complex system comprising 13 species and 19 parameters. The dynamics of the system are described by mass-action kinetics, with non-linear interactions among species. Under the assumption of full observation of all species, we can decompose the system into 13 subsystems, one per species. This simplifies inference, and allows us to investigate the local identifiability of parameters in this model. The full system of ODEs looks as follows:

$$\frac{d[R]}{dt} = -k_1^f[R] + k_1^b[R^*] + k_{11}[SOCS3.R^*] + k_{14}[SOCS3.STAT3.R^*] \tag{5.41}$$

$$\begin{aligned}\frac{d[R^*]}{dt} &= k_1^f[R] - k_1^b[R^*] - k_2^f[STAT3][R^*] + k_2^b[STAT3.R^*]+ \\ &\quad k_3[STAT3.R^*] - k_{10}^f[SOCS3][R^*] + k_{10}^b[SOCS3.R^*]\end{aligned} \tag{5.42}$$

$$\begin{aligned}\frac{d[STAT3]}{dt} &= -k_2^f[STAT3][R^*] + k_2^b[STAT3.R^*] + k_{14}[SOCS3.STAT3.R^*]+ \\ &\quad 2k_7[2STAT3^*\_N]\end{aligned} \tag{5.43}$$

$$\begin{aligned}\frac{d[STAT3.R^*]}{dt} &= k_2^f[STAT3][R^*] - k_2^b[STAT3.R^*] - k_{13}[SOCS3][STAT3.R^*]- \\ &\quad k_3[STAT3.R^*]\end{aligned} \tag{5.44}$$

$$\frac{d[STAT3^*]}{dt} = -2k_4^f[STAT3^*][STAT3^*] + 2k_4^b[2STAT3^*\_C] + k_3[STAT3.R^*] \tag{5.45}$$

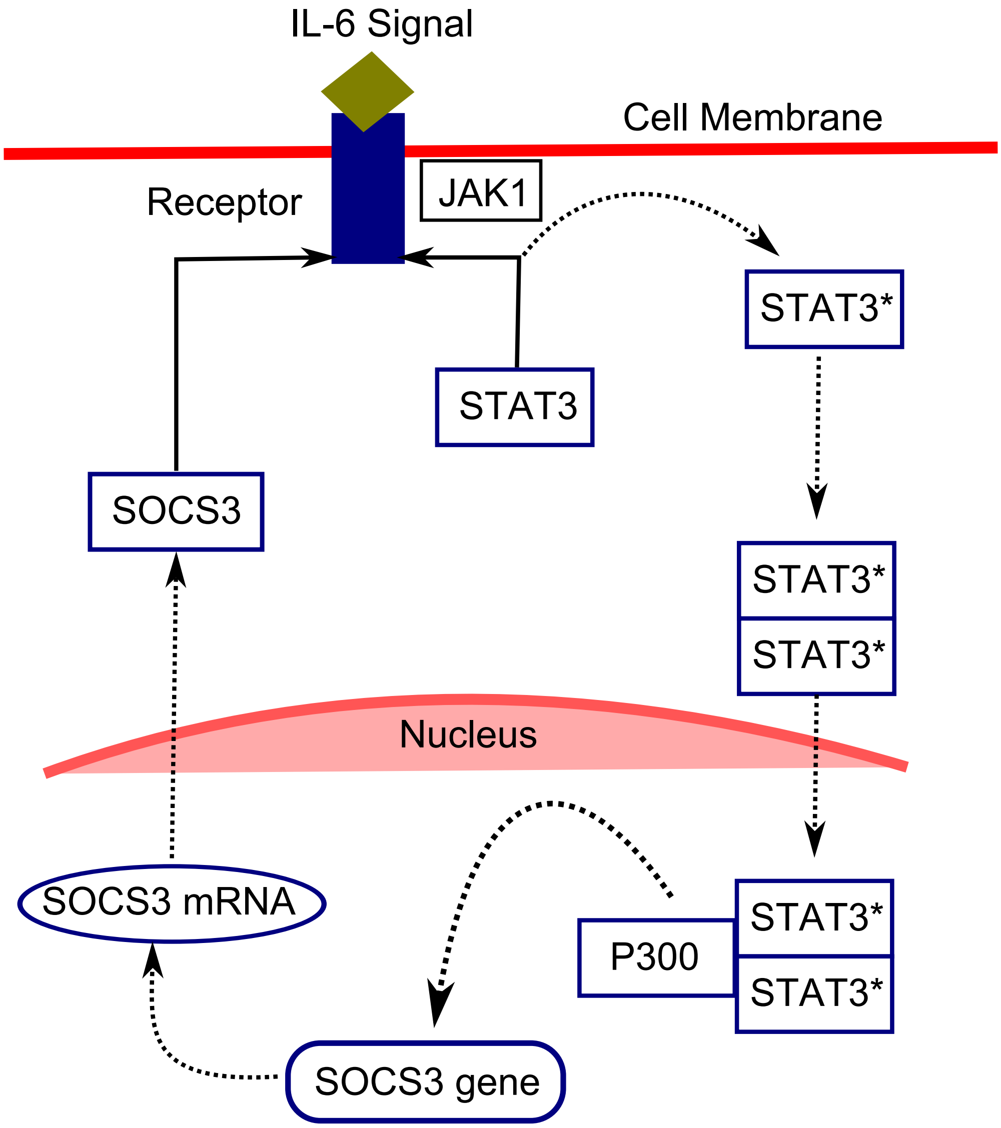


Figure 5.12: Schematic representation of interleukin-6 (IL-6) signalling in the JAK/STAT pathway. For a detailed description of this system, see Section 5.5.1. Note that the protein JAK is not modelled in the ODE system, but is assumed to be always present.

$$\frac{d[2STAT3^*\_C]}{dt} = k_4^f[STAT3^*][STAT3^*] - k_4^b[2STAT3^*\_C] - k_5[2STAT3^*\_C] \tag{5.46}$$

$$\begin{aligned}\frac{d[SOCS3]}{dt} = {} & k_{11}[SOCS3.R^*] - k_{12}[SOCS3] - k_{13}[SOCS3][STAT3.R^*] + \\ & k_9[SOCS3mRNA] + k_{14}[SOCS3.STAT3.R^*] - \\ & k_{10}^f[SOCS3][R^*] + k_{10}^b[SOCS3.R^*]\end{aligned} \tag{5.47}$$

$$\frac{d[SOCS3.R^*]}{dt} = -k_{11}[SOCS3.R^*] + k_{10}^f[SOCS3][R^*] - k_{10}^b[SOCS3.R^*] \tag{5.48}$$

$$\begin{aligned}\frac{d[2STAT3^*\_N]}{dt} = {} & -k_6^f[2STAT3^*\_N][P300] + k_6^b[2STAT3^*\_N.P300] + \\ & k_5[2STAT3^*\_C] - k_7[2STAT3^*\_N] + k_8[2STAT3^*\_N.P300]\end{aligned} \tag{5.49}$$

$$\begin{aligned}\frac{d[P300]}{dt} = {} & -k_6^f[2STAT3^*\_N][P300] + k_6^b[2STAT3^*\_N.P300] + \\ & k_8[2STAT3^*\_N.P300]\end{aligned} \tag{5.50}$$

$$\begin{aligned}\frac{d[2STAT3^*\_N.P300]}{dt} = {} & k_6^f[2STAT3^*\_N][P300] - k_6^b[2STAT3^*\_N.P300] - \\ & k_8[2STAT3^*\_N.P300]\end{aligned} \tag{5.51}$$

$$\frac{d[SOCS3.STAT3.R^*]}{dt} = -k_{14}[SOCS3.STAT3.R^*] + k_{13}[SOCS3][STAT3.R^*] \tag{5.52}$$

$$\frac{d[SOCS3mRNA]}{dt} = -k_9[SOCS3mRNA] + k_8[2STAT3^*\_N.P300] \tag{5.53}$$

5.5.2 Subsystem Inference

Trying to do parameter inference on the whole JAK/STAT system is very challenging. In our preliminary runs, even running the MCMC simulation for two weeks non-stop did not allow us to reach convergence (see Section 5.5.3.1), possibly due to the highly complex posterior landscape created by an ODE system with 19 parameters and 13 species. Note that this weakness is not particular to adaptive gradient matching; most methods based on MCMC would have difficulty to reach convergence.

It is therefore necessary to break up the inference task in some way. We decided that the most natural way to break up the inference would be to treat each equation as its own ODE subsystem. Georgoulas et al. (2012) describe a subsystem approach for parameter inference that works as follows: first a Gaussian process is fitted to the existing data for the trajectory of each species. These Gaussian processes are then used as the input to each subsystem that models part of the whole ODE system. Inference is done in parallel in the subsystems, and the resulting parameter samples are used to generate new estimates for the trajectories. These new estimates are then again used as inputs for parameter inference in the subsystems, and the process is repeated until it converges.

In theory we could do parameter inference in the same way, using the parameters inferred from the previous cycle through the subsystems to obtain the trajectories that form the inputs for the next cycle. However, in this work, I have not investigated the latter approach; instead I have assumed that we already have the trajectories that are needed as inputs for each subsystem, and I have investigated the feasibility of inferring the subsystem parameters from these perfect inputs. As we will see, this is a challenging problem on its own, and it encounters one of the problems that Georgoulas et al. (2012) mention in their discussion, to do with parameters that are shared among subsystems.

When you treat each equation as a subsystem, some of the ODE parameters become unidentifiable; only the sum of the parameters remains identifiable. We therefore have to reparameterise the system as follows:

$$\frac{d[R]}{dt} = -k_1^f[R] + k_1^b[R^*] + k_{11}[SOCS3.R^*] + k_{14}[SOCS3.STAT3.R^*] \tag{5.54}$$

$$\begin{aligned}\frac{d[R^*]}{dt} &= k_1^f[R] - k_1^b[R^*] - k_2^f[STAT3][R^*] + \mathbf{k^b_{2+3}[STAT3.R^*]} - \\ &\quad k_{10}^f[SOCS3][R^*] + k_{10}^b[SOCS3.R^*]\end{aligned} \tag{5.55}$$

$$\begin{aligned}\frac{d[STAT3]}{dt} &= -k_2^f[STAT3][R^*] + k_2^b[STAT3.R^*] + k_{14}[SOCS3.STAT3.R^*] + \\ &\quad 2k_7[2STAT3^*\_N]\end{aligned} \tag{5.56}$$

$$\frac{d[STAT3.R^*]}{dt} = k_2^f[STAT3][R^*] - \mathbf{k^b_{2+3}[STAT3.R^*]} - k_{13}[SOCS3][STAT3.R^*] \tag{5.57}$$

$$\frac{d[STAT3^*]}{dt} = -2k_4^f[STAT3^*][STAT3^*] + 2k_4^b[2STAT3^*\_C] + k_3[STAT3.R^*] \tag{5.58}$$

$$\frac{d[2STAT3^*\_C]}{dt} = k_4^f[STAT3^*][STAT3^*] - \mathbf{k^b_{4+5}[2STAT3^*\_C]} \tag{5.59}$$

$$\frac{d[SOCS3]}{dt} = \mathbf{k^b_{11+10}[SOCS3.R^*]} - k_{12}[SOCS3] - k_{13}[SOCS3][STAT3.R^*] + k_9[SOCS3mRNA] + k_{14}[SOCS3.STAT3.R^*] - k^f_{10}[SOCS3][R^*] \tag{5.60}$$

$$\frac{d[SOCS3.R^*]}{dt} = -\mathbf{k^b_{11+10}[SOCS3.R^*]} + k^f_{10}[SOCS3][R^*] \tag{5.61}$$

$$\frac{d[2STAT3^*\_N]}{dt} = -k^f_6[2STAT3^*\_N][P300] + \mathbf{k^b_{6+8}[2STAT3^*\_N.P300]} + k_5[2STAT3^*\_C] - k_7[2STAT3^*\_N] \tag{5.62}$$

$$\frac{d[P300]}{dt} = -k^f_6[2STAT3^*\_N][P300] + \mathbf{k^b_{6+8}[2STAT3^*\_N.P300]} \tag{5.63}$$

$$\frac{d[2STAT3^*\_N.P300]}{dt} = k^f_6[2STAT3^*\_N][P300] - \mathbf{k^b_{6+8}[2STAT3^*\_N.P300]} \tag{5.64}$$

$$\frac{d[SOCS3.STAT3.R^*]}{dt} = -k_{14}[SOCS3.STAT3.R^*] + k_{13}[SOCS3][STAT3.R^*] \tag{5.65}$$

$$\frac{d[SOCS3mRNA]}{dt} = -k_9[SOCS3mRNA] + k_8[2STAT3^*\_N.P300] \tag{5.66}$$

where the parts that have changed have been highlighted in bold red. This reparameterisation introduces 3 new parameters: k^b_{2+3}, k^b_{4+5} and k^b_{11+10}, and replaces k^b_6 by k^b_{6+8}, where the sub- and superscripts indicate which parameters from the original system have been summed. This brings the total number of parameters to 22.

We generate simulation data using the parameter settings:

$\{k^f_1 = 0.1, k^b_1 = 0.1, k_{11} = 0.003, k_{14} = 0.03, k^f_2 = 0.08, k^b_2 = 0.008, k_3 = 0.4, k_1 0^f = 0.5, k_1 0^b = 0.1, k_7 = 0.005, k_{13} = 0.02, k_9 = 0.01, k^f_4 = 0.2, k^b_4 = 0.01, k_5 = 0.5, k_{12} = 0.005, k^f_6 = 0.003, k^b_{6+8} = 0.013, k_8 = 0.01, k^b_{2+3} = 0.408, k^b_{4+5} = 0.51, k^b_{11+10} = 0.103\}$,

which give realistic behaviour for the system. We record 18 datapoints at time points which give greater weight to the intimal period where more variability is present:

$\{0, 10, 20, 30, 40, 50, 60, 80, 100, 120, 180, 240, 300, 360, 420, 480, 540, 600\}$.

The initial concentrations for the 13 species at time point 0 are:

$\{[R] = 4, [R^*] = 0, [STAT3] = 100, [STAT3.R^*] = 0, [STAT3^*] = 0, [2STAT3^*\_C] = 0, [SOCS3] = 0, [SOCS3.R^*] = 0, [2STAT3^*\_N] = 0, [P300] = 1, [2STAT3^*\_N.P300] = 0, [SOCS3.STAT3.R^*] = 0.1, [SOCS3mRNA] = 0.1\}$.

5.5.3 Results

We apply the adaptive gradient matching method from Section 5.2.2 to the problem of parameter inference in the JAK/STAT signalling pathway. We use the sigmoidal kernel, as this empirically gave us the best results. Throughout this section, where the subsystem approach is used, we assume that we have perfect inputs for each subsystem, and only the subsystem parameters need to be inferred. We will show that even in this optimistic scenario, we encounter certain problems related to the subsystem approach that are hard to overcome.

5.5.3.1 Parameter Inference for the Complete System

We first tried to apply AGM to the whole system (i.e., we did not use the subsystem approach). We ran five independent simulations to do convergence tests.

Figure 5.13 shows the convergence tests using scatterplots of the ODE parameters and the potential scale reduction factor (PSRF) Gelman and Rubin (1992), as well as an example trace of the log likelihood. For these simulations, we have chosen a burnin period of 10 million iterations in order to be conservative, and we have sampled a further 10000 iterations after burnin. As Figure 5.13 demonstrates, even an MCMC run of that length, which took more than two weeks to run on a standard desktop computer, did not converge. The PSRF values for all 22 parameters are very far from the value of 1.1 that is usually taken as indication of convergence. Furthermore, the log-likelihood still seems to be increasing even after a long burnin phase. The scatterplots, which show sampled values for three example parameters from two independent runs, reveal that some of the sampled parameter distributions seem to have two modes in one run, but only one mode in the other, another indication of insufficient convergence.

5.5.3.2 Perfect vs Noisy Observations

Figures 5.14 compares the predictions obtained by the sampled parameters, as well as the latent variables sampled during the MCMC, in the case of perfect observations (no noise) with the case of noisy observations, where we have added Gaussian noise with standard deviation 0.05 to the subsystem species only. Here we only show the results for a subset of species, for full results, see Appendix D. Figures 5.15 and 5.16 show the distributions of the sampled parameters for a couple of example subsystems. Appendix D contains the distributions of sampled parameters of the remaining subsystems.

We can see that the subsystem approach works as long as we assume perfect observation for the subsystem species, but breaks down when observation noise is added to the system. The natural question to ask is what could be causing this breakdown. We have done a perturbation study to elucidate this aspect.

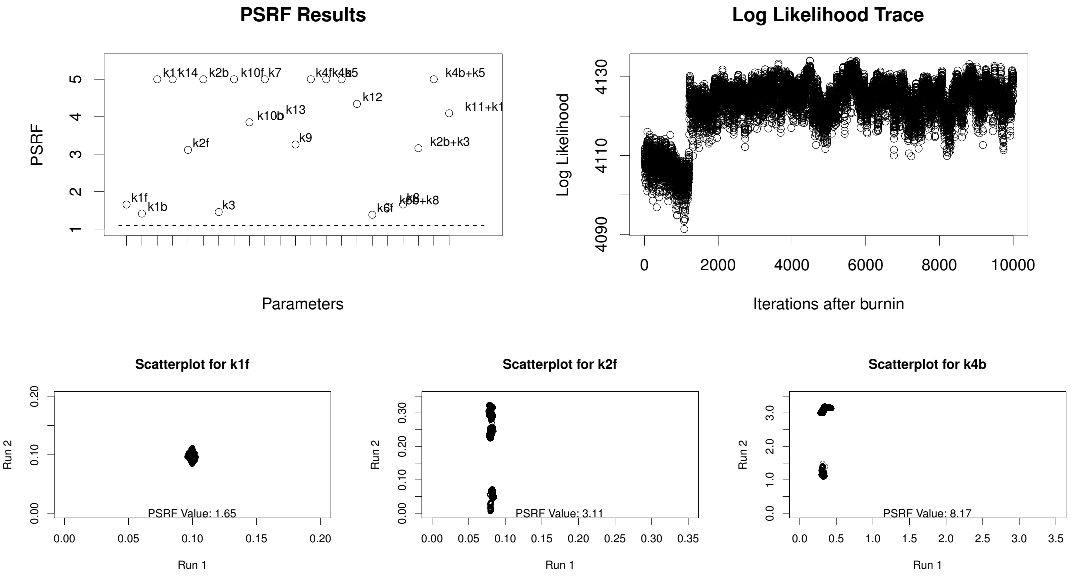


Figure 5.13: Convergence tests for inference on the complete JAK/STAT system. Top Right: Potential scale reduction factor (PSRF) values for all 22 parameters. Top Left: Trace of the (unnormalised) log likelihood after burnin. Bottom: Scatterplots showing sampled values from three example parameters over two independent MCMC runs.

5.5.3.3 Perturbation Study

There are two likely factors that could be causing the breakdown in the face of noise: 1) the switch from a global approach to a subsystems approach, or 2) the use of gradient matching as an approximation to solving the ODE system explicitly. To figure out which is at fault, we proceeded as follows. For each of the 22 parameters, we investigated the effect of perturbing that parameter by setting it to a value on the interval $[0,1]$ while keeping the other parameters fixed at the true values, and calculating four different likelihoods[4]:

- The global profile log likelihood, comparing data generated from the true ODE system with data generated from the perturbed ODE system.
- The global gradient log likelihood, comparing gradients calculated using data generated from the true ODE system with the gradients calculated using data generated from the perturbed ODE system.
- The local profile log likelihood, comparing data generated from the true ODE

[4] We assume a Gaussian likelihood, and hence calculate the log likelihood via the mean squared deviation of each species from the ODE solution using the original parameters.

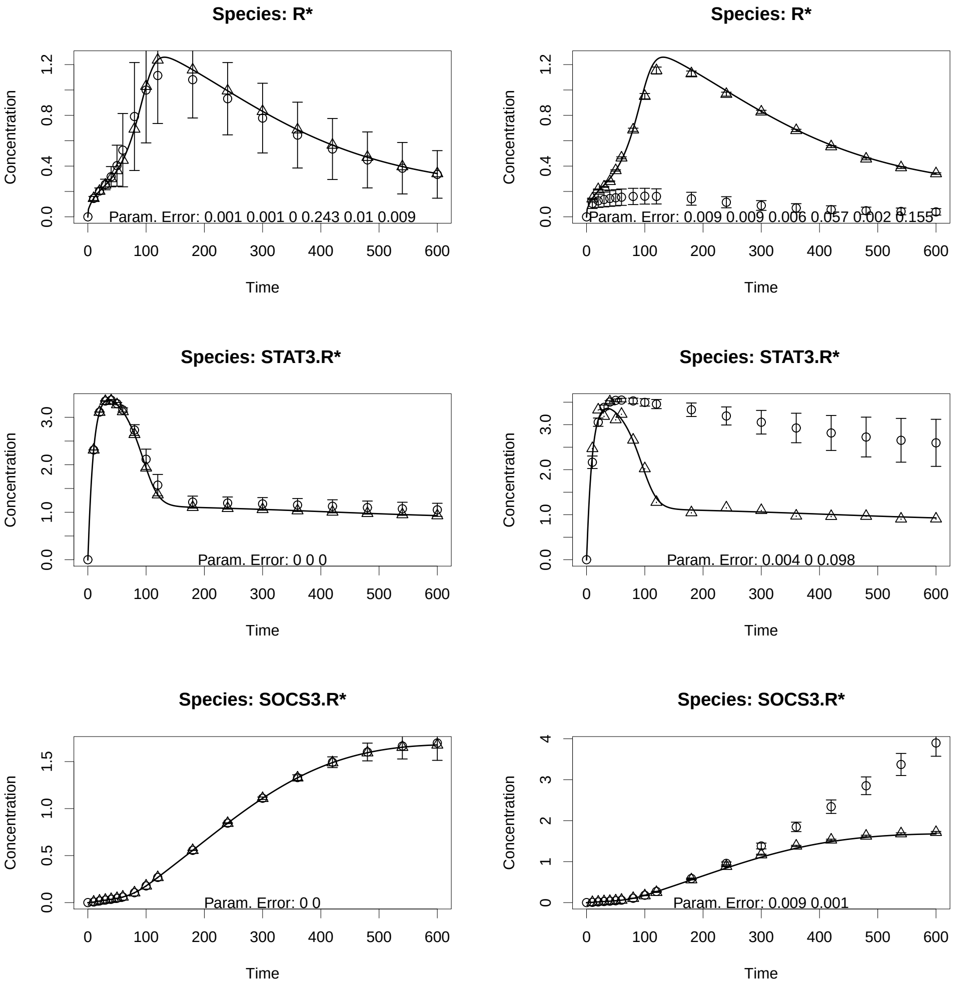


Figure 5.14: Comparison of parameter inference with and without noise for species 2, 4 and 8. Left: Perfect Observations. Right: Noisy observations (Gaussian noise with standard deviation 0.05). We present the true (solid line) and inferred (circles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. We also show the latent variables sampled from the posterior (triangles with error bars). The error bars show one standard deviation.

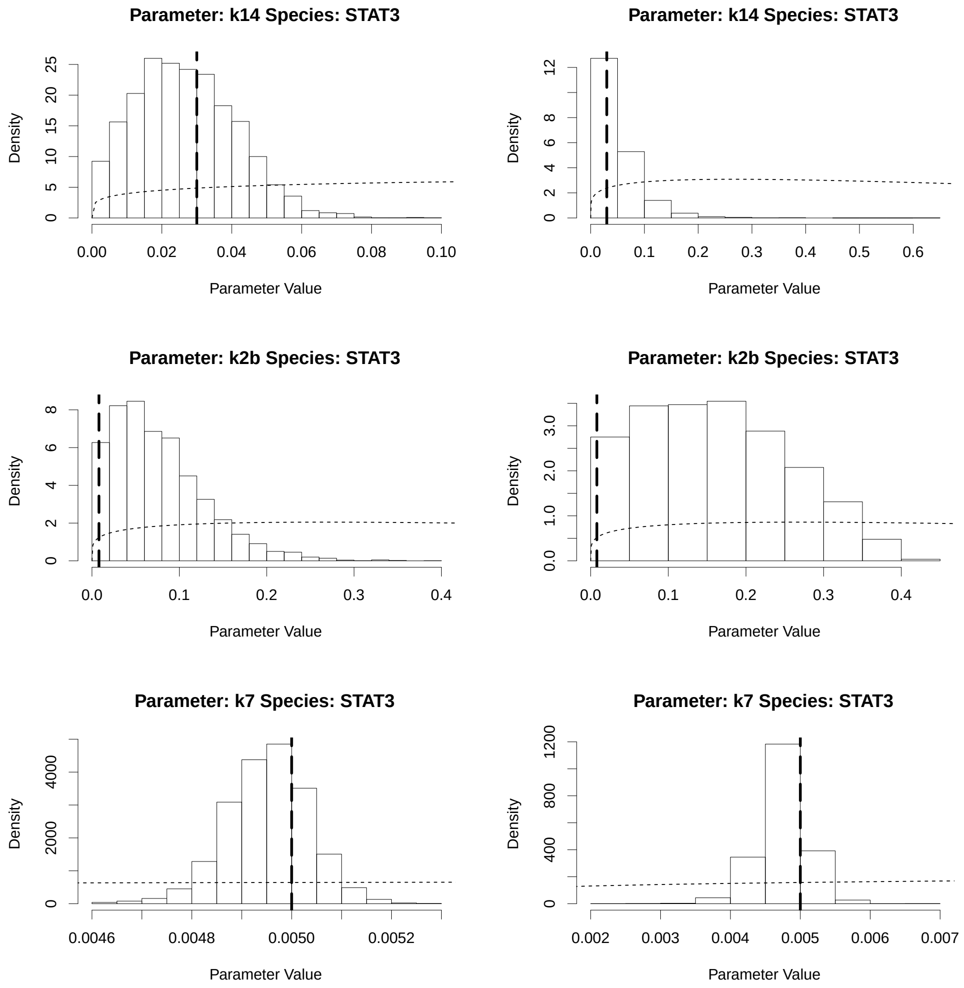


Figure 5.15: STAT3 - Species 3. Left: Perfect Observations. Right: Noisy Observations. The histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the vertical line indicates the true value of the parameter. Note that some of the parameters of this subsystem have been omitted for space reasons; for the remaining histograms, see Appendix D.

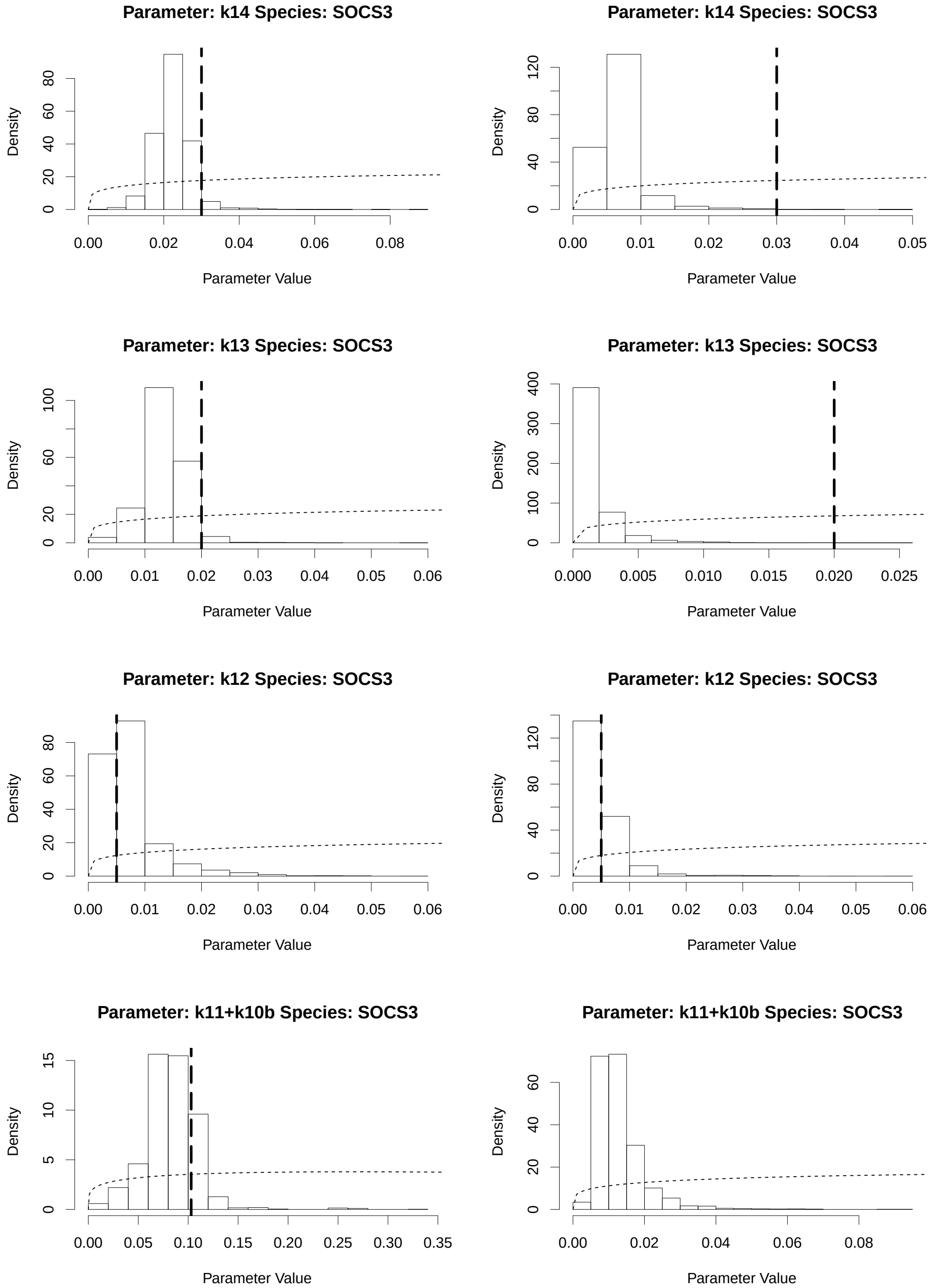


Figure 5.16: SOCS3 - Species 7. Left: Perfect Observations. Right: Noisy Observations. The histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the vertical line indicates the true value of the parameter. Note that some of the parameters of this subsystem have been omitted for space reasons; for the remaining histograms, see Appendix D.

system with data generated from the perturbed ODE **sub**system, conditional on the concentrations of species outside the subsystem staying the same.

- The local gradient data log likelihood, comparing gradients calculated using data generated from the true ODE system with gradients calculated using data generated from the perturbed ODE **sub**system, conditional on the concentrations of species outside the subsystem staying the same.

Note that although the first two likelihoods are called 'global', we have nevertheless only calculated them for a subsystem (i.e. one species), not for the whole system. They are global in the sense that the whole ODE system is solved with the perturbed parameter value. The other likelihoods are local because the ODE system is only solved for the subsystem species, and thus the solution is conditional on the other species staying the same. This closely mimics what happens in the subsystem parameter inference method.

Figures 5.17, 5.18 and 5.19 show the effect of the perturbations on a selection of subsystems. Results for the remaining subsystems can be found in Appendix D; the results are qualitatively similar. We can see that while there is almost no qualitative difference between the profile log likelihood and the global log likelihood, there is a significant difference between using the global likelihood and the local likelihood. The likelihood landscape for the global log likelihood is very peaked at the correct parameter values, while the landscape for the local log likelihood seems quite flat in comparison. This explains why the parameter inference breaks down in the presence of noise; the smaller peak will get lost among spurious peaks created by noisy data.

Why would the likelihood landscapes be flatter for a local solution? The answer lies in the effect of the parameters. For a local solution, changing one parameter only affects the subsystem species, and leaves the inputs to the subsystem unchanged. This is not the case for a global solution; changing one parameter can have an effect on the whole system, sometimes causing quite drastic changes in the inputs to the subsystem. Naturally this will have an equally drastic effect on the likelihood landscape. Unfortunately the problem cannot be solved by switching from a local to a global solution, as the global approach is not compatible with doing inference in the subsystem only. Therefore, as soon as we break the system up into subsystems, we will run into this kind of problem.

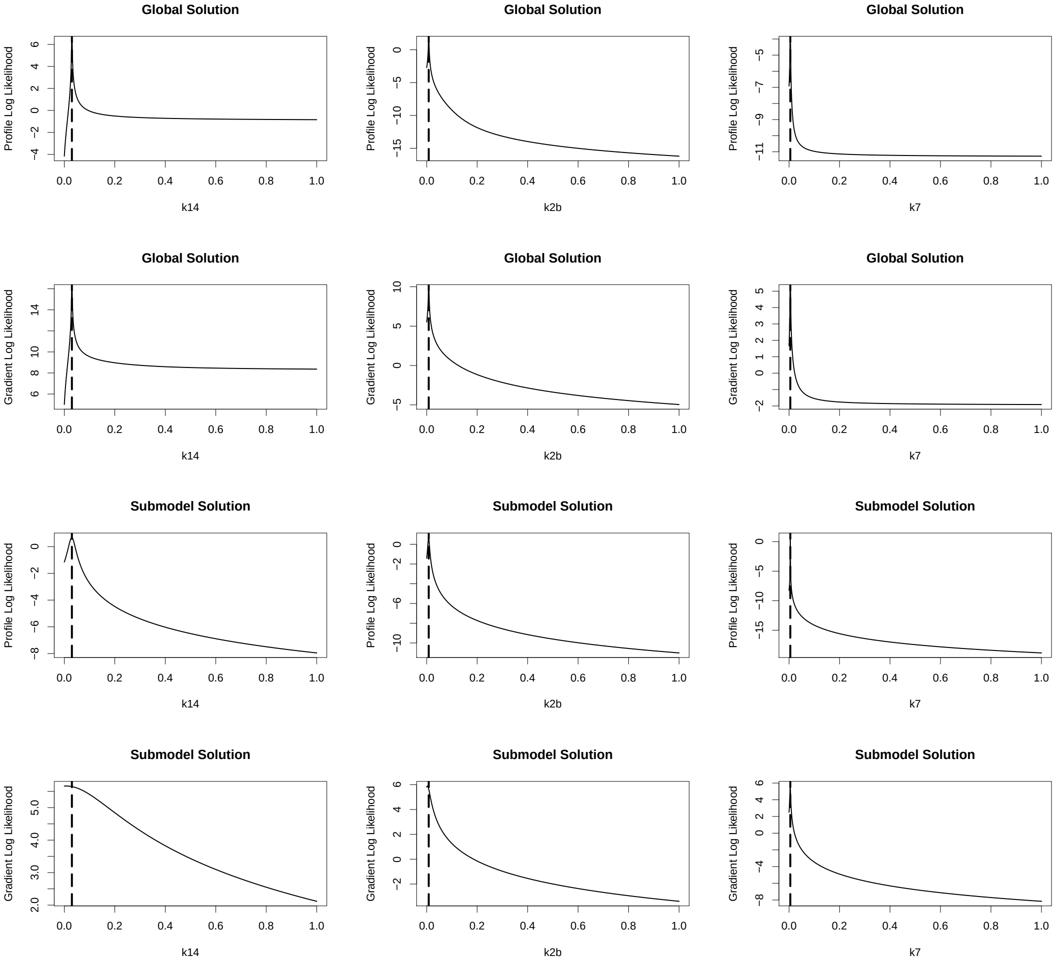


Figure 5.17: Profile and Gradient Likelihood Comparison of STAT3. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter. Note that some of the parameters of this subsystem have been omitted for space reasons; for the remaining histograms, see Appendix D.

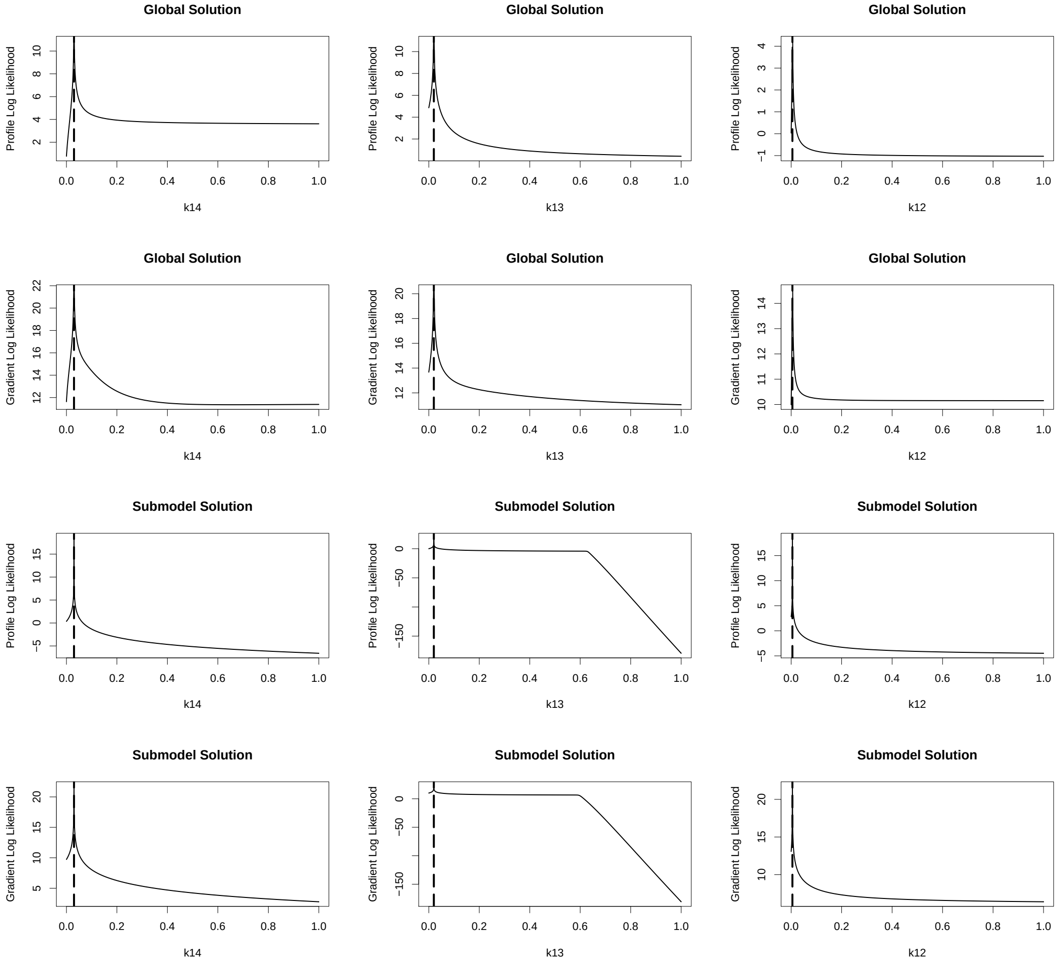


Figure 5.18: Profile and Gradient Likelihood Comparison of SOCS3 (a). From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter. Note that some of the parameters of this subsystem have been omitted for space reasons; for the remaining plots, see Appendix D.

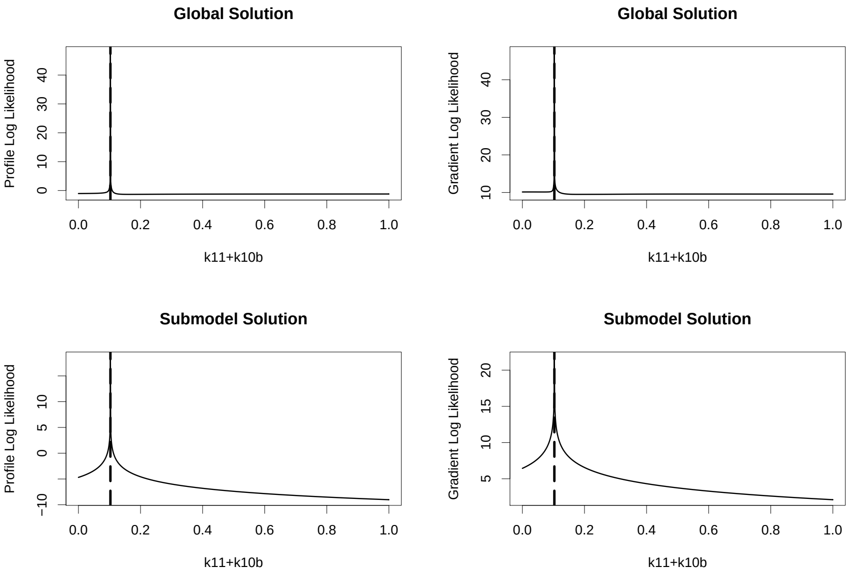


Figure 5.19: Profile and Gradient Likelihood Comparison of SOCS3 (b). Top Row: Profile Likelihood Global, Gradient Likelihood Global. Bottom Row: Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter. Note that some of the parameters of this subsystem have been omitted for space reasons; for the remaining plots, see Appendix D.

5.6 Discussion

In this chapter, I have described an adaptive gradient matching approach for parameter inference in ODE systems based on Calderhead et al. (2008). Adaptive gradient matching avoids the need for explicitly solving the ODE at each MCMC sampling step, which significantly reduces the computational burden. In the method of Calderhead et al., an adaptation of the ODE parameters has no influence on the inference of the GP hyperparameters. This corresponds to a unidirectional information flow from GP interpolation to parameter inference in the system of ODEs. We have developed a methodological improvement that infers both GP hyperparameters and ODE parameters jointly from the posterior distribution, and where due to conditional dependence between both groups, the latter may exert an influence on the former. This closes the inference procedure by effectively introducing an important information feedback loop from the ODE system back to the GP interpolation.

We have applied adaptive gradient matching to three model systems from ecology and systems biology, and have demonstrated that our method outperforms Calderhead et al. (2008) and performs on a par with a sampler which explicitly solves the ODE system at each step. Compared to explicitly solving the ODE system, our adaptive gradient matching approach is up to an order of magnitude faster, where the speedup in practice will depend on the size and stiffness of the ODE system. The method by Calderhead et al. is asymptotically as fast as adaptive gradient matching, but tends to be marginally faster per MCMC iteration in practice, as the inference of the latent variables can be done in a Gibbs step, which is not possible if we want to jointly infer the latent variables, GP hyperparameters and ODE parameters. However, we have shown that at least for the benchmark systems considered in this chapter, the approach of Calderhead et al. often fails to converge at all, thus rendering the computational complexity and the speed per MCMC iteration moot.

A close relative of our work is the recently published method of functional tempering (Campbell and Steele, 2012), which is based on the same gradient matching paradigm as our approach, but uses B-splines instead of Gaussian processes for data interpolation. The approach in Campbell and Steele (2012) has one vector of regularization parameters, which corresponds to our hyperparameter vector $\boldsymbol{\gamma}$ and penalizes the mismatch between the gradients. Our model additionally profits from the hyperparameters of the Gaussian process, $\boldsymbol{\phi}$, which define the flexibility of the interpolant and are automatically inferred from the data, while in the model of Campbell and Steele (2012) this flexibility is defined by the B-splines basis and has to be set in advance. An interesting difference is the tempering scheme of Campbell and Steele (2012), which applied to our model corresponds to gradually forcing $\boldsymbol{\gamma}$ to zero rather than inferring it from the posterior distribution. A comparative evaluation is the subject of our future research.

We have applied adaptive gradient matching to a more complex ODE model of molecular signalling in the JAK/STAT pathway. Here, we have shown that while inference for the full system is computationally not feasible, inference over subsystems can be fruitful under certain conditions. If there is no observational noise, then under the assumption of perfectly observed inputs to the subsystem, we can infer the right parameters. However the subsystem approach is hampered by shallow likelihood landscapes that make finding the true peak difficult in the presence of noise. A pure subsystem approach is therefore of limited use when tackling complex ODE models; more sophisticated approaches will be needed, possibly based on grouping parameters or sampling subsystems to obtain a more peaked likelihood landscape.

Chapter 6

Conclusion and Further Work

This chapter serves to sum up the main outcomes of the thesis (Section 6.1), both in terms of new models, techniques and insights (Subsection 6.1.1), and in terms of applications to challenging real-world datasets (Subsection 6.1.2). I will also discuss promising avenues for further research (Section 6.2).

6.1 Main Outcomes

6.1.1 Methodological Advances

In this thesis, I have covered a range of state-of-the-art network structure and parameter inference techniques in different models, from regression methods and static Bayesian networks (Chapter 2) over time-varying dynamic Bayesian networks (Chapters 3 and 4) to parameterised ODE systems (Chapter 5). The main methodological achievements can be summed up as follows:

- Comparison of network reconstruction methods for species absence/presence data in ecology. As described in Chapter 2, I compared the performance of sparse Bayesian regression, lasso regression, graphical Gaussian models and static Bayesian networks for reconstructing species interaction networks using data from a realistic stochastic simulation model.

- Development of the spatial autocorrelation model for ecological species absence/presence data, and integration into the framework of regression and of static Bayesian networks. As shown in Chapter 2, modelling the spatial autocorrelation explicitly aids in the network reconstruction process by taking account of the dependence that arises from using absence/presence data of species from spatially-correlated neighbouring locations.

- Development of hierarchical global and sequential information sharing models for a non-homogeneous time-varying dynamic Bayesian network, as described in Chapter 3. I compared the two types of information sharing with each other, on both simulated data and real-world applications, and identified their strengths and weaknesses, as well as showing that they improved on competing approaches.

- Further development and comparison of different functional forms (exponential versus binomial) and inter-node coupling strategies (soft versus hard) for the sequential information sharing priors. In Chapter 4, I presented a detailed comparison of the different approaches in terms of network reconstruction performance, and I investigated the effect of different hyperparameter settings. I highlighted some problems with the exponential prior formulation, and pointed out how they could be resolved by the binomial prior. Finally, I applied the sequential information sharing approaches to two real-world gene expression datasets and showed that they outperformed competing network reconstruction methods.

- Development of an improved MCMC sampler for the information sharing model. In Chapter 4, I demonstrated that the standard sampler employed in Lèbre et al. (2010) is not appropriate when used in conjunction with the hierarchical information sharing model, as it can get stuck in local optima. I described an improved, randomised multi-segment network structure move that resolves this difficulty, and leads to much better mixing and convergence.

- Improvement to the inference procedure in the Gaussian process ODE parameter inference model of Calderhead et al. (2008), by reformulating it in a principled way to allow for joint inference of the hyperparameters of the GP and the parameters of the ODE system. This allows the GP hyperparameters to be adapted to the ODE parameters and vice-versa. I tested this adaptive gradient matching approach using three small-to-medium sized ODE systems and compared it with the approach in Calderhead et al. (2008) as well as with a method using explicit integration of the ODE system. I showed that adaptive gradient matching performs competitively, improving on Calderhead et al. (2008) in parameter inference and on explicit integration in speed and computational complexity.

Note that the improved MCMC sampler for inference of time-varying dynamic Bayesian networks with information sharing, as described in Chapter 4, has been implemented as the R software package EDISON, which is described in Appendix F and is freely available on the Comprehensive R Archive Network (CRAN).

6.1.2 Real-World Applications

The application to real-world systems is an essential part of any research into new statistical and machine learning methods. In order to demonstrate the viability of the new methodologies described above, I have applied them to a broad selection of real-world datasets:

- In Chapters 2, the network reconstruction methods have been applied to bird species absence/presence data of 39 warbler species from the European Bird Atlas (Hagemeijer and Blair, 1997), covering Europe west of 30°E and including all probable and confirmed breeding records. As described in that chapter, my collaborator Ali Faisal dealt with the application to the bird data, with the final research being the product of our collaboration and discussions.

- In Chapter 3, I used a collection of four gene expression datasets of nine circadian clock genes in *Arabidopsis thaliana* (Mockler et al., 2007; Edwards et al., 2006; Grzegorczyk et al., 2008a), obtained under different experimental conditions, to test the global information sharing method for non-homogeneous dynamic Bayesian networks, and showed that information sharing leads to better agreement among the reconstructed networks.

- In Chapter 3 and 4, I used gene expression measurements in *Dropsophila melanogaster* (Arbeitman et al., 2002), from which we extracted a dataset of eleven wing muscle genes, to demonstrate that sequential information sharing infers changepoint locations that give better agreement with the known morphological changes than competing methods, and that many of the retrieved gene interactions have been described in the literature.

- In Chapter 4, I used gene expression data from a five-gene synthetic biology network in *Saccharomyces cerevisiae* (Cantone et al., 2009) with known structure to demonstrate that sequential information sharing outperforms competing methods in terms of network reconstruction accuracy.

- Finally, in Chapter 5, I applied the adaptive gradient matching method for ODE parameter inference to simulated time-course protein concentration data obtained from a realistic model of the JAK/STAT pathway compiled from the literature.

In addition, I participated in the DREAM 5 Network Inference Challenge, where my team used a variant of the model described in Chapter 3, as described in Appendix C. The datasets for this challenge consisted of a large simulated gene expression dataset, as well as gene expression data from *Staphylococcus aureus*, *Saccharomyces cerevisiae* and *Escherichia coli*, varying in size, but each containing several hundred measurements of several thousand genes. The competition organisers went on to publish a consortium paper in Nature Methods (Marbach et al., 2012), which includes a contribution from my team.

6.2 Future Research Directions

The work in this thesis was motivated by the observation that network structure and parameter inference is a crucial problem in biology. I have developed new techniques that tackle important challenges such as inference of time-varying networks, and efficient parameter inference in ODE systems. There are many other possible avenues for research in network inference; here I will highlight a few that I consider important and that are related to or arise from the ideas and methods described in this thesis.

Spatial models for species interaction networks in ecology. The work described in Chapter 2 has highlighted the importance of proper spatial modelling when dealing with species data in ecology. Modelling the spatial autocorrelation led to a significant improvement in the network reconstruction performance. Our paper (Faisal et al., 2010) has already been followed up by research done by another PhD student, Andrej Adherhold (Aderhold et al., 2012), which shows that using a 2-dimensional spatial model allows for reliable network inference in the presence of spatial heterogeneity. Further research in this vein, allowing, for example, for non-rectangular segmentations, is likely to produce even better spatial models.

New functional and structural forms for information sharing priors. In this thesis, I have presented five different models for information sharing priors: the independent edge prior for global information sharing (Chapter 3), and four hierarchical priors for sequential information sharing based on an exponential versus binomial distribution, with or without gene-specific hyperparameters (Chapter 4). Wang et al. (2011) have experimented with a different approach, which effectively combines the exponential prior with an additional factor that encourages network sparsity. One could think of other formulations that incorporate information sharing of networks aspects beyond edge comparisons; for instance, conserving network motifs across network segments.

Global information sharing for data integration. In Chapter 3, I demonstrated the usefulness of global information sharing for reconstructing networks from gene expression datasets obtained under different experimental conditions, a frequent problem in systems biology. Although my subsequent research focused more on sequential information sharing, the sequential information sharing priors that I developed in Chapter 4 could easily be adapted for the global case, by following a strategy similar to Werhli and Husmeier (2008) and using a learned hypernetwork to encode common edges. In fact, Penfold et al. (2012) recently presented a formulation that uses an exponential prior in a hierarchical (or global) manner to learn networks from multiple perturbed time-series using a Gaussian process model.

Information sharing in non-DBN models. The information sharing priors from Chapters 3 and 4 are not exclusive to the dynamic Bayesian network model, and can in principle be adapted for any network model with discrete edges. Information sharing can also be of interest for non-Bayesian network models without discrete edges, such as the graphical lasso (Meinshausen and Bühlmann, 2006). In Danaher et al. (2011), the problem of conserving information while jointly estimating several graphical Gaussian models was solved by introducing fused and group lasso penalties over the elements of the precision matrix.

Online learning for time-varying networks. Online learning is an increasingly important topic in fields such as finance, where data may not always be available in batch format. A recently published Masters thesis (Hongo, 2012) has built on the work described in Chapters 3 and 4 to develop an online learning framework for learning time-varying networks from financial data, using a particle filtering approach. The model in Wang et al. (2011), mentioned above, also uses particle filtering to allow for online learning.

Modelling interventions in systems biology. Interventions on specific genes or proteins are a useful tool for testing causal hypotheses in systems biology. Knocking out a gene or inhibiting the binding activity of a protein allows direct observation of the effect that this intervention has on the rest of the system. There already exist methods that take advantage of the extra information provided by interventions, such as the nested effects models (Markowetz et al., 2005) described in Chapter 1 and Appendix E. It should be possible to model interventions within the framework of dynamic Bayesian networks, either by placing restrictions on allowed edges, or by the use of appropriate priors, such as the one in Lo et al. (2012), which is very similar to our global information sharing prior, and could be used to down-weight the probability of edges that are inhibited by the intervention. This could lead to improved network reconstruction accuracy when interventional data is available.

Improving scalability and computational complexity. For the network structure inference methods considered in Chapters 2, 3 and 4, one could think of improving on the exact but comparatively slow inference via RJMCMC, and using variational approximations to scale up to inference on larger gene networks. If one does not want to abandon exact inference, then perhaps better variable selection techniques and faster-converging samplers, such as e.g. Riemannian manifold Hamiltonian Monte Carlo (Girolami and Calderhead, 2011) could be investigated. We have already shown in applying a variant of our model to the DREAM5 network inference challenge (as reported in Appendix C) that with some restrictions on the possible parent sets, exact inference is feasible for large datasets. As for the adaptive gradient matching method for ODE parameter inference in Chapter 5, this method is already quite efficient compared to a sampler using the explicit integration of the ODE system, but it could also benefit from better samplers in the Hamiltonian Monte Carlo vein. Another area for development would be finding a workable approach for splitting larger ODE systems into subsystems, which would allow for solving the subsystems in parallel, as the results in Chapter 5 have shown that the naive approach is insufficient.

Parameter inference in stochastic differential equations (SDEs). Parameter estimation in stochastic differential equations suffers from similar problems to parameter estimation in ODEs, but is intrinsically more difficult, because computing the likelihood exactly would require the solution of path integrals, which are usually intractable. Instead, Monte Carlo techniques are required to obtain the numerical solution of the

SDE. One popular approach to parameter inference in SDEs is to use a Kalman filter, a linear state space model with conditional Gaussian distributions for modelling the noise and the dynamics of the unkown true states. A set of ODEs is used to determine the first and second moments for the SDEs. See e.g. Møller et al. (2011). The ODEs will in general be solved numerically, which leads immediately to the possibility of applying adaptive gradient matching as described in Chapter 5. More involved would be finding a way of translating the adaptive gradient matching approach to the SDE setting directly.

Integration of probabilistic and mechanistic models. So far I have made the distinction between probabilistic models such as DBNs, that make simplifying assumptions (e.g. linearity) when modelling network dynamics, and mechanistic models such as ODE systems that try to explicitly model the processes that generate these dynamics. Ideally, one would like to combine both models under one framework, where the simplifying assumptions of probabilistic methods can be used for modelling the higher-level network dynamics, while the mechanistic model is used for representing the finer-grained interactions that we are most interested in. As pointed out in Chapter 1, some efforts have already been made to combine mechanistic and probabilistic aspects in one framework, most notably Äijö and Lähdesmäki (2009). In Liu et al. (2012), ODEs are approximated by dynamic Bayesian networks, which allows for easier inference. However, in some ways one would like to achieve the opposite; start from a probabilistic model, and determine only the mechanistic equations that govern system behaviour for a subsystem, where inputs to the subsystem would be determined using the probabilistic model. This would allow one to more easily scale up ODE models to very large systems, as only specific interactions of interest would have to be modelled explicitly, with the increase in parameter space that that entails. Adaptive gradient matching as described in Chapter 5 might give an idea of how to achieve this; if predictions from the Gaussian process could somehow be used as inputs, then we could choose whether we want to learn ODE parameters for a given subsystem, or whether we only wish to learn the Gaussian process model. While this is no doubt a challenging research problem, I believe that solving it could lead to the creation of an extremely useful model for systems biology.

Appendix A

Additional Results on the European Bird Atlas Data

This appendix contains additional results on the application of network inference to the European Bird Atlas Data, as described in Chapter 2. There results were originally included as supplementary material for the paper: ‘Inferring species interaction networks from species abundance data: A comparative evaluation of various statistical and machine learning methods”, which appeared in Ecological Informatics (Faisal et al., 2010). The work presented here was not done by me, but by Ali Faisal in collaboration with Colin Beale. It is included for completeness, as some sections of Chapter 2 refer to it, and because my work on the simulation study of ecological networks as described in Chapter 2 formed the basis for some of the decisions taken when applying the methods to the European Bird Atlas Data.

A.1 *A priori* network construction

To construct the *a priori network*, we used two sources: knowledge from the literature, and expert judgement.

First, we searched the ecological literature using ISI Web of Knowledge [1] (accessed on 10/5/09). For each species, we searched for all articles using the complete scientific name. If more than 100 articles were returned, we refined the search adding the terms ’interaction’ or ’competition’. We studied all abstracts and identified papers containing information about interspecific interactions for detailed reading. We identified 30 interactions using this method.

[1] Found at `http://www.isiwebofknowledge.com/`.

For the remaining 711 pairwise interactions we used our expert judgement to answer the question: In areas where these species occur in close proximity, is it plausible that one of the species would become more abundant or expand into different habitats if the other species were absent? In cases where we considered this likely we recorded an interaction in the network.

The final network can be found at `http://www.bioss.ac.uk/students/frankd.html`.

A.2 Phylogenetic distance analysis

To calculate the phylogenetic distances between warbler species, we first needed to get general information on warbler phylogeny. To that end, we searched the taxonomic literature (e.g. Alstroem et al. (2006)) and 'Tree of Life' servers (such as The Tree of Life Web Project in Maddison et al. (2007)). A conservative consensus tree was generated depicting relationships between the 39 warbler species as in Figure A.1.

As path lengths were unavailable we computed a range of distances using the method advocated by Grafen (1989) with values of ρ of 1, 0.6 and 0.3. Although correlations between the phylogenetic distance and recovered interaction scores were not qualitatively different when these different distances were assumed, they are arbitrary choices none the less. Consequently, we repeated the correlation analysis using Kendall's τ as a measure of rank correlation that is unaffected by assumed branch lengths. Again, results were qualitatively similar; they can be found in Table A.1. For the correlation analyses we used only data from the upper triangle of the distance matrices.

A.3 Ecological distance analysis

Ecological trait data for each of the 39 species is presented in Table A.2. From the habitat and migration status data we generated indicator variables identifying species with shared habitat and shared migration strategy. We combined these indicator variables with the morphological data and clutch size, centred and scaled each variable and calculated the Euclidian distance. As with the phylogenetic distance analysis, we used only data from the upper triangle of the distance matrix in correlation analyses.

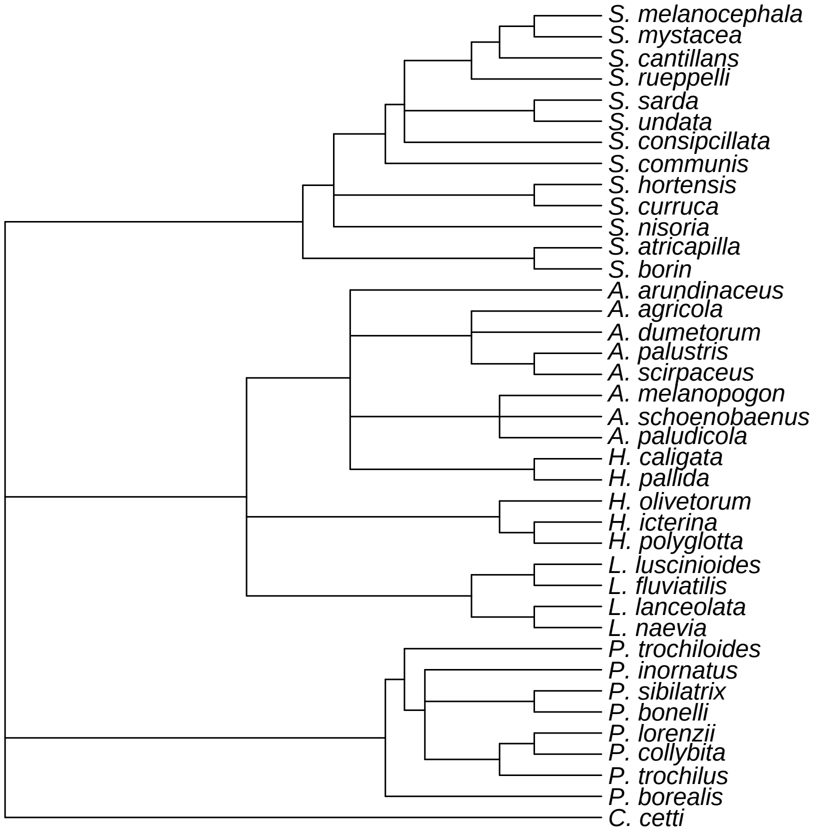


Figure A.1: Phylogenetic tree for the warbler species in our study.

Network	**ρ**	**Correlation**
Basic	1	-0.12 (-0.18, -0.04)
	0.6	-0.11 (-0.18, -0.04)
	0.3	-0.12 (-0.19, -0.05)
	Kendall's τ	-0.08
Spat. Autocorr	1	-0.12 (-0.19, -0.05)
	0.6	-0.12 (-0.19, -0.05)
	0.3	-0.12 (-0.19, -0.05)
	Kendall's τ	-0.08
Spat. Autocorr. and Bio-Climate Covariates	1	-0.14 (-0.21, -0.07)
	0.6	-0.14 (-0.21, -0.07)
	0.3	-0.12 (-0.22, -0.07)
	Kendall's τ	-0.09

Table A.1: Correlation coefficients of reconstructed networks with the phylogenetic tree whose branch lengths have been generated with different values of ρ, or with Kendall's τ. Numbers in brackets show the confidence intervals at 95%. None of the confidence intervals includes zero, indicating that the correlations are significant.

Species	Length	Mass	Wingspan	Clutch	Migrant status	Preferred Habitat
Acrocephalus agricola	13	11	16	4.5	Long Distance	Resident
Acrocephalus arundinaceus	20	33	26	4.5	Long Distance	Resident
Acrocephalus dumetorum	13	12	18	5	Long Distance	Resident
Acrocephalus melanopogon	12	12	16	4.5	Short Distance	Resident
Acrocephalus paludicola	13	12	18	5	Long Distance	Resident
Acrocephalus palustris	13	13	20	4.5	Long Distance	Shrub
Acrocephalus schoenobaenus	13	12	19	5	Long Distance	Shrub
Acrocephalus scirpaceus	13	13	19	4	Long Distance	Resident
Cettia cetti	14	13.5	17	4.5	Resident	Resident
Hippolais icterina	14	13	22	4.5	Long Distance	Broad-leaf Forest
Hippolais olivetorum	15	18	25	3.5	Long Distance	Broad-leaf Forest
Hippolais pallida	13	11	20	2.5	Long Distance	Shrub
Hippolais polyglotta	13	13	18	4	Long Distance	Broad-leaf Forest
Locustella fluviatilis	13	17	20	6	Long Distance	Shrub
Locustella luscinioides	14	18	20	5	Long Distance	Resident
Locustella naevia	13	14	17	5.55	Long Distance	Shrub
Phylloscopus collybita collybita	10	9	18	5.49	Short Distance	Broad-leaf Forest
Phylloscopus bonelli bonelli	12	9	18	5	Long Distance	Broad-leaf Forest
Phylloscopus borealis	11	10	19	5.5	Long Distance	Pine Forest
Phylloscopus trochiloides	10	8	18	5.49	Long Distance	Pine Forest
Phylloscopus sibilatrix	12	10	22	5.77	Long Distance	Broad-leaf Forest
Phylloscopus trochilus	11	10	19	5.93	Long Distance	Broad-leaf Forest
Sylvia atricapilla	13	21	22	4.56	Short Distance	Shrub

Sylvia borin	14	19	22	4.32	Long Distance	Shrub
Sylvia cantillans	12	11	17	4	Long Distance	Garrigue
Sylvia communis	14	16	20	4.64	Long Distance	Shrub
Sylvia conspicillata	12	10	15	4.5	Short Distance	Garrigue
Sylvia curruca	13	12	18	4.67	Long Distance	Shrub
Sylvia hortensis	15	21	22	5	Long Distance	Broad-leaf Forest
Sylvia melanocephala	14	13	16	4.5	Resident	Garrigue
Sylvia nisoria	16	25	25	4.5	Long Distance	Shrub
Sylvia rueppelli	14	14	20	5	Long Distance	Broad-leaf Forest
Sylvia sarda	12	10	16	4	Resident	Garrigue
Sylvia undata	12	10	16	4	Resident	Garrigue
Sylvia mystacea	14	10	17	4.5	Short Distance	Shrub
Hippolais caligata	12	10	20	3.5	Long Distance	Shrub
Phylloscopus inornatus	10	7	17	4	Long Distance	Pine Forest
Phylloscopus lorenzii	10	8	18	5.5	Short Distance	Broad-leaf Forest
Locustella lanceolata	12	12	15	5	Long Distance	Shrub

Table A.2: Ecological traits for the warbler birds.

	A priori net	Phylogenetic Dist.	Ecological Dist.
A priori net	1	0.38 (0.73, 0.02)	0.08 (0.21, -0.05)
Phylogenetic Dist.		1	0.28 (0.21, 0.34)
Ecological Dist.			1

Table A.3: Results of comparison between the ecological measures represented by the *a priori* interaction network, phylogenetic distance and ecological distance. *A priori* comparisons made with logistic regression are the regression coefficient, other results are Pearson's correlation coefficients, all with 95% confidence intervals

A.4 Comparison of Ecological Measures

We have three different ecological indicators that we can compare our reconstructed networks to: The a priori network, the phylogenetic distance and the ecological distance. The correlation of these indicators with the reconstructed networks presented in Chapter 2 is always significant, but also far from perfect correlation. This can be explained by the fact that these measures are not a true gold standard. In fact, each measure captures different aspects of the true relationships between species. In Table A.3 we present the correlation coefficients between the three ecological measures and show that they are also small but (mostly) significant.

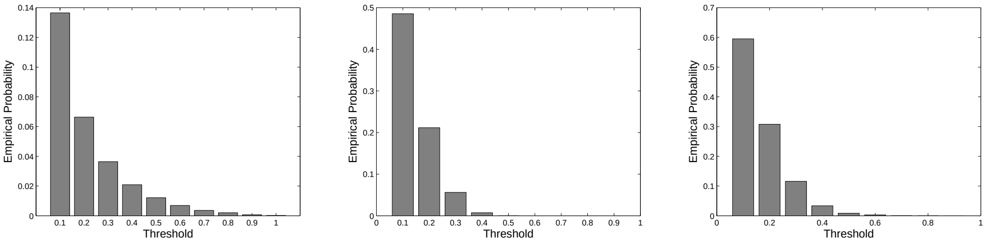
Another way to compare the ecological indicators is by taking the a priori network as a gold standard, and calculating the AUC and TPFP5 values for the phylogenetic and ecological distance measures. In effect, we are treating these distance measures as inverse edge scores. The results are shown in Table A.4. Again, The scores are better than random expectation (AUC=0.5, TPFP5=0.05), but far from perfect (AUC=TPFP5=1.0). This indicates that the various measures capture relevant, but only partial aspects of the unknown true interaction network.

A.5 Thresholding on Edge Interactions

To produce a single, interpretable network from the edge interaction strengths, we need to set a threshold to discard edges with low values. Recall that the "interaction strengths" are of different nature: marginal posterior probabilities for Bayesian net-

	AUC	TPFP5
Phylogenetic Distance	0.79	0.37
Ecological Distance	0.67	0.22

Table A.4: Comparison between the ecological measures by computing AUC and TPFP5 scores for phylogenetic and ecological distance measures, using the *a priori* interaction network as a gold standard.



(a) BN (b) LASSO (c) Consensus

Figure A.2: Distribution of edge strengths/posterior probabilities under the null hypothesis, averaged over 15,210 random species interactions from permuted data.

works, and regularised regression coefficients for LASSO. We would like to map them to p-values, which are more commonly used in statistics. To this end, we carried out a randomisation test. The rows and columns of the original warbler data were permuted ten times, and on each of these replications we carried out the same inference as for the original data. Since the permutation destroys all genuine associations among the species, the distribution of "interaction strengths" represents the null hypothesis of no species interaction. From this distribution, the p-value is easily computed as the probability of exceeding a given threshold.

Figure A.2 shows the null distributions obtained for Bayesian networks (left panel), LASSO (centre panel), and the consensus network (right panel). Table A.5 shows the "interaction strengths" corresponding to p-values of 0.1 and 0.01. Note that the p-values are used as descriptive measures, and no Bonferroni correction (which would be too conservative) was carried out.

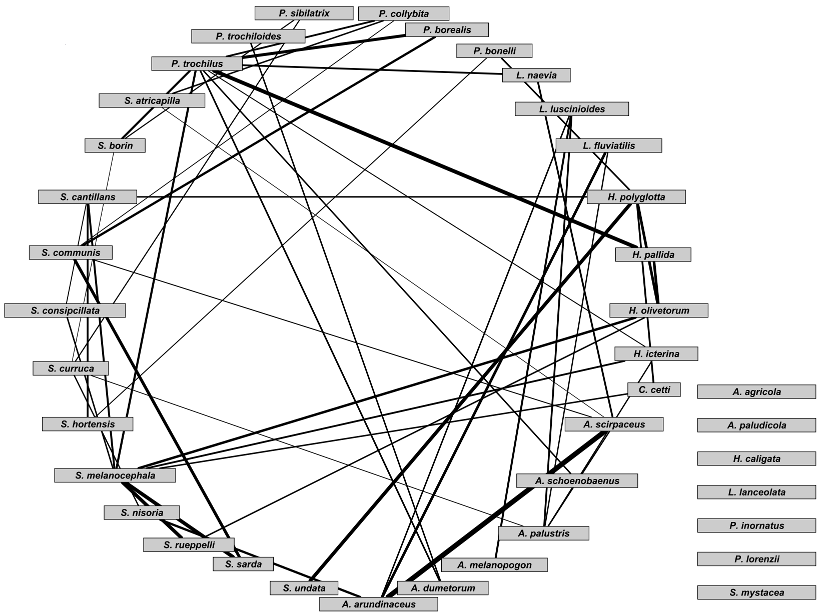


Figure A.3: Consensus network recovered from the basic dataset (without spatial autocorrelation or bio-climate covariates). The edges are pruned by placing a threshold value of 0.5 on the consensus network, which corresponds to a p-value of 0.01. See Section A.5 for a description of how these p-values were calculated. The boxes on the right show unconnected species.

A.6 Recovered Networks

Figures A.3-A.5 shows the consensus networks that were recovered from the warbler data. We get three different networks: one for the basic dataset, one for a dataset where we have modelled spatial autocorrelation as described in Section 2.3.2.1, and one for a dataset where we have included both spatial autocorrelation and two bio-climate covariates: temperature and availability of water. Details on how the sparsity and the correlation with the ecological measures vary for the different networks can be found in the main thesis (Section 2.7).

A.7 Network Characterisation

Studies have shown that molecular regulatory networks have degree distributions that approximately follow a power-law (Wagner, 2001; Guelzim et al., 2002; May, 2006). Loosely speaking, this means that there are many nodes with only one or few con-

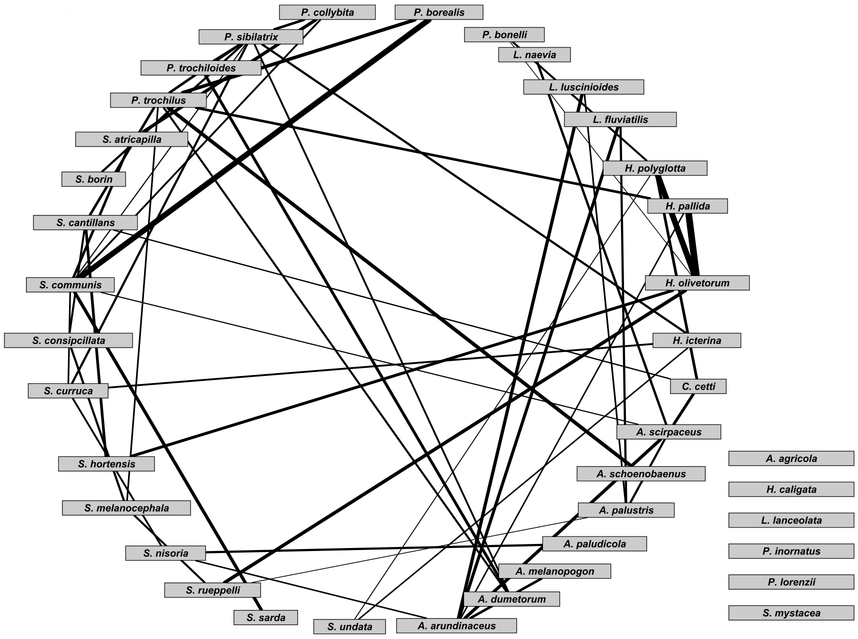


Figure A.4: Consensus networks recovered from the dataset with spatial autocorrelation included (but without bio-climate covariates). The edges are pruned by placing a threshold value of 0.5 on the original consensus network, which corresponds to a p-value of 0.01. See Section A.5 for a description of how these p-values were calculated. The boxes on the right show unconnected species.

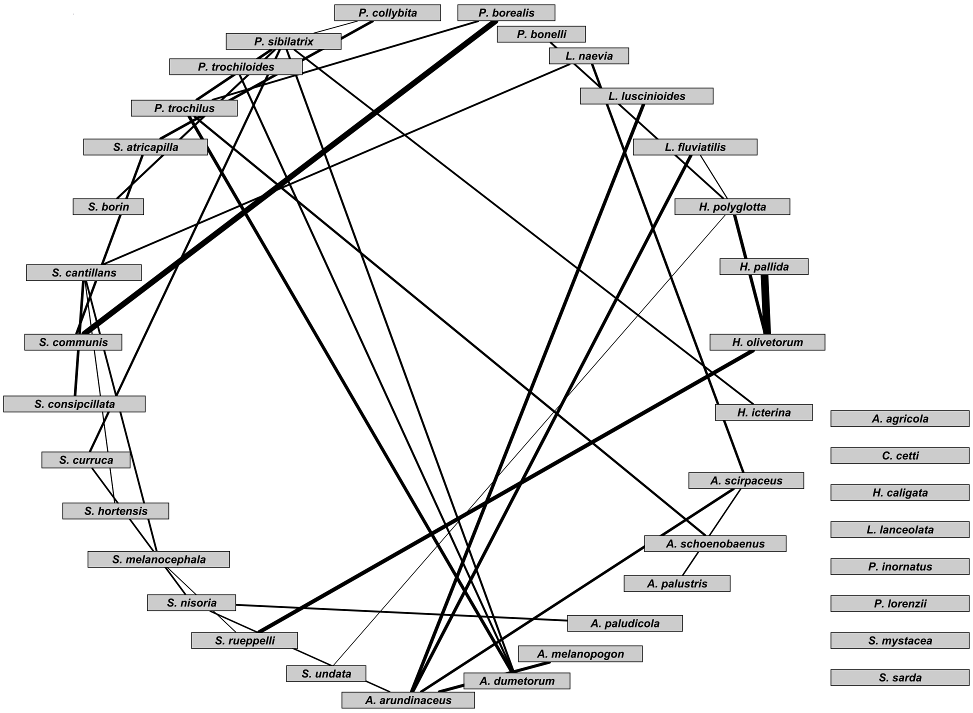


Figure A.5: Consensus networks recovered from the dataset with both spatial autocorrelation and bio-climate covariates included. The edges are pruned by placing a threshold value of 0.5 on the original consensus network, which corresponds to a p-value of 0.01. See Section A.5 for a description of how these p-values were calculated. The boxes on the right show unconnected species.

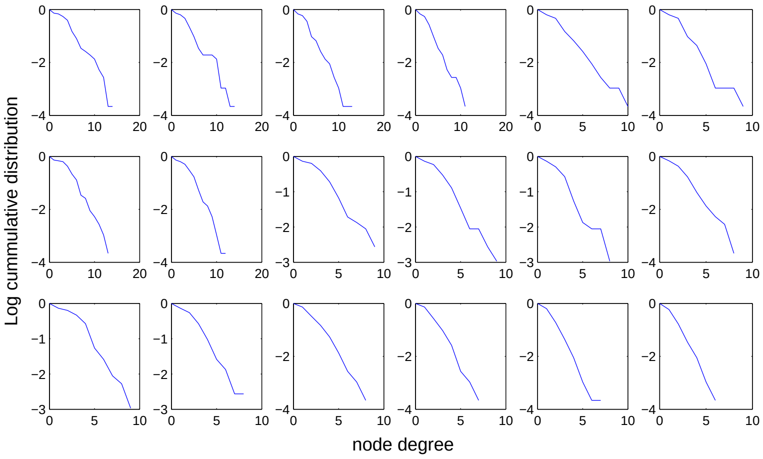


Figure A.6: Cumulative degree distribution for the consensus networks on the **log-linear scale** as the threshold varies. (Top) Basic bird data, (Middle) Bird data with spatial autocorrelation model added, (Bottom) Birds with spatial autocorrelation and bio-climate covariates. From left to right the thresholds are set at p-values 0.2, 0.15, 0.1, 0.05, 0.02, and 0.01.

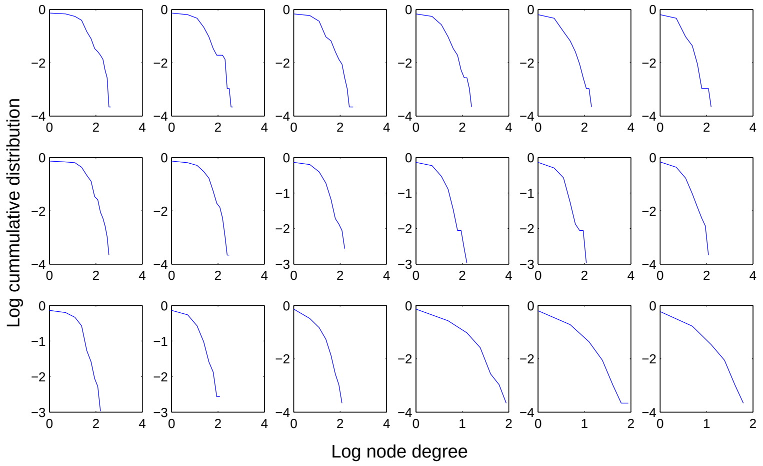
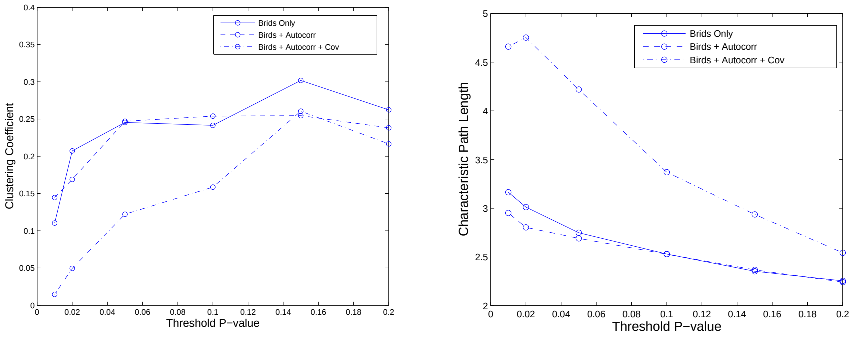


Figure A.7: Cumulative degree distribution for the consensus networks on the **log-log scale** as the threshold varies. (Top) Basic bird data, (Middle) Bird data with spatial autocorrelation model added, (Bottom) Birds with spatial autocorrelation and bio-climate covariates. From left to right the thresholds are set at p-values 0.2, 0.15, 0.1, 0.05, 0.02, and 0.01.



(a) Clustering Coefficient

(b) Network Diameter

Figure A.8: Variation of the clustering coefficient and network diameter for the consensus networks as the threshold varies.

nections, but also some nodes with many more connections than the average degree. Studies on food webs generally agree that the degree distribution is not Poisson (Proulx et al., 2005), however they disagree on whether the degree distributions are best fit by a power-law or by some other distribution. The existence of a variety of distributions has been shown, including power-law, truncated power-law and exponential (Dunne et al., 2002; Jordano et al., 2003; Laird and Jensen, 2006). In our study we observe that the distributions are closer to linear on the log-linear plot of the cumulative degree distribution (Fig. A.6), than on the log-log plot (Fig. A.7). Linearity on the log-log plot would be characteristic of a power-law distribution, but linearity on the log-linear plot shows that the network exhibits a near exponential distribution. The data also displays the insensitivity of this behaviour to varying the threshold.

Figure A.8 shows the variation of the clustering coefficient and the network diameter (characteristic path length) as the threshold varies. There is no discernible trend, which may mean that these particular statistics are not useful characterisations of the types of networks that we are considering.

p-value	**BN**	**LASSO**	**Consensus**
0.1	0.2	0.3	0.4
0.01	0.5	0.4	0.5

Table A.5: Mapping from p-value thresholds to edge strengths/posterior probabilities.

Appendix B

Additional Simulation Results for Global Information Sharing

Note: The results in this appendix have been adapted from the paper ”Dynamic Bayesian networks in molecular plant science: Inferring gene regulatory networks from multiple gene expression time series” (Dondelinger et al., 2012a).

B.1 Motivation

This appendix contains additional network inference experiments using the model from Chapter 3. More specifically, we test the global information sharing approach (TVDBN-GI) from Section 3.3.2, and compare it with the model without information sharing (TVDBN-0). The simulation model described below in Section B.2 is subtly different from the one in Section 3.4, in that here the changes are not applied sequentially, but are applied to the same underlying network for each new time series that is generated. This results in a setup that is better described as several datasets recorded under different conditions, rather than one time-varying dataset. However, the results using this simulation model are qualitatively similar to those reported in Section 3.6.1, in that TVDBN-GI outperforms TVDBN-0 due to the advantage conferred by information sharing. For this reason, I have not included these results in Chapter 3.

B.2 Simulation Model

The simulation model produces a time series of data points, each of which represents the normalised expression values of a gene. We start with a network M with the num-

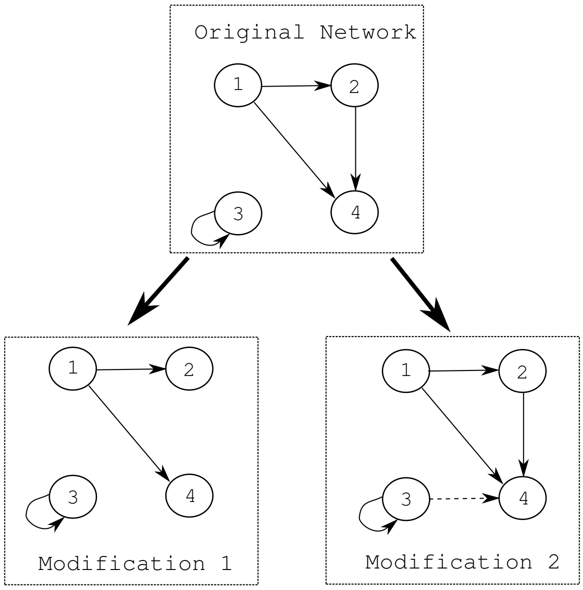


Figure B.1: Simulation model process: The original network is modified to obtain two new networks, each with a different change with respect to the original network.

ber of parents for each node drawn from a sparse Poisson prior (to keep the number of interactions low). Each directed interaction from gene A (the parent) to gene B (the child) has a weight that measures how much gene A will influence gene B. To ensure that the expression values stay at equilibrium, we test if the absolute value of all eigenvalues of the matrix of weights is less than 1, and remove interactions randomly until this condition is satisfied. The value $x_i(t)$ of each variable at time t is calculated using a linear regression, as in Section 3.4.

To simulate measurements under different experimental conditions, we applied two strategies: Changing the standard deviation σ_i of the noise in the regression in order to reflect the fact that the measurement noise may vary across time, and using a modified network that introduces a small number of changes with respect to the originally generated network (adding and removing interactions), reflecting the assumption that not all pathways are active all of the time. Fig. B.1 shows an example network with four nodes and the modified networks that have been generated from it.

For our experiments, we generated two kinds of datasets: One with long time series to test how well our methods can reconstruct the network when a lot of data is available, and one which replicates the Arabidopsis data (see Section 3.5.3), to see how the performance changes when there are fewer observations available. For the long time series, we generated networks with 10 genes and time series with 50 time steps. We generated 10 different datasets under two conditions: In the first case we did not modify the original network structure, while in the second case, we introduced an average of two changes for the modified networks. We changed the noise level for each network; we used $\sigma = 1$ as the starting value and added a value in the range $[-0.5, 0.5]$, drawn from a uniform random distribution. For the Arabidopsis-like time series, we generated networks with 9 genes and time series with 13 time steps. We generated 4 different datasets under the same two conditions as before, and used the same procedure to vary the noise levels.

B.3 Recovering Simulated Networks

Setup We used the simulation model presented in Section B.2 to generate time series from an underlying network under two different conditions:

1. All time series are generated using the structure of the underlying network, but varying the interaction weights and noise level.

2. Each time series is generated from a different network where we introduce a small number of changes (10%) with respect to the structure of the underlying network. We also vary the interaction weights and noise level.

The second condition should provide a more difficult inference problem than the first one, since there is less scope for information sharing. For both cases, we generate ten independent datasets, each with a different underlying network, to allow us to carry out paired t-tests for significance.

Results We used three measures to evaluate the performance of our methods: The area under the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate versus the false positive rate, the area under the precision-recall (PR) curve, which plots precision (fraction of true positives out of detected interactions) versus recall (fraction of true positives out of actual interactions; another name for the true positive rate) and the true positive rate at a false positive rate of 5% (TPFP5). We obtain the curves for the first two scores by varying a threshold on the marginal posterior probability of the interactions, and by only keeping those interactions that lie above the threshold at each point. The ROC curve will always be increasing from (0,0) to (1,1), while the precision-recall curve does not have to follow such a clear

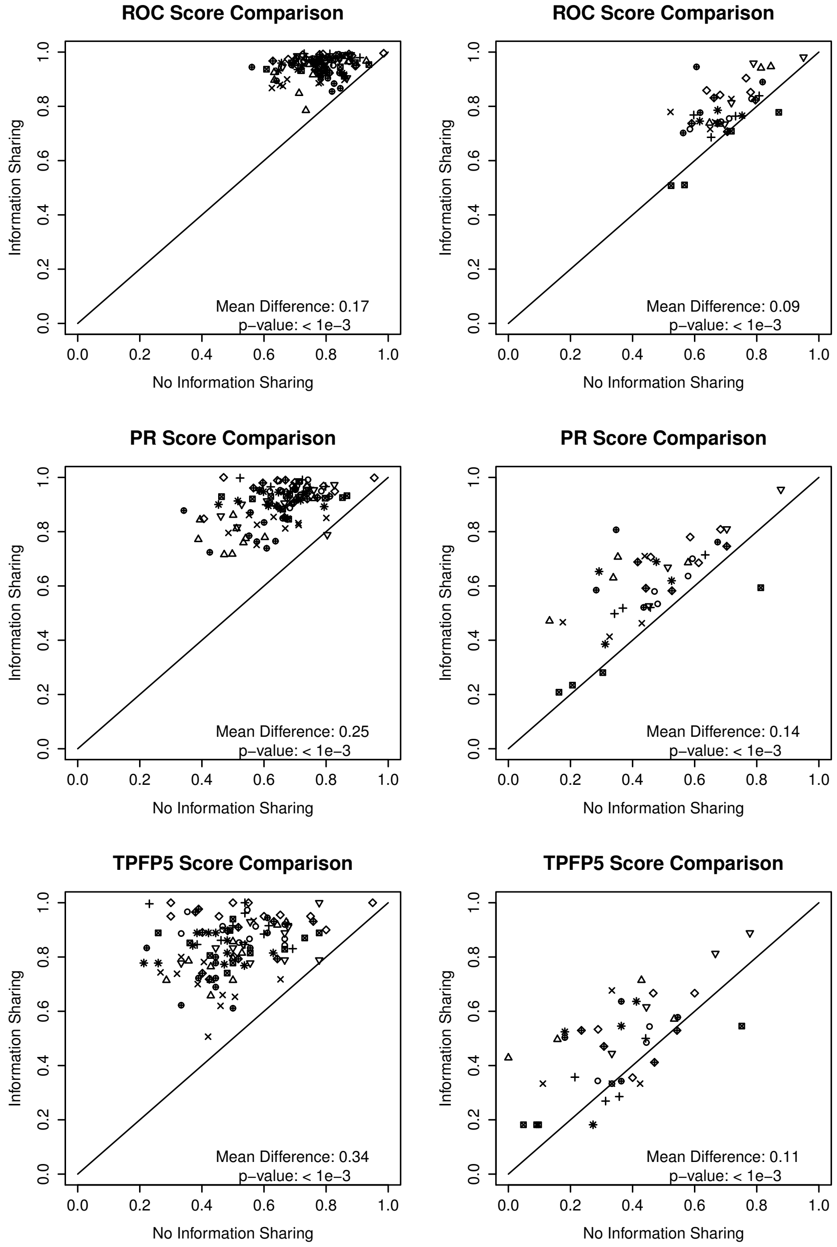


Figure B.2: **Same Structures**: Comparison of network reconstruction performance using the DBN model with global information sharing (TVDBN-GI) and without information sharing (TVDBN-0). Left Column: For each underlying network, we generated 10 time-series of length 50 without changing the network structures. Right Column: For each underlying network, we generated 4 time-series of length 13 without changing the network structures. Top Row: Area under the ROC curve score. Middle Row: Area under the precision-recall curve. Bottom Row: True positive rate at 5% false positives. In each case, a score of 1 denotes perfect reconstruction of the network. Points with the same symbol are the scores of networks reconstructed from different time series associated with a single underlying network.

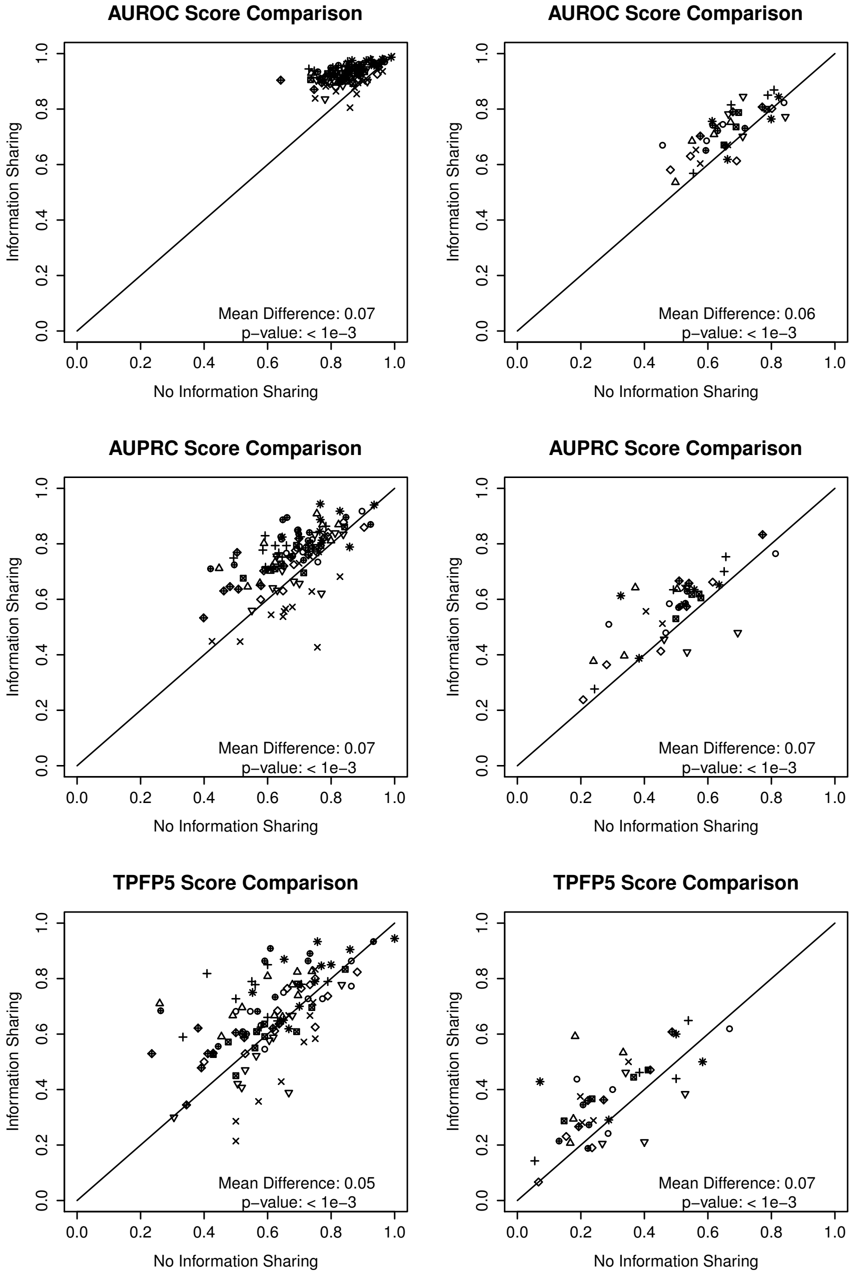


Figure B.3: **Different Structures**: Comparison of network reconstruction performance using the DBN model with global information sharing (TVDBN-GI) and without information sharing (TVDBN-0). Left Column: For each underlying network, we generated 10 time-series of length 50, changing about 10% of the network structure each time. Right Column: For each underlying network, we generated 4 time-series of length 13, changing about 10% of the network structure each time. Top Row: Area under the ROC curve score. Middle Row: Area under the precision-recall curve. Bottom Row: True positive rate at 5% false positives. In each case, a score of 1 denotes perfect reconstruction of the network. Points with the same symbol are the scores of networks reconstructed from different time series associated with a single underlying network.

trend (although precision will generally decrease as recall increases). Taking the area under the curve allows us to reduce the curve to one number that indicates the overall performance[1]. A perfect score for all three methods is a score of 1, which means that we always retrieve all of the true positives, and don't retrieve any false positives at the highest threshold.

These measures are interesting for different reasons: The ROC curve describes the overall performance of the network reconstruction method over positives and negatives, while the precision-recall curve is of practical interest because it does not include the true negatives, and hence focuses on how well the true edges are reconstructed. The TPFP5 score gives the fraction of true edges that we could expect to retrieve at a reasonable fraction of false positives.

Looking at the comparison in Figs. B.2-B.3, it is clear that the global information sharing model (TVDBN-GI) outperforms the DBN approach without information sharing (TVDBN-0). In every case, there is a significant improvement in the score when we apply information sharing. The improvement is most drastic when the underlying network structure is unchanged (Fig. B.2). Applying small changes to the network structure for each new simulated time-series (Fig. B.3) leads to a smaller, but still significant improvement.

The relative performance between the model with and without information sharing is similar whether we use long time series (left column in Figs. B.2-B.3) or time series that have the same length as the Arabidopsis data (right column). The improvement is larger for longer time series, however, most likely due to there being more datasets available that can benefit from the information sharing (10 rather than 4). In absolute terms, the performance increases when the time series are longer, which is reasonable because it means that more data is available. Nevertheless, the performance with simulated time series of the same length as the Arabidopsis data is still reasonable, and there is a definite increase in the accuracy of the reconstructed networks when using information sharing. This is an encouraging finding, which motivates the application of our method to the Arabidopsis gene expression time series in Chapter 3. Note that for the latter, an objective evaluation in terms of network reconstruction scores was not feasible owing to the lack of a proper gold standard.

We notice that overall the PR scores are less impressive than the ROC scores. This is a consequence of the sparseness in the model; we have more non-interactions than

[1] In order to calculate the area, we need to interpolate to find additional points of the curve. For the ROC curve, this is a straightforward linear interpolation, while for the precision-recall curve, we follow Davis and Goadrich (2006).

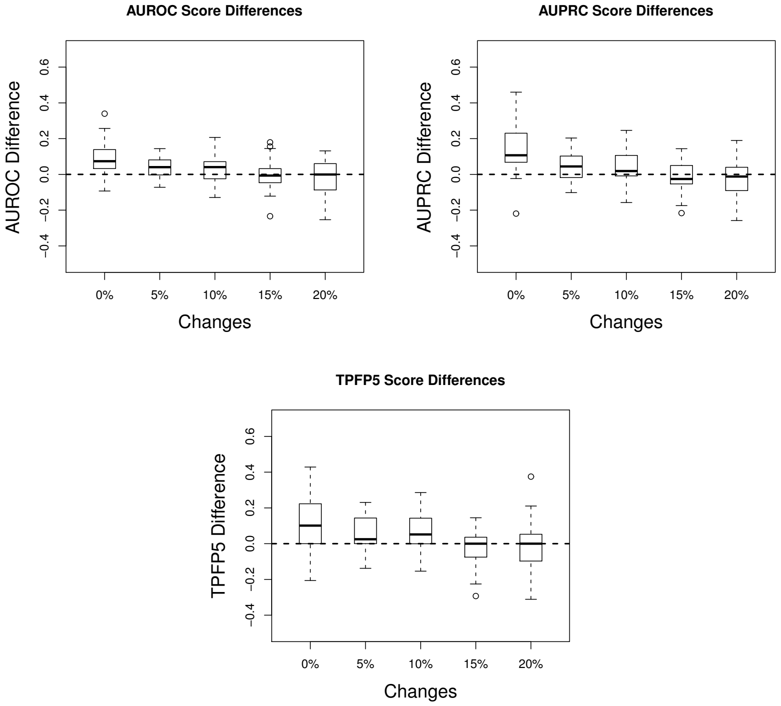


Figure B.4: **Influence of the Number of Changes**: We vary the number of changes applied to each network from 0% to 20%. Network reconstruction performance is measured using the area under the ROC curve (AUROC) score, Area under the precision-recall curve (AUPRC) score, and the true positive rate at 5% false positives (TPFP5).The boxplots show the difference of the network reconstruction scores with global information sharing (TVDBN-GI) to those without (TVDBN-0); larger differences indicate better performance of information sharing, 0 means they perform equally well. The horizontal bar of each boxplot shows the median, the box margins show the 25th and 75th percentiles, the whiskers indicate data within 2 times the interquartile range, and circles are outliers.

interactions in the simulated network, and our DBN model favours fewer interactions, which means that we are more likely to detect true negatives correctly. This improves the false positive rate, but has no effect on the precision, meaning that the PR curve is not going to reflect this. This makes the PR curve (and the TPFP5 score) a better measure if we are more interested in the retrieved interactions than in the non-interactions. The trend is the same, however, in that the improvement when using information sharing is smaller (but still significant) when we apply small changes to the underlying network before simulating the data.

Further simulations, where we increased the noise levels to $\sigma = 2$ and doubled the number of changes in the network structure, showed that the benefit obtained through information sharing is robust to noise, but does not persist when the number of changes becomes too large, as could be expected. It is reasonable to ask how much of a topology disturbance we can have while still getting a significant improvement with information sharing. Fig. B.4 plots the difference in network reconstruction scores between no information sharing and information sharing as the number of changes varies between 0% of the network and 20%. Note that the sparseness of gene regulatory networks means that 20% of the network represents a sizeable portion of the gene interactions. For example, if the original network has 10 genes, then 20% represents 20 interactions that change, so on average each gene will change two of its regulators. The crossover point where information sharing no longer gives a significant improvement seems to be around 15% of the network changing.

Appendix C

DREAM 5

C.1 Introduction

Note: The material in this appendix is closely based on the conference paper ”A Bayesian regression and multiple changepoint model for systems biology”, presented at the International Workshop on Statistical Modelling 2011 (Dondelinger et al., 2011). Most of the material has been reprinted verbatim.

In this appendix, I describe a Bayesian regression and multiple changepoint model, with Bayesian inference based on reversible jump Markov chain Monte Carlo (RJMCMC) (Green, 1995). This work was done in collaboration with Andrej Aderhold at Biomathematics and Statistics Scotland, Sophie Lèbre at the University of Strasbourg, and Marco Grzegorczyk at the University of Dortmund. The model is essentially identical to the one described in Chapters 3 and 4, but instead of a regression based on the previous timepoint, as in a DBN, the regression is based on the current timepoint. This allows us to reconstruct networks from non-timecourse data. My team developed this model to participate in a gene regulatory network prediction competition (DREAM 5), which ensured that the comparative evaluation with other methods was done objectively. A consortium paper about the challenge has since been published in Nature Methods (Marbach et al., 2012), which includes a contribution from our team. Results from the consortium paper are discussed at the end of the appendix in Section C.4.

C.2 Model

Multiple changepoints: Let p be the number of target genes, whose expression values $y = \{y_i(t)\}_{1 \le i \le p, 1 \le t \le N}$ are measured on N separate chips. $\mathcal{M}_i$ is the set of parents (regulators) associated with target gene i in the gene regulatory network. We model the differences in the regulatory relationships measured by different chips (assumed to be in some natural order, e.g. a time series) with a multiple changepoint process. For each target gene i, an unknown number k_i of changepoints define k_i+1 non-overlapping segments. Segment $h \in \{1,..,k_i+1\}$ starts at changepoint ξ_i^{h-1} and stops before ξ_i^h, so that $\xi_i = (\xi_i^0,...,\xi_i^{h-1},\xi_i^h,...,\xi_i^{k_i+1})$ with $\xi_i^{h-1} < \xi_i^h$.

This changepoint process induces a partition of the chip ordering, $y_i^h = (y_i(t))_{\xi_i^{h-1} \leq t < \xi_i^h}$. The network structure $\mathcal{M}_i$ remains the same for each segment h, but the other parameters of the model can vary.

Regression model: For all genes i, the random variable $Y_i(t)$ refers to the expression of gene i on chip t. Within any segment h, the expression of gene i at chip t depends on the gene expression values on chip t of a set R_i of m potential regulator genes (parents), with $i \notin R_i$. We define a regression model by (a) the set of s_i parents denoted by $\mathcal{M}_i = \{j_1,...,j_{s_i}\} \subseteq R_i$, and (b) a set of parameters $((a_{ij}^h)_{j \in R_i}, \sigma_i^h)$; $a_{ij}^h \in \mathbb{R}$, $\sigma_i^h > 0$. For all $j \neq 0$, $a_{ij}^h = 0$ if $j \notin \mathcal{M}_i$. For all genes i, for all chips t in segment h ($\xi_i^{h-1} \leq t < \xi_i^h$), the random variable $Y_i(t)$ depends on the m variables $\{Y_j(t)\}_{j \in R_i}$ according to

$$Y_i(t) \ = \ a_{i0}^h + \textstyle\sum_{j \in \mathcal{M}_i} a_{ij}^h \, Y_j(t) \ + \varepsilon_i(t) \qquad \text{(C.1)}$$

where the noise $\varepsilon_i(t)$ is assumed to be Gaussian with mean 0 and variance $(\sigma_i^h)^2$, $\varepsilon_i(t) \sim N(0,(\sigma_i^h)^2)$. We define $a_i^h = (a_{ij}^h)_{j \in R_i}$.

Prior: The k_i+1 segments are delimited by k_i changepoints, where k_i is distributed a priori as a truncated Poisson random variable with mean λ and maximum $\overline{k} = N-2$: $P(k_i|\lambda) \propto \frac{\lambda^{k_i}}{k_i!} \mathbb{1}_{\{k_i \leq \overline{k}\}}$. Conditional on k_i changepoints, the changepoint positions vector $\xi_i = (\xi_i^0, \xi_i^1, ..., \xi_i^{k_i+1})$ takes non-overlapping integer values, which we take to be uniformly distributed a priori. For all genes i, the number s_i of parents for node i follows a truncated Poisson distribution[1] with mean Λ and maximum $\overline{s} = 5$: $P(s_i|\Lambda) \propto \frac{\Lambda^{s_i}}{s_i!} \mathbb{1}_{\{s_i \leq \overline{s}\}}$. Conditional on s_i, the prior for the parent set $\mathcal{M}_i$ is a uniform distribution over all parent sets with cardinality s_i: $P(\mathcal{M}_i \,||\mathcal{M}_i| = s_i) = 1/\binom{p}{s_i}$. The overall prior on the network structures is given by marginalization:

$$P(\mathcal{M}_i|\Lambda) = \textstyle\sum_{s_i=1}^{\overline{s}} P(\mathcal{M}_i|s_i) P(s_i|\Lambda) \qquad \text{(C.2)}$$

Conditional on the parent set $\mathcal{M}_i$ of size s_i, we assume for the prior distribution $P(a_i^h|\mathcal{M}_i, \sigma_i^h)$ of the s_i+1 regression coefficients for each segment h a zero-mean multivariate Gaussian with covariance matrix $(\sigma_i^h)^2 \Sigma_{a_i^h}$, where following Andrieu and Doucet (1999) we set $\Sigma_{a_i^h}^{-1} = \delta^{-2} D_{a_i^h}^{\dagger}(y) D_{a_i^h}(y)$, and $D_{a_i^h}(y)$ is the $(\xi_i^h - \xi_i^{h-1}) \times (s_i+1)$ matrix

[1] A restrictive Poisson prior encourages sparsity of the network, and is therefore comparable to a sparse exponential prior, or an approach based on the LASSO.

whose first column is a vector of 1 (for the constant in model (C.1)) and each $(j+1)^{th}$ column contains the observed values $(y_j(t))_{\xi_i^{h-1}-1\leq t<\xi_i^h-1}$ for all regulatory genes j in $\mathcal{M}_i$. Finally, the conjugate prior for the variance $(\sigma_i^h)^2$ is the inverse gamma distribution, $P((\sigma_i^h)^2) = \mathcal{IG}(\upsilon_0, \gamma_0)$. Following Lèbre et al. (2010), we set the hyperparameters for shape, $\upsilon_0 = 0.5$, and scale, $\gamma_0 = 0.05$, to fixed values that give a vague distribution. The terms λ and Λ can be interpreted as the expected number of changepoints and parents, respectively, and δ^2 is the expected signal-to-noise ratio. These hyperparameters are drawn from vague conjugate hyperpriors, which are in the (inverse) gamma distribution family: $P(\Lambda) = P(\lambda) = \mathcal{G}a(0.5,1)$ and $P(\delta^2) = \mathcal{IG}(2,0.2)$.

Posterior: Equation (C.1) implies that

$$P(y_i^h|\xi_i^{h-1},\xi_i^h,\mathcal{M}_i,a_i^h,\sigma_i^h) \quad \propto \quad \exp\left(-\frac{(y_i^h - D_{a_i^h}(y)a_i^h)^\dagger\ (y_i^h - D_{a_i^h}(y)a_i^h)}{2(\sigma_i^h)^2}\right) \qquad (C.3)$$

From Bayes theorem, the posterior is given by the following equation:

$$P(k,\xi,\mathcal{M},a,\sigma,\lambda,\Lambda,\delta^2|y) \propto P(\delta^2)P(\lambda)P(\Lambda)\prod_{i=1}^{p}P(k_i|\lambda)P(\xi_i|k_i)P(\mathcal{M}_i|\Lambda) \qquad (C.4)$$

$$\prod_{h=1}^{k_i}P([\sigma_i^h]^2)P(a_i^h|\mathcal{M}_i,[\sigma_i^h]^2,\delta^2)P(y_i^h|\xi_i^{h-1},\xi_i^h,\mathcal{M}_i,a_i^h,[\sigma_i^h]^2)$$

Inference: An attractive feature of the chosen model is that the marginalization over the parameters a and σ in the posterior distribution of (C.4) is analytically tractable: $P(k,\xi,\mathcal{M},\lambda,\Lambda,\delta^2|y) = \int P(k,\xi,\mathcal{M},a,\sigma,\lambda,\Lambda,\delta^2|y)da d\sigma$ See Andrieu and Doucet (1999), Lèbre et al. (2010) for details and an explicit expression. The number of changepoints and their location, k, ξ, the network structure $\mathcal{M}$ and the hyperparameters λ, Λ and δ^2 can be sampled from the posterior $P(k,\xi,\mathcal{M},\lambda,\Lambda,\delta^2|y)$ with RJMCMC. A detailed description can be found in Lèbre et al. (2010). The posterior probabilities of the gene interactions submitted to DREAM are obtained from the posterior sample of network structures $\mathcal{M}$ by marginalization.

C.3 Simulations and Result

To assess the performance of the proposed method we participated in a competition organised by the DREAM (Dialogue for Reverse Engineering Assessments and Methods) consortium in autumn of 2010. The goal was to reverse engineer gene regulatory networks from gene expression data sets. Participants were given four microarray compendia and were challenged to infer the structure of the underlying transcriptional regulatory networks. The first compendium was based on an in-silico (i.e.

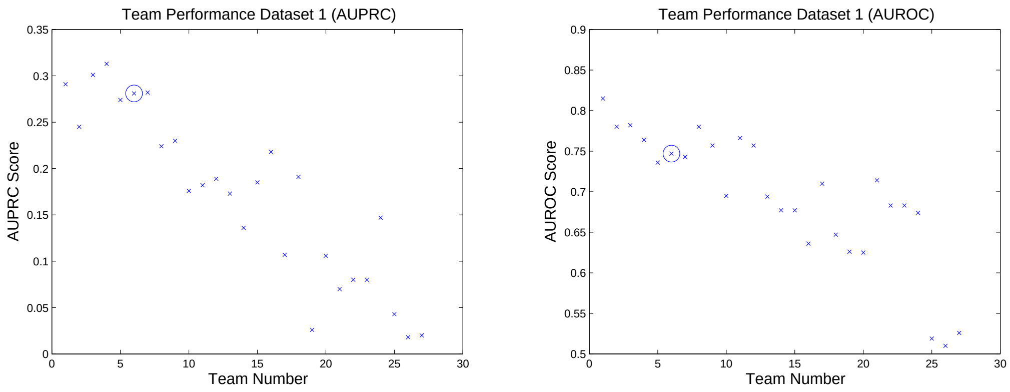


Figure C.1: Areas under the precision recall (left) and ROC (right) curves obtained on an in silico data set by all teams participating in the DREAM 5 competition. The circles indicate the performance of our proposed method.

simulated) network, the other three compendia were obtained from microorganisms. Each compendium consisted of hundreds of microarray experiments, which included a wide range of genetic, drug, and environmental perturbations. More information is available in Table C.1 and at `http://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM_Project`. Network predictions were evaluated by the organisers on a subset of known interactions for each organism, or on the known network for the in-silico case (which is more objective).

Our method assumes an ordering of the microarray chips. While this condition is naturally met for time course experiments, it does not hold for the varying experimental conditions of the DREAM data. We therefore resorted to the heuristic pre-processing step of mapping the high-dimensional gene expression profiles onto a one-dimensional self-organising map (SOM) initialized by the first principal component. We applied the software package *som* in R with default parameter settings. To reduce the computational complexity of the RJMCMC simulations we applied a pre-filtering step based on TESLA (Ahmed and Xing, 2009), a time-varying network inference method based on L1-regularised linear regression. For each gene we identified a set of 20 potential candidate regulators, based on the 20 regression coefficients with the largest modulus.

We assessed the convergence of our simulations with standard diagnostics based on Gelman-Rubin potential scale reduction factors (PSRF). Owing to unexpected downtime of the computer cluster we were using, only the simulations on the first two data sets showed a sufficient degree of convergence (PSRF$\leq$ 1.2); for the latter data sets

Table C.1: This table summarises the information about the DREAM 5 Network Inference Challenge data sets. For each data set, we show which organism it came from, how many genes were measured, how many of those genes were identified as transcription factors (possibly regulatory genes) and how many chips (datapoints) were included.

Data Set	Organism	Genes	Transcription Factors	Chips
1	Synthetic	1643	195	806
2	*S. Aureus*	2810	99	160
3	*E. Coli*	4511	334	805
4	*S. Cerevisiae*	5950	333	536

we submitted the results from TESLA. The second data set was later removed from the evaluation by the organisers. Figure C.1 shows the results for the in silico data set obtained from the rankings of interactions submitted by all participating teams, using two criteria: the area under the precision-recall curve (AUPRC), and the area under the receiver-operator characteristic (AUROC) curve. As discussed in Davis and Goadrich (2006), AUPRC gives a more faithful indication of the network reconstruction accuracy than AUROC, and it is thus seen that our method clearly lies in the group of the 5 top-ranked models. This suggests that it compares favourably with the majority of existing schemes and provides a useful tool for contemporary research in systems biology.

C.4 Consortium Paper and DREAM 5 Outcomes

Following the DREAM 5 Network Inference Challenge, the challenge organisers went on to publish a consortium paper in Nature Methods (Marbach et al., 2012), which includes a contribution from my team. To sum up the paper briefly, they compared over 30 network inference methods (the participating teams plus some off-the-shelf methods) on the four datasets, and showed that while no single method gave the best performance on all datasets, they could achieve robust performance by combining several inference methods. Subsequent experiments showed that of 53 previously unknown regulatory interactions that were predicted with high confidence, 23 could be experimentally validated. More details can be found in Marbach et al. (2012).

These findings indicate that our approach of combining two methods (TESLA for pre-selection, followed by the Bayesian regression and changepoint model) has promise, in that it leverages the strengths of these two approaches. In general, regression methods performed quite well on the simulated data. The best performance overall was achieved by two novel methods, one based on random forests, and the other based on ANOVA. Our method performed third best out of the regression methods, improving on several LASSO-based methods. One of the regression methods that consistently outperformed our method employed the group lasso (Yuan and Lin, 2005) to combine steady-state and time-series data, indicating that perhaps our pre-processing could have been improved by a better grouping. The organisers note, however, that no single class of methods seemed to give best performance, and that performance seemed implementation-dependent.

Participation in this challenge proved to be a very valuable experience, in that it allowed us to compare our methods to the state-of-the-art in a real-world situation, and showed that we can perform competitively with the best the field has to offer. By taking part in DREAM 5, we not only proved that our methods work, but we also contributed to the ongoing effort to improve network inference and to learn more about the problem itself. At the same time, the challenge highlighted the inherent difficulty of network inference (even the best combined predictor only achieved a precision of about 50%), which indicates that there is much work yet to be done.

Appendix D

Parameter Inference in the JAK/STAT pathway

This appendix contains additional results on ODE parameter inference in the JAK/STAT pathway. These are the results for all 13 species, which have been omitted from Chapter 5 in favour of only including a selection of example species to illustrate each observation. This was done to keep the presentation clear and concise. Here I present the full results for completeness. Section D.1 shows the complete results for parameter inference using AGM with the sigmoid kernel. Section D.2 shows the full results of the perturbation study described in Chapter 5, Section 5.5.3.3. The dataset used for all the experiments in this appendix is identical to the one presented in Chapter 5, Section 5.5.

D.1 Full Results for Sampled Parameters

This section shows the full results for sampled parameters using AGM that have been omitted from Chapter 5 for conciseness and clarity of presentation.

D.1.1 Perfect Observation (No Noise)

For data with no added Gaussian observation noise, Figures D.1-D.13 show the predictions obtained by the sampled parameters, as well as the latent variables sampled during the MCMC run, compared to the true noiseless profile for that species. They also show the distributions of the sampled parameters.

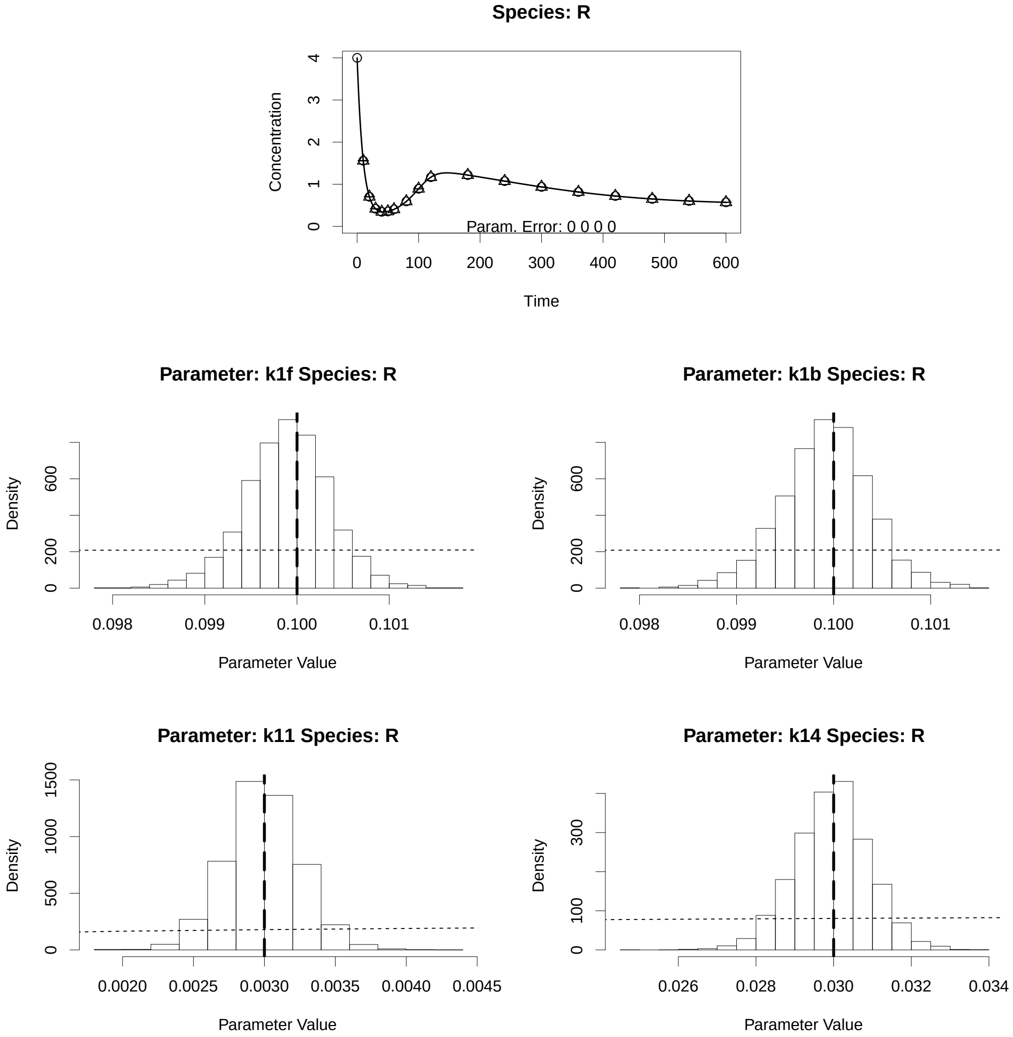


Figure D.1: R - Species 1, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

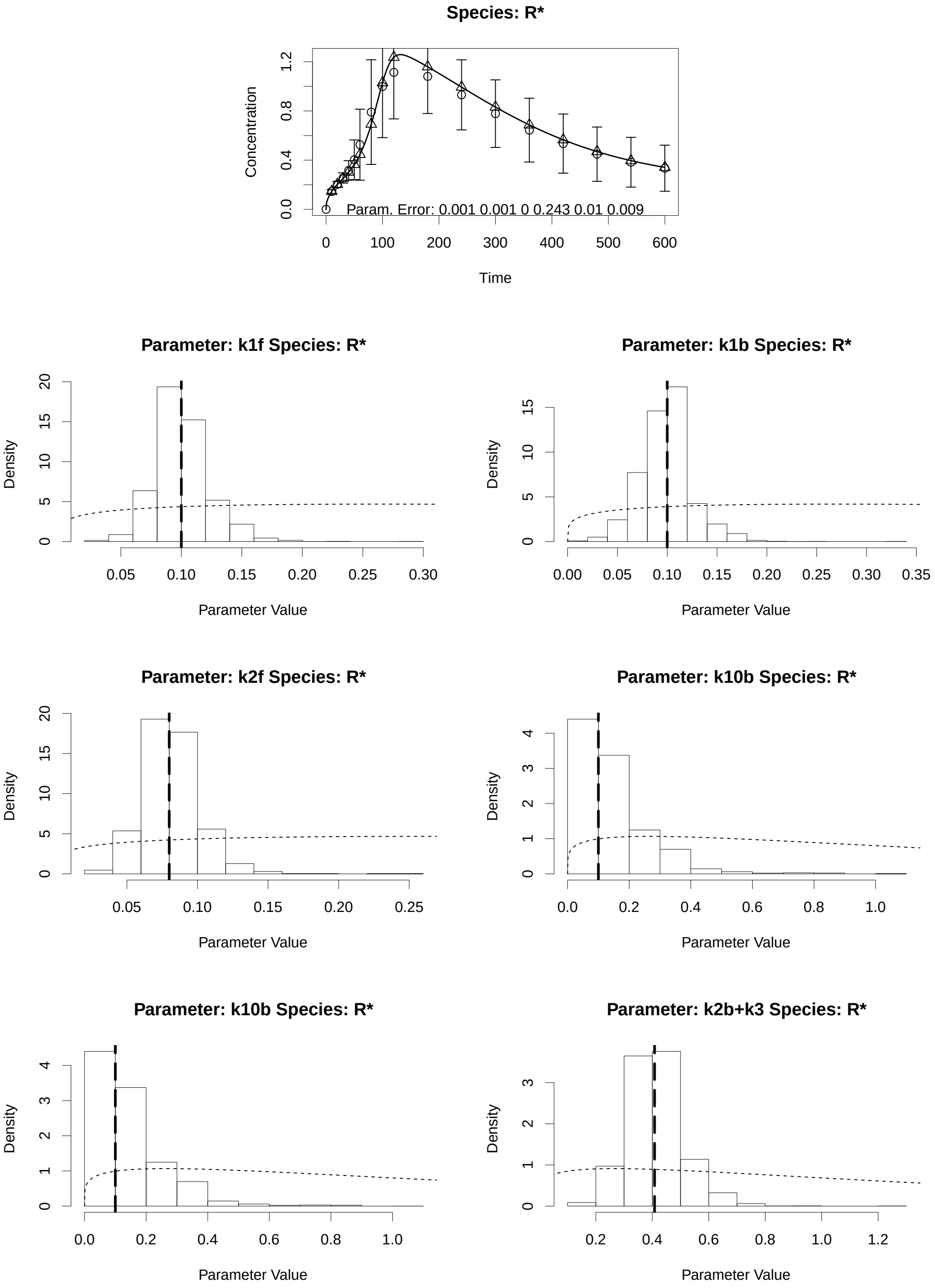


Figure D.2: R* - Species 2, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

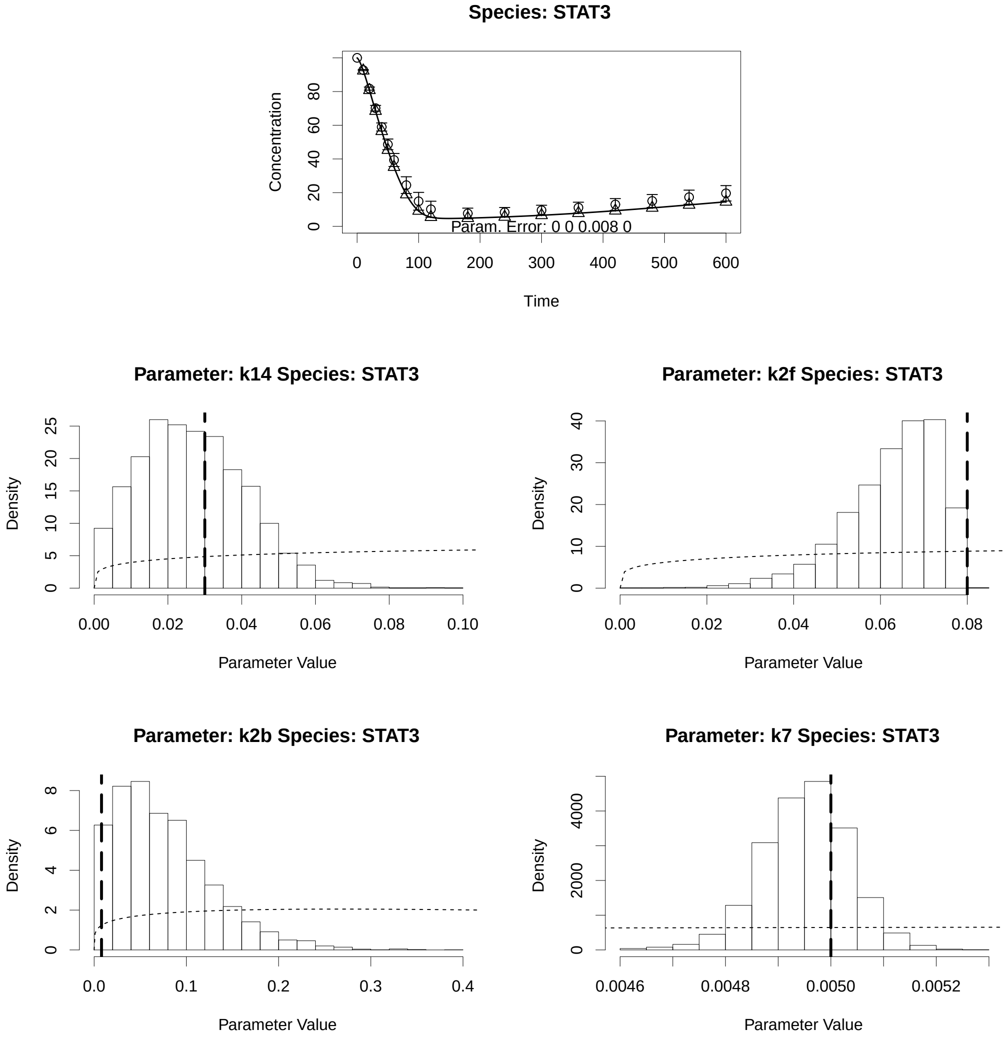


Figure D.3: STAT3 - Species 3, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

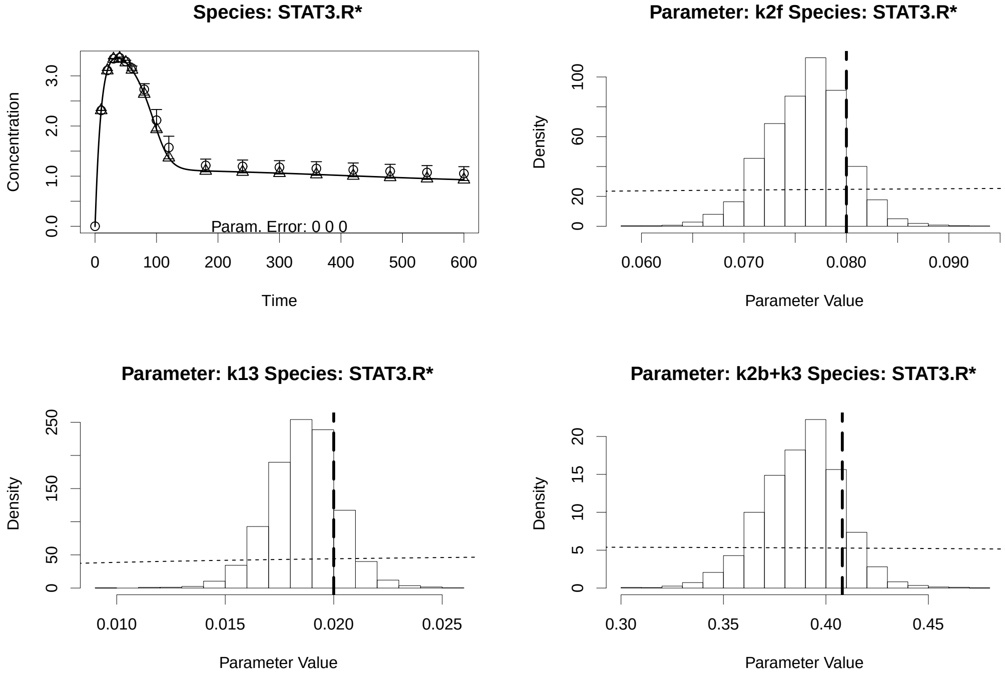


Figure D.4: STAT3.R* - Species 4, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

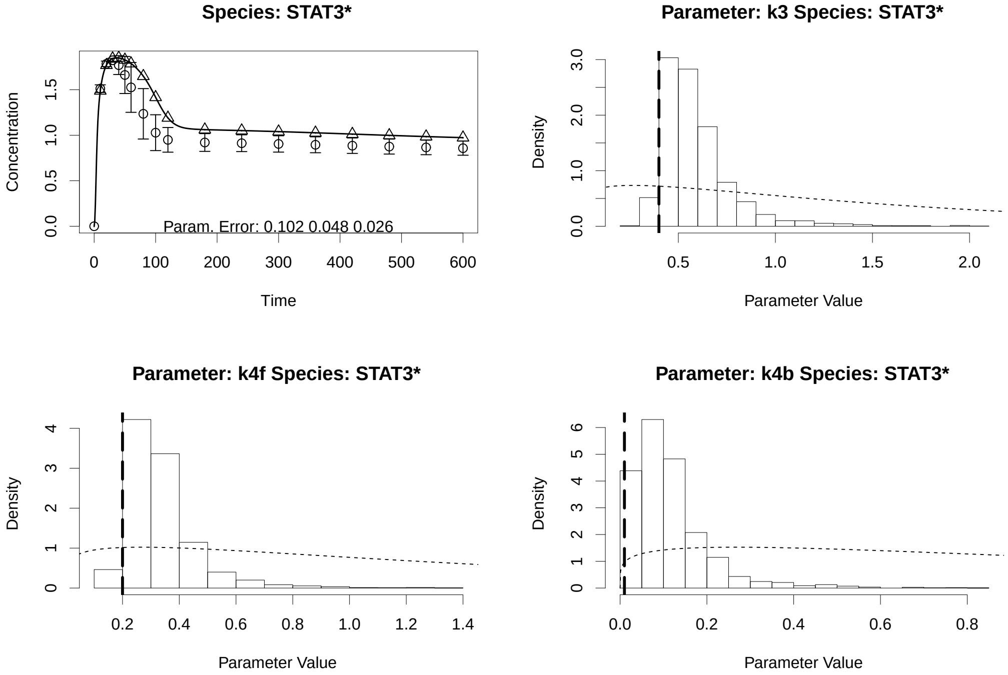


Figure D.5: STAT3* - Species 5, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

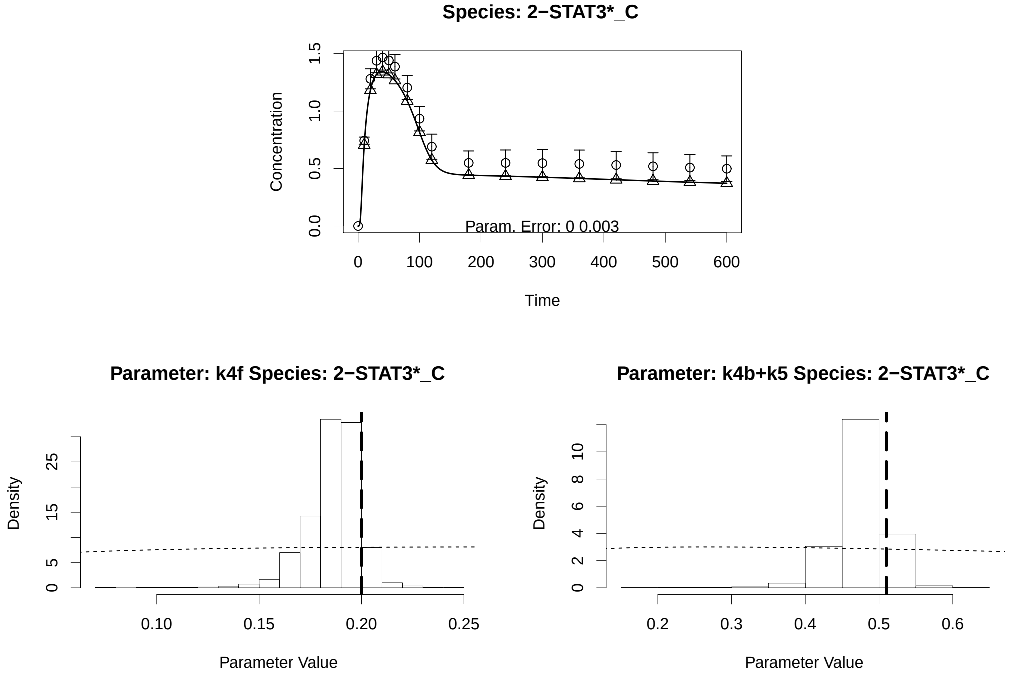


Figure D.6: 2-STAT3*_C - Species 6, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

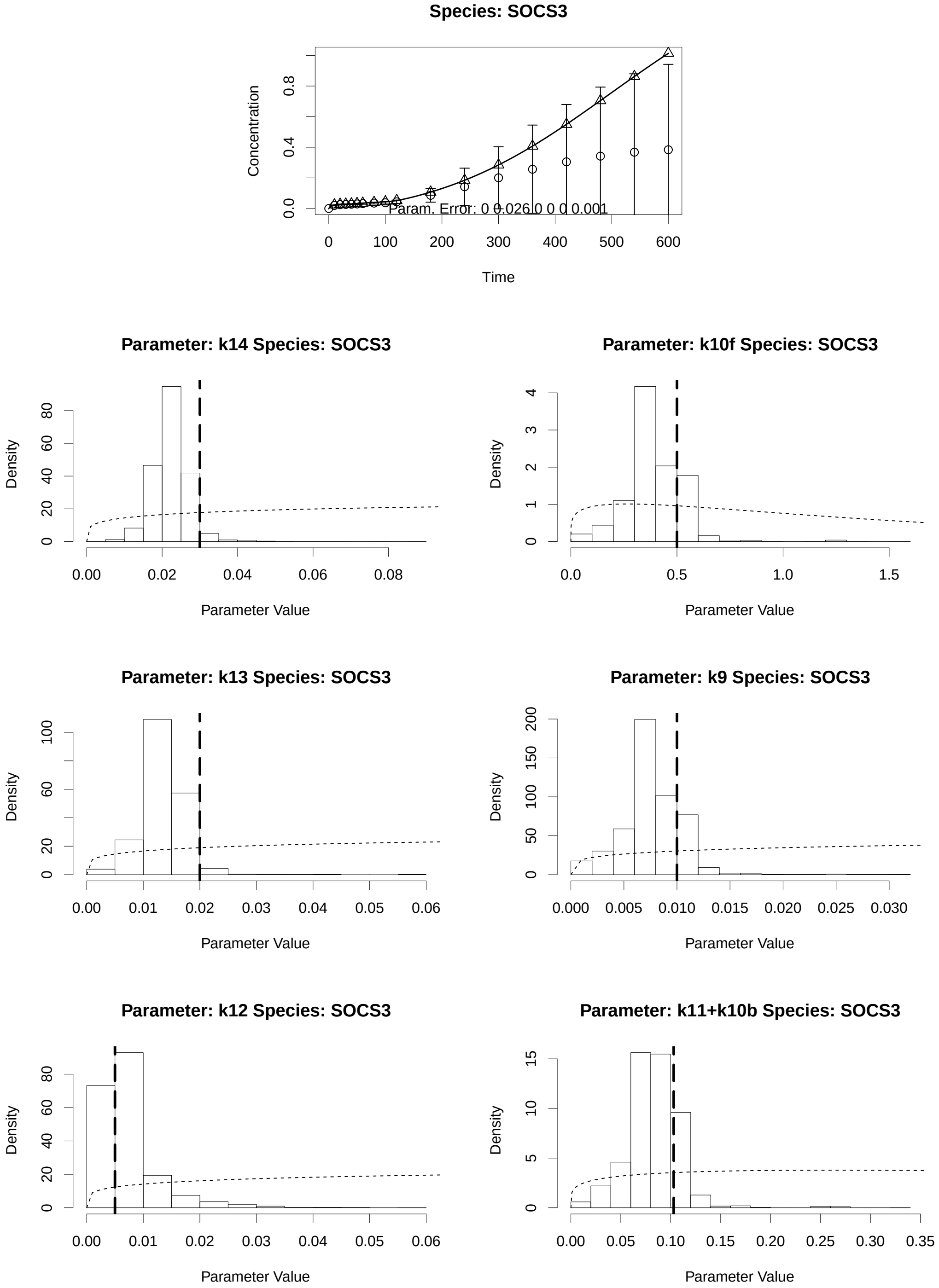


Figure D.7: SOCS3 - Species 7, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

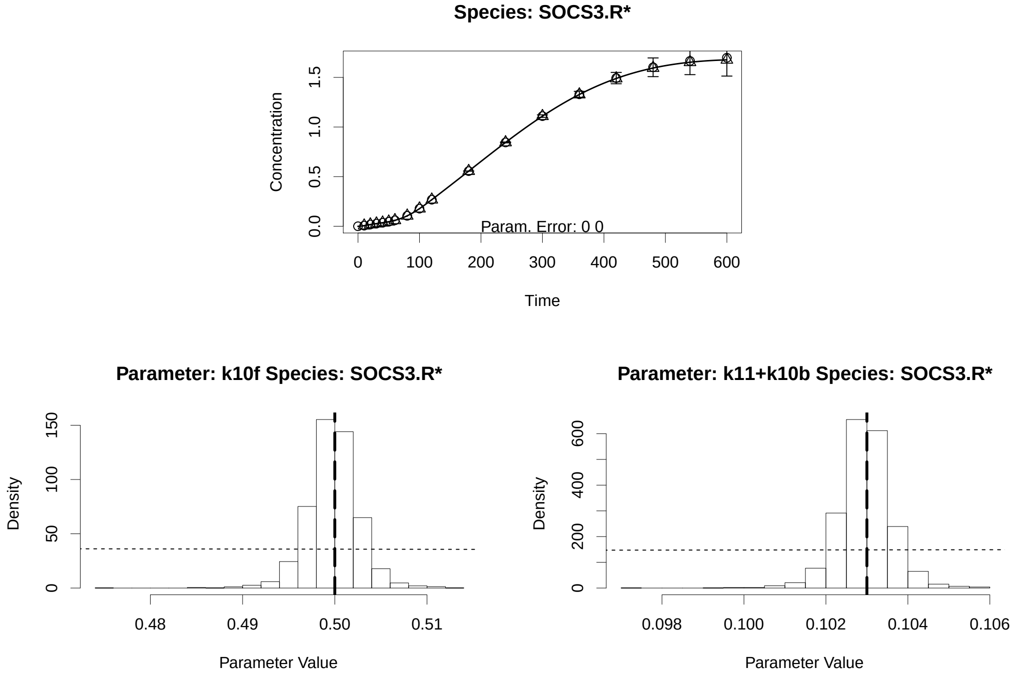


Figure D.8: SOCS3.R* - Species 8, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

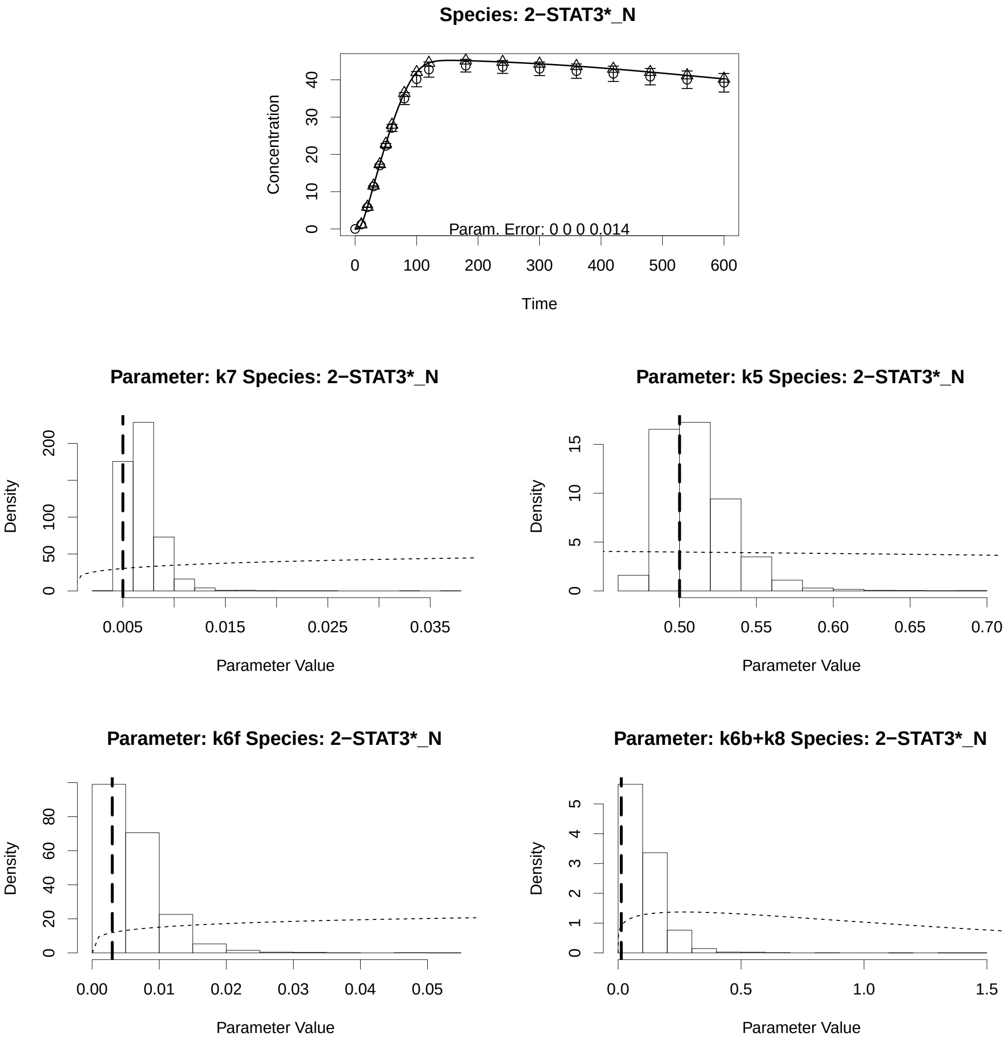


Figure D.9: 2-STAT3*_N - Species 9, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

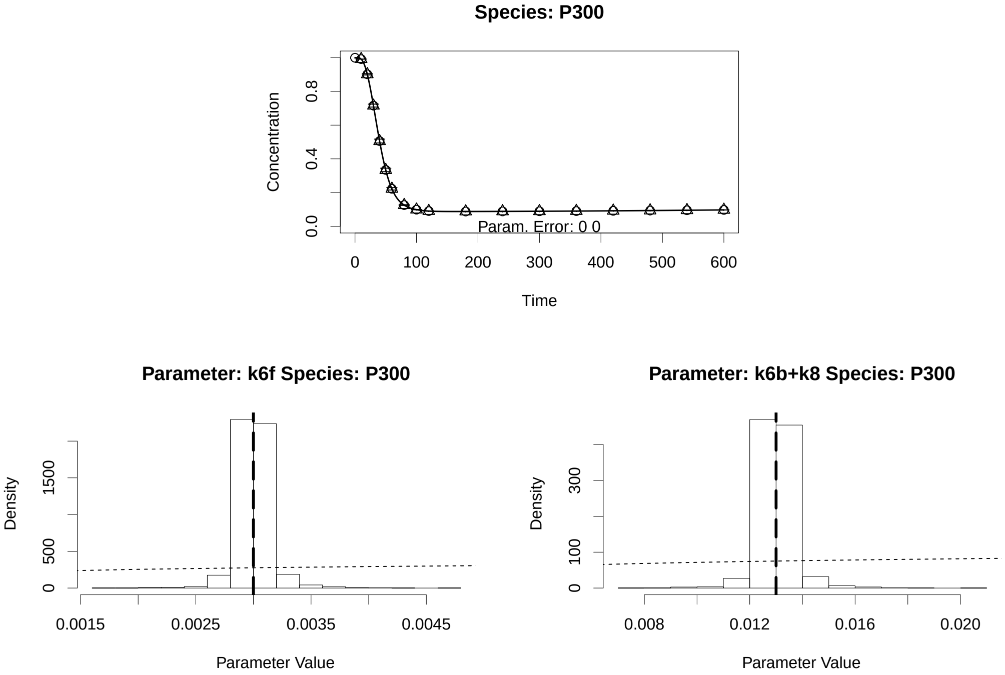


Figure D.10: P300 - Species 10, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

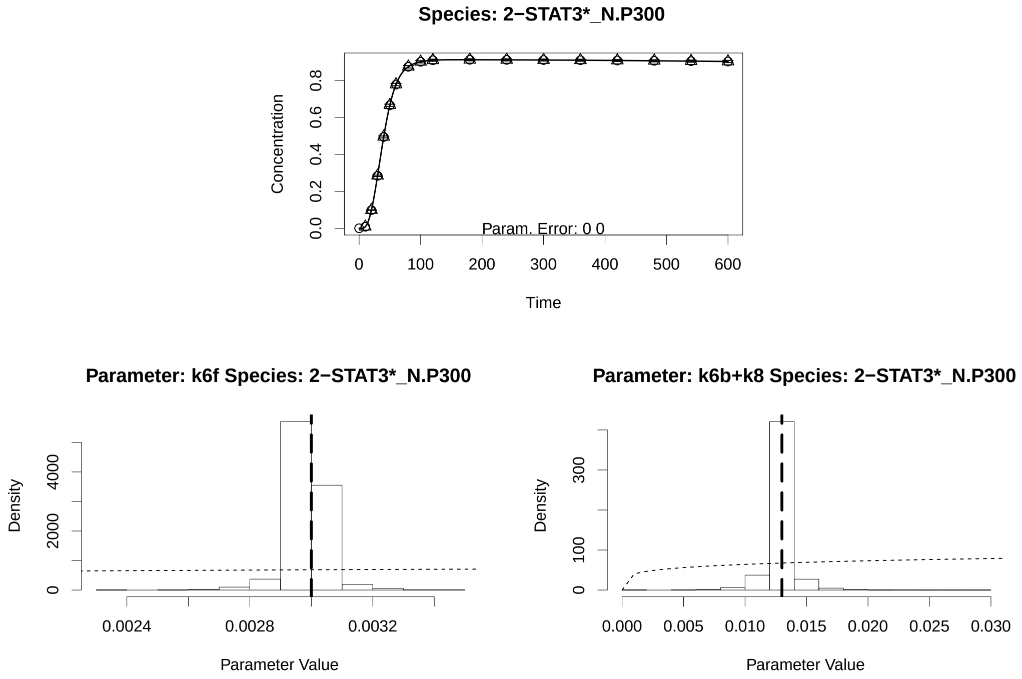


Figure D.11: 2-STAT3*_N.P300 - Species 11, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

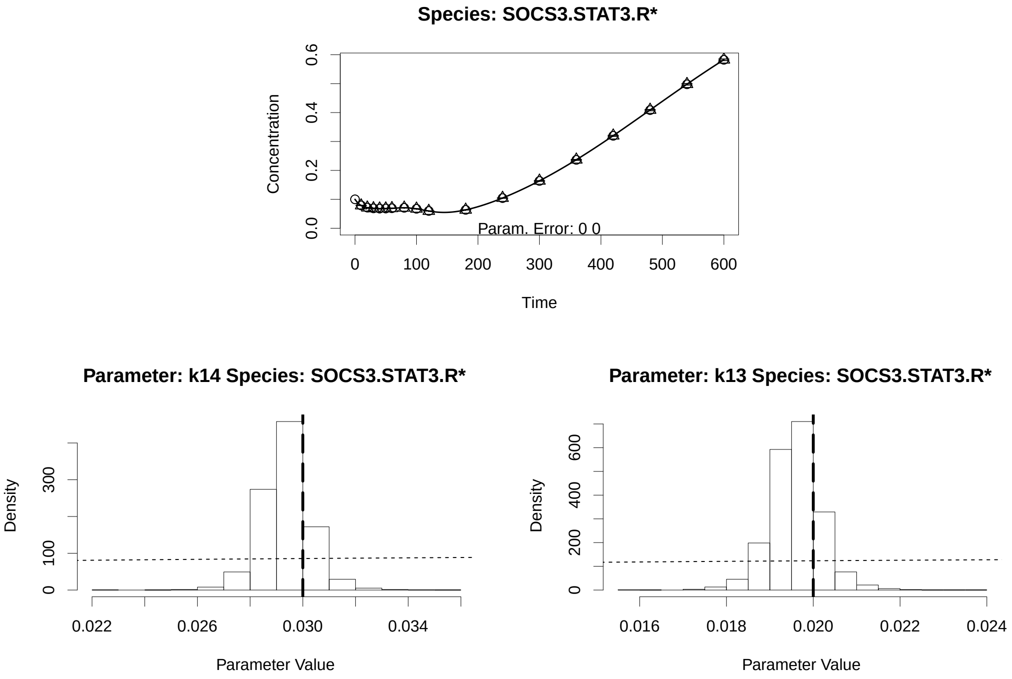


Figure D.12: SOCS3.STAT3.R* - Species 12, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

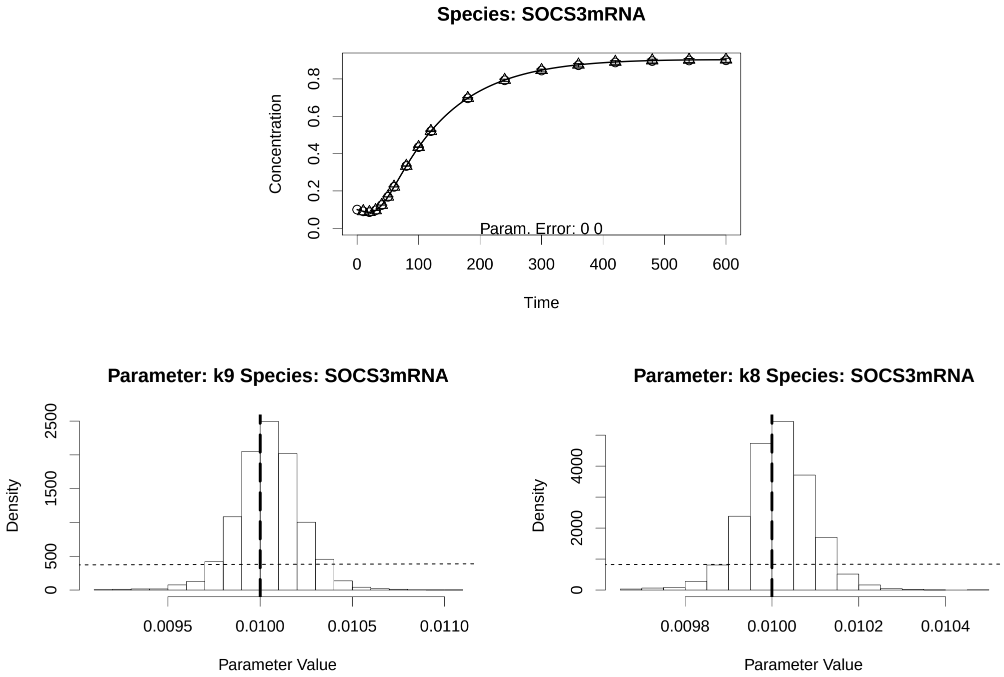


Figure D.13: SOCS3mRNA - Species 13, Noise 0. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

D.1.2 Noisy Observations

For data with Gaussian observation noise with std. dev. 0.05, Figures D.14-D.26 show the predictions obtained by the sampled parameters, as well as the latent variables sampled during the MCMC run, compared to the true noiseless profile for that species. They also show the distributions of the sampled parameters.

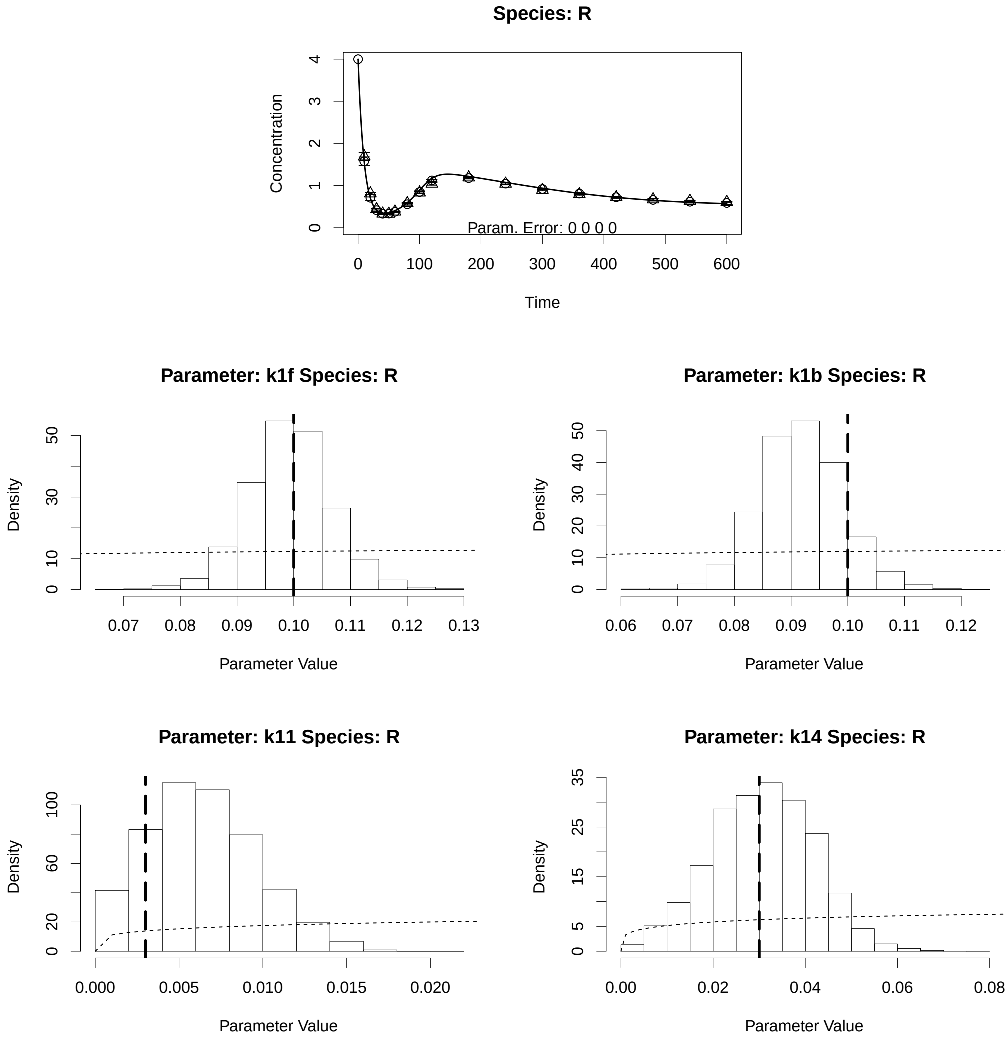


Figure D.14: R - Species 1, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

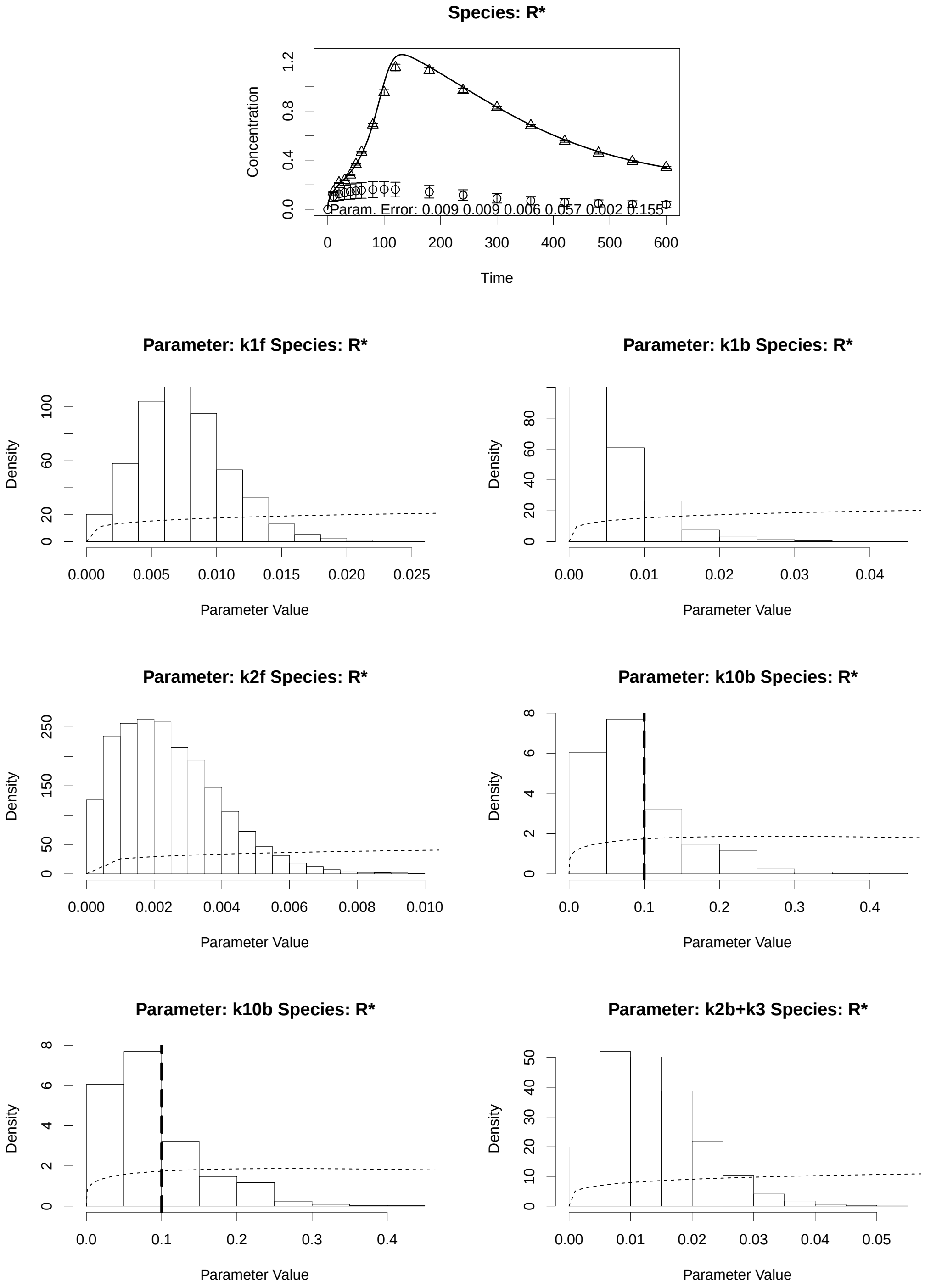


Figure D.15: R* - Species 2, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

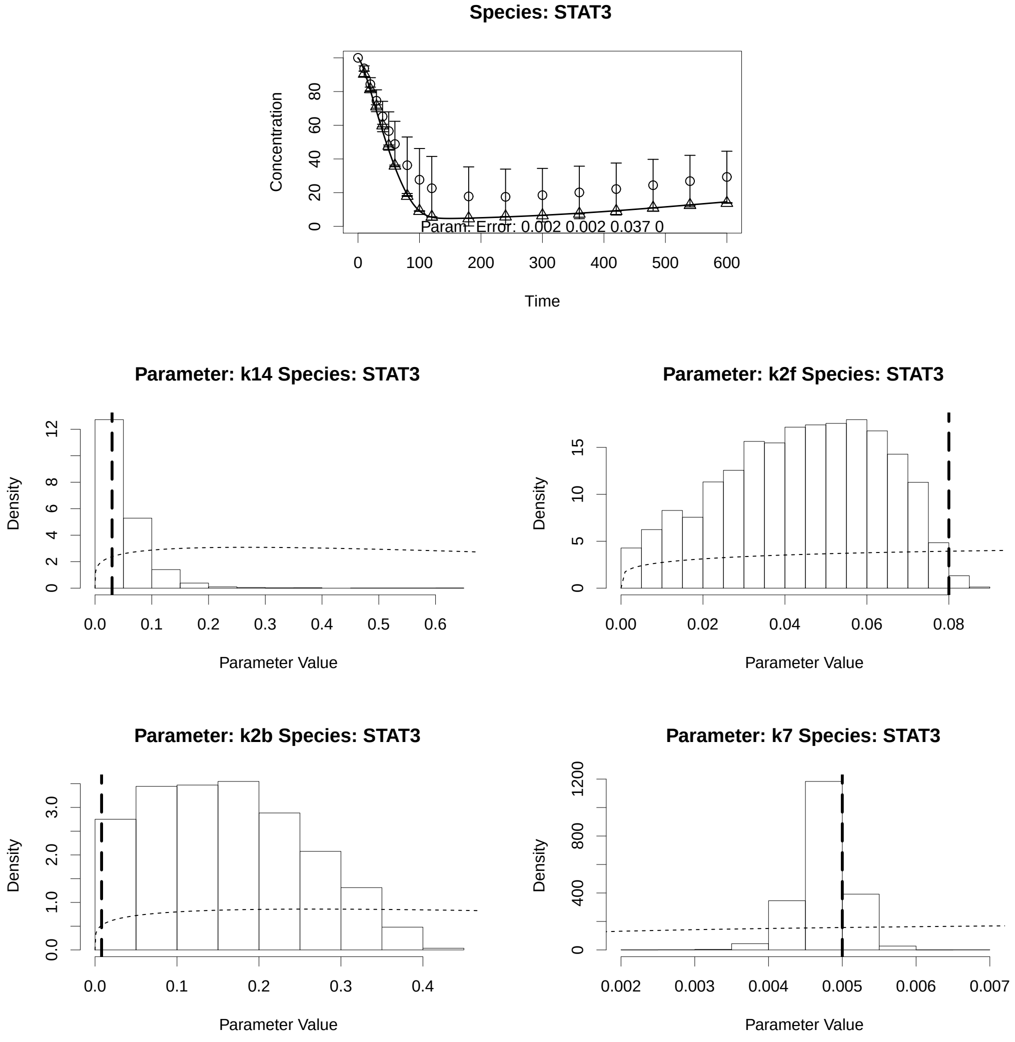


Figure D.16: STAT3 - Species 3, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

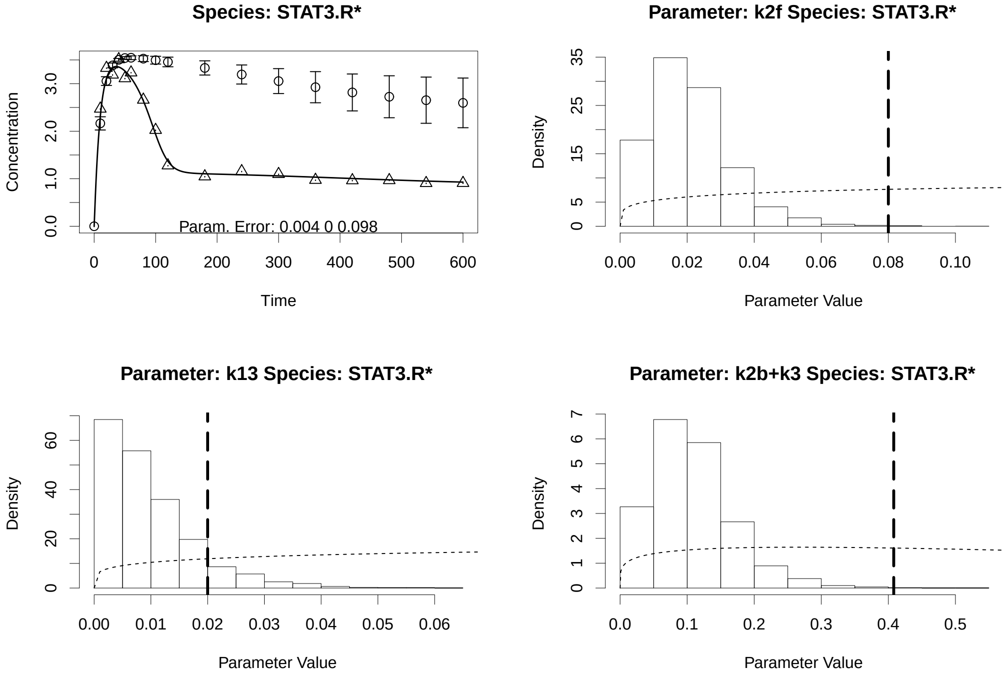


Figure D.17: STAT3.R* - Species 4, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

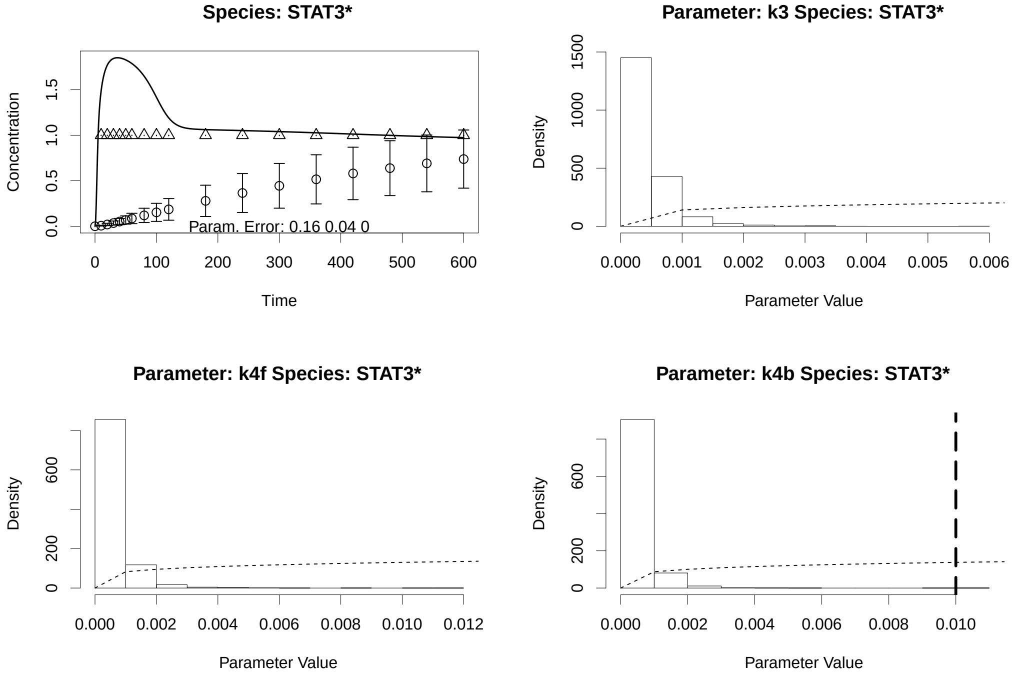


Figure D.18: STAT3* - Species 5, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

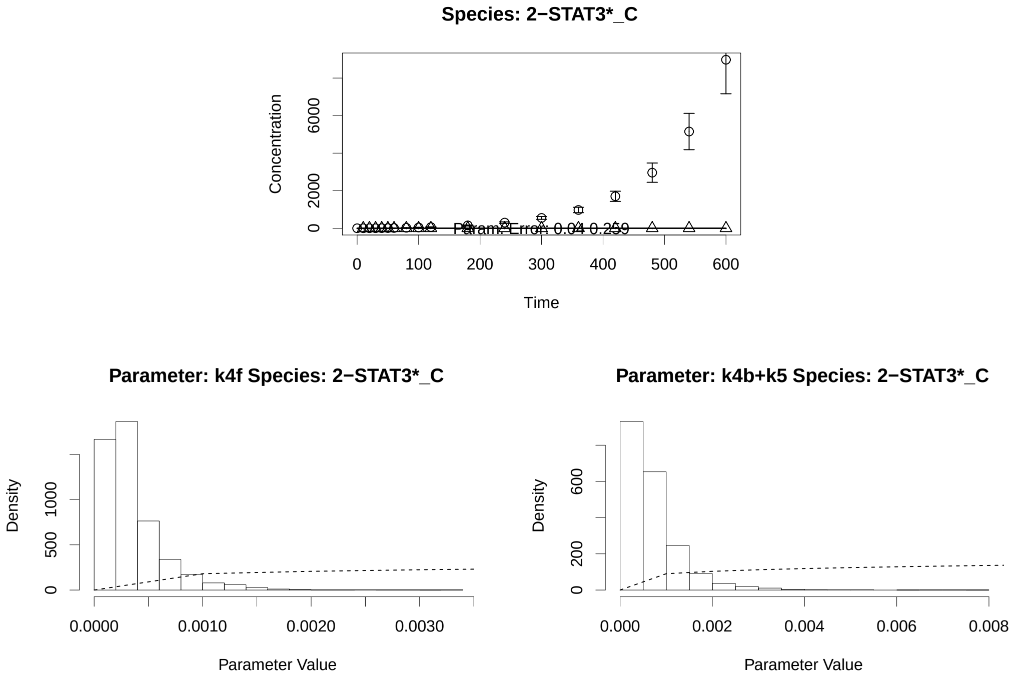


Figure D.19: 2-STAT3*_C - Species 6, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in a n idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

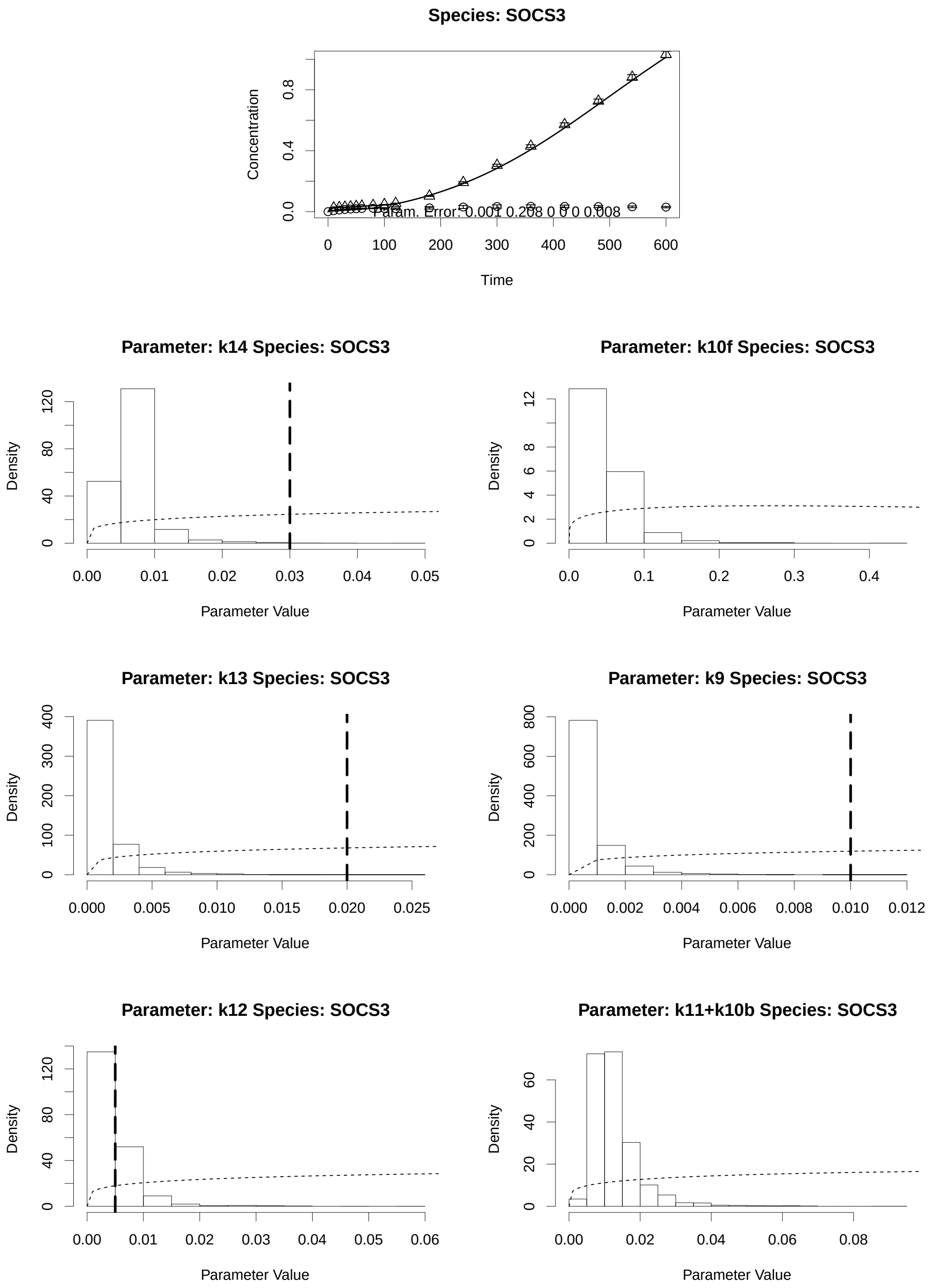


Figure D.20: SOCS3 - Species 7, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

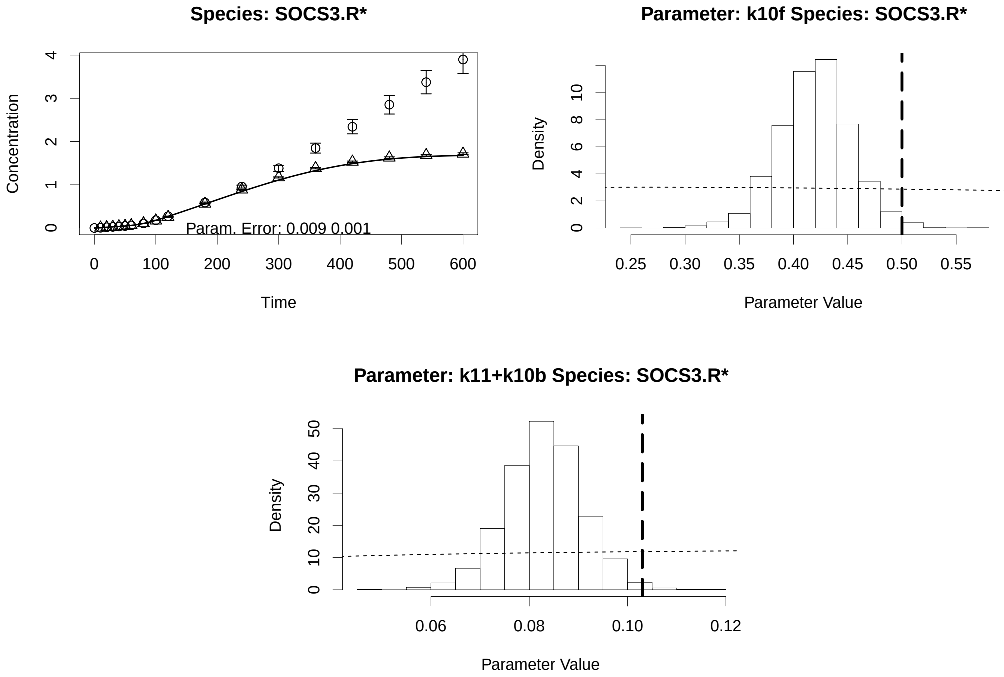


Figure D.21: SOCS3.R* - Species 8, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

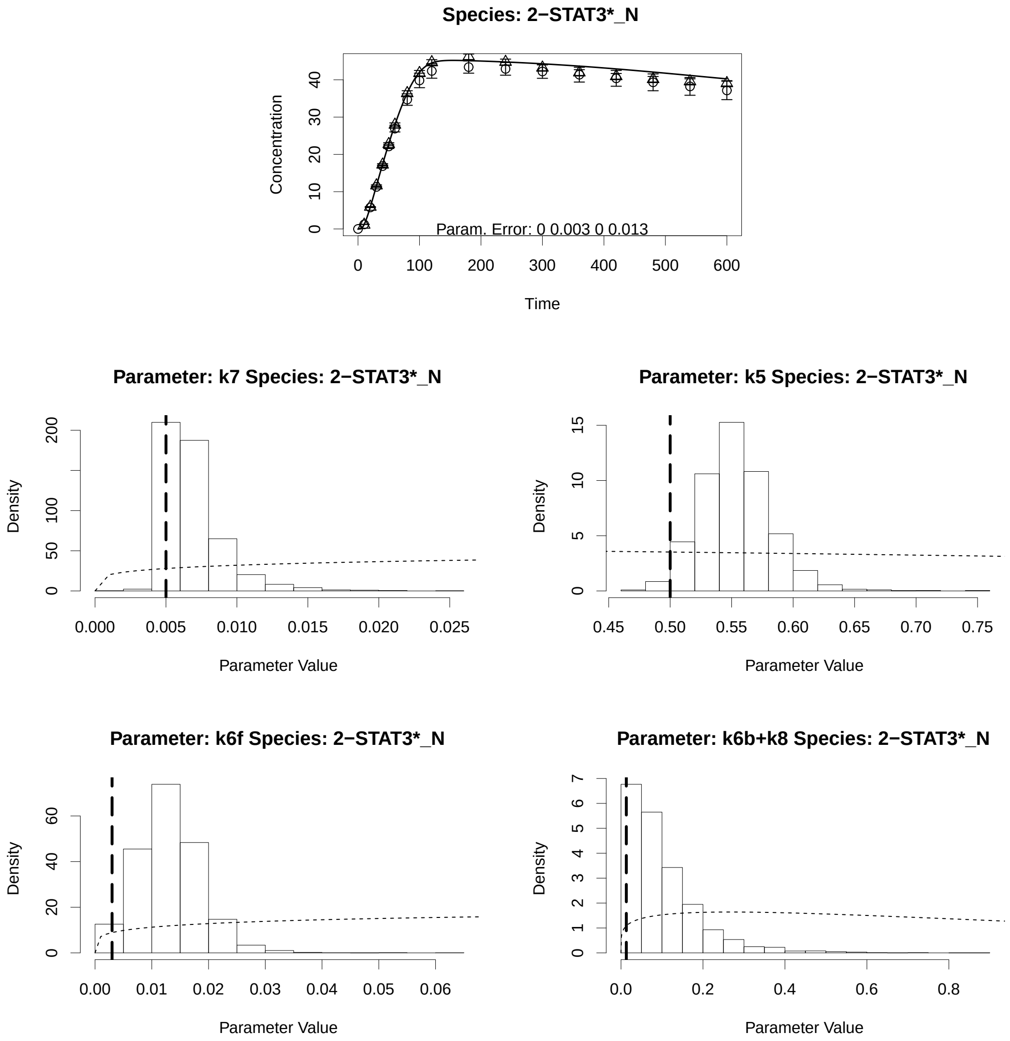


Figure D.22: 2-STAT3*_N - Species 9, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

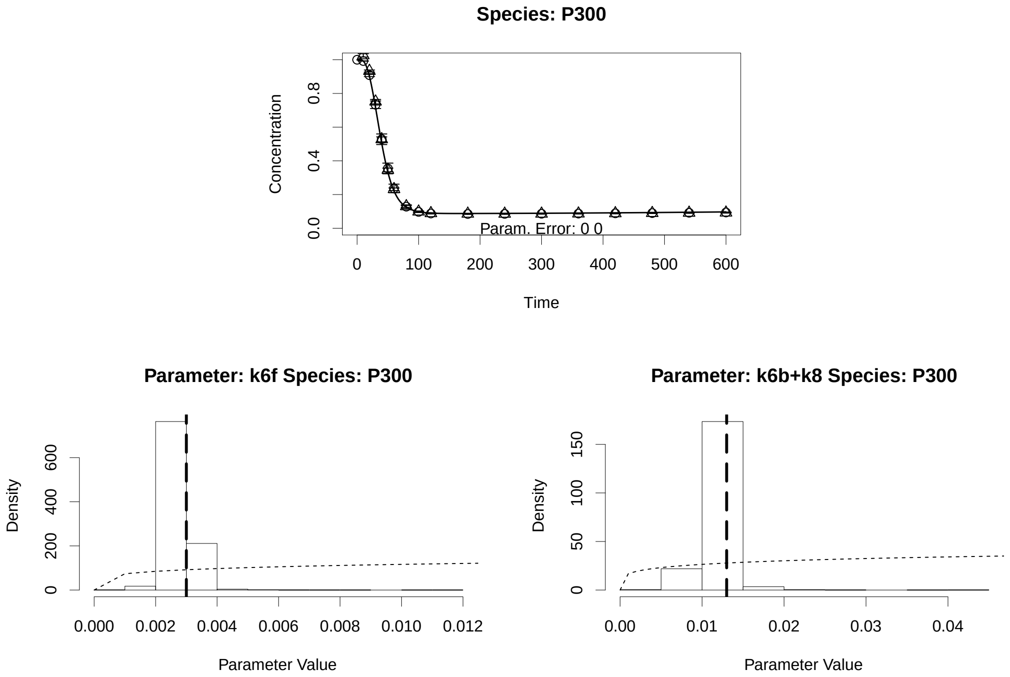


Figure D.23: P300 - Species 10, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

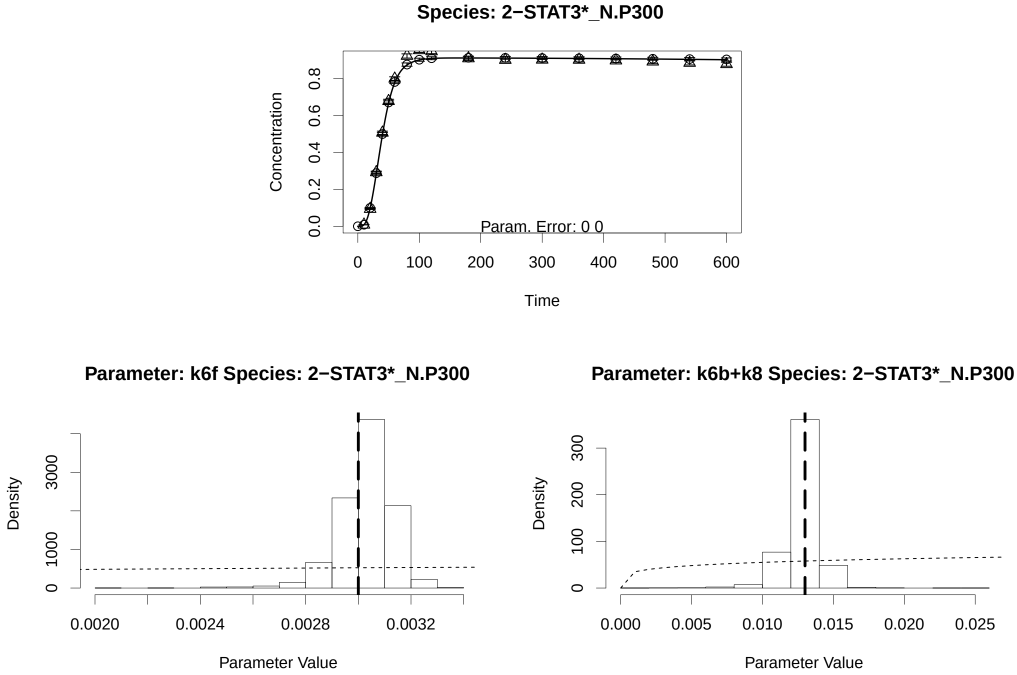


Figure D.24: 2-STAT3*_N.P300 - Species 11, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

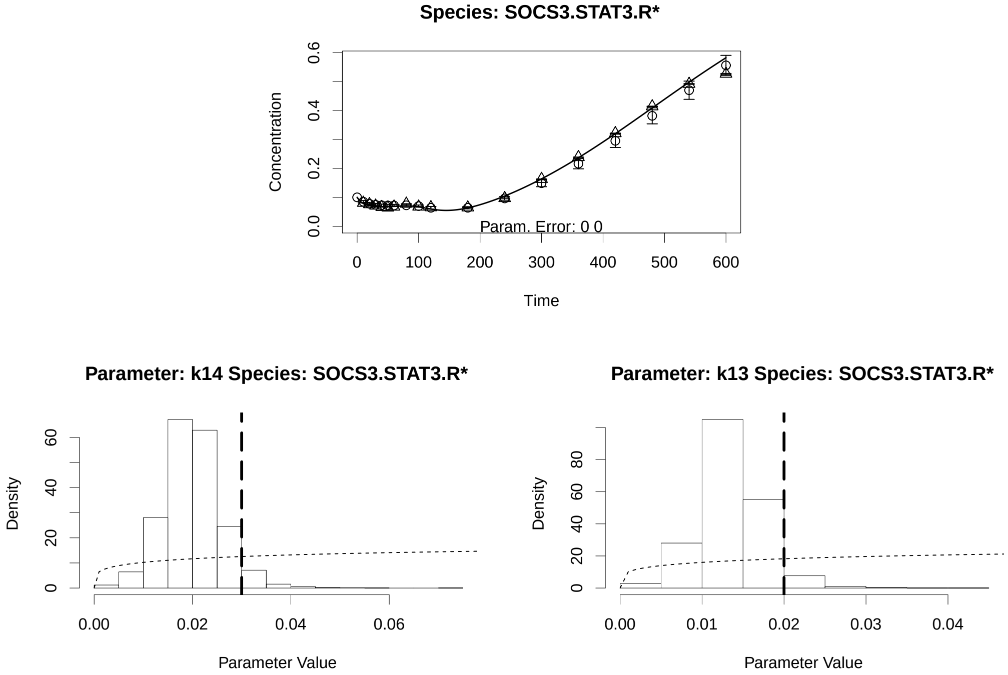


Figure D.25: SOCS3.STAT3.R* - Species 12, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

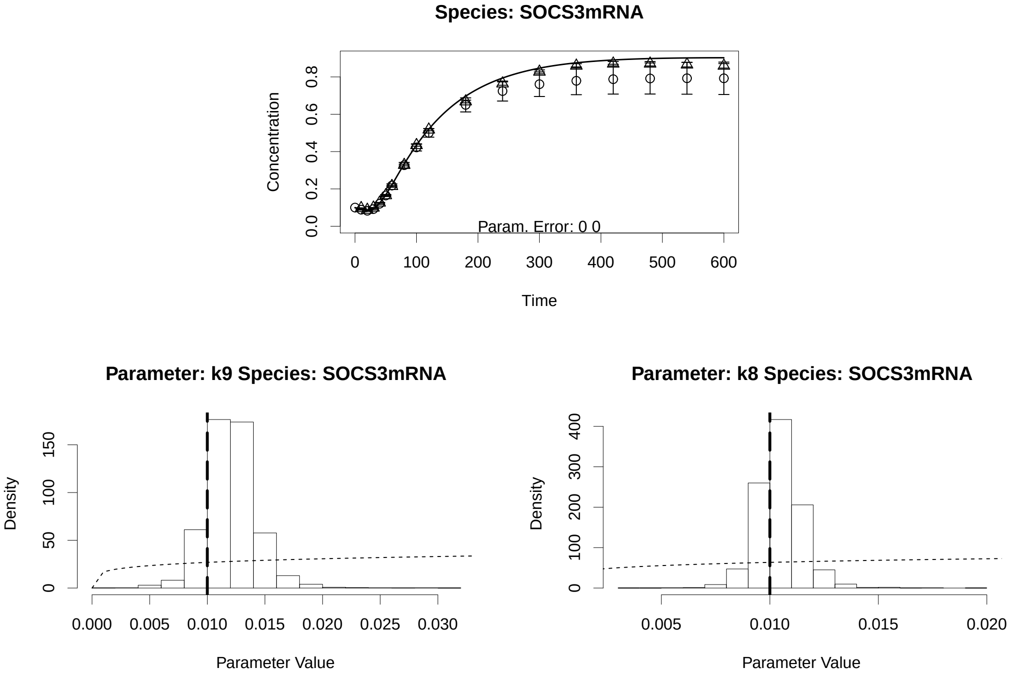


Figure D.26: SOCS3mRNA - Species 13, Noise 0.05. True (solid line), inferred (circles with error bars) and sampled latent (triangles with error bars) concentrations for each species in an idealised subsystems approach with perfect inputs to each subsystem. The error bars show one standard deviation. Histograms show the distribution of sampled parameters, the dashed line is the gamma prior on the parameters, the horizontal line indicates the true value of the parameter.

D.2 Perturbation Experiments

As described in Chapter 5, for each of the 22 parameters of the JAK/STAT system, I investigated the effect of perturbing the parameters by setting the parameter to a value on the interval $[0,1]$ while keeping the other parameters fixed at the true values, and calculating four different likelihoods: The global profile log likelihood, the global gradient log likelihood, the local profile log likelihood and the local gradient log likelihood. See Section 5.5.3.3 for a description of these quantities. Below I present the full comparison of the four likelihoods under perturbation for each species/subsystem and each parameter.

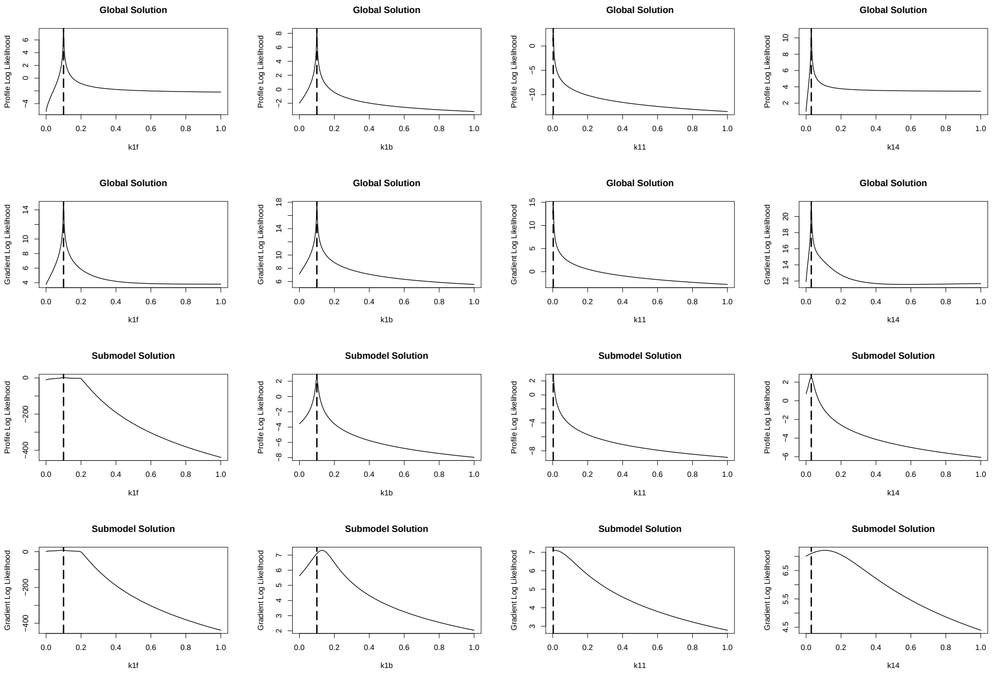


Figure D.27: Profile and Gradient Likelihood Comparison of Species 1. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

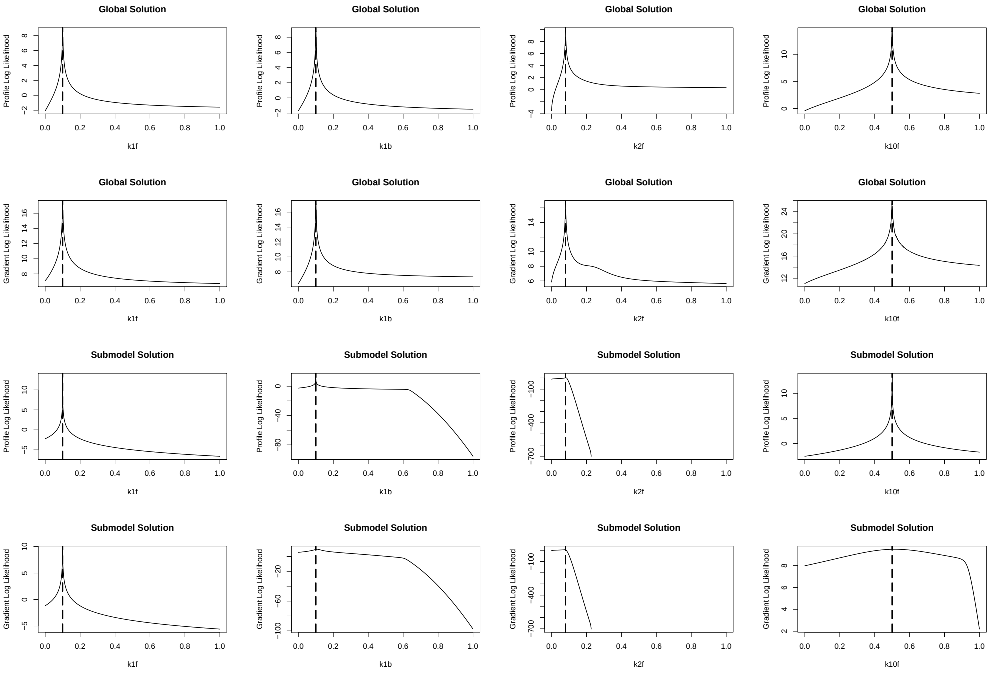


Figure D.28: Profile and Gradient Likelihood Comparison of Species 2 (a). From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

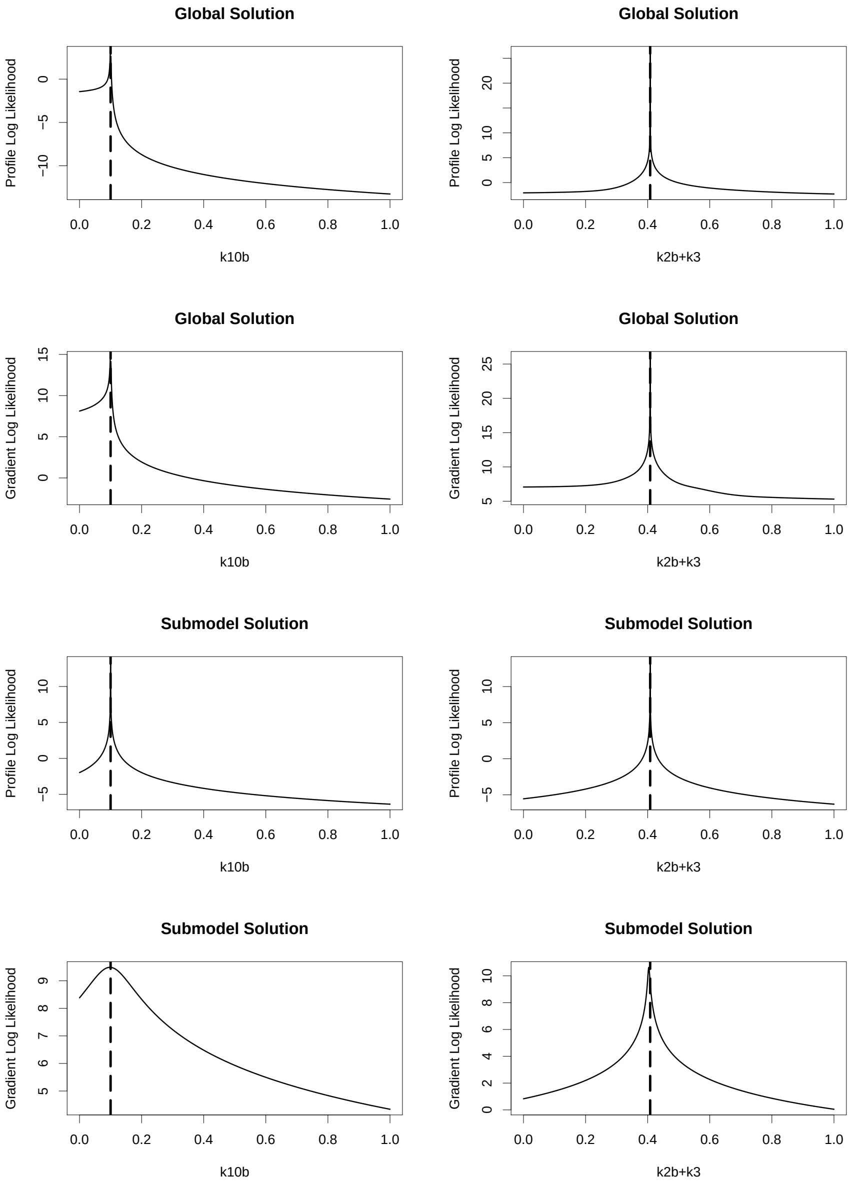


Figure D.29: Profile and Gradient Likelihood Comparison of Species 2 (b). From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

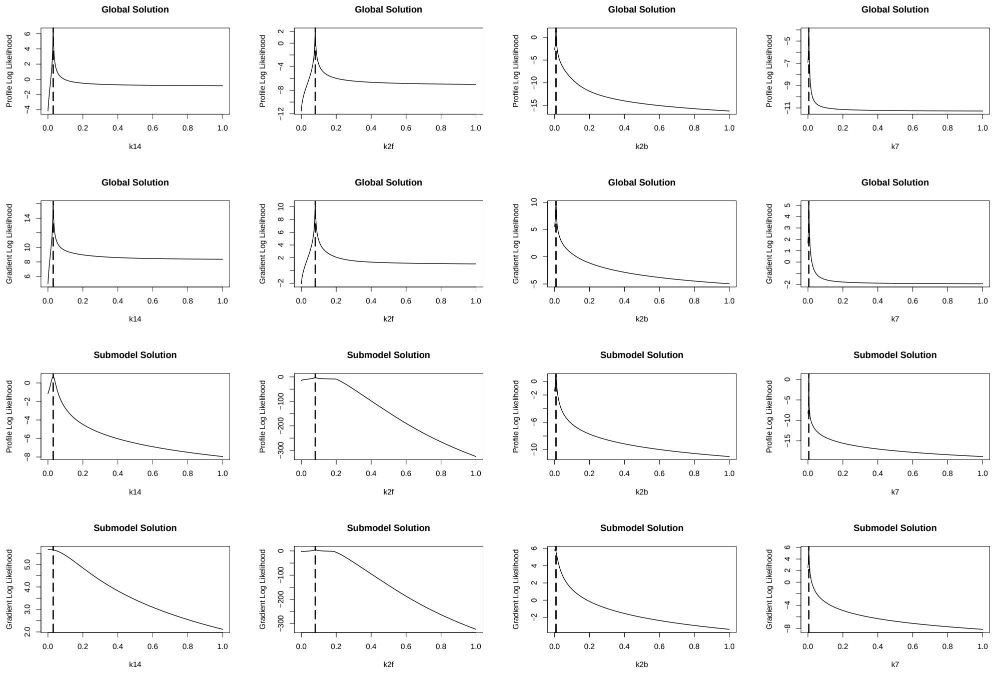


Figure D.30: Profile and Gradient Likelihood Comparison of Species 3. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

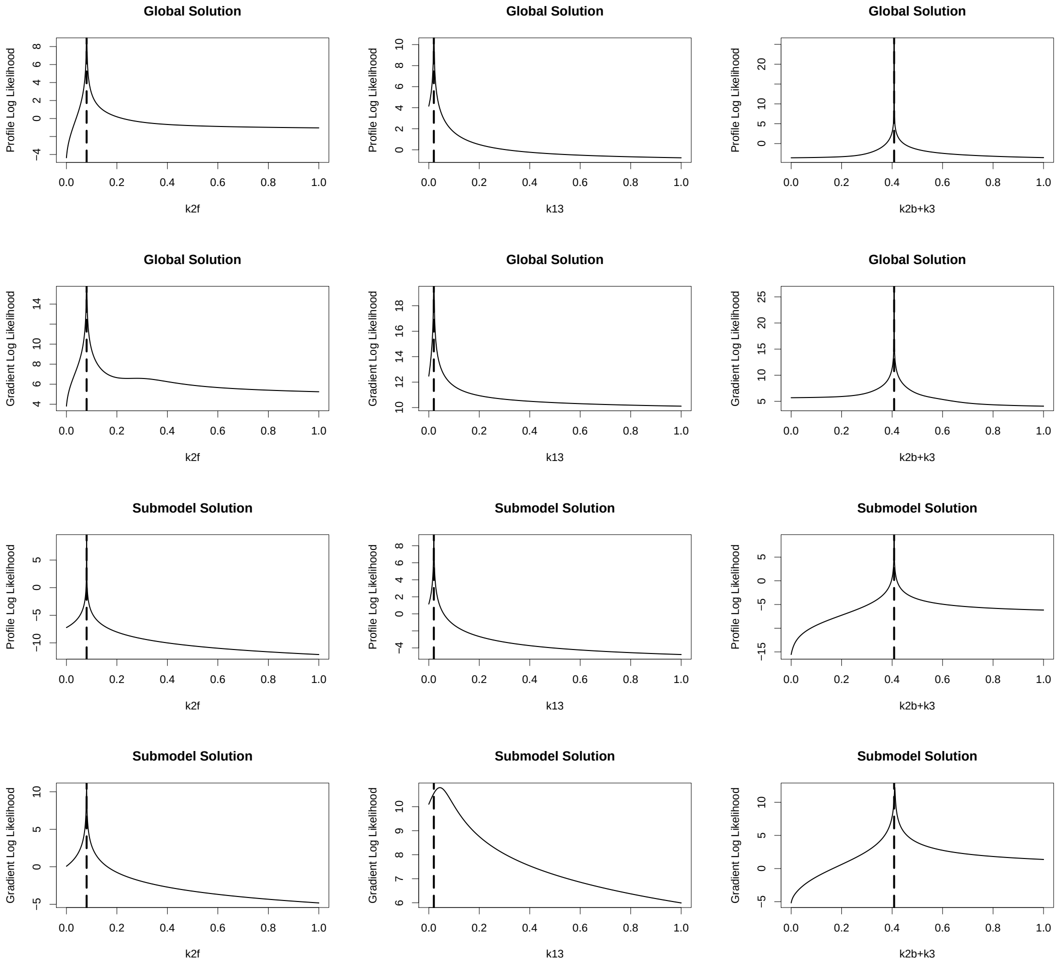


Figure D.31: Profile and Gradient Likelihood Comparison of Species 4. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

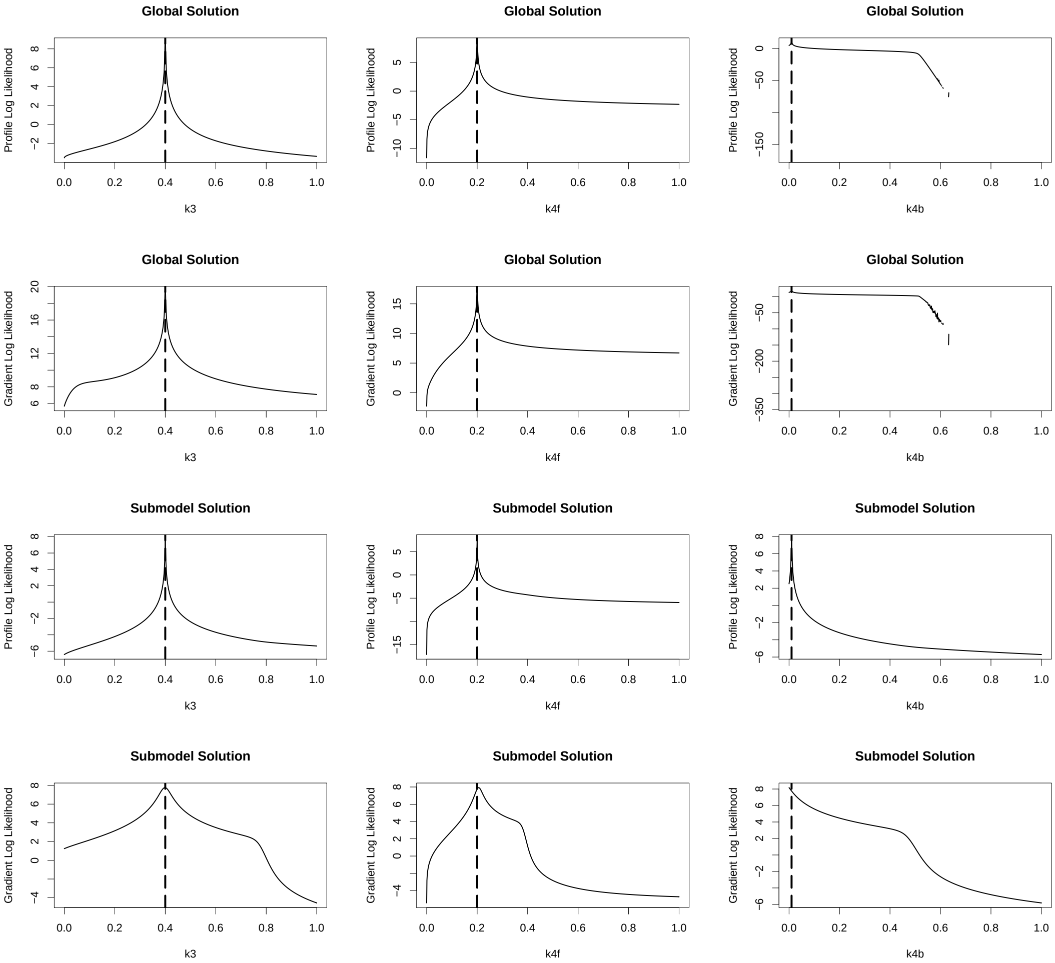


Figure D.32: Profile and Gradient Likelihood Comparison of Species 5. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

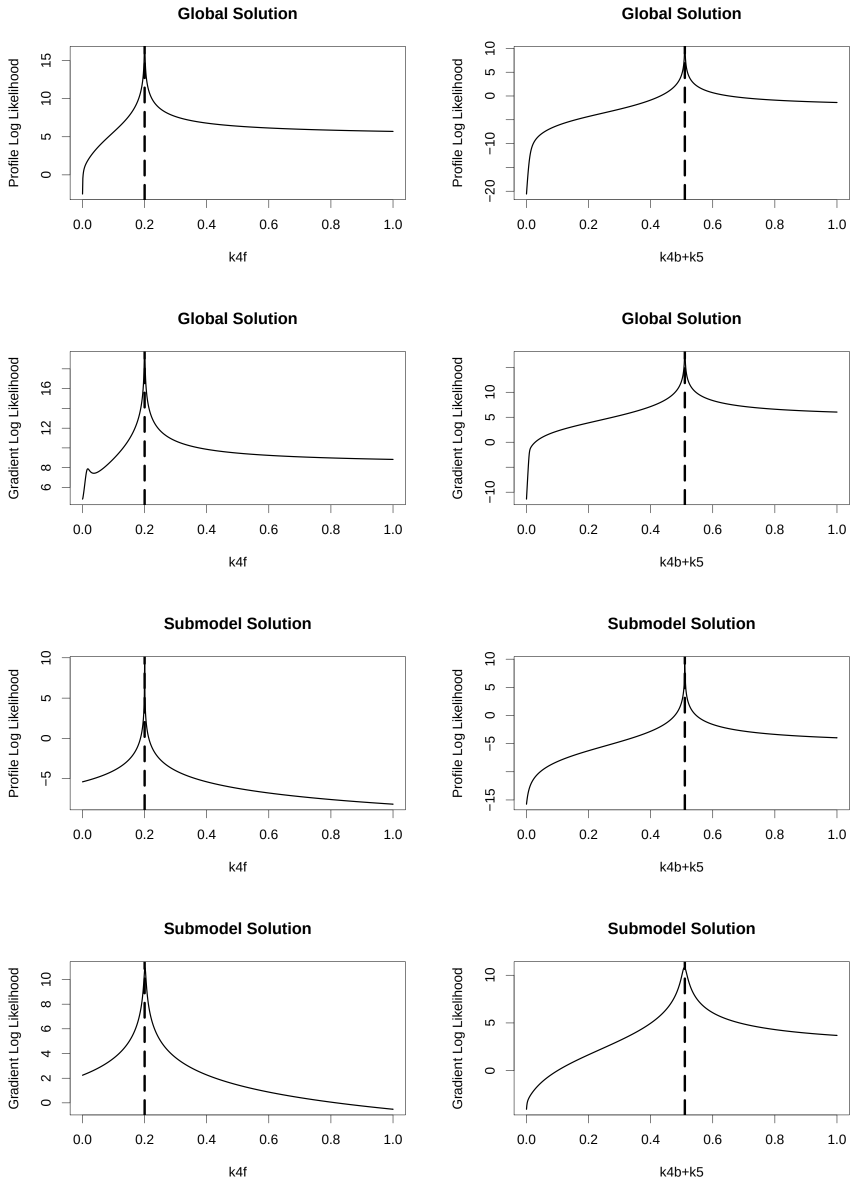


Figure D.33: Profile and Gradient Likelihood Comparison of Species 6. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

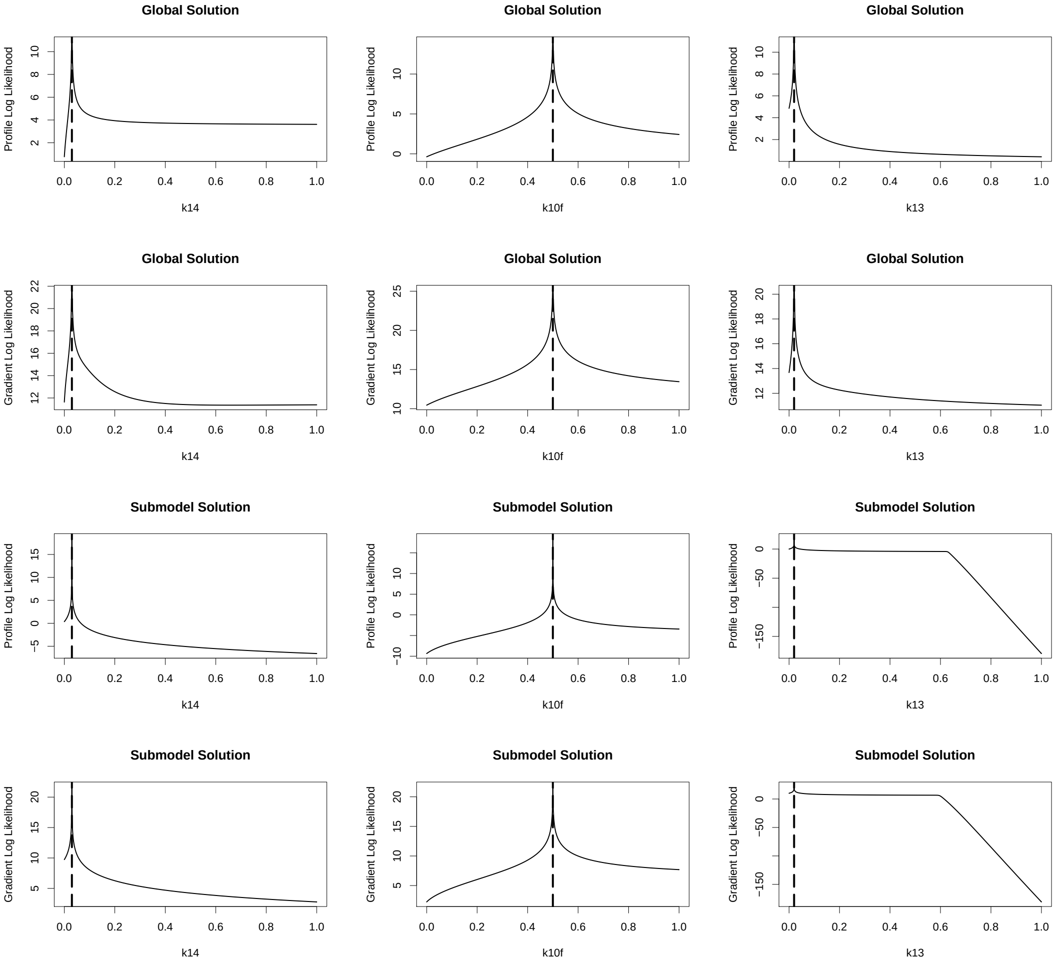


Figure D.34: Profile and Gradient Likelihood Comparison of Species 7 (a). From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

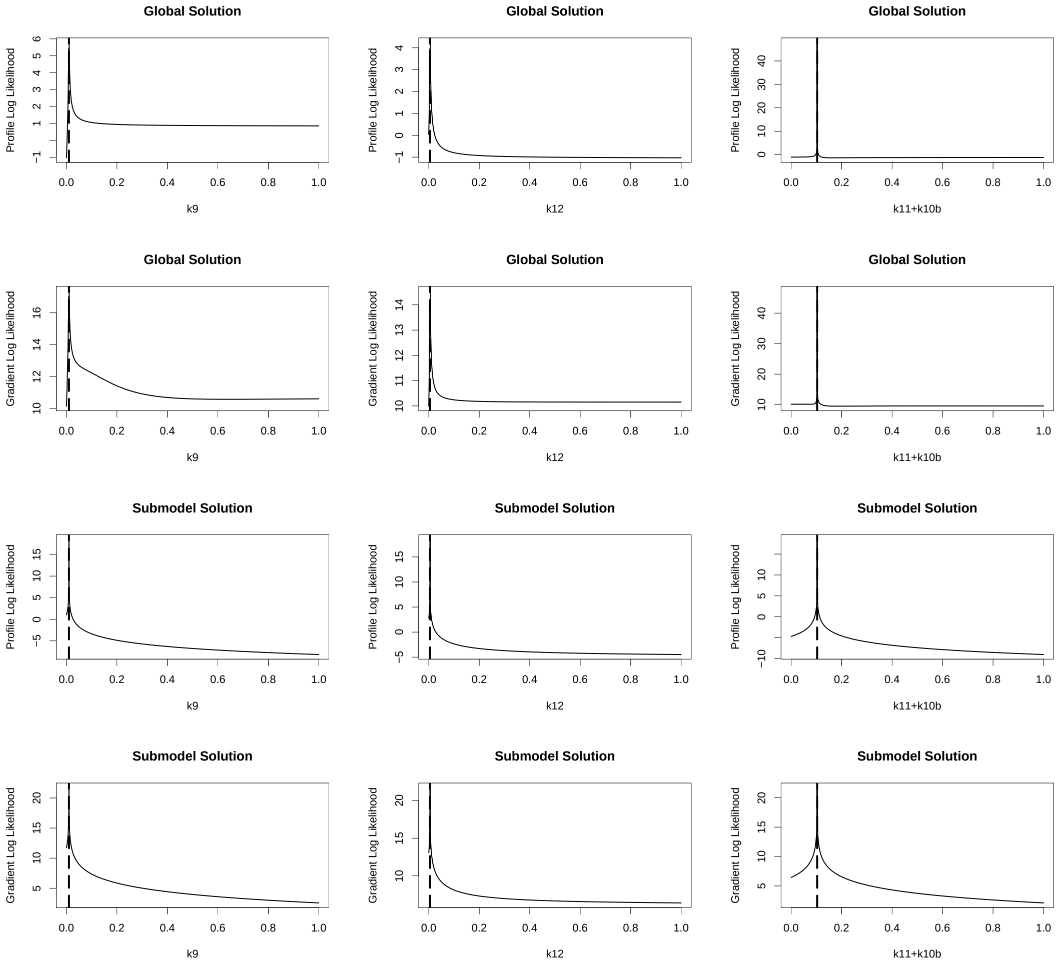


Figure D.35: Profile and Gradient Likelihood Comparison of Species 7 (b). From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

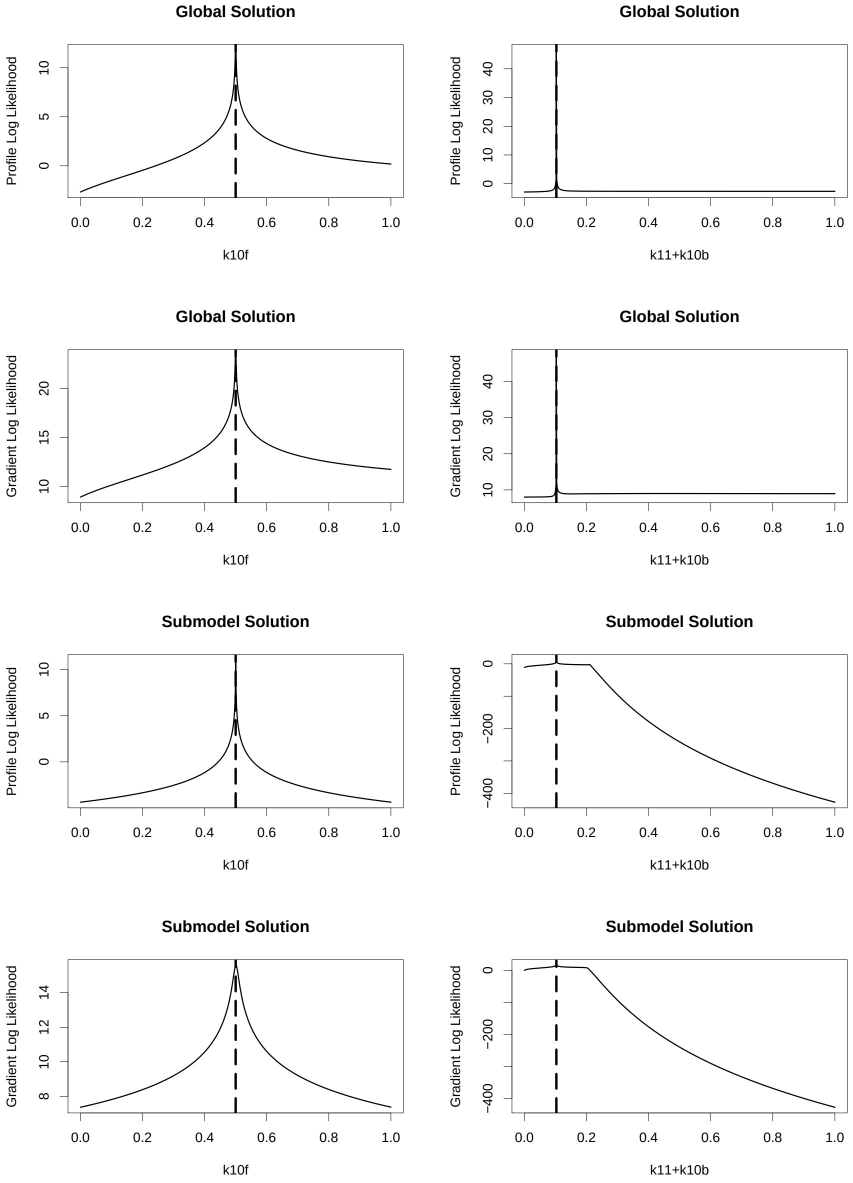


Figure D.36: Profile and Gradient Likelihood Comparison of Species 8. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

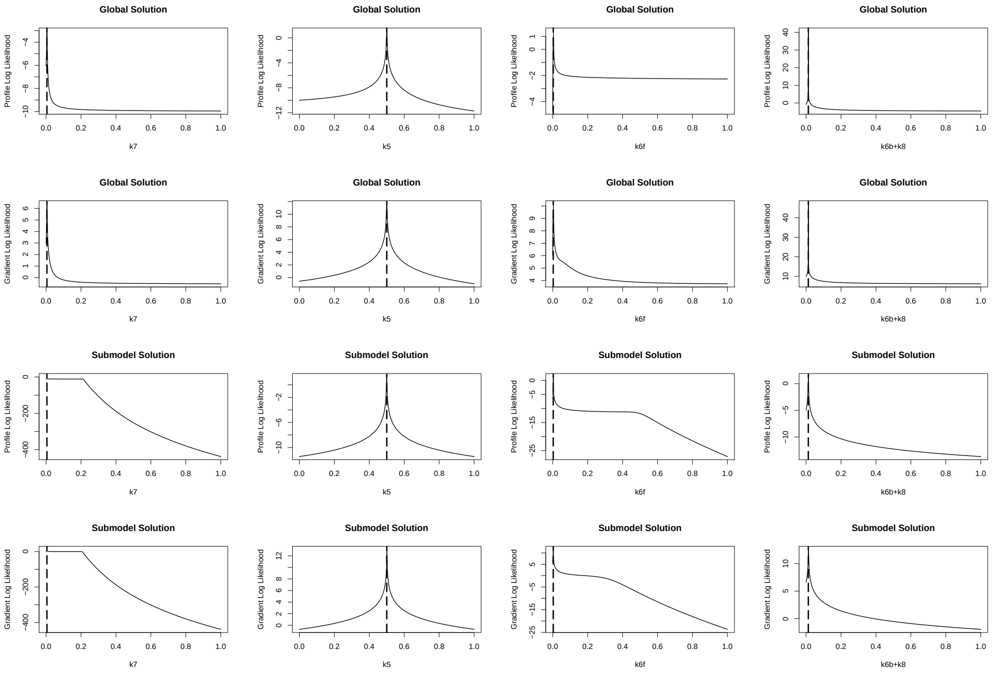


Figure D.37: Profile and Gradient Likelihood Comparison of Species 9. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

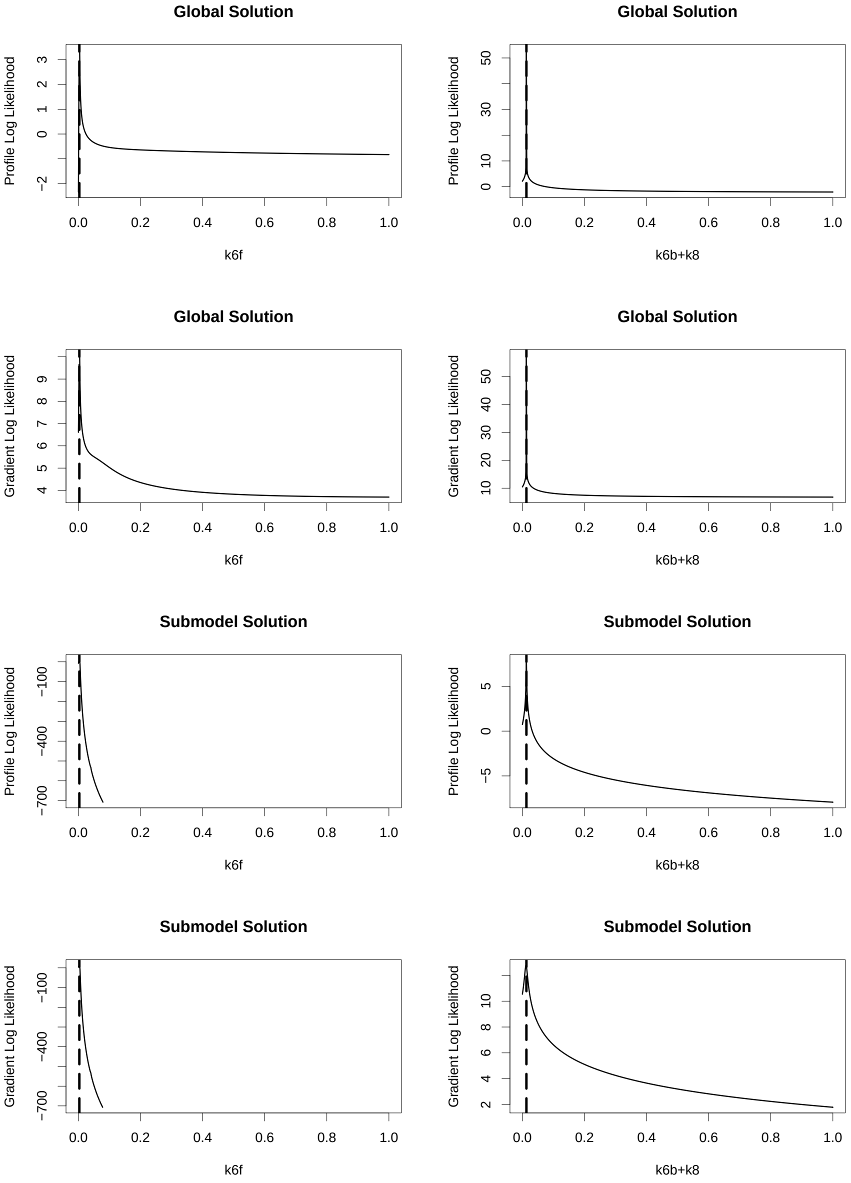


Figure D.38: Profile and Gradient Likelihood Comparison of Species 10. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

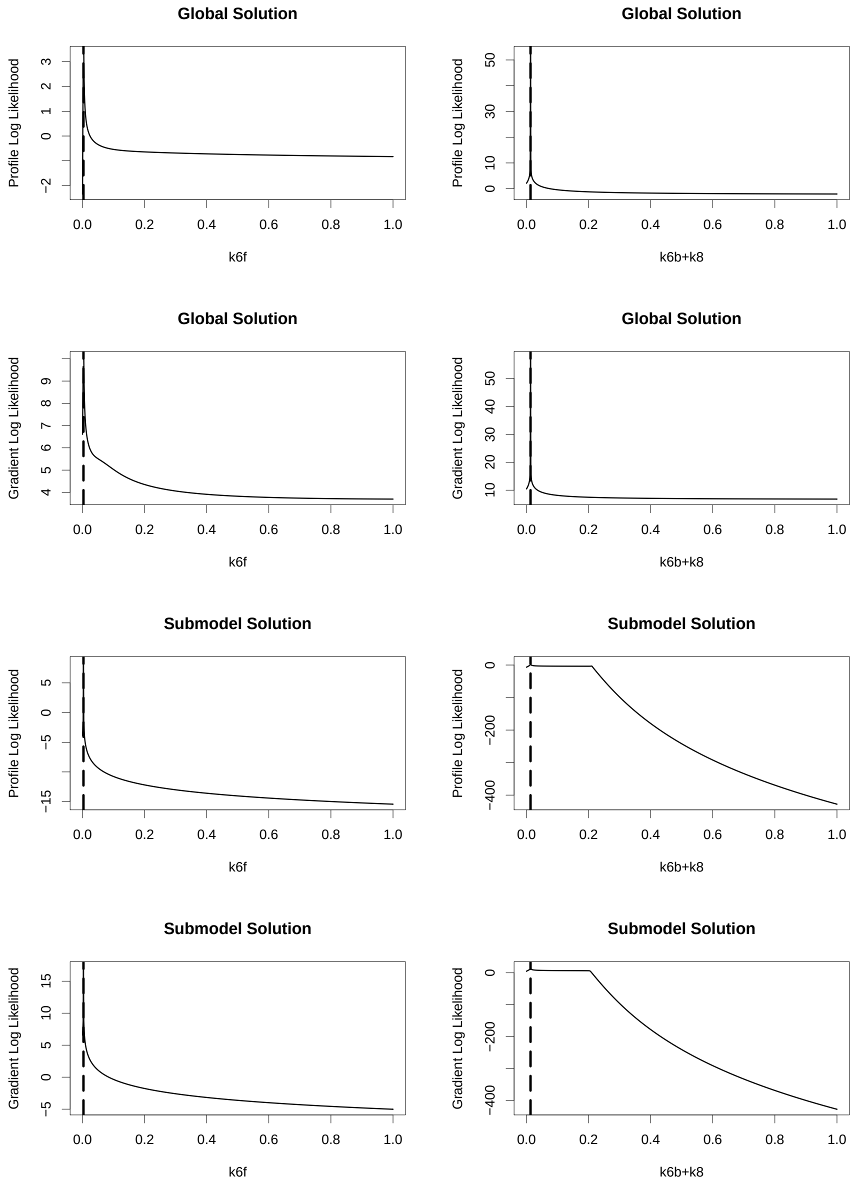


Figure D.39: Profile and Gradient Likelihood Comparison of Species 11. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

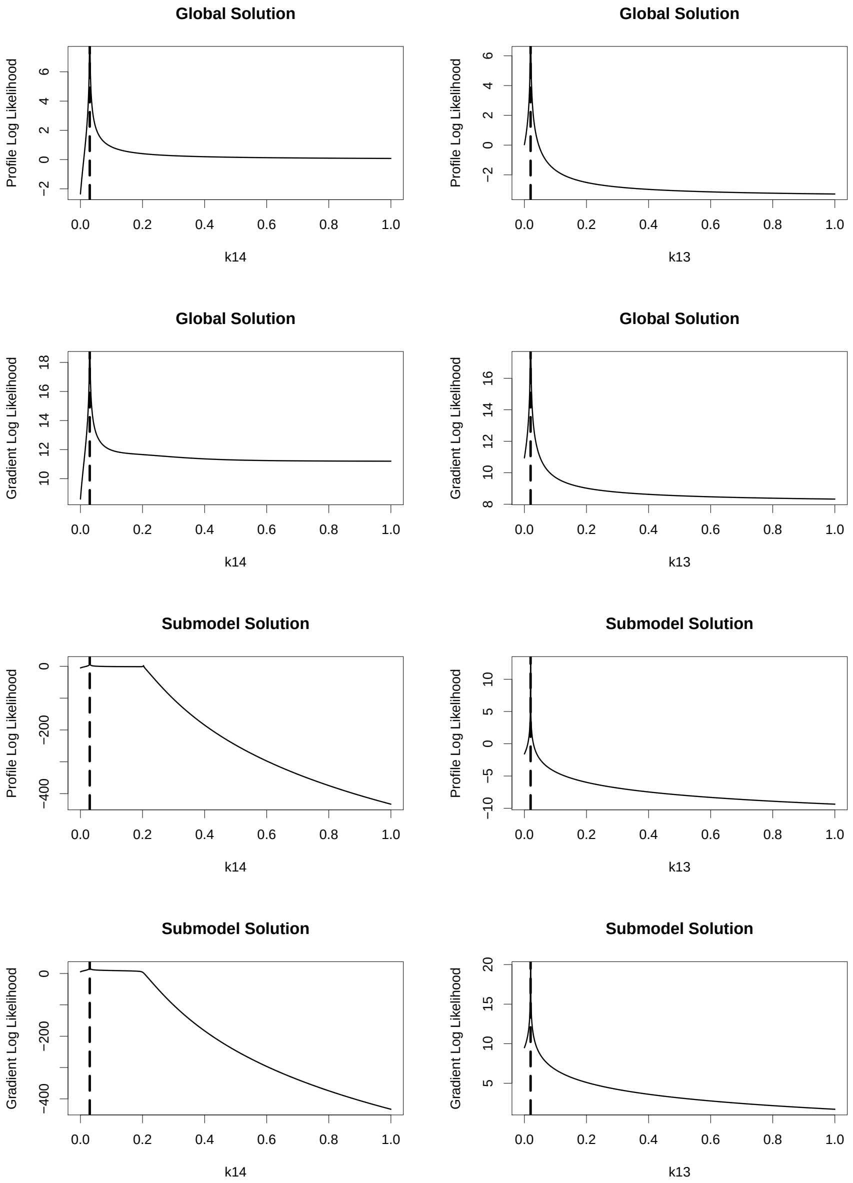


Figure D.40: Profile and Gradient Likelihood Comparison of Species 12. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

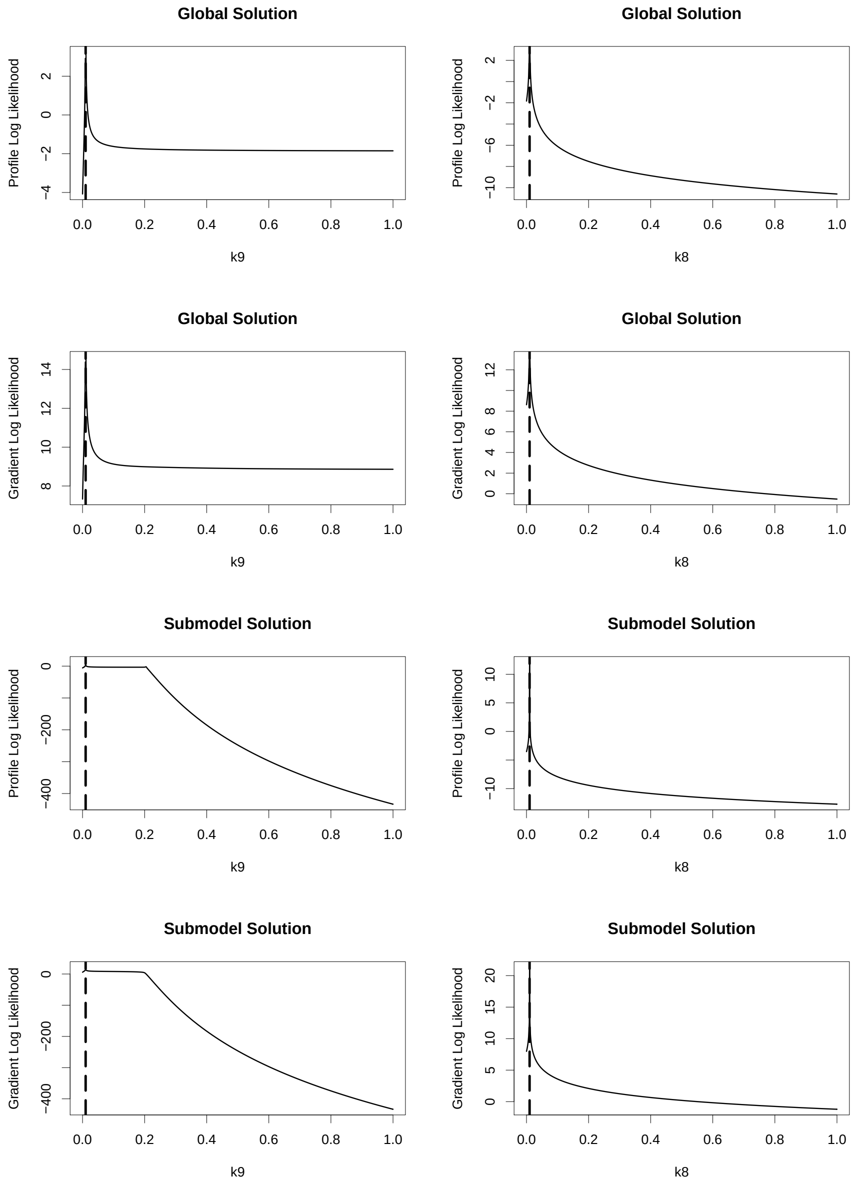


Figure D.41: Profile and Gradient Likelihood Comparison of Species 13. From top to bottom: Profile Likelihood Global, Gradient Likelihood Global, Profile Likelihood Local, Gradient Likelihood Local. The dashed vertical line denotes the true value of the parameter.

Appendix E

Nested Effects Models for Gene Regulatory Networks

E.1 Introduction

This project was motivated by work in Liu et al. (2008) on Pectobacterium atrosepticum (Pba), a plant pathogen which attacks the potato plant by synthesising plant cell wall degrading enzymes. The bacteria coordinate their attack on a plant via a regulatory mechanism called quorum sensing. In quorum sensing, the bacteria communicate via signalling molecules that allow them to determine when a critical population density has been reached, at which point they attack.

Liu et al. found that Pba uses quorum sensing to make an attack on the plant's defences while simultaneously attacking the cell wall. They showed that genes within the quorum sensing regulon also regulate virulence. Following up on this study, Kuang Lin at BioSS worked on applying various methods from computational statistics and machine learning to reconstruct possible gene regulatory networks from gene expression profiles of Pba. His work has been published in Lin et al. (2010); here I will only describe the results that were contributed by my own sub-project.

The data consisted of mRNA expression profiles for 9 knock-out mutations. The knock-outs were obtained via transposon mutagenesis, whereby transposons are inserted into the chromosome using a plasmid vector. This insertion disrupts function of an existent gene, creating the knock-out mutation. In this case the 8 genes expM, hor, hrpL, expI, expR, aepA, virR and virS were knocked out. The ninth mutation was a double knockout of virR and expM. Both wild type and mutants were grown under the same conditions and were then used to inoculate sterilised potato tubers. 12 hours after inoculation, the mRNA levels were measured using microarrays, and relative gene expression levels with respect to the wild type were obtained. These measurements were further preprocessed to remove outliers and achieve normalisation.

While most of Lin et al. (2010) deals with determining the relationship between clusters of genes measured in the microarray experiment, it is also interesting to look at the relationship between the genes that were knocked out, which were known (or at least suspected) to be regulators of the other genes. The question here is whether these knocked-out genes are regulators of each other as well, and what their interactions are. In other words, we want to construct a gene regulation network of the knocked-out genes. As it happens, nested effect models are the perfect method for determining this.

E.2 Methodology

General Framework Nested effect models (NEMs) are a model for determining the relationship between genes. Rather than looking at the expression levels of these regulating genes (called S-genes for signalling genes), NEMs look at the effect that knocking out each of these genes has on the expression levels of the genes that they regulate (called E-genes for effect reporting genes).

This means that we need two sets of parameters for a NEM: A network hypothesis Φ, that describes the relationship between the S-genes, and a model Θ for the regulation of the E-genes, where $\theta_i = j$ if E-gene i is regulated by S-gene j. We assume that an E-gene can only be regulated by one S-gene and use model averaging to account for all possibilities.

Using Bayes' theorem, the score for a network hypothesis given data $\mathbf{D}$ is:

$$P(\Phi|\mathbf{D}) = \frac{P(\mathbf{D}|\Phi)P(\Phi)}{P(\mathbf{D})} \tag{E.1}$$

If we assume that the observations of each E-gene, the parameters θ_i and the knockout experiments are independent, then the likelihood $P(\mathbf{D}|\Phi)$ for a dataset consisting of m E-genes and n S-genes decomposes as:

$$P(\mathbf{D}|\Phi) = \prod_{i=1}^{m}\sum_{j=1}^{n}\prod_{k=1}^{n} P(D_{ik}|\Phi,\theta_i = j)P(\theta_i = j|\Phi) \tag{E.2}$$

where $P(D_{ik}|\Phi,\theta_i = j)$ is the likelihood of the effect observed at E-gene i when knocking out S-gene k and $P(\theta_i = j|\Phi)$ is the prior probability of E-gene i being regulated by S-gene j. More details can be found in Markowetz et al. (2005) and Fröhlich et al. (2008).

Modelling Effects In order to find the likelihood of observing an effect at E-gene i when knocking out S-gene k, Markowetz et al. (2005, 2007) first used a discretisation scheme based on thresholding to transform the continuous expression values of the E-genes into binary indicators. Then they calculated the likelihood based on the expected false positive and false negative rates. This has the potential to lose information, and also requires both positive and negative controls to estimate the error rates, which may not always be available.

Fröhlich et al. (2008) developed an alternative method which uses p-values that correspond to the likelihood of an E-gene i being differentially expressed when S-gene k is knocked out. They obtain the raw p-value using limma (Smyth et al., 2004), and fit a three-component Beta-uniform mixture (BUM) model to those values. The BUM model consists of a uniform distribution (reflecting the null hypothesis) and two Beta distributions such that:

$$P(D_{ik}) = \pi_{1k} + \pi_{2k} Beta(D_{ik}, \alpha_k, 1) + \pi_{3k} Beta(D_{ik}, 1, \beta_k) \tag{E.3}$$

where D_{ik} is the p-value of E_i at knockout S_k, the π_{*k} are the mixing coefficients and we have the constraints that $\alpha_k < 1$ and $\beta_k > 2$. If $\hat{\pi} = P(D_{ik} = 1)$ is the maximum uniform part of the model, then we have:

$$P(D_{ik}|\Phi, \theta_i) = \begin{cases} \frac{P(D_{ik})-\hat{\pi}}{1-\hat{\pi}} & \text{if } \Phi \text{ predicts an effect} \\ 1 & \text{otherwise} \end{cases} \tag{E.4}$$

A Priori Filtering of Effects A typical microarray experiment can measure the expression levels of thousands of genes, not all of which will be affected by the knockout of an S-gene. For that reason, it makes sense to apply an a priori filtering step to remove E-genes that only show random effects. Fröhlich et al. (2008) use a scheme that finds patterns of differentially expressed genes that are statistically significant. Given the multiple-testing corrected p-value p_k of an E-gene expression level in experiment k, and a false positive rate α, we can set:

$$b_k = \begin{cases} 1 & \text{if } p_k < \alpha \\ 0 & \text{otherwise} \end{cases} \tag{E.5}$$

If s_k is the number of significant genes in experiment k, then the probability of observing a pattern $\mathbf{b} = \{b_1, ..., b_n\}$ under the null hypothesis H_0 is:

$$P(\mathbf{b}|H_0) = \prod_k^n (b_k \alpha \frac{s_k}{M} + (1 - b_k)(1 - \alpha)\frac{M - s_k}{M}) \tag{E.6}$$

where M is the total number of E-genes. This allows us to calculate the number of times that we should expect to see b by chance. Using a binomial test, we can calculate the statistical significance of seeing b more often than expected, and keep only those effects which show a significant pattern.

Network Inference Methods We now know how to calculate the likelihood for a given network hypothesis. Unfortunately, unless the number of S-genes is very small, it is impractical to score all possible network structures. To circumvent this problem, Markowetz et al. (2007) developed a method based on scoring networks consisting of triples and combining them, while Froehlich et al. developed two alternative methods: a greedy hillclimbing approach, and a method he called module networks which relies on hierarchical clustering (Fröhlich et al., 2008).

In the triples approach, we consider all possible triples of S-genes and score the networks that can be formed using only three nodes. Then we select the highest-scoring network for each triple and use model averaging to combine them into a complete networks. We calculate the frequency of each edge and include all the edges whose frequency exceeds a certain threshold.

Greedy hillclimbing is a more basic approach where we simply start from a network (usually with no edges) and at each step add the edge that gives the biggest improvement to the score. If no more improvements are possible, the algorithm terminates. This only gives us a local optimum, so it is usually advisable to use bootstrapping (repeat the greedy hillclimbing algorithm several time, each time sampling with replacement from the E-genes) to get a measure of the confidence we have in each edge.

The module networks method starts out by creating a hierarchical clustering of the gene expression profiles using a standard clustering method (Froehlich suggests average linkage). Then, starting from the top, we look for clusters containing at most 4 S-genes. When the network has been decomposed into non-overlapping clusters (or modules) of at most 4 S-genes, we find the highest-scoring network for each cluster using an exhaustive search. Finally, the modules are connected using a constrained greedy hillclimbing approach, which only adds edges between S-genes in different modules.

E.3 Experiments and Results

Simulation As we saw in section E.2, there is more than one way of using NEMs, and it is not immediately obvious which method is the most reliable. For that reason, I started off by doing a small simulation study to compare the most promising approaches: using triples vs using a greedy search with bootstrapping, with or without a priori filtering of effects. I took one of the networks that had been inferred from the data (the one shown in figure E.3a, in fact), and used this as the underlying network for a simulation. The advantage of this approach, other than its simplicity, is that we can evaluate the performance of the NEMs on a realistic network, rather than the transitively closed ideal networks of Fröhlich et al. (2008).

Like Froehlich et al., we sample p values for each knockout from the mixture distribution in equation E.3. Each S-gene is linked to 100 E-genes. The p values for E-genes where we do not expect an effect due to the network structure are sampled from the uniform distribution. There is a slight subtlety in when to expect an effect if there are different paths between two S-genes (e.g. between expM and aepA in the network we use here). If one path is disabled by a network, do we expect to see an effect downstream (AND-model) or will the signal simply travel via the alternative path (OR-model)? We chose to go with the AND-model.

For each S-gene where one would expect an effect, we calculate the probability of observing that effect, based on the distance of the current S-gene to the knockout gene. The observed effects are sampled from the beta distributions according to the mixing coefficients π_{1k} and π_{2k}, while for unobserved effects we sample from the uniform distribution. For each knockout, all parameters are drawn from the same ranges as in Fröhlich et al. (2008), and for each E-gene a small amount of Gaussian noise is added to these parameters.

Figure E.1 shows the results of the simulation study. There is no significant difference between the two optimization schemes: triples versus greedy search, whereas the filtered versus unfiltered scheme shows a significant difference (at the 0.05 significance level). Surprisingly, the effect of filtering is not consistent, leading to an improvement in the performance of the triple method, but a deterioration in the performance of the greedy search. Given these inconsistencies, we decided to apply all four methods to the real data.

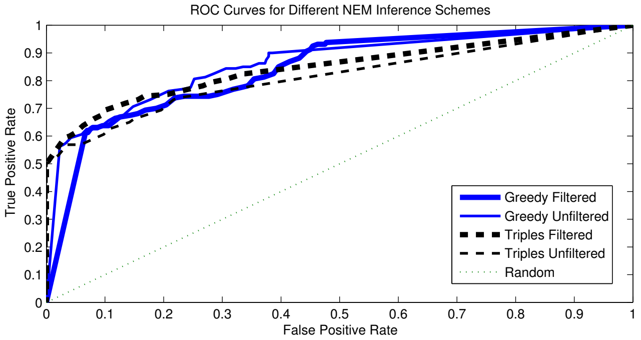


Figure E.1: ROC curves showing the performance of NEMs on simulation data when using different optimization schemes.

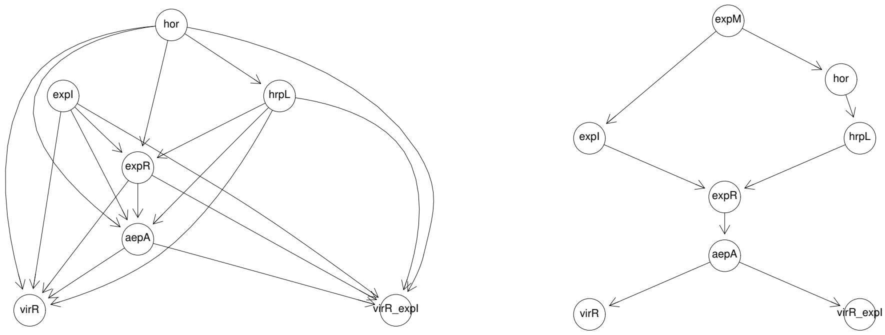
Microarray Data The networks found by the different methods are shown in Figures E.2 and E.3. Because of the intrinsic symmetries of the scoring scheme, NEMs cannot distinguish between two network structures that are equivalent after transitive closure. This implies that the network structures in Figures E.2a and E.2b are score equivalent, that is, they cannot be distinguished on the basis of the inference scheme. As such, the method only infers regulatory hierarchies rather than actual interaction networks.

To resolve ambiguities, and make the networks interpretable, I only present the transitive reduction of each graph, that is, the most parsimonious graph of an equivalence class. This means that whenever two nodes are connected by a path, an interaction via a shortcut path is supported by the data as well.

Figures E.2 and E.3 show the networks obtained with different network inference schemes and different pre-filtering methods. The amount of variation between the graphs gives an indication of the robustness of the inference scheme. In one of the networks we removed the virS data from the dataset; this was due to the fact that virS knockout experiment was carried out under different conditions, which might add an unwanted source of noise.

Unlike for the simulated data, we do not have a gold standard for evaluating how well the true network is predicted using NEMs. However, the literature tells us that there is evidence for some of the regulatory interactions that were predicted. The current knowledge, summarised in Figure 5 in Liu et al. Liu et al. (2008), shows that expI is upstream of both virR and aepA in the regulatory hierarchy. Figures E.2 and E.3 suggest that this order is, in fact, consistently predicted by all the graphs learned in our study.

Moreover, in none of the predicted networks does the double knockout expI/virR



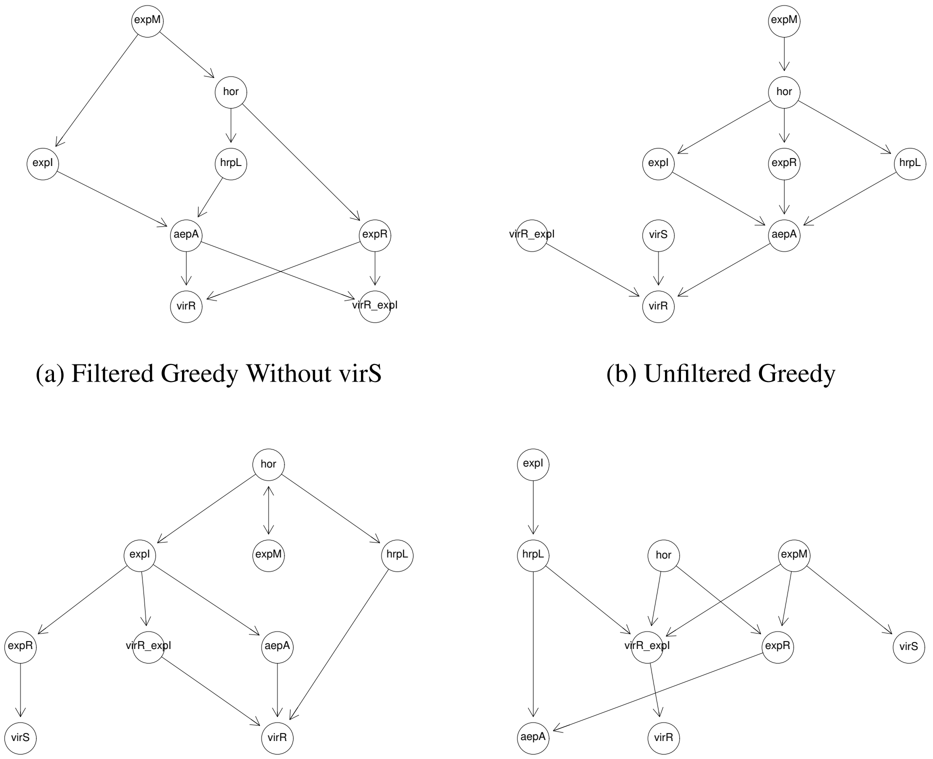
(a) Filtered Greedy Full

(b) Filtered Greedy Reduced

Figure E.2: Filtered Greedy Full - This figure shows the network obtained using a greedy search with bootstrapping, where only the edges that were present in all the bootstrap samples have been retained. The set of effector genes used for the search has been pre-filtered to retain only those genes that show a non-random expression pattern over all knock-outs. The genes virS and expM are not included in this network for visibility purposes (expM was a universal regulator, and virS was regulated by every other node). Because of the nature of NEMs, the full network in (a) is transitively closed. This is not a realistic assumption in most pathways. For that reason, (b) is a transitive reduction of (a) where shortcuts have been eliminated. This makes it easier to interpret the graph, but one should keep in mind that this is only an approximation, and that some of the edges that have been eliminated may have been true interactions.

appear above both the individual knock-outs expI and virR. This can only be explained by some antagonism between expI and virR, which is again in agreement with the regulatory structure reported in Liu et al. (2008). There are also some interesting deviations, though. Liu et al. predict expM to be quite low in the regulatory hierarchy. However, all the graphs learned in our study concur in predicting expM to be at the top of the regulatory hierarchy. This finding might point to some flaws in the current hypotheses about the regulatory mechanisms in Pba, indicating that it may be possible to obtain a revised and improved model of gene regulatory interactions using NEMs or other network inference methods.

A comparison of the predicted graphs points to some disagreement between them. This is an inevitable consequence of the noise in the data, the complexity of the inference problem, and the different nature of the approaches adopted for dealing with both. This work is one of the first studies to investigate the robustness of learning NEMs from real data, and provides insight into the degree of variation in graph structure that results



(a) Filtered Greedy Without virS

(b) Unfiltered Greedy

(c) Filtered Triples

(d) Unfiltered Triples

Figure E.3: (a) is a transitive reduction of the network found using the same method as for figure E.2b, but excluding the virS knockout data. There was reason to believe that the virS data could affect filtering (and possibly network construction) because it was generated under different conditions than the other knock-outs, which might lead to spurious effects. (b) is a transitive reduction of the network found using the same method as for figure E.2b, but without filtering the genes. (c) is a transitive reduction of the network found using the triples scoring method. For this graph, only edges with 100% support have been retained. The same gene filtering method as for figure E.2b has been applied. (d) is a transitive reduction of the network found using the same method as for (c), but without filtering the genes. For this network, it was possible to retain all edges with more than 50% support (because it was sparser than the network in (c)).

from a variation of the learning algorithm and prefiltering scheme.

E.4 Conclusions

The Pba dataset was ideally suited for the NEM, in that it consisted of microarray measurements of knockout mutants. This made it easy to apply NEMs; however, the difficulty was in deciding which variant of the NEM method to apply. The simulation did not show a clear preference, indicating that the differences in accuracy between different methods were small. Since the networks produced vary quite a bit, this variation is probably mainly due to different treatment of the noise in the data.

Nevertheless, we managed to reproduce some of the interactions that have been reported in the literature. The relationships that seem to contradict the literature, such as the regulatory role of expM, may yet be demonstrated, or they may be due to some limitation of NEMs that can be overcome with more sophisticated techniques. In either case, further experiments could probably confirm or deny these relationships.

Appendix F

Software Package EDISON

F.1 Description

As part of the work described in Chapters 3 and 4, I developed an R software package for dynamic Bayesian network inference. The software package EDISON (Estimation of Directed Interactions from Sequences Of Nonhomogeneous gene expression) enables the reconstruction of time-varying gene regulatory networks from non-homogeneous gene expression data, using hierarchical sequential information sharing priors.

F.2 Features

EDISON offers the following functionalities to researchers interested in the reconstruction of time-varying networks:

Network Reconstruction. The software runs reversible-jump MCMC simulations of the hierarchical Bayesian model and returns the sampled gene networks. These can then be processed further using the evaluation functions of the package.

Changepoint Detection. Along with the sampled networks, the software also returns the sampled changepoints for each gene. The changepoint densities, both for individual genes and for the whole network, can be calculated using the evaluation functions.

Information Sharing. I included four varieties of information sharing between successive networks in this release: Exponential versus binomial priors with hard coupling (shared hyperparameters) or soft coupling (gene-specific hyperparameters with common hyperprior) among genes. The package also gives the option of no information sharing, in which case the model is essentially equivalent to ARTIVA in Lèbre et al. (2010).

Convergence Monitoring. MCMC convergence is monitored using the potential scale reduction factor (Gelman and Rubin, 1992).

Network Generation. The package includes functions for generating sequences of random networks with structure changes, and simulating data from them by using a regression model.

F.3 Implementation and Use

The software package is implemented in R, and will run under Window, Linux or OS X in the presence of an installation of R 2.13.1 or later. A simple example for using the software follows:

```
> library('EDISON')
> dataset <- simulateNetwork(l=30, cps=c(10, 20))
> results <- EDISON.run(dataset, num.iter=1e5, information.sharing='bino_hard')
> cps <- calculateCPProbabilities(results)
> networks <- calculateEdgeProbabilites(results)
```

Here, `dataset` is a dataset of length 30 with changepoints at timepoints 10 and 20, simulated from random networks with the default size of 10 nodes and default number of 2 network structure changes per node at each changepoint. Variable `results` will contain the network and changepoint samples collected during the 1e5 RJMCMC simulation steps, and `cps` and `networks` contain the marginal posterior probabilities of the changepoints and interactions in the gene network, respectively. One can apply a threshold to the latter to obtain a discrete network.

The R code has been made publicly available on the Comprehensive R Archive Network (CRAN). Software and documentation can be found at `http://cran.r-project.org/web/packages/EDISON/`.

Bibliography

Aderhold, A., Husmeier, D., Lennon, J., Beale, C., and Smith, V. (2012). Hierarchical Bayesian models in ecology: Reconstructing species interaction networks from non-homogeneous species abundance data. *Ecological Informatics*.

Ahmed, A. and Xing, E. P. (2009). Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106:11878–11883.

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

Albert, R. and Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.

Alstroem, P., Ericson, P., Olsson, U., and Sundberg, P. (2006). Phylogeny and classification of the avian superfamily *Sylvioidea*. *Mol. Phylogenet. Evol.*, 38(2):381–397.

Andrianantoandro, E., Basu, S., Karig, D., and Weiss, R. (2006). Synthetic biology: new engineering rules for an emerging discipline. *Molecular Systems Biology*, 2(1).

Andrieu, C. and Doucet, A. (1999). Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing*, 47(10):2667–2676.

Araujo, M. B., Pearson, R. G., Thuiller, W., and Erhard, M. (2005). Validation of species-climate impact models under climate change. *Global Change Biology*, 11(9):1504–1513.

Arbeitman, M., Furlong, E., Imam, F., Johnson, E., Null, B., Baker, B., Krasnow, M., Scott, M., Davis, R., and White, K. (2002). Gene expression during the life cycle of Drosophila melanogaster. *Science*, 297(5590):2270–2275.

Archambeau, C., Cornford, D., Opper, M., and Shawe-Taylor, J. (2007). Gaussian process approximations of stochastic differential equations. In *Journal of Machine Learning Research Workshop and Conference Proceedings*, volume 1, pages 1–16.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J., and Blom, J. (2009). Systems biology: parameter estimation for biochemical models. *FEBS Journal*, 276(4):886–902.

Augustin, N. H., Mugglestone, M. A., and Buckland, S. T. (1996). An autologistic model for the spatial distribution of wildlife. *J. Appl. Ecol.*, 33(2):339–347.

Bansal, M., Della Gatta, G., and Di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, 22(7):815–822.

Barabási, A., Albert, R., and Schiffer, P. (1999). The physics of sand castles: maximum angle of stability in wet and dry granular media. *Physica A*, 266(1-4):366–371.

Batjes, N. H. (1996). Development of a world data set of soil water retention properties using pedotransfer rules. *Geoderma*, 71(1-2):31–52.

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

Beale, C. M., Lennon, J. J., and Gimona, A. (2008). Opening the climate envelope reveals no macroscale associations with climate in European birds. *Proc. Natl. Acad. Sci.*, 105(39):14908–14912.

Beisner, B. E., Haydon, D. T., and Cuddington, K. (2003). Alternative stable states in ecology. *Front. Ecol. Environ.*, 1(7):376–382.

Blüthgen, N., Menzel, F., and Blüthgen, N. (2006). Measuring specialization in species interaction networks. *BMC Ecology*, 6(1):9.

Bonneau, R., Reiss, D., Shannon, P., Facciotti, M., Hood, L., Baliga, N., Thorsson, V., et al. (2006). The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol*, 7(5):R36.

Brunel, N. (2008). Parameter estimation of ODEs via nonparametric estimators. *Electronic Journal of Statistics*, 2:1242–1267.

Butte, A. and Kohane, I. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, 5:418–429.

Butte, A. S. and Kohane, I. S. (2003). Relevance networks: A first step toward finding genetic regulatory networks within microarray data. In Parmigiani, G., Garett, E. S., Irizarry, R. A., and Zeger, S. L., editors, *The Analysis of Gene Expression Data*, pages 428–446, New York. Springer.

Calderhead, B., Girolami, M. A., and Lawrence, N. D. (2008). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. *Neural Information Processing Systems (NIPS)*, 22.

Campbell, D. and Steele, R. (2012). Smooth functional tempering for nonlinear differential equation models. *Statistics and Computing*, pages 1–15.

Cantone, I., Marucci, L., Iorio, F., Ricci, M. A., Belcastro, V., Bansal, M., Santini, S., di Bernardo, M., di Bernardo, D., and Cosma, M. P. (2009). A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, 137(1):172181.

Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian network structures. *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 11:87–98.

Cho, H., Lee, K., and Fadali, M. (2009). Online learning algorithm of dynamic Bayesian networks for nonstationary signal processing. *International Journal of Innovative Computing, Information and Control*, 5(4):1027–1042.

Cohen, J. E., Schoenly, K., Heong, K. L., Justo, H., dArida, G., Barrion, A. T., and Litsinger, J. (1994). A food-web approach to evaluating the effect of insecticide spraying on insect pest population-dynamics in a Philippine irrigated rice ecosystem. *J. Appl. Ecol.*, 31:747–763.

Dale, M. R. T. and Fortin, M. J. (2002). Spatial autocorrelation and statistical tests in ecology. *Ecoscience*, 9(2):162–167.

Danaher, P., Wang, P., and Witten, D. (2011). The joint graphical lasso for inverse covariance estimation across multiple classes. *arXiv preprint arXiv:1111.0324*.

Davis, A. J., Jenkinson, L. S., Lawton, J. H., Shorrocks, B., and Wood, S. (1998). Making mistakes when predicting shifts in species range in response to global warming. *Nature*, 391(6669):783–785.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, page 240. ACM.

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9(1):67–103.

de Silva, E. and Stumpf, M. P. H. (2005). Complex networks and simple models in biology. *Journal of the Royal Society Interface*, 2(5):419.

Dobra, A. and Lenkoski, A. (2011). Copula Gaussian graphical models and their application to modeling functional disability data. *The Annals of Applied Statistics*, 5(2A):969–993.

Dondelinger, F. (2008). Inferring ecological networks from species abundance data: evaluation on simulated data. Master's thesis, School of Informatics, University of Edinburgh.

Dondelinger, F., Aderhold, A., Grzegorczyk, M., Husmeier, D., et al. (2011). A Bayesian regression and multiple changepoint model for systems biology. *International Workshop on Statistical Modelling (IWSM)*, (2008):1–5.

Dondelinger, F., Husmeier, D., and Lèbre, S. (2012a). Dynamic Bayesian networks in molecular plant science: inferring gene regulatory networks from multiple gene expression time series. *Euphytica*, 183(3):361–377.

Dondelinger, F., Lèbre, S., and Husmeier, D. (2010). Heterogeneous continuous dynamic Bayesian networks with flexible structure and inter-time segment information sharing. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*.

Dondelinger, F., Lèbre, S., and Husmeier, D. (2012b). Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*.

Dondelinger, F., Rogers, S., Filippone, M., Cretella, R., Palmer, T., Smith, R., Millar, A., and Husmeier, D. (2012c). Parameter inference in mechanistic models of cellular regulation and signalling pathways using gradient matching. In *9th International Workshop on Computational Systems Biology (WCSB)*.

Doshi-Velez, F., Wingate, D., Tenenbaum, J., and Roy, N. (2011). Infinite dynamic Bayesian networks. In *Proceedings of the 28th Annual International Conference on Machine Learning (ICML)*. International Machine Learning Society.

Dunne, J., Williams, R., and Martinez, N. (2002). Food-web structure and network theory: the role of connectance and size. *Proc. Natl. Acad. Sci.*, 99(20):12917–12922.

Edwards, D. M. (2000). *Introduction to Graphical Modelling*. Springer Verlag, New York.

Edwards, K., Anderson, P., Hall, A., Salathia, N., Locke, J., Lynn, J., Straume, M., Smith, J., and Millar, A. (2006). FLOWERING LOCUS C mediates natural variation in the high-temperature response of the Arabidopsis circadian clock. *The Plant Cell Online*, 18(3):639.

Elgar, S., Han, J., and Taylor, M. (2008). mef2 activity levels differentially affect gene expression during Drosophila muscle development. *Proceedings of the National Academy of Sciences*, 105(3):918.

Elle, O. (2003). Quantification of the integrative effect of ecotones as exemplified by the habitat choice of Blackcap and Whitethroat (*Sylvia atricapilla* and *S. communis*, Sylviidae). *J. Ornithol.*, 144(3):271–283.

Emilsson, V., Thorleifsson, G., Zhang, B., Leonardson, A., Zink, F., Zhu, J., Carlson, S., Helgason, A., Walters, G., Gunnarsdottir, S., et al. (2008). Genetics of gene expression and its effect on disease. *Nature*, 452(7186):423–428.

Faisal, A. (2008). Inferring ecological networks from species abundance data: application to the European bird atlas data. Master's thesis, School of Informatics, University of Edinburgh.

Faisal, A., Dondelinger, F., Husmeier, D., and Beale, C. (2010). Inferring species interaction networks from species abundance data: A comparative evaluation of various statistical and machine learning methods. *Ecological Informatics*, 5(6):451–464.

Ferrazzi, F., Rinaldi, S., Parikh, A., Shaulsky, G., Zupan, B., and Bellazzi, R. (2008). Population models to learn Bayesian networks from multiple gene expression experiments. http://www.labmedinfo.org/biblio/author/326.

Formstecher, E., Aresta, S., Collura, V., Hamburger, A., Meil, A., Trehin, A., Reverdy, C., Betin, V., Maire, S., Brun, C., et al. (2005). Protein interaction mapping: a Drosophila case study. *Genome research*, 15(3):376.

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

Friedman, N. and Koller, D. (2003). Being Bayesian about network structure. *Machine Learning*, 50:95–126.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620.

Friedman, N., Murphy, K., and Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In *Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI 98)*, pages 139–147.

Fröhlich, H., Fellmann, M., Sültmann, H., and Beissbarth, T. (2008). Estimating signaling networks through nested effects models. *Lifestat 2008*, page 10.

Fröhlich, H., Praveen, P., and Tresch, A. (2011). Fast and efficient dynamic nested effects models. *Bioinformatics*, 27(2):238–244.

Garcia, E. (1983). An experimental test of competition for space between blackcaps *Sylvia atricapilla* and garden warblers *Sylvia borin* in the breeding season. *J. Anim. Ecol.*, 52(3):795–805.

Gardner, T., di Bernardo, D., Lorenz, D., and Collins, J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling.

Gaston, K. J. (2003). *The structure and dynamics of geographic ranges*. Oxford University Press.

Geiger, D. and Heckerman, D. (1994). Learning Gaussian networks. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 235–243, San Francisco, CA. Morgan Kaufmann.

Gelman, A. and Rubin, D. (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 7(4):457–472.

Georgoulas, A., Clark, A., Ocone, A., Gilmore, S., and Sanguinetti, G. (2012). A subsystems approach for parameter estimation of ODE models of hybrid systems. *Electronic Proceedings in Theoretical Computer Science*, 92.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Golightly, A. and Wilkinson, D. (2005). Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788.

Grafen, A. (1989). The phylogenetic regression. *Phil. Trans. R. Soc. B*, 326:119–157.

Grandvalet, Y. and Canu, S. (1999). Outcomes of the equivalence of adaptive ridge with least absolute shrinkage. *Adv. Neural. Inf. Process. Syst.*, 11:445–451.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732.

Grzegorczyk, M. and Husmeier, D. (2009). Non-stationary continuous dynamic Bayesian networks. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 682–690.

Grzegorczyk, M. and Husmeier, D. (2011). Non-homogeneous dynamic Bayesian networks for continuous data. *Machine Learning*, pages 1–65.

Grzegorczyk, M., Husmeier, D., Edwards, K., Ghazal, P., and Millar, A. (2008a). Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Bioinformatics*, 24(18):2071–2078.

Grzegorczyk, M., Husmeier, D., and Werhli, A. (2008b). Reverse engineering gene regulatory networks with various machine learning methods. In Emmert-Streib, F. and Dehmer, M., editors, *Analysis of Microarray Data: A Network-Based Approach*, pages 101–142, Weinheim. Wiley-VCH.

Guelzim, N., Bottani, S., Bourgine, P., and Képès, F. (2002). Topological and causal structure of the yeast transcriptional regulatory network. *Nat. Genet.*, 31(1):60–63.

Guo, F., Hanneke, S., Fu, W., and Xing, E. (2007). Recovering temporally rewiring networks: A model-based approach. In *Proceedings of the 24th international conference on Machine learning*, page 328. ACM.

Hagemeijer, W. J. M. and Blair, M. J. (1997). *The EBCC atlas of European breeding birds: their distribution and abundance*. Poyser London.

Hartemink, A. J. (2001). *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. PhD thesis, MIT.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Heckerman, D. (1999). A tutorial on learning with Bayesian networks. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 301–354, Cambridge, Massachusetts. MIT Press.

Heckerman, D. and Geiger, D. (1994). Learning Bayesian Networks.

Heckerman, D., Geiger, D., and Chickering, D. (1995a). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995b). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:245–274.

Heinrich, P., Behrmann, I., Müller-Newen, G., Schaper, F., and Graeve, L. (1998). Interleukin-6-type cytokine signalling through the gp130/Jak/STAT pathway. *Biochemical Journal*, 334(Pt 2):297.

Henneman, M. L. and Memmott, J. (2001). Infiltration of a Hawaiian community by introduced biological control agents. *Science*, 293(5533):1314–1316.

Holt, R. D. and Barfield, M. (2009). Trophic interactions and range limits: the diverse roles of predation. *P. Roy. Soc. B*, 276(1661):1435–1442.

Homyk Jr, T. and Emerson Jr, C. (1988). Functional interactions between unlinked muscle genes within haploinsufficient regions of the Drosophila genome. *Genetics*, 119(1):105.

Hongo, Y. (2012). *Online Learning of Non-Stationary Networks, with Application to Financial Data*. PhD thesis, Duke University.

Huntley, B., Collingham, Y. C., Willis, S. G., and Green, R. E. (2008). Potential impacts of climatic change on European breeding birds. *PLoS One*, 3(1).

Hurn, A. and Lindsay, K. (1999). Estimating the parameters of stochastic differential equations. *Mathematics and computers in simulation*, 48(4-6):373–384.

Husmeier, D., Dondelinger, F., and Lèbre, S. (2010). Inter-time segment information sharing for non-homogeneous dynamic Bayesian networks. In Lafferty, J. e. a., editor, *Proceedings of the twenty-fourth annual conference on Neural Information Processing Systems (NIPS)*, volume 23, pages 901–909. Curran Associates.

Husmeier, D., Dybowski, R., and Roberts, S. (2005). *Probabilistic Modeling in Bioinformatics and Medical Informatics*. Advanced Information and Knowledge Processing. Springer, New York.

Husmeier, D. and McGuire, G. (2003). Detecting recombination in 4-taxa DNA sequence alignments with Bayesian hidden Markov models and Markov chain Monte Carlo. *Molecular Biology and Evolution*, 20(3):315–337.

Ings, T. C., Montoya, J. M., Bascompte, J., Bluthgen, N., Brown, L., Dormann, C. F., Edwards, F., Figueroa, D., Jacob, U., Jones, J. I., Lauridsen, R. B., Ledger, M. E., Lewis, H. M., Olesen, J. M., van Veen, F. J. F., Warren, P. H., and Woodward, G. (2009). Review: Ecological networks beyond food webs. *J. Anim. Ecol.*, 78:253–269.

Jasra, A., Stephens, D., and Holmes, C. (2007). On population-based simulation for static inference. *Statistics and Computing*, 17(3):263–279.

Jordano, P., Bascompte, J., and Olesen, J. (2003). Invariant properties in coevolutionary networks of plant-animal interactions. *Ecol. Lett.*, 6(1):69–81.

Kolar, M., Song, L., and Xing, E. (2009). Sparsistent learning of varying-coefficient models with structural changes. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1006–1014.

La Sorte, F. A., Lee, T. M., Wilman, H., and Jetz, W. (2009). Disparities between observed and predicted impacts of climate change on winter bird assemblages. *Proceedings of the Royal Society B*, 276(1670):3167.

Lähdesmäki, H. and Shmulevich, I. (2008). Learning the structure of dynamic Bayesian networks from time series and steady state measurements. *Machine Learning*, 71(2):185–217.

Laird, S. and Jensen, H. (2006). The Tangled nature model with inheritance and constraint: Evolutionary ecology restricted by a conserved resource. *Ecol. Complex.*, 3(3):253–262.

Lande, R., Engen, S., and Saether, B. (2003). *Stochastic Population Dynamics in Ecology and Conservation*. Oxford University Press.

Larget, B. and Simon, D. L. (1999). Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, 16(6):750–759.

Lèbre, S. (2007). *Stochastic process analysis for Genomics and Dynamic Bayesian Networks inference.* PhD thesis, Université d'Evry-Val-d'Essonne, France.

Lèbre, S., Becq, J., Devaux, F., Lelandais, G., and Stumpf, M. (2010). Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, 4(130).

Lèbre, S., Dondelinger, F., and Husmeier, D. (2012). Nonhomogeneous dynamic Bayesian networks in systems biology. *Methods in molecular biology (Clifton, NJ)*, 802:199.

Lee, M., Ye, A., Gardino, A., Heijink, A., Sorger, P., MacBeath, G., and Yaffe, M. (2012). Sequential application of anticancer drugs enhances cell death by rewiring apoptotic signaling networks. *Cell*, 149(4):780–794.

Legendre, P. (1993). Spatial autocorrelation: trouble or new paradigm? *Ecology*, 74(6):1659–1673.

Lennon, J. J. (2000). Red-shifts and red herrings in geographical ecology. *Ecography*, 23:101–113.

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

Liang, F., Paulo, R., Molina, G., Clyde, M. A., and Berger, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, 103(481).

Liang, H. and Wu, H. (2008). Parameter estimation for differential equation models using a framework of measurement error in regression models. *Journal of the American Statistical Association*, 103(484):1570–1583.

Lin, K., Husmeier, D., Dondelinger, F., Mayer, C., Liu, H., Prichard, L., Salmond, G., Toth, I., and Birch, P. (2010). Reverse engineering gene regulatory networks related to quorum sensing in the plant pathogen *Pectobacterium atrosepticum. Computational Biology*, pages 253–281.

Liu, B., Hagiescu, A., Palaniappan, S., Chattopadhyay, B., Cui, Z., Wong, W., and Thiagarajan, P. (2012). Approximate probabilistic analysis of biopathway dynamics. *Bioinformatics*, 28(11):1508–1516.

Liu, H., Coulthurst, S., Pritchard, L., Hedley, P., Ravensdale, M., Humphris, S., Burr, T., Takle, G., Brurberg, M., Birch, P., et al. (2008). Quorum sensing coordinates brute force and stealth modes of infection in the plant pathogen Pectobacterium atrosepticum. *PLoS Pathogens*, 4(6).

Liu, H., Xu, M., Gu, H., Gupta, A., Lafferty, J., and Wasserman, L. (2010). Forest density estimation. *arXiv preprint arXiv:1001.1557.*

Lo, K., Raftery, A., Dombek, K., Zhu, J., Schadt, E., Bumgarner, R., and Yeung, K. (2012). Integrating external biological knowledge in the construction of regulatory networks from time-series expression data. *BMC Systems Biology*, 6(1):101.

Locke, J., Kozma-Bognár, L., Gould, P., Fehér, B., Kevei, E., Nagy, F., Turner, M., Hall, A., and Millar, A. (2006). Experimental validation of a predicted feedback loop in the multi-oscillator clock of Arabidopsis thaliana. *Molecular Systems Biology*, 2(1).

Locke, J., Southern, M., Kozma-Bognar, L., Hibberd, V., Brown, P., Turner, M., and Millar, A. (2005). Extension of a genetic network model by iterative experimentation and mathematical analysis. *Molecular Systems Biology*, 1:(online).

Logsdon, B., Hoffman, G., and Mezey, J. (2012). Mouse obesity network reconstruction with a variational bayes algorithm to employ aggressive false positive control. *BMC Bioinformatics*, 13(1):53.

Losos, J. B. (2008). Phylogenetic niche conservatism, phylogenetic signal and the relationship between phylogenetic relatedness and ecological similarity among species. *Ecol. Lett.*, 11(10):995–1003.

Lotka, A. (1932). The growth of mixed populations: two species competing for a common food supply. *Journal of the Washington Academy of Sciences*, 22(461-469):461–469.

Luce, R. and Perry, A. (1949). A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116.

MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Comput.*, 4:415–447.

MacKay, D. J. C. (1996). Hyperparameters: optimize, or integrate out. In Heidbreder, G., editor, *Maximum Entropy and Bayesian Methods*, pages 43–59, Santa Barbara. Kluwer Academic Publisher.

Maddison, D., Schulz, K., and Maddison, W. (2007). The tree of life web project. *Zootaxa*, 1668:19–40.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *Int. Stat. Rev.*, 63:215–232.

Marbach, D., Costello, J., Küffner, R., Vega, N., Prill, R., Camacho, D., Allison, K., Kellis, M., Collins, J., Stolovitzky, G., et al. (2012). Wisdom of crowds for robust gene network inference. *Nature Methods*.

Markowetz, F., Bloch, J., and Spang, R. (2005). Non-transcriptional pathway features reconstructed from secondary effects of rna interference. *Bioinformatics*, 21(21):4026–4032.

Markowetz, F., Kostka, D., Troyanskaya, O., and Spang, R. (2007). Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, 23(13):i305–i312.

May, R. (2006). Network structure and the biology of populations. *Trends Ecol. Evol.*, 21(7):394–399.

McClung, C. R. (2006). Plant circadian rhythms. *Plant Cell*, 18(4):792–803.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Memmott, J. (1999). The structure of a plant-pollinator food web. *Ecology Letters*, 2(5):276–280.

Memmott, J., Fowler, S., Paynter, Q., Sheppard, A., and Syrett, P. (2000). The invertebrate fauna on broom, *Cytisus scoparius*, in two native and two exotic habitats. *Acta Oecol.*, 21(3):213–222.

Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002). Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824.

Mockler, T., Michael, T., Priest, H., Shen, R., Sullivan, C., Givan, S., McEntee, C., Kay, S., and Chory, J. (2007). The diurnal project: Diurnal and circadian expression profiling, model-based pattern matching and promoter analysis. *Cold Spring Harbor Symposia on Quantitative Biology*, 72:353–363.

Møller, J., Madsen, H., and Carstensen, J. (2011). Parameter estimation in a simple stochastic differential equation for phytoplankton modelling. *Ecological Modelling*, 222(11):1793–1799.

Montana, E. and Littleton, J. (2004). Characterization of a hypercontraction-induced myopathy in Drosophila caused by mutations in mhc. *The Journal of Cell Biology*, 164(7):1045.

Murphy, K. and Mian, S. (1999). Modelling gene expression data using dynamic Bayesian networks. *University of California, Berkeley.*

Murray, I. and Adams, R. (2010). Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 23.

Murray Jr, B. G. (1988). Interspecific territoriality in *Acrocephalus*: A critical review. *Ornis Scand.*, 19(4):309–313.

Needham, C., Bradford, J., Bulpitt, A., and Westhead, D. (2007). A primer on learning in Bayesian networks for computational biology. *PLoS Comput. Biol.*, 3(8):1409–1416.

New, M., Hulme, M., and Jones, P. (1999). Representing twentieth-century space–time climate variability. Part I: Development of a 1961–90 mean monthly terrestrial climatology. *J. Climate*, 12(3):829–856.

Nobile, A. and Fearnside, A. (2007). Bayesian finite mixtures with an unknown number of components: the allocation sampler. *Stat. Comput.*, 17(2):147–162.

Nodelman, U., Shelton, C., and Koller, D. (2002). Learning continuous time Bayesian networks. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 451–458. Morgan Kaufmann Publishers Inc.

Nongthomba, U., Cummins, M., Clark, S., Vigoreaux, J., and Sparrow, J. (2003). Suppression of muscle hypercontraction by mutations in the myosin heavy chain gene of *Drosophila melanogaster*. *Genetics*, 164(1):209.

Oates, C., Hennessy, B., Lu, Y., Mills, G., and Mukherjee, S. (2012a). Network inference using steady-state data and Goldbeter–koshland kinetics. *Bioinformatics*, 28(18):2342–2348.

Oates, C., Hill, S., Mukherjee, S., et al. (2012b). Comment on 'large-scale dynamic gene regulatory network inference combining differential equation models with local dynamic Bayesian network analysis'. *arXiv preprint arXiv:1201.3380.*

Opgen-Rhein, R. and Strimmer, K. (2007). From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Syst. Biol.*, 1(37).

Park, T. and Casella, G. (2008). The Bayesian Lasso. *J. Am. Stat. Assoc.*, 103(482):681–686.

Parkhurst, S. and Ish-Horowicz, D. (1991). wimp, a dominant maternal-effect mutation, reduces transcription of a specific subset of segmentation genes in Drosophila. *Genes & Development*, 5(3):341.

Pearl, J. (1985). Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society*.

Pearl, J. (2000). *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press.

Penfold, C., Buchanan-Wollaston, V., Denby, K., and Wild, D. (2012). Nonparametric Bayesian inference for perturbed and orthologous gene regulatory networks. *Bioinformatics*, 28(12):i233–i241.

Perrin, B., Ralaivola, L., Mazurie, A. E., Bottani, S., Mallet, J., and d'Alch Buc, F. (2003). Gene network inference using dynamic Bayesian networks. *Bioinformatics*, 19.

Pokhilko, A., Hodge, S., Stratford, K., Knox, K., Edwards, K., Thomson, A., Mizuno, T., and Millar, A. (2010). Data assimilation constrains new connections and components in a complex, eukaryotic circadian clock model. *Molecular Systems Biology*, 6(1).

Poyton, A., Varziri, M., McAuley, K., McLellan, P., and Ramsay, J. (2006). Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers & Chemical Engineering*, 30(4):698–708.

Prentice, I. C., Cramer, W., Harrison, S. P., Leemans, R., Monserud, R. A., and Solomon, A. M. (1992). A global biome model based on plant physiology and dominance, soil properties and climate. *Journal of Biogeography*, 19(2):117–134.

Prill, R. J., Marbach, D., Saez-Rodriguez, J., Sorger, P. K., Alexopoulos, L. G., Xue, X., Clarke, N. D., Altan-Bonnet, G., and Stolovitzky, G. (2010). Towards a rigorous assessment of systems biology models: The DREAM3 challenges. *PLoS ONE*, 5(2):e9202.

Proulx, S., Promislow, D., and Phillips, P. (2005). Network thinking in ecology and evolution. *Trends Ecol. Evol.*, 20(6):345–353.

Punskaya, E., Andrieu, C., Doucet, A., and Fitzgerald, W. (2002). Bayesian curve fitting using MCMC with applications to signal segmentation. *IEEE Transactions on Signal Processing*, 50(3):747–758.

Ramsay, J., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rau, A., Jaffrézic, F., Foulley, J., and Doerge, R. (2010). An empirical Bayesian method for estimating biological networks from temporal microarray data. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

Reshef, D., Reshef, Y., Finucane, H., Grossman, S., McVean, G., Turnbaugh, P., Lander, E., Mitzenmacher, M., and Sabeti, P. (2011). Detecting novel associations in large data sets. *science*, 334(6062):1518–1524.

Robinson, J. and Hartemink, A. (2010). Learning non-stationary dynamic Bayesian networks. *The Journal of Machine Learning Research*, 11:3647–3680.

Robinson, J. W. and Hartemink, A. J. (2009). Non-stationary dynamic Bayesian networks. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 1369–1376. Morgan Kaufmann Publishers.

Rogers, S. and Girolami, M. (2005). A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, 21(14):3131–3137.

Rolando, A. and Palestrini, C. (1991). The effect of interspecific aggression on territorial dynamics in *Acrocephalus* warblers in a marsh area of north-western Italy. *Bird Study*, 38(2):92–97.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Protein-signaling networks derived from multiparameter single-cell data. *Science*, 308:523–529.

Sanchez, C., Lachaize, C., Janody, F., Bellon, B., Roeder, L., Euzenat, J., Rechenmann, F., and Jacq, B. (1999). Grasping at molecular interactions and genetic networks in *Drosophila melanogaster* using FlyNets, an internet database. *Nucleic acids research*, 27(1):89.

Schäfer, J. and Strimmer, K. (2005a). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764.

Schäfer, J. and Strimmer, K. (2005b). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Stat. Appl. Genet. Mol. Biol.*, 4(1):Article 32.

Schäfer, T., Ledebur, G., Beier, J., and Leisler, B. (2006). Reproductive responses of two related coexisting songbird species to environmental changes: global warming, competition, and population sizes. *J. Ornithol.*, 147(1):47–56.

Schmitz, O. J., Krivan, V., and Ovadia, O. (2004). Trophic cascades: the primacy of trait-mediated indirect interactions. *Ecol. Lett.*, 7(2):153–163.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464.

Shachter, R. (1998). Bayes-ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proc. of the Conf. on Uncertainty in AI.*

Sims, D., Bursteinas, B., Gao, Q., Zvelebil, M., and Baum, B. (2006). FLIGHT: database and tools for the integration and cross-correlation of large-scale RNAi phenotypic datasets. *Nucleic acids research*, 34(suppl 1):D479.

Sinclair, A. R. E. and Byrom, A. E. (2006). Understanding ecosystem dynamics for conservation of biota. *J. Anim. Ecol.*, 75:64–79.

Smith, V., Yu, J., Smulders, T., Hartemink, A., and Jarvis, E. (2006). Computational inference of neural information flow networks. *PLoS Comput. Biol.*, 2(11):1436–1449.

Smyth, G. et al. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):3.

Snow, D. W. and Perrins, C. M. (1998). *The Birds of the Western Palearctic Concise Edition*. Oxford University Press, Oxford.

Soetaert, K., Petzoldt, T., and Setzer, R. (2010). Solving differential equations in R: Package deSolve. *Journal of Statistical Software*, 33(9):1–25.

Talih, M. and Hengartner, N. (2005). Structural learning with time-varying components: Tracking the cross-section of financial time series. *Journal of the Royal Statistical Society B*, 67(3):321–341.

Tenenhaus, A., Guillemot, V., Gidrol, X., and Frouin, V. (2010). Gene association networks from microarray data using a regularized estimation of partial correlation based on pls regression. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 7(2):251–262.

Thomas, C. D., Cameron, A., Green, R. E., Bakkenes, M., Beaumont, L. J., Collingham, Y. C., Erasmus, B. F. N., de Siqueira, M. F., Grainger, A., Hannah, L., Hughes, L., Huntley, B., van Jaarsveld, A. S., Midgley, G. F., Miles, L., Ortega-Huerta, M. A., Peterson, A. T., Phillips, O. L., and Williams, S. E. (2004). Extinction risk from climate change. *Nature*, 427(6970):145–148.

Thuiller, W., Lavorel, S., Araújo, M. B., Sykes, M. T., and Prentice, I. C. (2005). Climate change threats to plant diversity in Europe. *Proceedings of the National Academy of Sciences*, 102(23):8245.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc. B*, 58(1):267–288.

Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *JMLR*, 1(3):211–244.

Tipping, M. and Faul, A. (2003). Fast marginal likelihood maximisation for sparse Bayesian models. In Bishop, C. M. and Frey, B. J., editors, *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 9.

Tirelli, T., Pozzi, L., and Pessani, D. (2009). Use of different approaches to model presence/absence of Salmo marmoratus in Piedmont (Northwestern Italy). *Ecological Informatics*, 4(4):234–242.

Titsias, M. and Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24.

Tong, A. H. Y., Drees, B., Nardelli, G., Bader, G. D., Brannetti, B., Castagnoli, L., Evangelista, M., Ferracuti, S., Nelson, B., Paoluzi, S., Quondam, M., Zucconi, A., Hogue, C. W. V., Fields, S., Boone, C., and Cesareni, G. (2002). A Combined Experimental and Computational Strategy to Define Protein Interaction Networks for Peptide Recognition Modules. *Science*, 295(5553):321–324.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202.

Valiente, G. (2002). *Algorithms on trees and graphs*. Springer Verlag.

van Someren, E. P., Vaes, B. L. T., Steegenga, W. T., Sijbers, A. M., Dechering, K. J., and Reinders, M. J. T. (2006). Least absolute regression network analysis of the murine osterblast differentiation network. *Bioinformatics*, 22(4):477–484.

van Veen, F. J., Brandon, C. E., and Godfray, H. C. (2009). A positive trait-mediated indirect effect involving the natural enemies of competing herbivores. *Oecologia*, 160(1):195–205.

Varah, J. (1982). A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3:28.

Vázquez, D. P. and Simberloff, D. (2002). Ecological specialization and susceptibility to disturbance: Conjectures and refutations. *The American Naturalist*, 159(6):606–623.

Vyshemirsky, V. and Girolami, M. (2008). Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839.

Wagner, A. (2001). The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol. Biol. Evol.*, 18(7):1283–1292.

Wang, Z., Kuruoglu, E., Yang, X., Xu, Y., and Huang, T. (2011). Time varying dynamic Bayesian network for non-stationary events modeling and online inference. *IEEE Transactions on Signal Processing*, 4(59).

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Watts, M. and Worner, S. (2008). Comparing ensemble and cascaded neural networks that combine biotic and abiotic variables to predict insect species distribution. *Ecological Informatics*, 3(6):354–366.

Wei, Z. and Li, H. (2007). A Markov random field model for network-based analysis of genomic data. *Bioinformatics*, 23(12):1537.

Werhli, A. and Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Stat. Appl. Genet. Mol. Biol.*, 6(1):Article 15.

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

Werhli, A. V. and Husmeier, D. (2008). Gene regulatory network reconstruction by Bayesian integration of prior knowledge and/or different experimental conditions. *Journal of Bioinformatics and Computational Biology*, 6(3):543–572.

Werner, E. E. and Peacor, S. D. (2003). A review of trait-mediated indirect interactions in ecological communities. *Ecology*, 84(5):1083–1100.

Werner, T. (2010). Next generation sequencing in functional genomics. *Briefings in bioinformatics*, 11(5):499–511.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics*. Wiley New York.

Williams, P. M. (1995). Bayesian regularization and pruning using a Laplace prior. *Neural Comput.*, 7:117–143.

Williams, R. and Martinez, N. (2000). Simple rules yield complex food webs. *Nature*, 404(6774):180–183.

Xuan, X. and Murphy, K. (2007). Modeling changing dependency structure in multivariate time series. In Ghahramani, Z., editor, *Proceedings of the 24th Annual International Conference on Machine Learning (ICML 2007)*, pages 1055–1062. Omnipress.

Yeung, K., Dombek, K., Lo, K., Mittler, J., Zhu, J., Schadt, E., Bumgarner, R., and Raftery, A. (2011). Construction of regulatory networks using expression time-series data of a genotyped population. *Proceedings of the National Academy of Sciences*, 108(48):19436–19441.

Yuan, M. and Lin, Y. (2005). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In Goel, P. and Zellner, A., editors, *Bayesian Inference and Decision Techniques*, pages 233–243. Elsevier.

Zhao, W., Serpedin, E., and Dougherty, E. (2006). Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics*, 22(17):2129.

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