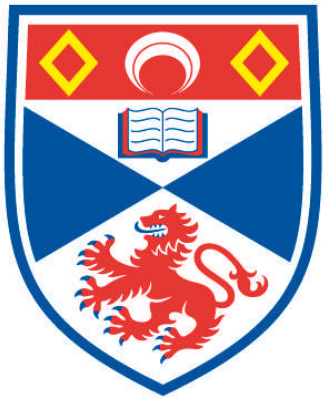
MACHINE LEARNING IN SYSTEMS BIOLOGY AT DIFFERENT SCALES : FROM MOLECULAR BIOLOGY TO ECOLOGY

Andrej Aderhold

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



2015

Machine Learning in Systems Biology at Different Scales: from Molecular Biology to Ecology

Andrej Aderhold

This thesis is submitted in partial fulfilment for the degree of

Ph.D.

at the

University of St Andrews

2015

Declaration

1. Candidate's declarations:

I, Andrej Aderhold, hereby certify that this thesis, which is approximately 64000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in August 2010 and as a candidate for the degree of Ph.D. in August 2011; the higher study for which this is a record was carried out in the University of St Andrews between 2010 and 2015.

Date 8th April 2015 signature of candidate _____

2. Supervisor's declaration:

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Ph.D. in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date 8th April 2015 signature of supervisor _____

3. Permission for publication:

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis: No embargo on print and electronic copy.

Date 8th April 2015 signature of candidate _____ signature of supervisor _____

Abstract

Machine learning has been a source for continuous methodological advances in the field of computational learning from data. Systems biology has profited in various ways from machine learning techniques but in particular from network inference, i.e. the learning of interactions given observed quantities of the involved components or data that stem from interventional experiments. Originally this domain of system biology was confined to the inference of gene regulation networks but recently expanded to other levels of organization of biological and ecolgical systems. Especially the application to species interaction networks in a varying environment is of mounting importance in order to improve our understanding of the dynamics of species extinctions, invasions, and population behaviour in general.

The aim of this thesis is to demonstrate an extensive study of various state-of-art machine learning techniques applied to a genetic regulation system in plants and to expand and modify some of these methods to infer species interaction networks in an ecological setting. The first study attempts to improve the knowledge about circadian regulation in the plant *Arabidopsis thaliana* from the view point of machine learning and gives suggestions on what methods are best suited for inference, how the data should be processed and modelled mathematically, and what quality of network learning can be expected by doing so. To achieve this, I generate a rich and realistic synthetic data set that is used for various studies under consideration of different effects and method setups. The best method and setup is applied to real transcriptional data, which leads to a new hypothesis about the circadian clock network structure.

The ecological study is focused on the development of two novel inference methods that exploit a common principle from transcriptional time-series, which states that expression profiles over time can be temporally heterogeneous. A correponding concept in a spatial domain of 2 dimensions is that species interaction dynamics can be spatially heterogeneous, i.e. can change in space dependent on the environment and other factors. I will demonstrate the expansion from the 1-dimensional time domain to the 2-dimensional spatial domain, introduce two distinct space segmentation schemes, and consider species dispersion effects with spatial autocorrelation. The two novel methods display a significant improvement in species interaction inference compared to competing methods and display a high confidence in learning the spatial structure of different species neighbourhoods or environments.

Contents

Preface

The content of this thesis is the result of my Ph.D. study that produced the following published papers and a book chapter.

- Aderhold, A., Husmeier, D., Lennon, J. J., Beale, C. M., and Smith, V. A. (2012). Hierarchical Bayesian models in ecology: reconstructing species interaction networks from non-homogeneous species abundance data. *Ecological Informatics*.
- Aderhold, A., Husmeier, D., and Smith, V. A. (2013). Reconstructing ecological networks with hierarchical Bayesian regression and Mondrian processes. *Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*.
- Aderhold, A., Husmeier, D., Smith, V. A., Millar, A. J., and Grzegorczyk, M. (2013). Assessment of regression methods for inference of regulatory networks involved in circadian regulation. *Tenth International Workshop on Computational Systems Biology (WCSB)*.
- Aderhold, A., Husmeier, D., and Grzegorczyk, M. (2014). Statistical inference of regulatory networks for circadian regulation. *Statistical applications in genetics and molecular biology (SAGMB)*.
- Grzegorczyk, M., Aderhold, A., Smith, V. A., and Husmeier, D. (2014). Inference of circadian regulatory networks. *Second International Work-conference on Bioinformatics and Biomedical Engineering (IWBBIO)*.
- Aderhold, A., Smith, V. A., and Husmeier, D. (2014). Biological Network Inference at Multiple Scales: From Gene Regulation to Species Interactions. *Pattern Recognition in Computational Molecular Biology: Techniques and Approaches, Wiley*. [book chapter in print]

From the time line of publication dates and the topics it can be observed that I started my work in ecology and later continued with computational molecular biology. The initial model that I used to develop the two proposed methods for the ecological application were based on the inference of gene regulation networks. I encountered several similarities or common principles that could be exploited to improve the network inference for the ecological data. To learn more about these principles we decided to take a step back and study the recent scientific progress in the inference of gene regulation networks, which culminated in the publication of three additional papers and an overview book chapter. I'm confident that the experience I gained from these studies will positively affect my research into ecological systems even though it has been dormant for a while. Furthermore, I learned that the complexity of molecular regulation mechanisms, in particular the integration of multiple levels of organization, presents a challenging field of research that is likely to further improve the modelling of molecular systems. Thus it is worthwhile to consider multiscale modelling approaches that take advantage of common principles and integrate different scales.

Acknowledgements

I'd like to warmly thank my Ph.D. supervisors V. Anne Smith and Dirk Husmeier for their encouragements, the many valueable discussions, and especially their patience. I would not have come so far without their persistent support. Furthermore, I'd like to thank all the other co-authors without whom the following scientific contributions would not have been possible: Marco Grzegorczyk, John J. Lennon, Colin M. Beale and Andrew J. Millar. A large hug goes to Christin who tolerated me in the ups and downs of my Ph.D. study.

Chapter 1

Introduction

Since the dawn of molecular biology and the recent compelling increase of biological data, it has become evident that the study of complex biological systems requires not only an understanding of the involved parts following a reductionist approach, but more importantly, the understanding of the causalities that describe how the parts interact with each other and how these interactions in turn affect other parts of the system that might exist on higher or lower levels of organization. A fundamental tool to reveal such causal links is the exploitation of information from interventional data such as the knock-down or over-expression of genes in molecular biology, or the perturbation of an organism through artificial changes of the environmental conditions [9]. Experimental design and hypothesis building in molecular and cell biology are substantially based on these approaches. In addition, from a holistic viewpoint it can be beneficial to combine and link multiple scales of a biological systems, from the scale of molecular interactions to the scale of cell behaviour, and further to the scale of cell populations [99]. In particular, it can be observed that common biological principles frequently dictate similar behaviour on different biological scales or even different systems. The knowledge about one system can be exploited to model the behaviour of different, but yet in principle also similar, systems. For instance, the mechanisms that govern invasion and establishment of foreign species in ecology can be applied to model the spread and growth of cancer cells in healthy tissue [83], or the observation that seemingly unrelated collective behaviours in protein families, network of neurons, and flocks of birds, manifest at the so called critical point of a phase transition [105], which indicates a deeper governing principle.

This thesis attempts to demonstrate how common principles in molecular biology can

improve the modelling of ecological systems. In particular, I will show how the time-varying nature of gene regulatory networks can be adapted to a two-dimensional spatial domain that defines the neighbourhoods or niches of dependent species. The thesis is structured in two main parts: The first part is a study of a circadian molecular system in plants that includes an extensive evaluation of various machine learning techniques that attempt to learn the structure of gene and protein regulation. The second part will present the adaptation of a state-of-the-art technique called the time-varying dynamic Bayesian network to an ecological setting that takes into account spatial heterogeneity and spatial autocorrelation. What follows is the motivation for systems biology in the next section and a brief overview of the challenges in molecular biology (Section 1.2) and ecology (Section 1.3). The introduction concludes with an overview of the thesis in Section 1.4.

1.1 Motivation for systems biology

The field of systems biology recently emerged as an interdisciplinary science that involves various fields, but in particular genomics, mathematics, statistics, and bioinformatics. In contrast to traditional non-quantitative approaches that often emphasized a reductionist view on biological systems, modern quantitative systems biology has undergone a paradigm shift that conceptualises molecular reactions in the cell, the elementary building block of life, as a complex underlying network of interactions or pathways. Besides aiming for a deeper theoretical understanding of molecular processes and their emergent properties, modern systems biology sees a huge range of potential applications, ranging from the targeted genetic modifications of plants for improved resistance, yield, and a variety of agronomic desired traits [135], to unravelling the causes of neurodegenerative diseases, cancer, and ageing (e.g. [112, 146]).

The new challenge for systems biology is to encompass all the facets of biology, from the molecular scale of gene regulatory networks to cell organization and movement, from signal transduction in neural synapses to the influence of species interactions on biodiversity, from chemotaxis of cancer cells during invasion and metastasis, to species dispersion and migration patterns. The ultimate quest is the elucidation of the common principles spanning several spatial and temporal scales. Important questions to be addressed are: Which common mechanisms determine both aberrant behaviour in groups of migrating animals and the movement of cancer cells in the human body? Which organisational principles are common to the response of eukaryote gene regula-

tory networks to environmental stress, and the response of trophic species interaction networks to climate change? What mathematical model commonalities can be defined that govern principle mechanisms in intra-cellular and inter-cellular signalling?

To find answers to these question, mathematics has been extensively applied to biological problems by reformulating the problem into the mathematical language. For instance, a biologist can ask a question about how something is built, how it works, and what it is for. For a mathematician, these questions are about structure, mechanism, and function [30]. Hence, applying mathematical principles on biological data given these questions can reveal otherwise hidden insights and produce meaningful biological interpretations and predictions. In the case of systems biology the dominant contemporary question is about the structure of organization of a biological system that can involve the temporal domain. Researchers have developed a variety of inference methods that attempt to learn the relations of elements and thus the structure. The inference problem can be generally defined as the process of "reverse engineering", or inferring, the interactions between components of a system given its observed quantities that can be obtained from a single or different conditions (e.g., interventional data). Interactions play such an important role because they carry out the processes that define the behaviour of a system and life itself. This can involve simple signalling, regulation, or control and further extend to emergent behaviour that only springs into existence because of complex patterns of interaction. Mathematics has yet to invent principles and formalisms that are able to describe complex biological challenges such as presented by emergent functionality [30].

The major challenge in systems biology arises from the fact that interactions have to be predicted by the observed quantities of the involved components alone. This is common for small systems where it is often difficult to observe a process directly because of technical limitations. For large systems, however, it can also be very demanding and laborious to identify all meaningful interactions by pure observation. In addition, factors that for instance link across different biological domains can significantly impact a system and, likewise, hidden factors that might play a crucial role but are not known have to be taken into account. Systems biology tries to address these challenges when it comes to the inference of biological structure.

1.2 The challenge in molecular biology

The most established application of systems biology is in genomics and particular in the study of interaction patterns as has been previously noted. One such pattern is manifested in gene regulatory networks, which are responsible for the control of the majority of molecular processes essential for growth and survival of any organism on earth. This can involve the control of organism development [8], response of the immune system to pathogens [126], or the adaptation to changing environmental conditions through stress responses [130].

The gene regulatory network defines the organizational structure that controls gene expression through various regulation or transcription factors including specific proteins or microRNAs [76]. Unfortunately, the actual regulatory processes can not be observed directly and have to be inferred by measuring the quantitive change of those components that are involved in the interactions. One key component that acts as the mediator of genetic expression is the messenger RNA (mRNA) which is a transcript from an activated gene that is translated into protein. The level of mRNA in a sample is the most common form of capturing the genetic activity. Several techniques exists to measure the amount of specific mRNA molecules, such as the "quantitative reverse transcription polymerase chain reaction" (qRT-PCR) method, and the real-time qRT-PCR method that permits the measurement of mRNA during the amplification process [23][1]. A more recent technique is "Whole Transcriptome Shotgun Sequencing" (WTSS), also called RNA sequencing (RNA-seq), which has the capability to capture RNA transcripts in a whole genome for a variety of different RNA types, e.g. mRNA, and non-coding such as miRNA [28]. In addition to strictly measuring RNA concentrations, it is often desireable to artificially manipulate the biological activity of RNA molecules to mimic loss of function for specific genes. Such molecular pertubations, which correspond to gene knock-outs or over-expression (depending on what function the pertubed RNA had), can be realized through the so called RNA interference (RNAi) technique. Data measurements that stem from such experiments are called interventional data [9] and play a crucial role in classic biology because they facilitate the identification and isolation of causalities by comparing the behaviour and expression of possibly related components in the perturbed system to the expression patterns in the unperturbed system. For instance, a common experimental setup in the identification of regulatory effects is to knock out a certain component and observe whether

[1] PCR was originally developed for the amplification of DNA, but can also be used for RNA.

hypothised target components are either over- or under-expressed compared to normal conditions. The former case would indicate an inhibitory relationship and the latter case an activitional effect of the component that was knocked out. Hence, it is beneficial for any methods that aims to identify regulatory patterns to include such interventional data together with pure observational data. In addition, certain prior knowledge about the system, such as an expected density of interconnectivity, validated interactions, e.g. from Chip-Seq experiments that provide protein to DNA binding information, or knowledge about time dependent changes in the system in form oscillations or time delayed reactions, add to the understanding of regulatory mechanisms. Despite these techniques, pattern identification on a broader scale can potentially contribute to the interpretation of experimental outcomes.

In the last decade, computational systems biology has been the driving force for the development of inference methods that attempt to automatically identify regulatory patterns. In light of the growing amount of biomolecular data from high-throughput techniques, such as DNA micro-array experiments [129], or from quantitative real-time polymerase chain reaction (qRT-PCR) data, the computational cost required to process this data is substantially increasing. Following up on the seminal paper by Friedman et al. [48], a variety of methods have been proposed [148], and several procedures have been pursued to objectively assess the network reconstruction accuracy [80, 149, 34], e.g. of the Raf pathway, a cellular signalling network in human immune system cells [126]. It has been demonstrated that machine learning techniques can not only serve to broaden our biological knowledge [44, 37], but also handle the increasing amount of data from high-throughput measurements in a more efficient way than was previously attempted [72].

I will describe various of these state-of-the art methods that are used for the inference of gene regulation networks in Chapter 2. In Chapter 4 these methods are applied to a system of circadian regulation in the plant *Arabidopsis thaliana*. I will test the performance of each method with a benchmark given by a realistic gene and protein regulation system that is produced by simulations from stochastic differential equation. Experimental protocols are closely followed, including the entrainment of seedlings to different light-dark cycles and the knock-out of various key regulatory genes. Furthermore, this study provides relative assessment scores for the comparison of the presented methods, and investigates the influence of systematically missing values related to unknown protein concentrations and mRNA transcription rates.

1.3 The challenge in ecological systems

While interaction networks at the molecular level have been at the forefront of modern biology, due to the ever increasing amount of available post-genomic data, interaction networks at other scales are drawing growing attention. This concerns, in particular, ecological networks, owing to their connection with climate change and biodiversity, which poses new challenges and opportunities for machine learning and computational statistics. Similar to molecular systems, ecological systems are complex dynamical systems with interconnected networks of interactions among species and abiotic factors of the environment. Following commonalities between molecular and ecological systems exist: Genes are the basic building blocks of gene regulation networks and can be compared to organisms as principle participants in the formation of ecological networks; gene profile measurements from DNA or mRNA assays can be compared to population data gathered in field surveys; expression profiles of genes or proteins can be matched with population densities or species coverage; gene regulation compares to species interactions, and different conditions compare to different environments. Hence, it appears natural to apply the same principle found in gene regulatory networks with certain adaptations, to the model of an ecological system.

What is the importance of deciphering the interconnectedness of an ecological system besides gaining knowledge about individual dependencies? As with molecular regulatory networks, the complexity of ecological networks is staggering, with hundreds or thousands of species interacting in multiple ways, from competition and predation to facilitation (whereby one species profits from the presence of another) and mutualism (where two species exist in a relationship in which each benefits from the other). These characteristics in synergy form ecological systems that are resilient and stable towards a multitude of variations occurring naturally in the environment and from within the dynamics of the system itself. However, an ecological system is also constantly confronted with the unexpected, such as extreme influences from outside or species/population behaviour deviating from the norm [77]. The origin of such perturbations can be hidden in the true interconnectedness of the system and sometimes reveals itself and leads to seemingly unpredictable behaviour: Changing the numbers of one species can influence unexpected others [74]; disturbing one interaction can impact the resilience mechanisms of a whole ecosystem [22]; a system can transit between different stable states [13], or permanently shift the state [128]. The most apparent causes for such behaviour are rapid climate change and aggressive invasive species. In addition, human actions such

as modern agriculture and fossil fuel combustion have also shown to substantially impact biodiversity and ecosystem stability [45, 144]. In order to identify the causes and effects for certain behaviours on these scales requires an understanding of the ecological networks underlying the system [36].

Inferring the interactions in complex ecosystems is not a straightforward task to accomplish. Direct observation requires minute observations and detailed fieldwork, and is capable of measuring only certain types of species interactions, like between predators and their prey, or between pollinators and their host plants. The majority of interactions are not directly observable. This restriction calls for the development of novel computational inference techniques to learn networks of species interactions directly from observed species concentrations. The challenges for computational inference specific to ecological systems are that, first, the interactions take place in a spatially explicit environment which must be taken into account, and second, the interactions can vary across this environment depending on the make-up of the elements (species and abiotic factors) present.

In this thesis I will meet these challenges by showing the necessary modifications to an inference method from systems biology [89] for temporally explicit (1-dimensional) gene expression data to infer ecological interactions from spatially explicit species abundance data on a 2-dimensional grid. I describe a non-homogeneous Bayesian regression model based on the Bayesian hierarchical regression model of Andrieu and Doucet [7], using a global multiple change-point process as introduced by Aderhold et al. [2]. I further modify the method with a Mondrian process that implements a spatial partitioning at different levels of resolution as introduced by Aderhold et al. [3]. I make further use of the spatially explicit nature of ecological data by correcting for spatial autocorrelation with a regulator node (in Bayesian network terminology) that explicitly represents the spatial neighbourhood of a node. The performance of these methods is demonstrated on synthetic and realistic simulated data, and I infer a network from a real world data set. The results show that ecological modelling could benefit from these types of methods, and that the required modifications do not conflict with, but extend the basic methodology used in systems biology.

1.4 Dissertation Overview

In this thesis I will demonstrate two extensive studies that involve various established methods from systems biology and two modifications to infer ecological networks. The

structure is in the following form: Chapter 2 introduces the mathematical notation and various state-of-the art machine learning and statistical inference methods. Chapter 3 describes the details of sparse regression with convex optimization and MCMC inference that constitute important techniques for network inference. The chapter concludes by presenting the evaluation methods used throughout the rest of the thesis involving scoring metrics to measure reconstruction accuracy. Chapter 4 demonstrates an extensive study of network learning accuracy of the methods presented in Chapter 2 with application to a genetic regulatory system called the circadian clock of the plant *Arabidopsis thaliana* (thale cress). This study presents results on simulated data and real-time polymerase chain reaction (qRT-PCR) gene expression data, and also proposes method modifications that take into account time dependent changes of the gene regulation. Chapter 5 describes the method modifications that allow me to apply the previously defined methods to an ecological problem setting. This is realised with the expansion of the data domain from 1-dimensional time to 2 dimensions of space. In addition, methods that can learn the spatial segmentation on a global scale and local scale using the Mondrian process are described. Finally, the findings of the previous two chapters and future work is presented in the Conclusion Chapter 6.

Chapter 2

Methods

This chapter introduces the notation used throughout this thesis together with a description of various methods that we will apply to infer gene regulatory networks in Chapter 4 and ecological species networks in Chapter 5. Note that the focus of the study in Chapter 4 is a broad comparison of all the presented methods under consideration of the special requirements of a molecular network setting. The only method that it not part of the evaluation in Chapter 4 is the Banjo (Section 2.14), which we unfortunately missed to consider at the time of the study. However, it is part of the ecological study in Chapter 5.

One of the main objectives of this thesis is to demonstrate the exploitation of common principles throughout multiple scales. A common principle in molecular and ecological systems can be found in the time-varying nature of genetic regulation and space-varying nature of species neighbourhoods or niches. For instance, the phenotypical changes of an insect with the distinct phases of an embryo, larva, pupa, and adult [8] are matched by genotypical changes in expression profiles and interconnectedness, e.g. of regulatory factors. In ecology, variations of population densities, growth rates, and species behaviour are matched by changes in the environment, species interaction patterns, and varying genetic traits in different populations.

To reflect these changes in a model, we require an approach that can handle *non-homogeneous* data that varies in time or space. In contrast to *homogeneous* approaches where data is treated as a single monolithic block, the non-homogeneous approach or model allows the partitioning/segmentation of the data to account for different phases and associated changes in the interaction networks and network parameters. The non-homogeneous hierarchical Bayesian (HBR) model described in Section 2.3

was previously applied to gene expression data [60] and showed the best performance in the molecular study of Chapter 4. I adapted this framework to the 2-dimensional spatial domain and propose two different segmentation procedures in Section 5.2.4 and Section 5.2.5. For the purpose of comparison, we included the well established sparse regression method Lasso (Section 2.4) and the previously mentioned Banjo. Hence, the goal of the ecological study is not to test the performance of all the methods defined in this Chapter, but focus on the modifications to the HBR method that lead to the spatially sensitive partitioning methods called BRAM and BRAMP.

An overview of the methods is presented in Table 2.1. It also lists the methods that are particularly modified to a gene regulation and ecological scenario, but are defined only in the corresponding Chapter 4 and Chapter 5. This includes the HBR modifications HBR-light, HBR-cps, and HBR-nl for gene regulation and BRAM, and BRAMP for ecology. Furthermore, some of the methods below lack the differentiation of response and predictor data as it is the convention throughout this thesis. Since, the affected methods are only applied to the gene regulation we will frequently refer to Equation (4.1) that defines this differentiation in the context of gene regulation[1]. I outline these necessary modifications in this chapter in order to avoid a fragmentation of method descriptions in the thesis.

Finally, note that the majority of the content in this Chapter was published by Aderhold et al. [1] in collaboration with Marco Grzegorczyk.

2.1 Notation

For the models that I will use to infer the network interactions, I have target variables y_n $(n = 1, \ldots, N)$, each representing the mRNA time derivative (gradient) of a particular gene n or the density of a species n in a particular location given a ecological setting. To generalize this representation I follow a graph theory terminology and call a gene or species a "node". The realizations of each target variable y_n can then be written as a vector $\mathbf{y}_n = (y_{n,1}, \ldots, y_{n,M})^\mathsf{T}$, where $y_{n,m}$ is the realization (or observation) of y_n at data point m. Whenever I consider sets of time series in a gene regulation setting I refer to the index m as time point and data point synonymously, in particular I also

[1] The convention is to always use a static approach without 'time-shift', i.e. variables at one time point (predictor) affect the variables at the same time point (target). This seems to contradict the dynamic nature of some of the methods defined in this chapter. However, by using a time derivative and simulating a time-shift we can adapt these methods to work with the convention of Equation (4.1). Refer to Section 4.2 for a detailed discussion of the regulation mechanism.

Abbreviation	Full Name, section, reference
HBR	Homogeneous hierarchical Bayesian regression, Section 2.2, Grzegorczyk and Husmeier [60]
nh-HBR	Non-homogeneous hierarchical Bayesian regression, Section 2.3, Grzegorczyk and Husmeier [60]
HBR-light	nh-HBR with time-varying light dependent change-points, Section 4.3.1,
HBR-cps	nh-HBR with change-points on the response gradient, Section 4.3.2,
HBR-nl	HBR and nh-HBR with additional non-linear terms, Section 4.3.3, (HBR-light,-cp, -nl published by Aderhold et al. [1])
BRAM	Bayesian regression and multiple global change-points, Section 5.2.4, Aderhold et al. [2]
BRAMP	Bayesian regression and Mondrian process change-points, Section 5.2.5, Aderhold et al. [3]
Lasso, Elastic Net	Sparse regression, Section 2.4, Tibshirani [136]
Tesla	Non-homogeneous sparse regression, Section 2.5, Ahmed and Xing [5]
ARD-SBR	Automatic Relevance determination (Sparse Bayesian regression), Section 2.6, Tipping [139]
GGM	Graphical Gaussian models, Section 2.7, Schäfer and Strimmer [127]
BSA	Bayesian spline autoregression, Section 2.8, Morrissey et al. [106]
SSM	State-space models, Section 2.9, Beal et al. [12]
GP	Gaussian processes, Section 2.10, Rasmussen and Williams [119]
ARACNE	Mutual information measure with pruning, Section 2.11, Margolin et al. [97]
MBN	Mixture Bayesian networks, Section 2.12, Ko et al. [84]
BGe	Gaussian Bayesian networks, Section 2.13, Geiger and Heckerman [49]
Banjo	Bayesian Inference with Java objects (Dynamic Bayesian networks with BDe), Section 2.14, Hartemink [67]

Table 2.1: List of all models included in this thesis.

say that $y_{n,m}$ is the observation of y_n at time index m. In the case of ecological data, m becomes a sample location in a two dimensional grid that maps into a coordinate system with (x, y).

For node n there are N_n *potential* regulators, $x_1^n, \ldots, x_{N_n}^n$, which are either gene, protein concentrations, or species densities.[2] The task is to infer a set of regulators $\boldsymbol{\pi}_n$ with $\boldsymbol{\pi}_n \subset \{x_1^n, \ldots, x_{N_n}^n\}$ for each target variable y_n. The collection of regulators $\{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N\}$ can then be thought of as a regulatory interaction graph, $\mathcal{G}$. In $\mathcal{G}$ the regulators and the target variables represent the nodes and from each regulator in $\boldsymbol{\pi}_n$ a directed edge is pointing to the target node y_n. Hence, in terms of graphical models the graph $\mathcal{G}$ possesses a bipartite structure, where the *potential* regulators $x_1^n, \ldots, x_{N_n}^n$ are the potential parent nodes of the target variable y_n $(n = 1, \ldots, N)$, and there is a directed edge from x_i^n to y_n in $\mathcal{G}$, symbolically $x_i^n \rightarrow y_n$, if $x_i^n \in \boldsymbol{\pi}_n$. In regression

[2] Note that the sets of potential regulators are defined for each node n specifically. That is, the potential regulators for two target variables y_n and $y_{n'}$ can be different, e.g. if certain (biologically-motivated) restrictions are imposed.

models the regulators are usually referred to as covariates, and throughout the thesis I therefore use the terms regulator(s), parent node(s) and covariate(s) interchangeably.

In regression models the observations of all the *potential* covariates of the target y_n can be collected in a design matrix $\mathbf{X}_n$ such that each row of $\mathbf{X}_n$ corresponds to a covariate and contains all M observations of that particular covariate. An additional row with constant elements equal to 1 is added to $\mathbf{X}_n$ to take the intercept into account. In addition, for a fixed subset of covariates, $\boldsymbol{\pi}_n$, I define $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$ to be the sub-matrix of the full design matrix, $\mathbf{X}_n$, where all rows that belong to covariates which are not in $\boldsymbol{\pi}_n$ have been deleted. To paraphrase that, in the restricted design matrix $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$ I keep only those rows of $\mathbf{X}_n$ that correspond either to the intercept or to the covariates in the set $\boldsymbol{\pi}_n$.

For non-regression models I additionally define two vectors. For $m = 1, \ldots, M$ let $\mathbf{x}_{n,m} := (x^n_{1,m}, \ldots, x^n_{N_n,m})^\mathsf{T}$ denote the vector of the concentrations/ densities of all N_n *potential* regulators for node n at the observation m, which corresponds to a column entry in the design matrix $\mathbf{X}_n$. Let $\mathbf{z}_{n,m} := (y_{n,m}, \mathbf{x}^\mathsf{T}_{n,m})^\mathsf{T}$ extend the vector $\mathbf{x}_{n,m}$ by including the value of the response $y_{n,m}$, i.e. the derivative of the concentration of the target node n at observation m ($m = 1, \ldots, M$). In addition, for a fixed subset of regulators, $\boldsymbol{\pi}_n$, I define $\mathbf{x}_{\boldsymbol{\pi}_n,m}$ and $\mathbf{z}_{\boldsymbol{\pi}_n,m}$ to be the corresponding sub-vectors of $\mathbf{x}_{n,m}$ and $\mathbf{z}_{n,m}$, respectively, where all elements that do not correspond to regulators in $\boldsymbol{\pi}_n$ have been deleted.

Finally, denote by $\mathbf{X}^\star_n$ and $\mathbf{X}^\star_{n[\boldsymbol{\pi}_n]}$ the sub-matrices of the design matrices $\mathbf{X}_n$ and $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$ in which the constant row for the intercept has been removed. For the state-space models (SSMs), described in Subsection 2.9, I define $\mathbf{x}_{.,m}$ as the vector of the observations of *all* potential regulators at observation m.[3] A complete overview of the notation is given in Table 2.2.

2.2 Hierarchical Bayesian regression models (HBR)

What follows is a brief definition of the hierarchical Bayesian regression (HBR) framework, which provides the basis for several Bayesian regression models in this thesis (see Table 2.1 for the overview). In particular the description of the BRAM and BRAMP that descend from the HBR delve into more methodological detail.

In the HBR approach I assume a linear regression model for the target vectors $\mathbf{y}_n$

[3] Note that vector $\mathbf{x}_{.,m}$ includes every available regulator without any dependency on the target node n.

Symbol	Short verbal description
n	target/ response node n ($n = 1, \ldots, N$)
m	sample/ observation at time or location m ($m = 1, \ldots, M$)
x_n	variable measuring the value of the node n
$x_{n,m}$	variable x_n of sample m
y_n	target (response) variable, gradient corresponding to target node n
$y_{n,m}$	target (response) variable y_n at sample m,
	derivative of x_n at sample m ($m = 1, \ldots, M$)
$\mathbf{y}_{\cdot,m}$	vector of all target variables (gradients) at sample m
	$\mathbf{y}_{\cdot,m} := (y_{1,m}, \ldots, y_{N,m})^\mathsf{T}$
$\mathbf{y}_n$	vector of all M samples for the target gene y_m
	$\mathbf{y}_n := (y_{n,1}, \ldots, y_{n,M})^\mathsf{T}$
N_n	the number of potential regulators for target node n
x_i^n	the i-th regulator for target node n ($n = 1, \ldots, N_n$)
$x_{i,m}^n$	the observation for the i-th regulator for target node n at sample m
$\boldsymbol{\pi}_n$	particular set of regulators (covariates, parent nodes) for target node n
	$\boldsymbol{\pi}_n \subset \{x_1^n, \ldots, x_{N_n}^n\}$
$\mathcal{G}$	the bipartite graph structure $\mathcal{G} = \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N\}$
$\mathbf{X}_n$	full design matrix for node n including all N_n potential regulators for n
$\mathbf{X}_{n[\boldsymbol{\pi}_n]}$	restricted design matrix for node n, $\mathbf{X}_n$ restricted to regulators in the set $\boldsymbol{\pi}_n$
$\mathbf{x}_{n,m}$	vector of samples at m for all N_n regulators of node n
	$\mathbf{x}_{n,m} := (x_{1,m}^n, \ldots, x_{N_n,m}^n)^\mathsf{T}$
$\mathbf{z}_{n,m}$	response variable for node n and concentrations of all its N potential
	regulators at sample m, $\mathbf{z}_{n,m} := (y_{n,m}, \mathbf{x}_{n,m}^\mathsf{T})^\mathsf{T}$
$\mathbf{x}_{\boldsymbol{\pi}_n,m}$	vector of samples at m for the $\lvert\boldsymbol{\pi}_n\rvert$ regulators in $\boldsymbol{\pi}_n$
$\mathbf{z}_{\boldsymbol{\pi}_n,m}$	response variable for node n and concentrations of its regulators in the
	set $\boldsymbol{\pi}_n$ at sample m, $\mathbf{z}_{\boldsymbol{\pi}_n,m} := (y_{n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m}^\mathsf{T})^\mathsf{T}$
$\mathbf{X}_n^\star$	the matrix (or set) of all M samples for the N_n potential regulators of n
	similar to the full design matrix $\mathbf{X}_n$, but without the row for the intercept
$\mathbf{X}_{n[\boldsymbol{\pi}_n]}^\star$	the matrix (or set) of all M samples for the regulators in $\boldsymbol{\pi}_n$
	similar to the restricted design matrix $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$, but without the intercept row
$\mathbf{x}_{\cdot,m}$	vector of samples at m of *all* potential regulators, i.e.
	this vector includes every available regulator, and it is not target-specific
$\mathcal{D}$	complete data set including all observations $\mathcal{D} = \{\mathbf{y}_{\cdot,m}, \mathbf{x}_{\cdot,m}\}_{\forall m}$ or
	$\mathcal{D} = \{\mathbf{x}_{\cdot,m}\}_{\forall m}$ for when the response is part of the regulation set
$\boldsymbol{\theta}$	vector of model parameters

Table 2.2: Overview of symbols, introduced in Section 2.1. These notations are used throughout the thesis. The methods for the ecological application expand the notation as listed in Table 5.1 and 5.2. For more detailed descriptions see main text in Section 2.1.

with values distributed according to a multivariate Gaussian distribution $\mathcal{N}$ with mean $(\mathbf{X}^{\mathsf{T}}_{n[\boldsymbol{\pi}_n]}\mathbf{w}_n)$ and covariance matrix $(\sigma_n^2\mathbf{I})$:

$$\mathbf{y}_n|(\mathbf{w}_n, \sigma_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]}) \sim \mathcal{N}(\mathbf{X}^{\mathsf{T}}_{n[\boldsymbol{\pi}_n]}\mathbf{w}_n, \sigma_n^2\mathbf{I}) \tag{2.1}$$

where $\mathbf{w}_n$ is the vector of regression parameters, $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$ is the restricted design matrix whose rows correspond to the variables in the covariate set $\boldsymbol{\pi}_n$ with an additional constant row for the intercept, and σ_n^2 is the noise variance. I impose a Gaussian prior on the regression parameter vector:

$$\mathbf{w}_n|(\sigma_n, \delta_n) \sim \mathcal{N}(\mathbf{0}, \delta_n\sigma_n^2\mathbf{I}) \tag{2.2}$$

The hyper-parameter δ_n can be interpreted as the 'signal-to-noise' (SNR) ratio [60]. For the posterior distribution I get, e.g. as described in Section 3.3 of [16]:

$$\mathbf{w}_n|(\sigma_n, \delta_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]}, \mathbf{y}_n) \sim \mathcal{N}(\boldsymbol{\Sigma}_n\mathbf{X}_{n[\boldsymbol{\pi}_n]}\mathbf{y}_n, \sigma_n^2\boldsymbol{\Sigma}_n) \tag{2.3}$$

where $\boldsymbol{\Sigma}_n^{-1} = \delta_n^{-1}\mathbf{I} + \mathbf{X}_{n[\boldsymbol{\pi}_n]}\mathbf{X}^{\mathsf{T}}_{n[\boldsymbol{\pi}_n]}$. The marginal likelihood, $p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \sigma_n^2, \delta_n)$, can be obtained by application of standard results for Gaussian integrals [e.g. 16, Appendix B]:

$$\begin{aligned} p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \sigma_n^2, \delta_n) &= \int p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \sigma_n^2, \mathbf{w}_n)p(\mathbf{w}_n|\sigma_n^2, \delta_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]})d\mathbf{w}_n \\ &= \mathcal{N}(\mathbf{y}_n|\mathbf{0}, \sigma_n^2(\mathbf{I} + \delta_n\mathbf{X}^{\mathsf{T}}_{n[\boldsymbol{\pi}_n]}\mathbf{X}_{n[\boldsymbol{\pi}_n]})) \end{aligned} \tag{2.4}$$

For σ_n^{-2} and δ_n^{-2} I choose conjugate gamma priors, $\sigma_n^{-2} \sim \text{Gam}(\nu, \nu)$, and $\delta_n^{-1} \sim \text{Gam}(A_\delta, B_\delta)$.[4] The integral resulting from the marginalization over σ_n^{-2},

$$p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \delta_n) = \int_0^\infty p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \sigma_n^2, \delta_n)p(\sigma_n^{-2}|\nu)d\sigma_n^{-2} \tag{2.5}$$

is a multivariate Student t-distribution with a closed-from solution [e.g. 16, 60].

Given the data for all the potential regulators of y_n, i.e. given the full design matrix $\mathbf{X}_n$, the objective is to infer the set of covariates $\boldsymbol{\pi}_n$ from the marginal posterior distribution:

[4] I set: $\nu = 0.005$, $A_\delta = 2$, and $B_\delta = 0.2$ for the gene regulation and ecological application, as from Grzegorczyk and Husmeier [60].

$$P(\boldsymbol{\pi}_n|\mathbf{X}_n, \mathbf{y}_n, \delta_n) = \frac{P(\boldsymbol{\pi}_n)p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \delta_n)}{\sum_{\boldsymbol{\pi}_n^\star} P(\boldsymbol{\pi}_n^\star)p(\mathbf{y}_n|\mathbf{X}^\star_{n[\boldsymbol{\pi}_n]}, \delta_n)} \propto P(\boldsymbol{\pi}_n)p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \delta_n) \tag{2.6}$$

where the sum in the denominator is over all valid covariate sets, $\boldsymbol{\pi}_n^\star$, $P(\boldsymbol{\pi}_n)$ is a uniform distribution over all covariate sets subject to a maximal cardinality, typically $|\boldsymbol{\pi}_n| \leq 3$ or $|\boldsymbol{\pi}_n| \leq 5$. I sample sets of covariates (or regulators) $\boldsymbol{\pi}_n$, signal-to-noise hyper-parameters δ_n, and noise variances σ_n^2 from the joint posterior distribution with Markov chain Monte Carlo (MCMC), following the Metropolis-Hastings within partially collapsed Gibbs scheme from Grzegorczyk and Husmeier [60]. Within that scheme, I sample covariate sets $\boldsymbol{\pi}_n$ from Equation (2.6) with Metropolis-Hastings, using the proposal mechanism from Grzegorczyk and Husmeier [60]: given the current covariate set $\boldsymbol{\pi}_n$, randomly propose a new covariate set from the system of all covariate sets such that it can be reached (i) either by removing a single covariate from $\boldsymbol{\pi}_n$, (ii) or by adding a single covariate to $\boldsymbol{\pi}_n$, (iii) or by a covariate flip move. The (hyper-)parameters δ_n^{-1}, $\mathbf{w}_n$, and σ_n^{-2} can be sampled with Gibbs sampling steps. As shown by Grzegorczyk and Husmeier [60], the full conditional distributions of δ_n^{-1} and $\mathbf{w}_n$ are given by:

$$\delta_n^{-1}|(\mathbf{w}_n, \sigma_n^2) \sim \text{Gam}\left(A_\delta + \frac{|\boldsymbol{\pi}_n|+1}{2}, B_\delta + \frac{1}{2\sigma_n^2}\mathbf{w}_n^\mathsf{T}\mathbf{w}_n\right) \tag{2.7}$$

$$\mathbf{w}_n|(\mathbf{y}_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]}, \sigma_n^2, \delta_n) \sim \mathcal{N}(\boldsymbol{\Sigma}_n^\star \mathbf{X}_{n[\boldsymbol{\pi}_n]}\mathbf{y}_n, \sigma_n^2\boldsymbol{\Sigma}_n^\star) \tag{2.8}$$

where $|\boldsymbol{\pi}_n|$ is the cardinality of the parent set, $\boldsymbol{\pi}_n$, and $\boldsymbol{\Sigma}_n^\star = \left(\delta_n^{-1}\mathbf{I} + \mathbf{X}_{n[\boldsymbol{\pi}_n]}\mathbf{X}_{n[\boldsymbol{\pi}_n]}^\mathsf{T}\right)^{-1}$. The inverse variance hyper-parameters, σ_n^{-2} can be sampled with a collapsed Gibbs sampling step, in which the regression parameter vectors, $\mathbf{w}_n$, have been integrated out. This marginalization yields [e.g. 60]:

$$\sigma_n^{-2}|(\mathbf{y}_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]}, \delta_n) \sim \text{Gam}\left(\nu + \frac{M}{2}, \nu + \frac{\mathbf{y}_n^\mathsf{T}\left(\mathbf{I} + \delta_n\mathbf{X}_{n[\boldsymbol{\pi}_n]}^\mathsf{T}\mathbf{X}_{n[\boldsymbol{\pi}_n]}\right)^{-1}\mathbf{y}_n}{2}\right) \tag{2.9}$$

where M is the number of samples or observations. A compact representation of the relationships among the (hyper-) parameters of the Bayesian regression model is given in the top panel of Figure 2.1.

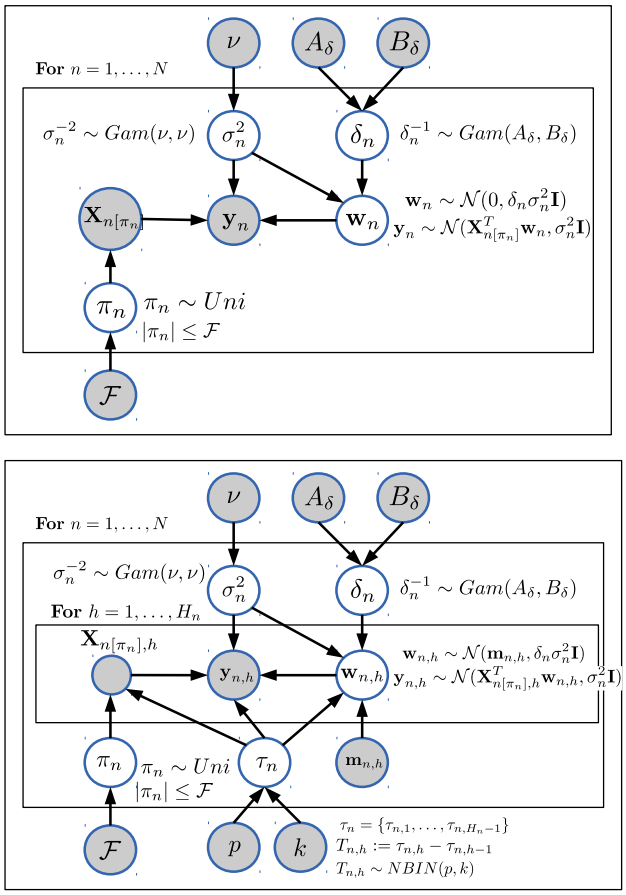


Figure 2.1: Representation of the hierarchical Bayesian regression models as graphical models. In both panels the grey circles refer to fixed hyper-parameters, while the white circles refer to flexible (hyper-)parameters, which are inferred from the posterior distribution with MCMC. *Top panel:* The homogeneous Bayesian regression model. The outer plate includes the complete model, and the centre plate refers to the target variables, $n = 1, \ldots, N$; see Subsection 2.2 for a detailed model description. *Bottom panel:* The uncoupled variant of the non-homogeneous Bayesian regression model. Variable-specific change-point sets, $\boldsymbol{\tau}_n$, divide the data into variable-specific segments. The additional inner plate refers to the variable-specific segments, $h = 1, \ldots, H_n$; see Sections 2.3 for more details. In the coupled variant of the non-homogeneous Bayesian regression model (not shown in this figure) the regression parameter vectors, $\mathbf{w}_{n,h}$ $(h = 1, \ldots, H_n)$, are sequentially coupled via Equations (2.11-2.12).

2.3 Non-homogeneous hierarchical Bayesian models

The non-homogeneous case of the HBR described in this Section is based on the assumption that the data is heterogeneous at a single (time) scale[5]. Such data is produced by continuously measuring gene expression profiles in certain time intervals. In the application to genetic data of plants in Chapter 4, e.g., the underlying regulatory relationships are non-linear and vary most evidently in dependence on the external light condition. Chapter 5 will present the expansion of the HBR to a spatial scale.

I follow Grzegorczyk and Husmeier [60] and combine the Bayesian regression model from Subsection 2.2 with a multiple change-point process. The change-point process imposes a set of $H_n - 1$ change-points, $\{\tau_{n,h}\}_{1 \leq h \leq (H_n-1)}$ with $\tau_{n,h} < \tau_{n,h+1}$, to divide the temporal observations of a variable into H_n disjunct segments. With the two pseudo-change-points $\tau_{n,0} := 1$ and $\tau_{n,H_n} := M$ each segment $h \in \{1, \ldots, H_n\}$ is defined by two demarcating change-points, $\tau_{n,h}$ and $\tau_{n,h+1}$. The vector of the target variable realizations, $\mathbf{y}_n = (y_{n,1}, \ldots, y_{n,M})^{\mathsf{T}}$, is thus divided into H_n sub-vectors, $\{\mathbf{y}_{n,h}\}_{h=1,\ldots,H_n}$, where each sub-vector corresponds to a temporal segment: $\mathbf{y}_{n,h} = (y_{n,(\tau_{n,h}+1)}, \ldots, y_{n,\tau_{n,h+1}})^{\mathsf{T}}$. In Grzegorczyk and Husmeier [60] the distances between two successive change-points, $M_{n,h} = \tau_{n,h+1} - \tau_{n,h}$, are assumed to have a negative binomial distribution, symbolically $M_{n,h} \sim NBIN(p, k)$; see Grzegorczyk and Husmeier [60] for the technical details and see Section 4.3.2 for my slightly different implementation of the response gradient $\mathbf{y}$.

I keep the covariate set, $\boldsymbol{\pi}_n$, fixed among the H_n segments, and I apply the linear Gaussian regression model, defined in Equation (2.1), to each segment h:

$$\mathbf{y}_{n,h} | (\mathbf{X}_{n[\boldsymbol{\pi}_n],h}, \mathbf{w}_{n,h}, \sigma_n^2) \sim \mathcal{N}(\mathbf{X}_{n[\boldsymbol{\pi}_n],h}^{\mathsf{T}} \mathbf{w}_{n,h}, \sigma_n^2 \mathbf{I}) \tag{2.10}$$

where $\mathbf{X}_{n[\boldsymbol{\pi}_n],h}$ is the segment-specific (restricted) design matrix, which can be built from the realizations of the covariate set $\boldsymbol{\pi}_n$ in segment h, and $\mathbf{w}_{n,h}$ is the vector of the segment-specific regression parameters for segment h. As in Section 2.2 I impose an inverse Gamma prior on σ_n^2, symbolically $\sigma_n^{-2} \sim \text{Gam}(\nu, \nu)$. For the segment-specific regression parameters, $\mathbf{w}_{n,h}$ $(h = 1, \ldots, H_n)$, I assume Gaussian priors:

$$\mathbf{w}_{n,h} | (\mathbf{m}_{n,h-1}, \sigma_n, \delta_n, \mathbf{X}_{\boldsymbol{\pi}_n,h}) \sim \mathcal{N}(\mathbf{m}_{n,h-1}, \delta_n \sigma_n^2 \mathbf{I}) \tag{2.11}$$

[5] Note that the change-point process is not limited to the (time) scale of observations but can partition the amplitude of a response in order to approximate intrinsic non-linearities as shown in Section 4.3.2 for the response variable.

with the hyper-prior $\delta_n^{-1} \sim \text{Gam}(A_\delta, B_\delta)$. As with Grzegorczyk and Husmeier [60] I distinguish two variants of the non-homogeneous Bayesian regression model. In the *uncoupled variant* I set $\mathbf{m}_{n,h} = \mathbf{0}$ for all $h \geq 0$. In the *sequentially coupled variant* I allow for information-sharing between the regression parameters of adjacent segments by setting $\mathbf{m}_{n,0} = \mathbf{0}$, and for $h \geq 1$:

$$\mathbf{m}_{n,h} = \mathbf{\Sigma}_{n,h}(\delta_n^{-1}\mathbf{m}_{n,(h-1)} + \mathbf{X}_{\boldsymbol{\pi}_n,h}\mathbf{y}_{n,h}). \tag{2.12}$$

with $\mathbf{\Sigma}_{n,h}^{-1} = \delta_n^{-1}\mathbf{I} + \mathbf{X}_{\boldsymbol{\pi}_n,h}\mathbf{X}_{n[\boldsymbol{\pi}_n],h}^{\mathsf{T}}$. As in the previous section, posterior inference is carried out with the Metropolis-Hastings within partially collapsed Gibbs sampling scheme from Grzegorczyk and Husmeier [60]. The marginal likelihood in Equation (2.6) has to be replaced by:

$$P(\boldsymbol{\pi}_n|\mathbf{X}_n, \delta_n, \{\tau_{n,h}\}_{1\leq h\leq (H_n-1)}) \propto P(\boldsymbol{\pi}_n)\prod_{h=1}^{H_n} p(\mathbf{y}_{n,h}|\mathbf{X}_{\boldsymbol{\pi}_n,h}, \delta_n) \tag{2.13}$$

where $p(\mathbf{y}_{n,h}|\mathbf{X}_{\boldsymbol{\pi}_n,h}, \delta_n)$ $(h = 1, \ldots, H_n)$ can be computed in closed-form; see Grzegorczyk and Husmeier [60] for a mathematical derivation. The full conditional distribution of $\mathbf{w}_{n,h}$ is now given by [60]:

$$\mathbf{w}_{n,h}|(\mathbf{y}_{n,h}, \mathbf{X}_{\boldsymbol{\pi}_n,h}, \sigma_n^2, \delta_n) \sim \mathcal{N}(\tilde{\mathbf{m}}_{n,h}, \sigma_n^2\mathbf{\Sigma}_{n,h}) \tag{2.14}$$

where $\mathbf{\Sigma}_{n,h}$ was defined below Equation (2.12). For the uncoupled variant of the model I have: $\tilde{\mathbf{m}}_{n,h} = \mathbf{\Sigma}_{n,h}\mathbf{X}_{\boldsymbol{\pi}_n,h}\mathbf{y}_{n,h}$. For the coupled variant of the model I have: $\tilde{\mathbf{m}}_{n,h} := \mathbf{m}_{n,h}$, where $\mathbf{m}_{n,h}$ was defined in Equation (2.12). The full conditional distribution of δ_n^{-1}, symbolically $p(\delta_n^{-1}|\sigma_n^2, \{\mathbf{w}_{n,h}\}_{h=1,\ldots,H_n})$, is a Gamma distribution whose closed-form solution can be found in [60]. The inverse variance hyper-parameters, σ_n^{-2}, can again be sampled with a collapsed Gibbs sampling step [60]:

$$\sigma_n^{-2}|(\mathbf{y}_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]}, \delta_n, \{\tau_{n,h}\}_{1\leq h\leq (H_n-1)}) \sim \text{Gam}\left(\nu + \frac{M}{2}, \nu + \frac{\sum_{h=1}^{H_n}\Delta_{n,h}^2}{2}\right) \tag{2.15}$$

with $\Delta_{n,h}^2 := (\mathbf{y}_{n,h} - \mathbf{X}_{n[\boldsymbol{\pi}_n],h}\mathbf{m}_{n,h-1})^{\mathsf{T}}\left(\mathbf{I} + \delta_n\mathbf{X}_{n[\boldsymbol{\pi}_n],h}^{\mathsf{T}}\mathbf{X}_{n[\boldsymbol{\pi}_n],h}\right)^{-1}(\mathbf{y}_{n,h} - \mathbf{X}_{n[\boldsymbol{\pi}_n],h}\mathbf{m}_{n,h-1})$, where $\mathbf{m}_{n,h-1}$ can be computed with Equation (2.12) in the coupled variant, and $\mathbf{m}_{n,h-1} = \mathbf{0}$ for all $h \geq 0$ in the uncoupled variant. A compact graphical representation of the relationships among the (hyper-)parameters of the uncoupled variant of

the non-homogeneous Bayesian regression model can be found in the bottom panel of Figure 2.1. Note that the coupled variant of the non-homogeneous Bayesian regression model cannot be represented properly as a graphical model, as the regression parameter vectors are *sequentially* coupled among adjacent segments via Equations (2.11-2.12).

Combining the linear regression model with a change-point process provides a natural mechanism to allow for temporal (longitudinal) relationships in the data. However, the data in the gene regulation study are a mixture of short time series from several independent experiments, where the overall temporal factor influencing the system is the light phase. In addition, I aim to draw on the change-point process as a mechanism to approximate the intrinsic non-linearities of the Michaelis-Menten kinetics via a piecewise linear model. I therefore treat the data as independent interchangeable realizations and regroup them prior to the application of the change-point process, as explained in Sections 4.3.1 and 4.3.2.

2.4 Sparse regression (Lasso and Elastic Net)

An efficient and widely applied linear regression method that provides network sparsity is the Least Absolute Shrinkage and Selection Operator (Lasso) introduced by Tibshirani [136]. The Lasso optimizes the regression parameters $\mathbf{w}_n$ of a linear model based on the residual sum of squares subject to an ℓ_1-norm penalty term, $\lambda_1 \|\mathbf{w}_n\|_1$, where λ_1 is a regularization parameter, and $\|\mathbf{w}_n\|_1$ is the sum of the absolute values of the components of $\mathbf{w}_n$:

$$\hat{\mathbf{w}}_n = \operatorname{argmin}\left\{ ||\mathbf{y}_n - \mathbf{X}_n^{\mathsf{T}}\mathbf{w}_n||_2^2 + \lambda_1 \|\mathbf{w}_n\|_1 \right\} \tag{2.16}$$

For definitions of the full design matrix $\mathbf{X}_n$ and the target gradient vector $\mathbf{y}_n$ see Table 2.2. Equation (2.16) is a convex optimization problem, for which a variety of fast and effective algorithms exist (see Section 3.1.5, and e.g. Hastie et al. [70]). The effect of Equation (2.16) is to simultaneously shrink and prune the parameters in $\mathbf{w}_n$, thereby promoting a sparse network. The degree of sparsity depends on the regularization parameter λ_1, which can be optimized with cross-validation or information criteria, like BIC (see Section 3.1.4).

The shortcomings are that the Lasso will only select one predictor from a set of highly correlated variables, and that it can maximally select M variables, thereby potentially suffering from saturation effects. These difficulties are addressed with the Elastic Net method, proposed by Zou and Hastie [157], which combines the Lasso penalty with a

ridge regression penalty that constitutes a squared ℓ_2-norm $\|\mathbf{w}_n\|_2^2$:

$$\hat{\mathbf{w}}_n = \text{argmin}\left\{ ||\mathbf{y}_n - \mathbf{X}_n^\mathsf{T}\mathbf{w}_n||_2^2 + \lambda_1\|\mathbf{w}_n\|_1 + \lambda_2\|\mathbf{w}_n\|_2^2 \right\} \tag{2.17}$$

Again, this is a convex optimization problem for which effective algorithms exist, and the regularization parameters λ_1 and λ_2 can be optimized with cross-validation or BIC. For these two approaches (Lasso and Elastic Net) I typically use the absolute values of the elements of the estimated regression parameter vectors $\hat{\mathbf{w}}_n$ to score the regulatory effects on y_n ($n = 1, \ldots, N$) with respect to their strengths. Refer to Section 3.1 for a more detailed discussion of sparse regression.

2.5 Non-homogeneous sparse regression (Tesla)

Ahmed and Xing [5] proposed a non-homogeneous generalization of sparse regression called Tesla. The idea is to divide a time series or any other set of observations into segments and perform sparse regression for each segment separately, subject to an additional sparsity constraint that penalizes differences between regression parameters associated with adjacent segments. Consider a set of observation values (such as a time series of gene expression data) for node n, which is divided into H_n disjunct segments, marked by $H_n + 1$ demarcation points $1 = \tau_{n,1} \leq \ldots \leq \tau_{n,h} \leq \ldots \leq \tau_{(H_n+1)} = M$. Each segment is associated with a different set of regression parameters, $\mathbf{w}_{n,h}$, where $h \in \{1, ..., H_n\}$ is a label that identifies the segment. To prevent over-complexity and avoid overfitting, an additional ℓ_1-norm penalty is imposed on the parameter differences for adjacent segments, i.e. $\mathbf{w}_{n,h} - \mathbf{w}_{n,h-1}$ for $h > 1$:

$$\hat{\mathbf{w}}_{n,1}, ..., \hat{\mathbf{w}}_{n,H_n} = \text{argmin}\left\{ \sum_{h=1}^{H_n} ||\mathbf{y}_{n,h} - \mathbf{X}_{n,h}^\mathsf{T}\mathbf{w}_{n,h}||_2^2 + \lambda_1 \sum_{h=1}^{H_n} \|\mathbf{w}_n\|_1 + \lambda_2 \sum_{h=2}^{H_n} \|\mathbf{w}_{n,h} - \mathbf{w}_{n,h-1}\|_1 \right\} \tag{2.18}$$

where $\mathbf{y}_{n,h} = (y_{n,(\tau_{n,h}+1)}, \ldots, y_{n,\tau_{n,h+1}})^\mathsf{T}$ is the sub-vector of observations in the temporal segment h, and $\mathbf{X}_{n,h}$ is the corresponding segment specific design matrix. Given the regularization parameters λ_1 and λ_2, the optimal regression parameters $\{\hat{\mathbf{w}}_{n,h}\}$ can be found with convex programming [5]. The regularization parameters themselves can be optimized with cross-validation or information criteria, like BIC. Note that different nodes n can have different time series segmentations, with different values of H_n, and that the segmentations have to be defined in advance. General guidelines for the choice of coarseness of segmentation can be found in the publication from Ahmed and Xing [5].

In my applications in Chapter 4 the segmentation is naturally suggested by the light phase, as I describe in more detail in Section 4.5.6.2. Also note that the original formulation of Tesla, proposed by Ahmed and Xing [5], is for logistic regression and binary data. The modification to linear regression, as in Equation (2.18), is straightforward and more appropriate for my application to non-binary data.

2.6 Automatic relevance determination (ARD-SBR)

The method of automatic relevance determination (ARD) in the context of sparse Bayesian regression (SBR) was proposed by Tipping [139], and I refer to this method as ARD-SBR. ARD-SBR was first applied to learning gene regulation networks by Rogers and Girolami [121]. It is related to the Bayesian regression method discussed in Section 2.2, with the following modification of the prior on the regression parameters $\mathbf{w}_n$: Equation (2.2) is replaced by

$$p(\mathbf{w}_n|\boldsymbol{\alpha}_n) \;=\; \mathcal{N}(\mathbf{0}, \mathrm{diag}[\boldsymbol{\alpha}_n]^{-1}) \tag{2.19}$$

where $\boldsymbol{\alpha}_n$ is a vector of interaction hyper-parameters of the same dimension as $\mathbf{w}_n$, and $\mathrm{diag}[\boldsymbol{\alpha}_n]$ is a diagonal matrix with $\boldsymbol{\alpha}_n$ in the diagonal. The marginal likelihood, Equation (2.4), now becomes

$$\begin{aligned} p(\mathbf{y}_n|\mathbf{X}_n, \sigma_n^2, \boldsymbol{\alpha}_n) &= \int p(\mathbf{y}_n|\mathbf{X}_n, \sigma_n^2, \mathbf{w}_n) p(\mathbf{w}_n|\boldsymbol{\alpha}_n) d\mathbf{w}_n \\ &= \mathcal{N}(\mathbf{y}_n|\mathbf{0}, \sigma_n^{-2}\mathbf{I} + \mathbf{X}_n^{\mathsf{T}}\mathrm{diag}[\boldsymbol{\alpha}_n]^{-1}\mathbf{X}_n) \end{aligned} \tag{2.20}$$

and is optimized with respect to the hyper-parameters $\boldsymbol{\alpha}_n$ in a maximum likelihood type-II manner. In the gene regulation study of Chapter 4 I follow Rogers and Girolami [121] and use a slightly modified version of the fast marginal likelihood algorithm from Tipping and Faul [140] for optimization. Note that as opposed to Equation (2.4), Equation (2.20) depends on the full design matrix $\mathbf{X}_n$, not the design matrix restricted to a subset of regulators $\boldsymbol{\pi}_n$, $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$, and the discrete search in structure space, $\boldsymbol{\pi}_n$, is replaced by a continuous search in hyper-parameter space, $\boldsymbol{\alpha}_n$, which is much faster. Hyper-parameters $\alpha_{n,i}$ associated with irrelevant regulators x_i^n will be driven to $\alpha_{n,i} \to \infty$, as explained in Section 13.7 in [108]. The consequence is that the associated regression parameters will be driven to zero, $w_{n,i} \to 0$, and irrelevant regulators x_i^n will effectively be pruned; hence the name 'automatic relevance determination' (ARD). For

fixed values of the hyper-parameters, the posterior of the regression parameters $\mathbf{w}_n$ can be obtained, and the method was therefore originally called 'sparse Bayesian regression' (SBR). However, as opposed to the proper Bayesian method discussed in Section 2.2, SBR-ARD is only 'Bayesian' about the values of the regression parameters $\mathbf{w}_n$ and does not reflect any uncertainty about $\boldsymbol{\alpha}_n$, which is typically of more interest. Hence, in comparison with Section 2.2, SBR-ARD gains computational speed at the expense of less thorough, approximate inference. How does SBR-ARD compare with the sparse regression methods of Section 2.4? As shown in Section 5 in [139], the interaction parameters $\boldsymbol{\alpha}_n$ can in principle be integrated out analytically (although this is not advisable for computational reasons). The resulting prior distribution of the regression parameters is $p(w_{n,i}) \propto \frac{1}{|w_{n,i}|}$, where $w_{n,i}$ is the i-th element of the regression parameter vector $\mathbf{w}_n$. The latter prior has more probability mass for $w_{n,i} \to 0$ than the Lasso prior, $p(w_{n,i}) \propto \exp(-|w_{n,i}|)$. Hence, SBR-ARD will lead to sparser network structures than Lasso. As for Lasso, I use the absolute values of the elements of the estimated regression parameter vectors, $\hat{\mathbf{w}}_n$, to score the regulatory effects on the target variable y_n $(n = 1, \ldots, N)$ with respect to their strengths.

2.7 Graphical Gaussian models (GGM)

The method of graphical Gaussian models (GGMs) is based on the insight that for random vectors $\mathbf{z}$ from a multivariate Gaussian distribution with zero mean and covariance matrix $\mathbf{C}$, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{C})$, the components z_n and $z_{n'}$, corresponding e.g. to two nodes n and n', are stochastically independent conditional on the remaining system

$$p(z_n, z_{n'}|\{z_i\}_{i \neq n,n'}) \;=\; p(z_n|\{z_i\}_{i \neq n,n'}) p(z_{n'}|\{z_i\}_{i \neq n,n'}) \tag{2.21}$$

if and only if the corresponding element (n, n') in the inverse covariance matrix $\mathbf{C}^{-1}$ is zero. Hence, if $\mathbf{C}$ is known, an undirected graph of gene dependence structures can be obtained by connecting all nodes (n, n') with $[\mathbf{C}^{-1}]_{n,n'} \neq 0$ by an (undirected) edge. In practice, $\mathbf{C}$ is unknown and has to be approximated by the empirical covariance matrix

$$\mathbf{S} \;=\; \frac{1}{(M-1)} \sum_{m=1}^{M} (\mathbf{z}_m - \overline{\mathbf{z}})(\mathbf{z}_m - \overline{\mathbf{z}})^{\mathsf{T}} \tag{2.22}$$

where $\mathbf{z}_1, \ldots, \mathbf{z}_M$ is an i.i.d. sample. If M is less than the dimension of $\mathbf{z}_m$, then the estimated covariance matrix $\mathbf{S}$ is rank deficient. To deal with this problem, two

main approaches have been proposed. The first approach, proposed by Schäfer and Strimmer [127], is to use shrinkage and replace the empirical covariance matrix $\mathbf{S}$ by the following regularized matrix:

$$\mathbf{S}^* \;=\; (1-\lambda)\mathbf{S} + \lambda\mathbf{I} \tag{2.23}$$

where $\mathbf{I}$ is the identity matrix [various alternatives are discussed in 127] and $\lambda > 0$ is a regularization parameter, which can be optimized with empirical risk minimization; see Equations (8) and (10) in [127] for explicit expressions. The second approach, proposed by Friedman et al. [46] and termed 'Glasso' (for 'Graphical Lasso') is to maximize the penalized log-likelihood subject to an ℓ_1-regularization term applied to the matrix elements:

$$\hat{\mathbf{\Theta}} = \underset{\mathbf{\Theta}}{\operatorname{argmax}}\{\log\det(\mathbf{\Theta}) - \operatorname{trace}(\mathbf{S\Theta}) - \lambda||\mathbf{\Theta}||_1\} \tag{2.24}$$

where $\hat{\mathbf{\Theta}}$ is the estimated inverse covariance matrix.

To apply GGMs to the reconstruction of gene regulatory networks in Chapter 4, I consider the random vectors $\mathbf{z}_{n,m} := (y_{n,m}, \mathbf{x}_{n,m}^{\mathsf{T}})^{\mathsf{T}}$, where $y_{n,m}$ is the time derivative of the mRNA concentration of target gene n at time m, see Equation (4.1), and $\mathbf{x}_{n,m}$ is the vector of the concentrations of the N_n potential regulators at time m ($m = 1, \ldots, M$). For each potential target variable y_n ($n = 1, \ldots, N$) I extract a GGM from the sample $\{\mathbf{z}_{n,m}\}_{m=1,\ldots,M}$. I then consider the first row (or column) of the resulting precision matrix. By standardization I obtain the partial correlations $\rho(y_n, x_j^n|\{x_i^n\}_{i\neq j})$ between the target variable y_n and its potential regulators x_j^n ($j = 1, \ldots, N_n$). Note that I ignore all correlations between potential regulators x_j^n since they are irrelevant for this application, although, the GGM is estimating these at the same time. This is necessary because the target variables y_n constitute a different type of measurement, i.e. concentration changes, as opposed to concentrations in the regulators x_j^n. The native GGM does not differentiate between targets and regulators since all variables have the same type of concentration measure thus all correlations are estimated. Since the direction of causality is always directed towards the target variable y_n in this application, the edges in the reconstructed graphs are directed, symbolically: $\{x_1^n, \ldots, x_{N_n}^n\} \rightarrow y_n$. The repeated bi-partitioning of the genes into targets and putative regulators renders Glasso equivalent to Lasso, i.e. regressing the targets on the regulators, as discussed on page 4 in [46]. The absolute values of the partial correlations $|\rho(y_n, x_j^n|\{x_i^n\}_{i\neq j})|$ can be used to score the regulatory interactions $x_j^n \rightarrow y_n$ ($n = 1, \ldots, N$ and $j = 1, \ldots, N_n$)

with respect to their strengths.

Furthermore, it must be noted that the vector $\mathbf{z}_{n,m}$ is not fully i.i.d. Normal because the concentration gradient vector $y_{n,m} \in \mathbf{z}_{n,m}$ introduces a time dependencies between variables at adjacent time points m. Although, this clashes with the GGM assumption I decided to include this method in order to estimate the performance loss compared to a linear regression method such as Lasso.

2.8 Bayesian spline autoregression (BSA)

The Bayesian spline autoregression method (BSA) proposed by Morrissey et al. [106] is related to the hierarchical Bayesian regression method of Section 2.2 with the essential difference that in the restricted design matrix $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$ the original covariates are augmented with q B-spline basis functions of degree l defined over a set of k evenly spaced knots, where (q, l, k) are user-defined parameters. Consequently, the strength of the interaction between a regulator x_i^n and the target variable y_n, which was modelled with a scalar in the method of Section 2.2, now becomes a vector. That is, each individual element $w_{n,i}$ of the regression parameter vector $\mathbf{w}_n := (w_{n,0}, w_{n,1}, \ldots, w_{n,N_n})^\mathsf{T}$, where $i = 0$ corresponds to the intercept, is substituted for a vector $\mathbf{w}_{n,i}$, spanning the entire range of B-spline basis functions. To deal with the increased dimension of the resulting total parameter vector $\mathbf{w}_n := (\mathbf{w}_{n,0}{}^\mathsf{T}, \mathbf{w}_{n,1}{}^\mathsf{T}, \ldots, \mathbf{w}_{n,N_n}{}^\mathsf{T})^\mathsf{T}$ and encourage network sparsity, a slab-and-stick-like Bayesian variable selection scheme, first proposed by Smith and Kohn [131], is used. Define $\mathbf{w}_{n,i} = \gamma_{n,i}\mathbf{u}_{n,i}$, where $\gamma_{n,i} \in \{0, 1\}$ is a binary variable to indicate whether the interaction $x_i^n \rightarrow y_n$ is on ($\gamma_{n,i} = 1$) or off ($\gamma_{n,i} = 0$). The indicator variables $\gamma_{n,i}$ are given a Beta-Bernoulli prior, meaning a Bernoulli prior on $\gamma_{n,i}$ with hyper-parameters from a Beta distribution. The higher-level hyper-parameters of the Beta distribution have a Jeffreys prior. The parameter vectors $\mathbf{u}_{n,i}$ are given a Gaussian prior to shrink them towards the origin:

$$p(\mathbf{u}_{n,i}|\tau_{n,i}) \;=\; \mathcal{N}(\mathbf{u}_{n,i}|\mathbf{0}, \tau_{n,i}\mathbf{K})$$

where the structure of the covariance matrix $\mathbf{K}$ is constructed from the second-order differences between adjacent coefficients, and $\tau_{n,i}$ is a smoothness hyper-parameter that defines the trade-off between fitting an interpolating spline ($\tau_{n,i} \rightarrow 0$) and a straight line ($\tau_{n,i} \rightarrow \infty$). Several priors for $\tau_{n,i}$ were tested in [106], with the best performance achieved with an inverted Pareto distribution. Like for the hierarchical Bayesian regression method of Section 2.2, there is no closed-form expression for the posterior

distribution, and MCMC sampling based on a Metropolis-within-Gibbs scheme is used: the technical details can be found in [106]. The resulting MCMC samples $\gamma_{n,i}^{(1)}, \ldots, \gamma_{n,i}^{(H)}$ ($g = 1, \ldots, G$ and $i = 1, \ldots, N_n$) are used to estimate the marginal posterior probability of the regulatory interactions $x_i^n \rightarrow y_n$:

$$P(x_i^n \rightarrow y_n) = \frac{1}{H} \sum_{h=1}^{H} \gamma_{n,i}^{(h)} \tag{2.25}$$

For the Bayesian spline autoregression method I use these marginal interaction posterior probabilities to score the regulatory interactions with respect to their strengths. The method was originally designed for time series data of the form of Equation (4.3). However the underlying approximation Equation (4.2) might be sub-optimal. For a fair comparison with the other methods, I have therefore applied it to target variables $y_{n,m}$ of the form of Equation (4.1).

2.9 State-space models (SSM)

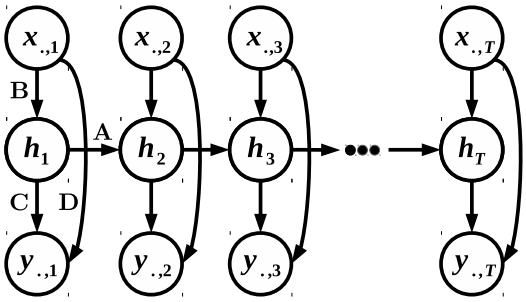


Figure 2.2: Graphical model representation of the state-space model (SSM). $\mathbf{y}_{.,m}$ represents the vector of all response variables (i.e. mRNA concentration derivatives) at observation m. $\mathbf{x}_{.,m}$ represents the vector of all potential regulators at observation m; depending on the problem, these are either mRNA or protein concentrations. $\mathbf{h}_m$ denotes the vector of unknown latent factors at observation m. The arrows indicate probabilistic dependence relations. The parameters of the model are the four transition matrices shown in capital letters **A**, **B**, **C**, and **D**. These parameters are given prior distributions, which depend on further hyper-parameters. For the full hierarchical Bayesian model representation, see [11]. The figure is adapted from Figure 5.2 in [11].

The state-space model (SSM) proposed by Beal et al. [12] is a Kalman filter with

additional Markovian dependence among the observation vectors, and additional dependence of the latent vectors on the observation vectors from the previous observation point; see Equations (6-7) in [12]. The parameters are estimated with variational Bayesian inference; since all distributions are multivariate Gaussian, this gives closed-form update equations that are carried out iteratively with a modified version of the expectation maximization algorithm. From these parameters, interaction strengths among the nodes can be derived; see Equation (8) in [12] for an explicit expression. The interactions contain two separate contributions: direct interactions, describing how node expression values at the previous observation point influence the current expression values, and indirect interactions, modelling node (gene) interactions mediated via the unobserved latent factors. The dimension of the latent vector is unknown and needs to be set using cross-validation or an estimate of the lower bound on the marginal likelihood. The intrinsic Markovian nature of the SSM from Beal et al. [12] is consistent with Equation (4.3), but not with Equation (4.1). However, a modification to my data format is straightforward by reverting to an alternative form of the SSM, proposed in [11], Chapter 5, and shown in Figure 2.2. In fact, the model in [12] is equivalent to the one in [11], with the external inputs replaced by the previous observations. The mathematical form of the model is as follows:

$$\begin{aligned} \mathbf{h}_{m+1} &= \mathbf{A}\mathbf{h}_m + \mathbf{B}\mathbf{x}_{\cdot,m} + \boldsymbol{\epsilon}_m \\ \mathbf{y}_{\cdot,m} &= \mathbf{C}\mathbf{h}_m + \mathbf{D}\mathbf{x}_{\cdot,m} + \boldsymbol{\xi}_m \end{aligned}$$

The symbols have the following meaning: $\mathbf{y}_{\cdot,m}$ is the vector of all response variables (i.e. mRNA concentration derivatives) at observation m. $\mathbf{x}_{\cdot,m}$ is the vector of all potential regulators at observation m; these are either mRNA or protein concentrations. $\mathbf{h}_m$ denotes the vector of unknown latent factors at observation m. $\boldsymbol{\epsilon}_m$ and $\boldsymbol{\xi}_m$ are vectors of i.i.d. white Gaussian noise. The parameters of the model are the transition matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$. These parameters are given prior distributions, which depend on further hyper-parameters. For the full hierarchical Bayesian model representation, see [11]. As described in [11], the posterior expectation of the interaction matrix $\mathbf{CB} + \mathbf{D}$ can be employed to assess the strengths of the individual network interactions; see Section 4.5.6.6 for details.

2.10 Gaussian processes (GP)

Gaussian processes provide a popular method in non-parametric Bayesian statistics for defining a prior distribution directly in the function space rather than the parameter space. By definition, a Gaussian process is a collection of random variables, of which any finite subset has a joint Gaussian distribution. For a node n the process can be fully represented by a mean function $m_n(.)$ and a covariance function $k_n(.,.)$:

$$f_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}) \sim \mathcal{GP}(m_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}), k_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m'})) \tag{2.26}$$

where $\mathbf{x}_{\boldsymbol{\pi}_n,m}$ and $\mathbf{x}_{\boldsymbol{\pi}_n,m'}$ are vectors of explanatory variables for target node n. In the case of gene regulation (Chapter 4) these are the gene expression values of the set of regulators $\boldsymbol{\pi}_n$, and $\mathbf{x}_{\boldsymbol{\pi}_n,m}$, $\mathbf{x}_{\boldsymbol{\pi}_n,m'}$ are the corresponding subsets of $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$; see Table 2.2 for an overview of the notation. The mean function $m_n(.)$ is usually set to zero, which presents prior ignorance about the trend (i.e. it is unsure that a trend is up or down). An important feature of Gaussian processes is that, due to the Gaussian assumption, marginalization integrals have closed form solutions. In particular, I get for the marginal likelihood, under the assumption of independent and identically distributed additive Gaussian noise with variance σ_n^2 [119]:

$$p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \boldsymbol{\theta}_n) \;=\; \frac{1}{\sqrt{(2\pi)^T|\mathbf{K}_n + \sigma_n^2\mathbf{I}|}} \exp\left(-\frac{1}{2}\mathbf{y}_n^{\mathsf{T}}(\mathbf{K}_n + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}_n\right) \tag{2.27}$$

where $\mathbf{y}_n = (y_{n,1}, \ldots, y_{n,M})^{\mathsf{T}}$ is a vector of target values for node n, and $\mathbf{K}_n$ is a M-by-M covariance matrix, with elements $K_{n,m,m'} = k_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m'})$. The arguments of the kernel function $k_n(.,.)$ are the vectors of concentration values associated with the putative regulators of node n, $\boldsymbol{\pi}_n$, taken at the observation points m and m'; these vectors are extracted from the (restricted) design matrix $\mathbf{X}_{n[\boldsymbol{\pi}_n]}$. The kernel function depends on certain hyper-parameters $\boldsymbol{\theta}_n$. For the widely applied squared exponential kernel

$$k_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m'}) \;=\; a_n \exp\left(-\frac{(\mathbf{x}_{\boldsymbol{\pi}_n,m} - \mathbf{x}_{\boldsymbol{\pi}_n,m'})^2}{2l_n^2}\right) \tag{2.28}$$

these are the length scale l_n and amplitude a_n: $\boldsymbol{\theta}_n = (l_n, a_n)$. For the kernel I follow Äijö and Lähdesmäki [6] and choose a Matérn class kernel

$$k_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m'}) \;=\; a_n \left(1 + \sqrt{\frac{3}{l_n^2}}(\mathbf{x}_{\boldsymbol{\pi}_n,m} - \mathbf{x}_{\boldsymbol{\pi}_n,m'})^\mathsf{T}(\mathbf{x}_{\boldsymbol{\pi}_n,m} - \mathbf{x}_{\boldsymbol{\pi}_n,m'})\right)$$
$$\exp\left(-\sqrt{\frac{3}{l_n^2}}(\mathbf{x}_{\boldsymbol{\pi}_n,m} - \mathbf{x}_{\boldsymbol{\pi}_n,m'})^\mathsf{T}(\mathbf{x}_{\boldsymbol{\pi}_n,m} - \mathbf{x}_{\boldsymbol{\pi}_n,m'})\right) \tag{2.29}$$

which provides a better compromise between smoothness and roughness. Like for the squared exponential kernel, the hyper-parameters $\boldsymbol{\theta}_n$ consist of a length scale and an amplitude parameter: $\boldsymbol{\theta}_n = (l_n, a_n)$. In order to apply Gaussian processes to the inference of gene regulatory networks, I follow the approach described by Äijö and Lähdesmäki [6]. The starting point is the mathematical formulation of transcriptional regulation of Equation (4.1), whose right-hand side can be reformulated as follows:

$$\tilde{f}_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}) \;=\; f_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}) + \mathbf{h}_n^\mathsf{T}\boldsymbol{\beta}_n \tag{2.30}$$

where $\boldsymbol{\beta}_n = (\alpha_n, \lambda_n)$ and $\mathbf{h}_n = (1, -x_{n,m})$. The approach taken by Äijö and Lähdesmäki [6] is to impose a normal distribution with mean vector $\mathbf{b}$ and covariance matrix $\mathbf{B} = \sigma_b^2\mathbf{I}$ on $\boldsymbol{\beta}_n$:

$$\boldsymbol{\beta}_n \;\sim\; N(\mathbf{b}, \mathbf{B}) \;=\; N(\mathbf{b}, \sigma_b^2\mathbf{I}) \tag{2.31}$$

It can then be shown [119] that a Gaussian process assumption for f_n

$$f_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}) \;\sim\; \mathcal{GP}(0, k_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m'})) \tag{2.32}$$

implies a Gaussian process for $\tilde{f}_n$ of the following form:

$$\tilde{f}_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}) \;\sim\; \mathcal{GP}(\mathbf{h}_n^\mathsf{T}\mathbf{b}, k_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m'}) + \mathbf{h}_n^\mathsf{T}\mathbf{B}\mathbf{h}_n) \tag{2.33}$$

This gives, in modification of Equation (2.27), a closed from expression for the marginal likelihood

$$p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \boldsymbol{\theta}_n, \sigma_n^2, \mathbf{b}, \sigma_b^2) \tag{2.34}$$

for which the explicit expression can be obtained from Äijö and Lähdesmäki [6]. Note that the target values $\mathbf{y}_n$ are time derivatives, which Äijö and Lähdesmäki [6] approximate by difference quotients. The hyper-parameters $\boldsymbol{\theta}_n = (a_n, l_n)$ and the noise variance σ_n^2 are optimized so as to maximize the marginal likelihood in Equation (2.34).

This can be achieved with the Polack-Ribiere conjugate gradient method, as described by Rasmussen and Williams [119]. To avoid negative values of $\boldsymbol{\beta}_n$, which are biologically implausible, Äijö and Lähdesmäki [6] suggested setting the hyper-parameters $\mathbf{b}$ and σ_b^2 to fixed values such that plausible values of $\boldsymbol{\beta}_n$ have high probability. To accomplish structure learning for a target variable y_n, the posterior probability for a selected set of regulators, $\boldsymbol{\pi}_n$, can be obtained from Bayes' theorem:

$$P(\boldsymbol{\pi}_n|\mathbf{y}_n, \mathbf{X}_n, \boldsymbol{\theta}_n, \sigma_n^2, \mathbf{b}, \sigma_b^2) = \frac{p(\mathbf{y}_n|\mathbf{X}_{n[\boldsymbol{\pi}_n]}, \boldsymbol{\theta}_n, \sigma_n^2, \mathbf{b}, \sigma_b^2)P(\boldsymbol{\pi}_n)}{\sum_{n'} p(\mathbf{y}_{n'}|\mathbf{X}_{n'[\boldsymbol{\pi}_{n'}]}, \boldsymbol{\theta}_{n'}, \sigma_{n'^2}, \mathbf{b}, \sigma_b^2)P(\boldsymbol{\pi}_{n'})} \tag{2.35}$$

where $P(\boldsymbol{\pi}_n)$ is the prior probability distribution on the set of potential regulators, for which Äijö and Lähdesmäki [6] chose a uniform distribution. The posterior probability of a particular gene interaction between the i-th regulator x_i^n and the target y_n is then given by marginalization:

$$P(x_i^n \rightarrow y_n|\mathbf{y}_n, \mathbf{X}_n, \boldsymbol{\theta}_n, \sigma_n^2, \mathbf{b}, \sigma_b^2) \;=\; \sum_{\boldsymbol{\pi}_n} I(x_i^n \in \boldsymbol{\pi}_n)P(\boldsymbol{\pi}_n|\mathbf{y}_n, \mathbf{X}_{n[\boldsymbol{\pi}_n]}, \boldsymbol{\theta}_n, \sigma_n^2, \mathbf{b}, \sigma_b^2) \tag{2.36}$$

where $I(x_i^n \in \boldsymbol{\pi}_n)$ is the indicator function, which is 1 if x_i^n is in the set of regulators $\boldsymbol{\pi}_n$, and zero otherwise. For larger networks, where a complete enumeration of all potential sets of regulators is computationally prohibitive, the common approach is to impose a fan-in restriction $\mathcal{F}$, e.g. of $\mathcal{F} = 3$, i.e. $P(\boldsymbol{\pi}_n) = 0$ if $|\boldsymbol{\pi}_n| > \mathcal{F}$, where $|\boldsymbol{\pi}_n|$ is the cardinality of the parent set. The posterior distribution of Equation (2.36) can be used to score the regulatory interactions with respect to their strengths. The Matlab software *GP4GRN* from Äijö and Lähdesmäki [6] implements the described framework and was used in my study.

2.11 Mutual information methods (ARACNE)

Consider three variables x_1, x_2 and x_3. The mutual information (MI) between x_1 and x_2 is then given by

$$I(x_1, x_2) \;=\; \int p(x_1, x_2) \log \left[\frac{p(x_1, x_2)}{p(x_1)p(x_2)}\right] dx_1 dx_2 \;\geq\; 0 \tag{2.37}$$

$I(x_1, x_2)$ is zero if the expression profiles of x_1 and x_2 are stochastically independent: $p(x_1, x_2) = p(x_1)p(x_2)$. The mutual information measures the degree of stochastic

dependence between x_1 and x_2, which in earlier work by Butte and Kohane [24] was used to provide a ranking of all potential gene interactions. A permutation test can then be used to set a threshold for discarding low-ranked interactions at a specified significance level. A shortcoming of this approach is the fact that direct interactions are not distinguished from indirect ones. Consider, for instance, a chain reaction

$$x_1 \longrightarrow x_2 \longrightarrow x_3$$

where node x_3 is indirectly regulated by node x_1 via the intermediary x_2, or the joint regulation of node x_1 and x_3 by node x_2:

$$x_1 \longleftarrow x_2 \longrightarrow x_3$$

In both scenarios the variables x_1 and x_3 are stochastically dependent, and $I(x_1, x_3)$ may be large despite the fact that there is no actual interaction between x_1 and x_3. To filter out such spurious interactions, a pruning mechanism was proposed by Margolin et al. [97], which is based on the data processing inequality: for the above interaction scenarios,

$$I(x_1, x_3) \;\leq\; \min\{I(x_1, x_2), I(x_2, x_3)\}$$

The proposed algorithm, called ARACNE, visits each node triplet in turn and removes the interaction with the smallest mutual information score. Each triplet is analysed irrespectively of whether its interactions have been marked for removal by prior pruning applications to different triplets, making the algorithm invariant with respect to a reordering of the nodes. A theoretical analysis of the types of networks that can be reconstructed with this algorithm can be found in [97]. The practical problem is related to the fact that Equation (2.37) cannot be computed exactly from a finite sample size but either requires a discretisation of the data (information loss), or the approximation of the probability densities $p(.)$ by a kernel density estimator; see [108], Chapter 14 for details. While the density itself depends critically on the bandwidth of this estimator, the ranking of mutual information scores has been found to be quite robust with respect to a variation of the bandwidth parameter; see Figure 1 in [97]. To apply ARACNE to gene expression time series in Chapter 2, a time delayed version has been proposed by Zoppoli et al. [156], which can deal with dynamic processes in the form of Equation (4.3). However, as will be discussed at the beginning of Section 4.2, the underlying approximation Equation (4.2) might be sub-optimal, and

I therefore apply ARACNE directly to Equation (4.1). That is, I apply ARACNE to each target variable y_n and its potential regulators $x_1^n, \ldots, x_{N_n}^n$ separately, to obtain mutual interaction scores $I^A(y_n, x_j^n)$ $(j = 1, \ldots, N_n)$, where $I^A(y_n, x_j^n) = I(y_n, x_j^n)$ or $I^A(y_n, x_j^n) = 0$ if the interaction has been pruned by the ARACNE algorithm. The ARACNE mutual interaction scores can then be interpreted in a bipartite manner, i.e. $I^A(y_n, x_j^n)$ is the strength of the regulatory interaction $x_j^n \to y_n$ $(n = 1, \ldots, N$ and $j = 1, \ldots, N_n)$.

2.12 Mixture Bayesian network models (MBN)

A flexible Gaussian mixture model approach for inferring non-linear network interactions has been proposed by Ko et al. [84, 85], which they call the "Mixture Bayesian network model"[6]. The key idea is to model each target node n conditional on its regulators in $\boldsymbol{\pi}_n$ with a conditional Gaussian mixture model. Given the vector of the variables in a regulator set $\boldsymbol{\pi}_n$ at sample m, symbolically $\mathbf{x}_{\boldsymbol{\pi}_n,m}$, I consider a Gaussian mixture model with C_n mixture components and the mixture weights $\alpha_{n,1}, \ldots, \alpha_{n,C_n}$ for the joint distribution of the target gene $y_{n,m}$ and its regulators $\mathbf{x}_{\boldsymbol{\pi}_n,m}$. Recalling the definition $\mathbf{z}_{\boldsymbol{\pi}_n,m} := (y_{n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m}^{\mathsf{T}})^{\mathsf{T}}$ from Table 2.2 I obtain:

$$p(\mathbf{z}_{\boldsymbol{\pi}_n,m}) = \sum_{c=1}^{C_n} \alpha_{n,c} f_{n,c}(\mathbf{z}_{\boldsymbol{\pi}_n,m}) \tag{2.38}$$

where each component-specific function $f_{n,c}(.)$ is the density function of a $(|\boldsymbol{\pi}_n|+1)$-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}_{n,c}$ and covariance matrix $\boldsymbol{\Sigma}_{n,c}$, and $\sum_{c=1}^{C_n} \alpha_{n,c} = 1$. The marginal distribution of the vector $\mathbf{x}_{\boldsymbol{\pi}_n,m}$ is then also a Gaussian mixture:

$$p(\mathbf{x}_{\boldsymbol{\pi}_n,m}) = \sum_{c=1}^{C_n} \alpha_{n,c} f_{n,c}^{\star}(\mathbf{x}_{\boldsymbol{\pi}_n,m}) \tag{2.39}$$

where the $|\boldsymbol{\pi}_n|$-dimensional Gaussian density functions $f_{n,c}^{\star}(.)$ have mean vectors $\boldsymbol{\mu}_{n,c}^{\star}$ and covariance matrices $\boldsymbol{\Sigma}_{n,c}^{\star}$ which are sub-vectors of $\boldsymbol{\mu}_{n,c}$ and sub-matrices of $\boldsymbol{\Sigma}_{n,c}$, respectively. More precisely, $\boldsymbol{\mu}_{n,c}^{\star}$ is obtained by deleting the element corresponding to the target variable $y_{n,m}$ in $\boldsymbol{\mu}_{n,c}$, and $\boldsymbol{\Sigma}_{n,c}^{\star}$ is obtained by deleting the row and the column corresponding to $y_{n,m}$ in $\boldsymbol{\Sigma}_{n,c}$. Considering $\mathbf{z}_{\boldsymbol{\pi}_n,m}$ $(m = 1, \ldots, M)$ as an i.i.d. sample and taking into account that the conditional distribution $p(y_{n,m}|\mathbf{x}_{\boldsymbol{\pi}_n,m})$

[6] I use the authors' terminology, although the model is not a proper Bayesian network.

is the ratio of the joint distribution in Equation (2.38) and the marginal distribution in Equation (2.39), the likelihood of the conditional Gaussian mixture model is given by:

$$LL(\mathbf{y}_n|\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]}, \boldsymbol{\theta}(\boldsymbol{\pi}_n, C_n)) = \frac{\prod_{m=1}^{M}\sum_{c=1}^{C_n}\alpha_{n,c}f_{n,c}(\mathbf{z}_{\boldsymbol{\pi}_n,m})}{\prod_{m=1}^{M}\sum_{c=1}^{C_n}\alpha_{n,c}f^{\star}_{n,c}(\mathbf{x}_{\boldsymbol{\pi}_n,m})} \tag{2.40}$$

where $\boldsymbol{\theta}(\boldsymbol{\pi}_n, C_n)$ denotes the set of mixture parameters, namely the mixture weights as well as the mean vectors and covariance matrices of the component-specific Gaussian distributions,
$designMrestrstar$ is the matrix of the observations of the regulators in $\boldsymbol{\pi}_n$, and $\mathbf{y}_n = (y_{n,1}, \ldots, y_{n,M})^{\mathsf{T}}$ is the vector of the target variable observations.

Given a fixed set of regulators, $\boldsymbol{\pi}_n$, and a fixed number of mixture components, C_n, the maximum likelihood (ML) estimates for the mixture parameters $\boldsymbol{\theta}(\boldsymbol{\pi}_n, C_n)$ can be obtained with the Expectation-Maximization (EM) algorithm, as described in detail by Ko et al. [85]. Keeping $\boldsymbol{\pi}_n$ fixed, ML estimates, $\hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C_n)$, can be computed for different numbers of mixture components C_n. Having estimates $\hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C_n)$ for $C_n = 1, \ldots, C_{MAX}$, where $C_{MAX} = 10$ is an imposed upper bound on the number of mixture components, the Bayesian Information Criterion (BIC) is employed to determine the *best* number of mixture components given $\boldsymbol{\pi}_n$:

$$C^{BIC}_{n|\boldsymbol{\pi}_n} = \underset{C_n}{\operatorname{argmin}}\{-2\log(LL(\mathbf{y}_n|\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]}, \hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C_n))) + \log(M)|\hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C_n)|\} \tag{2.41}$$

where M is the number of observations, $|\hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C_n)|$ is the number of the ML-estimated mixture parameters and the likelihood $LL(.|.)$ has been defined in Equation (2.40). With Equation (2.41) the best number of mixture components $C^{BIC}_{n|\boldsymbol{\pi}_n}$ can be determined for each potential regulator set $\boldsymbol{\pi}_n$. In my implementation I systematically compute $C^{BIC}_{n|\boldsymbol{\pi}_n}$ for each set $\boldsymbol{\pi}_n$ with a cardinality $|\boldsymbol{\pi}_n| \leq 3$. Finally, the best set of regulators $\boldsymbol{\pi}^{BIC}_n$ for target variable y_n minimizes the BIC criterion, and is thus given by:

$$\boldsymbol{\pi}^{BIC}_n = \underset{\boldsymbol{\pi}_n}{\operatorname{argmin}}\{-2\log(LL(\mathbf{y}_n|\mathbf{x}_{\boldsymbol{\pi}_n}, \hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C^{BIC}_{n|\boldsymbol{\pi}_n}))) + \log(M)|\hat{\boldsymbol{\theta}}(\boldsymbol{\pi}_n, C^{BIC}_{n|\boldsymbol{\pi}_n})|\} \tag{2.42}$$

I repeat the optimization procedure, outlined above, several times and I average

over the obtained results, as described in Subsection 4.5.6.9, to score the individual interactions, $x_i^n \to y_n$.

2.13 Gaussian Bayesian networks (BGe)

The BGe scoring metric was introduced by Geiger and Heckerman [49] and has become a standard modelling framework for static and dynamic Gaussian Bayesian networks.[7] For $m = 1, \ldots, M$ the common distribution of the target variable $y_{n,m}$ and its potential regulators $\mathbf{x}_{n,m}$ is assumed to be an i.i.d. sample from a $(N_n + 1)$-dimensional multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$p(\mathbf{z}_{n,m}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{N_n+1}{2}} \det(\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\{-\frac{1}{2}(\mathbf{z}_{n,m} - \boldsymbol{\mu})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{z}_{n,m} - \boldsymbol{\mu})\} \tag{2.43}$$

where N_n is the number of potential regulators for the target variable y_n, i.e. the length of the vector $\mathbf{x}_{\boldsymbol{\pi}_n,m}$, and $\mathbf{z}_{n,m} := (y_{n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m}^{\mathsf{T}})^{\mathsf{T}}$, as defined in Table 2.2. Onto the unknown parameters, namely the mean vector $\boldsymbol{\mu}$ and the precision matrix $\mathbf{W} := \boldsymbol{\Sigma}^{-1}$, a normal-Wishart prior is imposed, symbolically:

$$p(\mathbf{W}|\alpha, \mathbf{T_0}) = c(N_n, \alpha) \det(\mathbf{T_0})^{\frac{\alpha}{2}} \det(\mathbf{W})^{\frac{\alpha-N_n-1}{2}} \exp\{-\frac{1}{2}\text{trace}(\mathbf{T_0 W})\} \tag{2.44}$$

$$p(\mu|\mu_{\mathbf{0}}, (\nu\mathbf{W})^{-1}) = (2\pi)^{-\frac{N_n+1}{2}} \det(\nu\mathbf{W})^{\frac{1}{2}} \exp\{-\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu_0})^{\mathsf{T}} \nu\mathbf{W} (\boldsymbol{\mu} - \boldsymbol{\mu_0})\} \tag{2.45}$$

where

$$c(N_n, \alpha) = \left(2^{\frac{\alpha(N_n+1)}{2}} \pi^{\frac{(N_n+1)N_n}{4}} \sum_{i=1}^{N_n+1} \Gamma\left(\frac{\alpha + 1 - i}{2}\right)\right)^{-1} \tag{2.46}$$

and the hyper-parameters α, $\mathbf{T_0}$, ν and $\boldsymbol{\mu_0}$ of the normal-Wishart distribution are chosen fixed. [49] show that the marginal likelihood:

[7] Note that the abbreviation 'BGe' was introduced by Geiger and Heckerman [49] and stands for *Bayesian metric for Gaussian networks having score equivalence*; see [49] for more details.

$$p(\mathbf{z}_{n,1},\ldots,\mathbf{z}_{n,M}) = \int\int \left(\left\{ \prod_{m=1}^{M} p(\mathbf{z}_{n,m}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \right\} p(\boldsymbol{\mu}|\boldsymbol{\mu_0},(\nu\mathbf{W})^{-1}) p(\mathbf{W}|\alpha,\mathbf{T_0}) \right) d\boldsymbol{\mu} d\mathbf{W} \tag{2.47}$$

can then be computed in closed-form. If it is further assumed that the target variable y_n, conditional on the set of regulators $\boldsymbol{\pi}_n$, becomes statistically independent of all the other potential regulators, symbolically $p(\mathbf{y}_n|\mathbf{X}^{\star}_n,\boldsymbol{\pi}_n) = p(\mathbf{y}_n|\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]})$, then the conditional distributions

$$p(\mathbf{y}_n|\mathbf{X}^{\star}_n,\boldsymbol{\pi}_n) = p(\mathbf{y}_n|\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]}) = \frac{p(\mathbf{y}_n,\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]})}{p(\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]})} \tag{2.48}$$

can also be computed in closed-form for each regulator set $\boldsymbol{\pi}_n$, see [49] for details. Imposing uniform priors on the regulator sets, $\boldsymbol{\pi}_n$, subject to a maximal cardinality restriction $\mathcal{F}$, the posterior distribution of the regulator set $\boldsymbol{\pi}_n$ with $|\boldsymbol{\pi}_n| \leq \mathcal{F}$ is given by:

$$P(\boldsymbol{\pi}_n|\mathbf{y}_n,\mathbf{X}^{\star}_n) = \frac{p(\mathbf{y}_n|\mathbf{X}^{\star}_{n[\boldsymbol{\pi}_n]})}{\sum_{\tilde{\boldsymbol{\pi}}_n:|\tilde{\boldsymbol{\pi}}_n|\leq\mathcal{F}} p(\mathbf{y}_n|\mathbf{X}^{\star}_{n[\tilde{\boldsymbol{\pi}}_n]})} \tag{2.49}$$

where the sum in the denominator is over all valid regulator sets $\tilde{\boldsymbol{\pi}}_n$ whose cardinality is lower than or equal to the fan-in $\mathcal{F}$. The posterior probability of an interaction between x^n_i and y_n can then be computed by marginalization:

$$P(x^n_i \rightarrow y_n|\mathbf{y}_n,\mathbf{X}^{\star}_n) \;=\; \sum_{\boldsymbol{\pi}_n} I(x^n_i \in \boldsymbol{\pi}_n) P(\boldsymbol{\pi}_n|\mathbf{y}_n,\mathbf{X}^{\star}_n) \tag{2.50}$$

where $I(x^n_i \in \boldsymbol{\pi}_n)$ is the indicator function, which is 1 if x^n_i is in the set of regulators $\boldsymbol{\pi}_n$, and zero otherwise. I use the posterior probabilities in Equation (2.50) to score the regulatory interactions with respect to their strengths.

2.14 Dynamic Bayesian network with BDe score (Banjo)

The Banjo (Bayesian Inference with Java objects) is an implementation of a dynamic Bayesian network (DBN) inference algorithm using the Bayesian Dirichlet (BDe) scoring metric [67]. The DBN is a first order Markov model that has time-varying dependences and conditional independences of discrete variables, meaning that the variables at one time-point are affected by the variables of the immediate previous time-point.

The dependences that form the network are proposed in a greedy search procedure and the BDe metric scores how well the network represents the observed data. The strength and sign of the network dependences are determine through a additional influence score.

The DBN is defined by the pair $<\mathcal{G}, \mathbf{\Theta}>$. The graph $\mathcal{G}$ describes the dependence structure and the parameter set $\mathbf{\Theta}$ holds the probability distribution parameter vector $\boldsymbol{\theta}_{n|\boldsymbol{\pi}_n} = (\theta_{n,m|\boldsymbol{\pi}_n})_{\forall nm}$, where $m = (1, \ldots, M)$ refers in this context to the time points in the dynamic network. The parameter $\theta_{n,m|\boldsymbol{\pi}_n} = p(x_n|\boldsymbol{\pi}_n)$ for each node n and time point m depends on its corresponding parent set previously defined with $\boldsymbol{\pi}_n$. The joint probability distribution over all nodes is $p(\mathbf{x}) = P(x_1, \ldots, x_N) = \prod_{m=1}^{M} \prod_{n=1}^{N} P(x_n, m|\boldsymbol{\pi}_n)$, namely the probability for the variable x_n to take on a certain value, given the dependence on the incoming parent nodes. The score for the graph $\mathcal{G}$ given a data set $\mathcal{D}$ for all variables $x_{n,m}$ is the Bayesian score function:

$$logP(\mathcal{G}|\mathcal{D}) = logP(\mathcal{D}|\mathcal{G}) + P(\mathcal{G}) - logP(\mathcal{D}) \tag{2.51}$$

The evidence $p(\mathcal{D})$ can be neglected since its marginal probability is the same for all settings of $\mathcal{G}$. The prior over the graph $p(\mathcal{G})$ also vanishes since I assume no preference for a graph, thus yielding a uniform distribution. Solving the remaining log of the marginal likelihood $p(\mathcal{D}|\mathcal{G})$ requires the integration over all possible settings of the parameter set $\mathbf{\Theta}$, leading to the Bayesian Dirichlet score $s_{\mathcal{D}}(\mathcal{G})$:

$$BD: \quad s_{\mathcal{D}}(\mathcal{G}) = log\ p(\mathcal{D}|\mathcal{G}) = log \int p(\mathcal{D}|\mathcal{G}, \mathbf{\Theta}) p(\mathbf{\Theta}|\mathcal{G}) d\mathbf{\Theta} \tag{2.52}$$

The task is to find a graph $\mathcal{G}^*$ that satisfies $\mathcal{G}^* = \text{argmax}_{\mathcal{G}}\, s_{\mathcal{D}}(\mathcal{G})$. Assuming that $p(\mathbf{\Theta}|\mathcal{G})$ is a Dirichlet prior, the integral can be solved with

$$s_{\mathcal{D}}(\mathcal{G}) = log \prod_{n=1}^{N} \prod_{j=1}^{q_n} \left(\frac{\Gamma(\alpha_{nj})}{\Gamma(\alpha_{nj} + N_{nj})} \prod_{k=1}^{r_n} \frac{\Gamma(\alpha_{njk} + N_{njk})}{\Gamma(\alpha_{njk})} \right) \tag{2.53}$$

where q_n is the number of unique instantiations of $\boldsymbol{\pi}_n$, r_n is the number of discrete values in the data $\mathcal{D}$, $\Gamma(\cdot)$ is the gamma function, $\alpha_{nj} = \sum_k \alpha_{njk}$ and α_{njk} are the Dirichlet concentration hyper-parameters, N_{njk} is the number of times that x_m takes on the value k and the parents of x_n take on instantiation j, and $N_{nj} = \sum_k N_{njk}$.

A disadvantage of Banjo is that it is limited to discrete values, which requires a discretisation of my continuous data causing information loss. In Chapter 5 I use the quantile discretisation procedure described by Hartemink [66]. For a detailed account

of the method refer to the supplementary material S1 in [132] or the website[8].

To measure the confidence of the proposed interactions, I use the fact that Banjo produces a summary of the 100 highest scoring networks. Extracting the regulatory interactions between a predictor x_n and a target y_n from the 100 networks corresponds to marginalization over these high scoring networks. An estimator of marginal posterior probability of an interaction $x_n \rightarrow y_n$ is given by the fraction of networks that contain the interaction.

[8]Extensive documentation can be found at https://www.cs.duke.edu/ amink/software/banjo

Chapter 3

Inference and Evaluation

Applying probabilistic predictions to find proper parameters for a regression problem is a central paradigm in machine learning. The maximum likelihood estimate (MLE) and maximum a posteriori (MAP) estimate are the two most important and basic approaches to infer model parameters given only a likelihood (MLE) or posterior (MAP) probability estimate. With them it is easier to handle ambiguous cases by assigning a probability, i.e. a confidence, to the parameters in the model that map a predictive set of features to a dependent variable or response as it is defined in the widely used linear regression models. Given a data set $\mathcal{D} = (\mathbf{x}_1, .., \mathbf{x}_M)$ and the parameter vector $\boldsymbol{\theta}$, which can contain also a single parameter, the MLE is defined as

$$\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \Big\{ p(\mathcal{D}|\boldsymbol{\theta}) \Big\} \tag{3.1}$$

Hence $\boldsymbol{\theta}_{MLE}$ becomes a maximum likelihood estimate for the true parameter $\boldsymbol{\theta}$. The advantage of the MLE is that it is easy to compute and invariant under re-parametrization, i.e. if a function $g(\boldsymbol{\theta}_{MLE})$ is a MLE for $g(\boldsymbol{\theta})$ than it would still be a MLE if for instance the true parameter is squared with $g(\boldsymbol{\theta}^2)$. Furthermore, the MLE has several asymptotic properties such that it converges toward a normal distribution and with a large data size M converges to the true parameter $\boldsymbol{\theta}$. One of the major disadvantages of the MLE is that it tends to over-fit the model on the data. This means that the model might perfectly predict the data samples it was fitted to, but completely fails on a similar data set because it does not capture the uncertainty of the data but rather picks up the noise of the data samples. However, a penalized likelihood can prevent overfitting and is equivalent to the MAP where penalization is controlled

with a prior density. The MAP can be thought of as the maximum value of the joint posterior density $p(\boldsymbol{\theta}, \mathcal{D})$ for which the parameter $\boldsymbol{\theta}$ from the complete parameters set $\boldsymbol{\Theta}$ best explains the data $\mathcal{D}$.

$$\boldsymbol{\theta}_{MAP} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \Big\{ p(\boldsymbol{\theta}|\mathcal{D}) \Big\} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}} \Big\{ p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \Big\} \tag{3.2}$$

For a data set that has a large number of samples $M \to \infty$, the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ becomes dominant compared to the parameter prior $p(\boldsymbol{\theta})$. In this scenario the MAP tends to approach the solution of the MLE and also shares the same asymptotic properties. A disadvantage of the MAP is that it is not invariant to changes of the model parameter in contrast to MLE. Thus the MAP is suboptimal given the vast amount of possible posterior densities for varying parameter sets.

In Section 3.1 I will discuss feature selection techniques that control the number and choice of parameters in the set $\boldsymbol{\theta}$ with focus on least squares regularization. Section 3.2 describes the popular Marcov Chain Monte Carlo (MCMC) that infers marginal posterior probabilities by converging towards a true posterior density. Feature selection in a discrete sense can be achieved with the reversible jump MCMC (RJMCMC) that is described in the same section. Since, I use clustering to infer species similarities in terms of neighbourhood distributions (see Chapter 5), I will explain in Section 3.3 the k-means and Gap-statistics. Finally, Section 3.4 describes how I evaluate the learned network structures that are retrieved from the methods previously defined in Chapter 2.

3.1 Sparse regression

Sparse regression is a technique to prevent overfitting by decreasing the number of features in the model that act as predictors. The feature selection thus encourages sparsity of the model with the effect that full explanatory power is restricted to those features that are best suited to predict the model response. This is realized by ignoring certain features or setting elements of the weight values $\mathbf{w} = (w_1, .., w_N)$ to zero for those features j that should be excluded from the predictor set.

3.1.1 Spike and slab model

A Bayesian variable selection approach is the so called "spike and slab" model from Mitchell and Beauchamp [104], which has the posterior $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta}) p(\mathcal{D}|\boldsymbol{\theta})$. The model is parametrized with a set of relevant features $\boldsymbol{\theta}$, which are penalized with a so

called ℓ_0-pseudo-norm that regulates the number of selected features. The relevance of the features is indicated by a bit vector $\boldsymbol{\theta} = (\theta_1, .., \theta_N)$ with $\theta_j = 1$ in the case that a feature j is selected or relevant, and $\theta_j = 0$ if it is irrelevant. The ℓ_0-norm is formulated with $||\boldsymbol{\theta}||_0 = \sum_{j=1}^{N} \theta_j$ and penalizes the prior density of the bit vector with a Bernoulli distribution:

$$p(\boldsymbol{\theta}) = \prod_{j=1}^{N} Ber(\theta_j|\pi_0) = \pi_0^{||\boldsymbol{\theta}||_0}(1-\pi_0)^{N-||\boldsymbol{\theta}||_0} \tag{3.3}$$

where π_0 is the probability $p(\theta_j = 1)$ that a feature should be selected into the model. Hence low values of π_0 negatively penalize the number of features in $\boldsymbol{\theta}$ and high values promote a large number of selected features. The feature vector $\boldsymbol{\theta}$ affects the prior probability of the weights vector $\mathbf{w} = (w_1, .., w_N)$ by setting a weight w_j to zero if the corresponding feature j is defined irrelevant with $\theta_j = 0$. Whenever $\theta_j = 1$, the weight w_j can be expected to be non-zero. In this case a reasonable prior is defined by a normal distribution with a mean of zero and a variance $\sigma^2_{\mathbf{w}}$ that controls how strong the weight can fluctuate around the mean scaled by an additional noise variance variable σ^2:

$$p(w_j|\sigma^2, \theta_j) = (1-\theta_j)\delta_0(w_j) + \theta_j \mathcal{N}(w_j|0, \sigma^2\sigma^2_{\mathbf{w}}) \tag{3.4}$$

The first term $\delta_0(\cdot)$ is a point probability mass that causes a "spike" at zero and the second term is referred to as "slab" in the case when $\sigma^2_{\mathbf{w}} \to \infty$ and $\mathcal{N}(w_j)$ approaches a uniform distribution. The prior for the selected feature set $\boldsymbol{\theta}$ and the weights prior are combined in the full posterior with:

$$\begin{aligned} p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta}) &= p(\boldsymbol{\theta}|\pi_0)p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \\ &= p(\boldsymbol{\theta}|\pi_0) \int\int p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \boldsymbol{\theta})p(\mathbf{w}|\boldsymbol{\theta}, \sigma^2)p(\sigma^2)d\mathbf{w}d\sigma^2 \end{aligned} \tag{3.5}$$

A disadvantage of using the ℓ_0-pseudo-norm is that the values $||\boldsymbol{\theta}||_0$ are discrete which causes the objective function to become very non-smooth, i.e. non-convex. Hence replacing the discrete with a continuous prior leads to a convex approximation of the non-convex optimization problem.

3.1.2 ℓ_1 and ℓ_2 regularization

The posterior $p(\boldsymbol{\theta}|\mathcal{D})$ in Equation 3.5 has 2^N possible models that are computational expensive to explore given the fact that $\boldsymbol{\theta}$ is a discrete parameter vector. The "spike and slab" prior on $\boldsymbol{\theta}$ can be replaced with a prior of the continuous weight variables $\mathbf{w}$ by encouraging $w_j = 0$ with a distribution that centers a lot of probability density at zero. A Laplace distribution with a spike at the zero-mean ($\mu = 0$) and heavy tails that are parametrized with a regularization term can be formulated as

$$p(\mathbf{w}|\lambda) = \prod_{j=1}^{N} Lap(w_j|0, 1/\lambda) \propto \prod_{j=1}^{N} e^{-\lambda|w_j|} \tag{3.6}$$

The negative logarithm of this prior yields $\sum_{j=1}^{N} \lambda|w_j| = \lambda||\mathbf{w}||_1$, where $||\mathbf{w}||_1$ is the ℓ_1-norm of $\mathbf{w}$ and λ is the scaling parameter that controls the strength of regularization. This prior can be used to do MAP estimation because minimizing the negative log likelihood is equivalent to the MAP given a uniform prior $p(\boldsymbol{\theta})$. An estimate for the weight parameter $\hat{\mathbf{w}}$ can thus be formulated as the negative logarithm of the posterior in Equation 3.2:

$$\begin{aligned} \hat{\mathbf{w}}_{MAP} &= -log\Big(\underset{\mathbf{w}}{\text{argmax}}\Big\{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})\Big\}\Big) \\ &= \underset{\mathbf{w}}{\text{argmin}}\Big\{ - log\ p(\mathcal{D}|\mathbf{w}) - log\ p(\mathbf{w})\Big\} \end{aligned} \tag{3.7}$$

The first term in Equation 3.7 becomes $-1/(2\sigma^2)\sum_{n=1}^{M}(y_i - w_0 - \sum_{j=1}^{N} w_j x_j)^2$ in a linear regression scenario with Gaussian likelihood, and the second term is the previously described Laplace prior. By eliminating $-1/(2\sigma^2)$ from the first term one recovers the residual sum of squares that quantifies the loss of the linear model.

$$\begin{aligned} \hat{\mathbf{w}}_{MAP} &= \underset{\mathbf{w}}{\text{argmin}}\Big\{\sum_{n=1}^{M}(y_i - w_0 - \sum_{j=1}^{N} w_j x_j)^2 + \lambda^{'}\sum_{j=1}^{N}|w_j|\Big\} \\ &= \underset{\mathbf{w}}{\text{argmin}}\Big\{||\mathbf{y} - \mathbf{X}^{\mathsf{T}}\mathbf{w}||_2^2 + \lambda^{'}||\mathbf{w}||_1\Big\} \end{aligned} \tag{3.8}$$

where the penalty factor is $\lambda^{'} = 2\lambda\sigma^2$. This equation is also known as the Lasso described in Section 2.4 and represents in essence the Lagrangian form of a constrained optimization problem with the RSS corresponding to a quadratic objective function

subject to the constrain of the penalty term $||\mathbf{w}||_1$ under the boundary B:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}^\mathsf{T}\mathbf{w}||_2^2 \right\} \qquad s.t. \qquad ||\mathbf{w}||_1 \leq B \tag{3.9}$$

B is inversely related to the penalty λ and is an upper bound on the ℓ_1-norm constraint: a small value of B corresponds to a large value of λ hence the penalization of the weights $\mathbf{w}$ is stronger than with a relaxed constraint B.

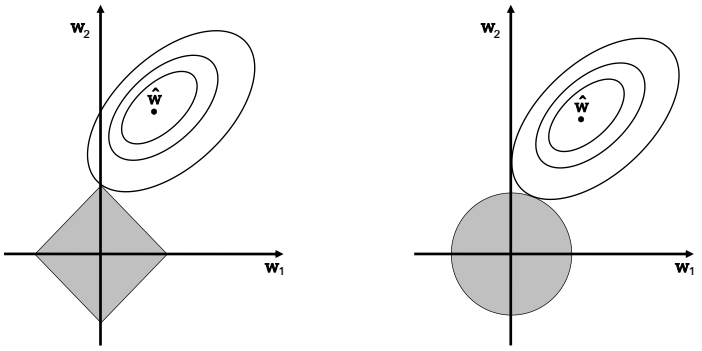


Figure 3.1: Geometric interpretation of ℓ_1 and ℓ_2-norm. Left plot illustrates the ℓ_1-norm with weight estimates $\hat{\mathbf{w}}$ "touching" the boundary of the diamond shaped constrained area. This is a solution to the optimization problem and will encourage weights to take on values of zero because of the particular shape of the constraint. The right plot shows the ℓ_2-norm constrained area that has a circle form. In this case no regularization of weights towards zero occurs because of the round shape. Based on Figure 3.12 from Hastie et al. [70].

The interpretation of B for the ℓ_1-norm is illustrated geometrically for a 2-dimensional weight vector in the left plot of Figure 3.1. The grey area in diamond shape is defined by the ℓ_1-norm whereas the size is determined by B. The area thus acts as the boundary that intersects the ellipse of estimated values $\hat{\mathbf{w}}$ of the objective function. Relaxing B causes the shape to grow in size until it "touches" the objective functions estimates. For small B and hence a small constraining area this is likely to occur along one of the axis, i.e. values of $w_j = 0$ will be encouraged because of the specific geometric shape of the diamond. In Figure 3.1 this is the case for $w_1 = 0$, whereas $w_2 \neq 0$. When $B \to 0$, the area becomes condensed at the origin zero and all weights approximate $\hat{\mathbf{w}} \to \mathbf{0}$.

The right plot in Figure 3.1 illustrates the case for ridge regression, that has a ℓ_2-norm:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}^{\mathsf{T}}\mathbf{w}||_2^2 \right\} \qquad s.t. \qquad ||\mathbf{w}||_2^2 \leq B \tag{3.10}$$

or in the Lagrangian form:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left\{ ||\mathbf{y} - \mathbf{X}^{\mathsf{T}}\mathbf{w}||_2^2 + \lambda ||\mathbf{w}||_2^2 \right\} \tag{3.11}$$

The area defined by the squared ℓ_2-norm $||\mathbf{w}||_2^2 = \sum_{j=1}^{N} |w_j|^2$ has a round shape. Because of this shape the boundary will likely intersect with estimates $\hat{w}_j$ that are not located exactly at one of the axis origins, which leads to the conclusion that the ℓ_2-norm is not encouraging the sparsity of $\mathbf{w}$. Hence the ℓ_1-norm is often preferred over the ℓ_2-norm because it produces sparse weights $\mathbf{w}$, which is essentially feature selection. A disadvantage of the absolute loss of the ℓ_1-norm is that it is not differentiable at the value $w_j = 0$ so that gradient based learning algorithms require modifications. Furthermore the ℓ_1-norm is biased towards signal reproduction (predictive model) instead of giving the best estimate for the true model (explanatory model). This is reflected in the way relevant weights are typically estimated as too small since they are simultaneously shrunken together with irrelevant weights. A de-biasing mechanism was proposed by Hastie et al. [70] and was tested on a synthetic data set in Section 4.6.1. The squared loss of the ℓ_2-norm has the disadvantage that outliers have the tendency to dominate the penalty because squared single large values of w_j have a much greater effect on the sum in $||\mathbf{w}||_2^2$ than smaller values. Both penalty norms can be combined in what was described as the Elastic Net method [157] in Section 2.4.

$$\hat{\mathbf{w}}_n = \operatorname{argmin} \left\{ ||\mathbf{y}_n - \mathbf{X}_n^{\mathsf{T}}\mathbf{w}_n||_2^2 + \lambda_1 \|\mathbf{w}_n\|_1 + \lambda_2 \|\mathbf{w}_n\|_2^2 \right\} \tag{3.12}$$

The Elastic Net will neutralize the tendency of the ℓ_1 penalty to select only a single predictor from a set of highly correlated variables through the additional ℓ_2 penalty that selects correlated variables together. The ℓ_2-norm also relaxes the ℓ_1 constraint on the maximum number of non-zero weights that equals M, i.e. the number of data samples. Hence more than M features can be selected from the feature set of size N whenever $M < N$.

3.1.3 Regularization path

Increasing the penalty term λ in Equation 3.8 and 3.11 has the tendency to increase the sparsity of the estimated weights $\hat{\mathbf{w}}$. Consequently one can plot each weight w_j as

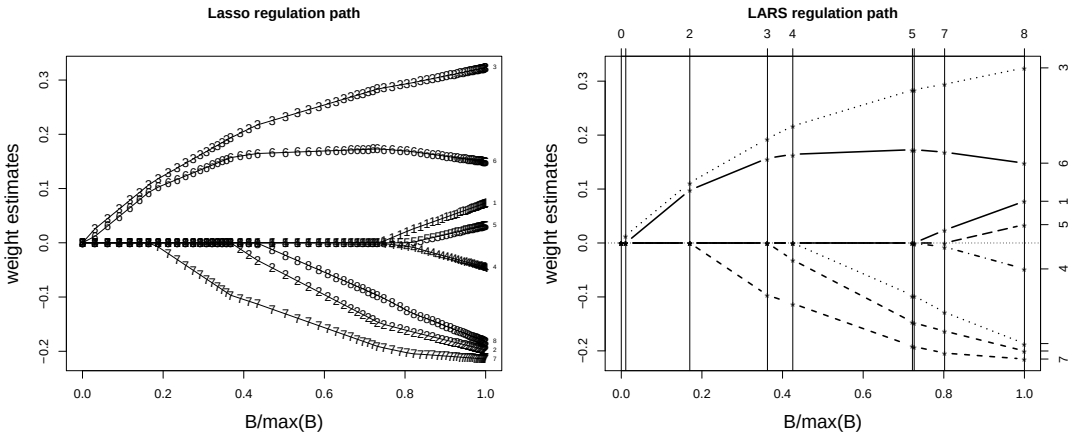


Figure 3.2: Profile of Lasso weights. The left plot shows the coefficients $\mathbf{w}$ for varying penalty parameter λ expressed as the shrinkage factor $(B/max(B))$. The numbers correspond to the different weights and each appearance marks the estimation for a given shrinkage factor. The right plot illustrates the piecewise linear regularization path of the LARS algorithm with vertical lines highlighting the critical values of B.

a function of the shrinkage factor $(B/max(B))$ to illustrate the effect of the boundary condition on the weights. This is known as the regularization path of the weights and is demonstrated for the ℓ_1-norm in the left panel of Figure 3.2. The plot displays the paths for 8 weights. The very left of the x-axis is attributed to $\lambda \to \infty$, which is proportional to the lowest boundary $B = 0$ with the effect that all weights are zero. Increasing the boundary B, i.e. decreasing the penalty parameter λ, causes an increase of the magnitude of the weights. While the change of the weights is not necessarily monotonic over the whole solution path, it was shown by Efron et al. [39] that the path is a piecewise linear function of B. This means that a critical set of values for B exists at which an element of $\mathbf{w}$ changes from a zero to a non-zero value. The values of all weights in between two adjacent critical values B will change linearly, which makes them easy to calculate. This is illustrated in the right plot of Figure 3.2 where vertical lines indicate the critical B that appear along the regularization path. It is possible to solve for these critical values analytically as it was proposed with the "least angle and shrinkage" (LARS) algorithm in [39] and thus retrieve the entire regularization path at a low computational cost.

3.1.4 Cross-Validation and BIC

In terms of selecting the optimal value for λ, cross-validation is a popular technique that improves the choice of λ by testing the estimated values $\hat{\mathbf{w}}$ against a validation set. Given a data set of size M and a number d of cross-validation samples, typically $d = 10$, the data is partitioned into a validation set of size $\lfloor M/d \rfloor$ and training set of size $M - \lfloor M/d \rfloor$. For each cross-validation sample d the content of the training and validation set are randomly reshuffled or picked in a unique fashion from the data set. A sufficiently large amount of λ values is evenly chosen from the regularization path. For each of the λ values the estimated weights $\hat{\mathbf{w}}_{train}$ are calculated given the training set. A mean of quantifying the robustness of these weights is to calculate the mean squared error (MSE) for the validation set based on $\hat{\mathbf{w}}_{train}$:

$$MSE = \frac{1}{M/d} \sum_{i \in val} (y_{i,val} - \hat{y}_{i,val})^2 = \sum_{i \in val} (y_{i,val} - \sum_{j=1}^{N} (x_{i,j} \hat{w}_{j,train}))^2 \tag{3.13}$$

where the symbol val refers to the data samples of the validation set and $\hat{y}$ is the predicted response of the validation set given the training set weight estimates and validation set predictor variables. In a linear regression setting the expression can also be seen as the RSS divided by the degrees of freedom, although the weight estimates come from a different data set. The MSE is calculated for each of the cross-validation samples and averaged. The λ with the lowest average MSE consequently indicates a sparse weight estimator $\hat{\mathbf{w}}$ with the highest robustness, i.e. predictive power, given new data.

The Bayesian information criterion (BIC) is another approach that can guide the choice to an optimal penalty parameter λ while penalizing over-complex models. In the context of regression analysis the BIC penalizes the log of the error variance, which is defined as the mean of the residual sum of squares, by the number of free parameters k, i.e. the non-zero weight values in the estimate $\hat{\mathbf{w}}$:

$$BIC = M \cdot log \, \frac{RSS(\hat{\mathbf{w}})}{M} + k \cdot log \, M \tag{3.14}$$

Since the BIC increases with an increase of the RSS and the number of free parameters k, a lower BIC score indicates a better fit, a less complex model, or both.

3.1.5 Optimization procedures

Several techniques exist that optimize the weights variables $\hat{\mathbf{w}}$ under the ℓ_1 penalty. The original approach by Tibshirani [136] used a quadratic program solver that converted the constraint in Equation 3.9 into a set of linear constraints that mark the boundary for the quadratic objective function. Unfortunately this method does not scale well with an increased number of variables $\mathbf{w}$. Coordinate descent tries to simplify this simultaneous variable optimization problem by optimizing each $w_j \in \mathbf{w}$ separately. The weight update can be done in a deterministic way, or randomly, whereas tracking the steepest gradient is a more advanced approach. However, coordinate descent can be slow to converge since each weight is updated at a time. Active set methods in contrast add or delete a weight variable from an active set of non-zero weight variables. They can also take a long time to converge if the active set is far away from the true solution. An advantage though is that for each active set a set of solutions for $\hat{\mathbf{w}}$ can be calculated quickly for various penalty terms $(\lambda_{min}, .., \lambda_{max})$. This is called warm starting and the popular LARS algorithm mentioned above exploits this technique and is implemented in the R package `lars`. Warm starting in combination with coordinate descent is implemented in the R package `glmnet` from Friedman et al. [47] and is used throughout this thesis together with cross-validation. Section 13.4 in the book by Murphy [108] discusses these and further techniques in more detail.

3.2 Inference with Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) algorithms are a family of estimation methods that sequentially sample random parameters with the goal to approximate a specific target probability distribution. There are various algorithms that accomplish this, such as the Metropolis-Hastings method, which is described in Section 3.2.1, Gibbs Sampling, described in Section 3.2.2, Hamiltonian sampling, slice sampling, and others. Regardless of the algorithm, in Bayesian inference the goal is always to obtain parameter samples from a target distribution that is equal to the true posterior distribution. MCMC is often used in settings where the integration over all model parameters is analytically not tractable because the model is too complex.

The MCMC originates in statistical physics and is frequently used in Bayesian inference to sample from a posterior by constructing a chain of MCMC states where the next state relies only on the current state (Markovian assumption). The Monte Carlo algorithm introduces a repeated computation of pseudo-random samples into the

Markov chain. Hence, the combination of Monte Carlo and Markov Chains allows the sequential draw of random variables that depend on past states. In particular, under these circumstances, a Markov chain produces a sequence of states (the chain) that tend to converge towards a stationary state, i.e. when the transition of one state does not change the resulting (following) state. At this stage the system is called to be in equilibrium and samples come from the true distribution. An intriguing feature is that the system at this point is independent of the starting condition, i.e. the initial state of the MCMC. Hence the equilibrium distribution has an invariant density. Samples from this equilibrium come from the target distribution which is the posterior.

In Bayesian terms this can be formulated in the following terms. Given a model with parameters $\boldsymbol{\theta}$ and the data $\mathcal{D}$, the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ is the probability of observing data $\mathcal{D}$ dependent on a model assumption, e.g. a normal distribution, and the corresponding model parameters $\boldsymbol{\theta}$. To obtain the posterior distribution, the likelihood is multiplied with the parameter prior probability $p(\boldsymbol{\theta})$ and normalized by integration over all possible parameter settings:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})d\boldsymbol{\theta}} \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta}) \tag{3.15}$$

Integrating over $\boldsymbol{\theta}$ is usually not feasible if the parameter is high-dimensional, but the posterior can still be approximated by sampling directly from $p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})$. This is accomplished with MCMC by sequentially producing samples of $\boldsymbol{\theta}$ from the un-normalized posterior and the previous state parameters. The distribution of the samples eventually converges towards the true distribution $p(\boldsymbol{\theta}|\mathcal{D})$. In some cases certain parameters in $\boldsymbol{\theta}$ can be dismissed when marginalization over the posterior for these parameters is analytically tractable. Thus, the posterior becomes a marginal posterior distribution and sampling from this distribution is limited to the remaining parameters.

In summary, a MCMC simulation constructs a chain that is a sequence of probability samples and comes from the equilibrium distribution that is equal to the true posterior distribution. Convergence must be established in order to retrieve samples from the true target distribution. It is typically reached after a sufficiently large number of samples have been sampled and the MCMC has reached a steady state. Thus, samples from the beginning of the MCMC chain should be dismissed and are labelled as belonging to the so called *burn-in* phase.

3.2.1 Metropolis-Hastings Algorithm

The Metropolis-Hastings (M-H) algorithm is named after Nicholas Metropolis and W. Keith Hastings and was proposed by Hastings [71]. The algorithm introduces a proposal and rejection mechanism that guides model parameter sampling in the MCMC chain and is quite efficient with high-dimensional models. The model parameters are typically proposed individually for the next state, thus, exploring a gradual update of parameter samples throughout the chain. In addition, the proposed parameter is accepted or rejected in dependence of a so called acceptance probability that is composed of a posterior ratio and a proposal ratio. Suppose $(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^{s-1}, \boldsymbol{\theta}^s, \ldots, \boldsymbol{\theta}^S)$ is the sequence of MCMC states described by the model parameters in a chain with length S. The following steps are necessary to generate a sample $\boldsymbol{\theta}^s$.

In the first step the state $\boldsymbol{\theta}^s$ is proposed from the state $\boldsymbol{\theta}^{s-1}$ with a so called proposal probability $q(\boldsymbol{\theta}^s|\boldsymbol{\theta}^{s-1})$. The proposal distribution can freely be chosen but will in practice effect the speed of MCMC convergence and the mixing of the samples in the chain. The mixing should be high to ensure a sufficient diversity of the samples[1]. For instance, a normal distribution that has $\boldsymbol{\theta}^{s-1}$ as a mean and an additional variance parameter will correspond to a random walk in the vicinity of the state $(s-1)$. In the second step the proposed parameter $\boldsymbol{\theta}^s$ is either rejected or accepted. The ratio of the posterior probabilities of the current and next parameter is multiplied with the ratio of the forward proposal density $q(\boldsymbol{\theta}^s|\boldsymbol{\theta}^{s-1})$ and the backward proposal density $q(\boldsymbol{\theta}^{s-1}|\boldsymbol{\theta}^s)$:

$$r = \frac{p(\boldsymbol{\theta}^s|\mathcal{D})}{p(\boldsymbol{\theta}^{s-1}|\mathcal{D})} \frac{q(\boldsymbol{\theta}^{s-1}, \boldsymbol{\theta}^s)}{q(\boldsymbol{\theta}^s, \boldsymbol{\theta}^{s-1})} \tag{3.16}$$

The sampled parameter in $\boldsymbol{\theta}^s$ is accepted if the acceptance probability $\alpha_{(s-1)\to s} = min(1, r)$ is larger than a uniformly sampled random number u in the interval $[0; 1]$, i.e. $\alpha_{(s-1)\to s} > u$. In the case that $\alpha_{(s-1)\to s} \leq u$, the sampled parameter is rejected and the new state retains the parameter of the old state with $\boldsymbol{\theta}^s := \boldsymbol{\theta}^{s-1}$. The definition of a backward proposal probability $q(\boldsymbol{\theta}^s, \boldsymbol{\theta}^{s-1})$ suggests that moving between different state has to be reversible. This is a necessary condition under which the converged chain satisfied the so called detailed balance for the target distribution $p(\boldsymbol{\theta})$ with the equilibrium $p(\boldsymbol{\theta}^s)q(\boldsymbol{\theta}^{s-1}, \boldsymbol{\theta}^s) = p(\boldsymbol{\theta}^{s-1})q(\boldsymbol{\theta}^s, \boldsymbol{\theta}^{s-1})$. Thus it can be assumed that the Markov chain will approach the stationary state with density $p(\boldsymbol{\theta})$.

[1] In fact, different strategies for mixing exist and the most common one is to promote strong mixing in the beginning of the MCMC to explore the posterior landscape extensively. Whereas, in later stages of the MCMC, when the chain converges to the posterior, the mixing is decreased to force a finer search in the posterior.

The basic form of the M-H algorithm is called Metropolis algorithm because it lacks the proposal ratio $q(\boldsymbol{\theta}^s, \boldsymbol{\theta}^{s-1})/q(\boldsymbol{\theta}^{s-1}, \boldsymbol{\theta}^s)$, which is also called the Hastings ratio, in the acceptance probability. In this case the proposal densities are symmetric, i.e. $q(\boldsymbol{\theta}^s, \boldsymbol{\theta}^{s-1}) = q(\boldsymbol{\theta}^{s-1}, \boldsymbol{\theta}^s)$. Hence, the M-H algorithms allows for asymmetric proposal densities and more flexibility. Several methods exist for the creation of an appropriate proposal function q.

3.2.2 Gibbs Sampling

The Gibbs sampler is a special case of the M-H algorithm and was introduced by Geman and Geman [53]. In contrast to the M-H algorithm, Gibbs sampling does not draw the values for the parameter vector $\boldsymbol{\theta}$ all at once, but samples each element in $\boldsymbol{\theta}$ seperately. Thus, unlike sampling the model parameters from the joint probability $p(\boldsymbol{\theta})$, each parameter $\theta_j \in \boldsymbol{\theta}$ is sampled from the full conditional distribution, i.e. conditional on the other paramters $\theta_{-j} \in \boldsymbol{\theta}$. This requires that the full conditional distribution asymptotically follows the true distribution, hence, the distribution must be known or at least very similar. If this is not the case, the Gibbs sampler should be dropped in favour of the M-H algorithm.

Let $\boldsymbol{\theta}^s = (\theta_1^s, .., \theta_N^s)$ be the parameter vector, where N is the size of the vector and s the state in the Markov chain. Each parameter in $\boldsymbol{\theta}^s$ is sampled from the conditional distribution $p(\theta_j^s | \theta_1^s, ..., \theta_{j-1}^s, \theta_{j+1}^{s-1}, ..., \theta_N^{s-1})_{j=1,..,N}$. The parameters that were already sampled for the state s, i.e. $(\theta_1^s, ..., \theta_{j-1}^s)$, influence the conditional probability of parameter j. Parameters that were not sampled in s are taken from the previous state $s-1$, i.e. the parameters $(\theta_{j+1}^{s-1}, ..., \theta_N^{s-1})$. The chain starts by setting the parameters in $\boldsymbol{\theta}^1$ to some arbitrary initial value and continues by sampling as described above. Under reasonable general conditions and a sufficiently large number of iterations, the Gibbs sampler converges towards the true distribution $p(\boldsymbol{\theta})$. This is because the full conditional distribution is proportional to the full joint distribution.

3.2.3 Reversible Jump MCMC

The previously discussed MCMC algorithms are limited to a constant size of the 'parameter space', i.e. the number of parameters in the model is constant. However, in some cases the true number of parameters can vary such as when more than one plausible model exists, e.g. multiple regression models with different predictor variables. In some cases it might also be an advantage to limit the number of parameters

with the aim to decrease model complexity, e.g. as a feature selection mechanism that infers predictors that best fit the posterior distribution while ignoring others. The latter example provides a mean by which over-fitting can be prevented and the speed of convergence increased.

The algorithm that defines these *trans-dimensional* changes is called the reversible jump MCMC (RJMCMC) and was introduces by Green [56]. The RJMCMC uses reversible jump sampling to change the dimensionality of the parameter space from one state in the chain to another. A parameter space is referred to as a candidate model M_k with the model indicator variable $k \in \{1, 2, ..., M_k\}$, where M_k can be finite or infinite. The model indicator k has the dimension d_k, which describes the dimensionality of the parameter space. The variable $\boldsymbol{\theta}_k$ is the parameter vector for the model M_k and has the dimensionality d_k. The key idea of reversible jump sampling is to propose a new model M_{k^*} from M_k that either decreases or increases the parameter dimension d_k while at the same time retaining the dimensionality of the parameter space across different models by using auxiliary random variables. For simplicity, assume that the value of the model indicator k matches the size of parameter $\boldsymbol{\theta}_k$ such that $k = d_k = |\boldsymbol{\theta}_k|$. Thus, a model M_k with indicator $k = 1$ only has a single parameter $\boldsymbol{\theta}_{k=1} = (\theta_1)$. A jump to the higher dimension with $k^* = 2$ creates an additional parameter θ_2 and the parameter vector expands to $\boldsymbol{\theta}_{k=2} = (\theta_1, \theta_2)$. To match this increase of dimensionality, the auxiliary variable u_1 is combined with the parameter $\boldsymbol{\theta}_{k=1}$ to produce the vector $\boldsymbol{\theta}_2$ with a deterministic function $f(u_1, \boldsymbol{\theta}_1)$, which can be defined for instance as $(\theta_1 + u_1, \theta_1 - u_1)$. In the case for when the dimensionality is reduced, the corresponding backward move requires a random variable u_2 that produces the parameter θ_1 with the inverse function $f^{-1}(u_2, \boldsymbol{\theta}_2)$. Note that the transformation function f has to be bijective so that there is only a single distinct solution given the variables u and $\boldsymbol{\theta}$. This condition ensures that a change of dimensionality always remains reversible, which is required to support convergence towards the stationary distribution.

In general terms, the proposal of a new model M_{k^*} from M_k can either decrease or increase the dimensionality d_k and is determined by the probability $p(k, k^*)$. The auxiliary variable u that expands or collapses the parameter space of M_k is generated with the proposal density $p(u|k, k^*, \boldsymbol{\theta}_k)$ and the parameters of the proposed state are then determined with $(u^*, \boldsymbol{\theta}_{k^*}) = f_{k \to k^*}(\boldsymbol{\theta}_k, u_k)$. Let $p(k)$ denote the prior probability of the model k, $p(\boldsymbol{\theta}_k|M_k)$ denote the prior for the parameters conditional on the model M_k, and $p(\mathcal{D}|\boldsymbol{\theta}_k, M_k)$ is the likelihood function. The ratio of likelihood, priors and proposals is defined as

$$r = \frac{p(\mathcal{D}|\boldsymbol{\theta}_{k^*}, M_{k^*})p(\boldsymbol{\theta}_{k^*}, M_{k^*})p(k^*)}{p(\mathcal{D}|\boldsymbol{\theta}_k, M_k)p(\boldsymbol{\theta}_k, M_k)p(k)} \frac{p(k^*, k)p(u^*|k^*, k, \boldsymbol{\theta}_{k^*})}{p(k, k^*)p(u|k, k^*, \boldsymbol{\theta}_k)} \left| \frac{f_{k^* \to k}(\boldsymbol{\theta}_{k^*}, u^{k^*})}{f_{k \to k^*}(\boldsymbol{\theta}_k, u^k)} \right| \quad (3.17)$$

The move is accepted with probability $\alpha_{k \to k^*} = min(1, r)$. In the case of rejection the parameter space remains unchanged and in the case of acceptance the model is updated $M_k \to M_{k^*}$ and the proposed parameter vector $\boldsymbol{\theta}_{k^*}$ replaces $\boldsymbol{\theta}_k$. The reversible jump thus generates samples from the posterior distribution $p(\boldsymbol{\theta}_k, k|\mathcal{D})$ and provides inference over the models M and the associated parameter spaces. A detailed account of MCMC inference can be found in the book on Bayesian data analysis by Gelman et al. [50].

3.3 Clustering with K-means and Gap-statistics

The K-means method is a non-hierarchical top-down clustering approach introduced by Hartigan and Wong [68]. It divides the complete data set in a predefined number of k clusters that are not bound to a hierarchy. The clusters each have a center position and the elements that are closest to a particular cluster center are assigned to it. The distance is typically measured with the squared euclidean distance, which the K-means tries to minimize by moving the centers such that the distances of its assigned elements decrease steadily. Initially, the positions of the k cluster centers are picked randomly followed by subsequent updates that move the centers to the mean position of its assigned elements. Following the center update, the elements are reassigned to the closest center until a maximum number of iterations is reached or another criterion is met, e.g. a very small movement of all centers. The K-means can get stuck in local optima whenever the centers form a cluster that is stable towards small changes of its center positions. Hence, it is recommended to repeat the K-means with different start settings and compare the results. To facilitate cluster formation, the initial center positions can also be derived from another clustering method such as a hierarchical method. This can provide starting values for the centers that are close to a true existing optima.

A disadvantage of the K-means algorithm is that it requires a predefined number of clusters. In many cases the best number of clusters is not known and has to be estimated, using techniques such as analysis of Gap-statistics [137]. This statistic compares for two settings of k the difference of the with-in cluster dispersion to the

dispersion of a reference distribution that has m samples. The K-means is run multiple times with different settings of the cluster size k, starting with the smallest value $k = 1$. For each setting of k the sum of pairwise distances is calculated, which defines the dispersion. Thus, the gap for k is calculated with

$$Gap(k) = E_m^*\{logW_k^*\} - logW_k \tag{3.18}$$

where W_k is the dispersion measure for k clusters defined as

$$W_k = \sum_{r=1}^{k} \frac{D_r}{2m_r} \tag{3.19}$$

with D_r as the sum of pairwise distances in cluster $r = (1, \ldots, k)$, and m_r as the number of elements in cluster r. The term E_m^* denotes the expectation of cluster dispersion over m samples of the reference distribution. If a gap $Gap(k)$ is greater than one standard deviation from the gap at $k + 1$, i.e. $G(k) > G(k + 1)$, then k can be considered a meaningful choice for the cluster size.

The K-means with Gap-statistics was applied to the ecological study in Chapter 5. A plot of the *Gap* function for different cluster sizes is displayed in Figure 5.18. The plot shows peaks for $k = 2$ and $k = 4$, which indicates that the data has potential clusters with these sizes.

3.4 Evaluation Methods

The major goal of the gene and ecological studies in this thesis is to learn underlying network structures given observed data that can also include interventional data. I can evaluate the performance of a method (from Chapter 2) by applying the method to synthetic data where the true underlying network is known (gold standard network). Given the gold standard and the learned network, the scoring schemes described in Section 3.4.1 calculate metrics that reflect the quality of true network recovery, and hence provide a mean by which method performance can be quantified and compared. In addition, some of the inference techniques used in this thesis employ MCMC sampling (e.g., the HBR methods, BGe, BRAM, BRAMP, see Table 2.1) to acquire samples of the network structure. These methods need to be evaluated in terms of proper convergence to the target distribution in order to avoid premature samples that poorly reflect the true underlying posterior distribution. It is therefor essential to obtain a robust estimate of the length of the previously mentioned burn-in phase and overall MCMC

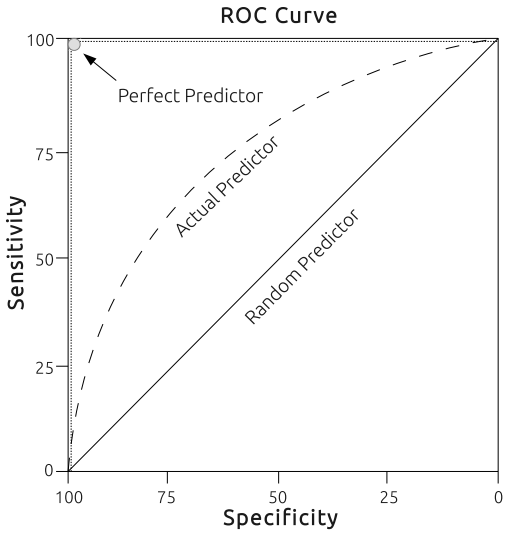


Figure 3.3: Receiver operating characteristic (ROC). A ROC curve for a perfect predictor, random expectation, and a typical predictor between these two extremes is shown. The area under the ROC curve (AUROC) is used as scoring metric.

chain length. This can be achieved by observing multiple independent MCMC chains in terms of the correlation or variation of the sampled model parameters as explained in Section 3.4.3. Another popular method is the potential scale reduction factor, which is described in Section 3.4.2 and quantifies the variation inside and between multiple MCMC chains.

3.4.1 Network Evaluation Metrics (AUROC & AUPREC)

The Receiver Operation Curve (ROC) and the Precision-Recall Curve (PREC) are the most important tools to measure the performance of network inference used in this thesis. By numerical integration of both curves one can obtain the area under the ROC curve (AUROC) and area under the PREC curve (AUPREC) as a global measure of network reconstruction accuracy, where larger values indicate a better performance. AUROC scores of 0.5 indicate random expectation and bad network recovery and AUROC scores of 1 mean perfect recovery and the best possible method performance. The interpretation of AUPREC scores is not as straight forwards and, hence, they should be always considered together with AUROC scores or in comparison with each other.

All the methods described in Chapter 2 provide a means by which interactions be-

tween genes and proteins or species can be ranked in terms of their significance or influence. If the true network is known, this ranking defines the ROC curve, where for all possible threshold values, the sensitivity or recall is plotted against the complementary specificity. The *sensitivity* is the proportion of true interactions that have been detected, i.e. the True Positive Rate, and the *specificity* is the proportion of non-interactions that have been avoided, i.e. the False Positives Rate. To obtain the area under the ROC curve I plot *sensitivity* rates along the x-axis and the *specificity* rate along the y-axis as shown in Figure 3.3. These rates can be obtained by the following equations:

$$\begin{aligned} False\ Positive\ Rate &= \frac{FP}{FP+TN} \\ True\ Positive\ Rate &= \frac{TP}{TP+FN} \end{aligned} \tag{3.20}$$

where FP are the false positives, TN are the true negatives, TP are the true positives and FN are the false negatives. In order to construct a curve for different degrees of sensitivity and specificity, the ROC method operates in the following way: Assume a ranked vector of indicators symbolized with $(p_1, \ldots, p_I)$, and a binary vector of the same size with $(b_1, \ldots, b_I)$, where I refers to the total number of indicators[2] and an element in b is 1 if an indicator is positive or existing, or 0 if the indicator is negative or non-existing. The latter vector is called the gold standard and is used as the benchmark against the former vector, which is typically inferred. Furthermore, define a threshold interval $\tau \in (0; 1]$ that is iteratively increased from a low value, such as 0.01, to the maximum of 1. In each iteration of τ, mark an indicator i as existing (1) if $p_i \leq \tau$, and non-existing (0) if $p_i < \tau$. Compare these indications to the real indicator b_i. In the case the indicators match and exist, i.e. have the value 1, a true positive is recovered. In the case that the indicators do not match and $b_i = 0$, a false positive is recovered. Similarly the false negative and false positive values are recovered, and consequently Equation 3.20 can be used to calculate the False Positive Rate (FPR) and True Positive Rate (TPR) for all values of τ. The FPR is plotted along the x-axis and the TPR along the y-axis. Gaps in the curve can occur and should be interpolated in a linear fashion. Finally, all points are connected and the area under the curve is calculated.

The Precision-Recall curve is constructed in the same way as the ROC curve but the

[2]The indicators can be posterior probabilities or confidences derived from an inference method. The vector indicator includes the interactions of the fully connected network.

x-axis corresponds to the Recall and the y-axis to the Precision. Both rates are defined as follows

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP}. \end{aligned} \tag{3.21}$$

This curve has the advantage over the ROC curve that the influence of a large amount of false positive edges can be better identified through the precision, whereas, a large amount of false positives will have a small effect on the False Positive Rate because it appears both in the numerator and denominator of Equation 3.20. Note that the interpolation of the curve is not linear as with the ROC curve because a change of the Recall value does not necessary mean a linear change of the Precision. Instead, I use the scheme proposed in Section 4 from Davis and Goadrich [34] to interpolate intermediate values. Inconsistencies in the AUPREC score can arise under certain circumstances as demonstrated in Appendix B. In these cases, missing values need to be extrapolated by using the interpolation procedure in [34].

Note that for the gene regulation study in Chapter 4 I only use AUROC scores following the suggestions of a study by Grzegorczyk and Husmeier [61] that observes little difference of AUROC and AUPREC scores for networks with low complexity. For the ecological study I use both scores since I found no evidence that supports the same claim for ecological data.

3.4.2 Potential Scale Reduction Factor

The Potential Scale Reduction Factor (PSRF) is a measure that quantifies the convergence of multiple MCMC chains and was described by Gelman and Rubin [51]. Given the assumption that several MCMC chains depend on the same data and are initialized with different model parameters, the variation between the chains and within each single chain reflects the degree of convergence. The variation between the chains is used to overestimate the so called target variance, which is regarded as the true sequence variance of a chain assuming the MCMC has reached convergence. The ratio of the target variance to the within sequence variance is the PSRF factor, i.e. the reduction factor that becomes smaller when the within sequence variance approaches the target variance. In other words: A low difference of the between chain sequence variance and

within chain sequence variance indicates chain convergence of the MCMC.

The PSRF calculation requires the output of at least two chains $c \geq 2$. The chain sequence must be represented by a single value for each chain, which is called the scalar estimate x. Given c chain sequences of length n, the between sequence variance B is calculated with

$$B = \frac{n}{c-1} \sum_{i=1}^{c} (\overline{x}_i - \overline{x})^2 \tag{3.22}$$

where $\overline{x}$ is the mean of the scalar estimate over all sequences $i = (1, .., c)$ and $\overline{x}_i$ is the mean of x in sequence i. The within sequence variance is defined as

$$W = \frac{1}{m} \sum_{i=1}^{c} \sum_{j=1}^{n} \frac{1}{n} (x_{ij} - \overline{x}_i)^2 \tag{3.23}$$

where the last sum is the sequence variance and x_{ij} annotates the scalar estimate with index $j = (1, .., n)$ in sequence i. The target variance is overestimated by adding up the weighted variances: $\hat{\sigma}^2 = \frac{n-1}{n} W + \frac{1}{n} B$. Finally, the PSRF is calculated with $PSRF = \sqrt{\hat{R}} = \sqrt{\hat{\sigma}^2 / W}$. As long as the chains have not converged, W is smaller than $\hat{\sigma}^2$ but with increasing n and longer chains, W approaches the target variance and the PSRF approaches 1 from above. Values of the PSRF that are greater than 1.2 indicate weak convergence and the simulation should be run longer. Values below 1.05 are a sign of strong convergence. Note that the PSRF is typically calculated for a sliding window of sub-sequences from the chain with a sufficiently large size n. This is to avoid interference of earlier iterations that could be badly converged and might account for samples from the burn-in phase (see left plot in Figure 5.7). In the case that multiple scalar estimates are present in a chain as it is for instance with a predictor vector in $\boldsymbol{\theta}$, the PSRF values can be summarizes in percentage of values below a certain threshold as displayed in the right plot of Figure 5.7.

3.4.3 Interaction Posterior Probability Correlation

The interaction posterior probability is a measure that quantifies the chance of observing regulator to response interactions $x_i^n \rightarrow y_n$ (see Section 4.3.4). In the case of MCMC simulations, such as with the Bayesian regression methods, the probability is marginalized over a specific number of samples, typically with 2000 iterations in size or larger. The convergence of a MCMC can be evaluated by comparing the probabilities of different MCMC chains. The interactions posteriors can be displayed in a scatter

plot that compares two MCMC chains as shown in the left plot of Figure 5.6, or as a time-varying trajectory of the correlation value between the interaction probabilities of two chains as shown in the right plot of Figure 5.6. Convergence is indicated with higher correlation values, which means that the sampled interactions of two chains are likely to occur at the same rate, independent of the starting condition.

Chapter 4

Statistical Inference of Gene Regulatory Networks

In this chapter I assess the accuracy of various state-of-the-art statistics and machine learning methods described in Chapter 2 for reconstructing gene and protein regulatory networks in the context of circadian regulation. The study draws on the increasing availability of gene expression and protein concentration time series for key circadian clock components in *Arabidopsis thaliana*. In addition, gene expression and protein concentration time series are simulated from a recently published regulatory network of the circadian clock in *A.thaliana*, in which protein and gene interactions are described by a Markov jump process based on Michaelis-Menten kinetics. I closely follow recent experimental protocols, including the entrainment of seedlings to different light-dark cycles and the knock-out of various key regulatory genes. The study provides relative network reconstruction accuracy scores for a critical comparative performance evaluation, and sheds light on a series of highly pertinent questions: it investigates the influence of systematically missing values related to unknown protein concentrations and mRNA transcription rates, it quantifies the dependence of the performance on the network topology and the degree of recurrency, it provides deeper insight into when and why non-linear methods fail to outperform linear ones, and it provides a comparison between inferred and published gene regulatory network structures.

This study was published in collaboration with Marco Grzegorczyk in [1]. Both of us contributed in the discussion of the experimental setup, in running simulations, and evaluating the results. Marco Grzegorzyk performed simulations with the methods MBN, SSM, BGe, and ARD-SBR and I performed simulations with the remaining

methods. The work is based on preliminary studies that was published in [4] and [57].

4.1 Introduction

Plants have to carefully manage their resources. The process of photosynthesis allows them to utilize sunlight to produce essential carbohydrates during the day. However, the earth's rotation predictably removes sunlight, and hence the opportunity for photosynthesis, for a significant part of each day, and plants need to orchestrate the accumulation, utilization and storage of photosynthetic products in the form of starch over the daily cycle to avoid periods of starvation, and thus optimize growth rates.

In the last few years, substantial progress has been made to model the central processes of circadian regulation, i.e. the mechanism of internal time-keeping that allows the plant to anticipate each new day, at the molecular level [113, 62]. Moreover, simple mechanistic models have been developed to describe the feedback between carbon metabolism and the circadian clock, by which the plant adjusts the rates of starch accumulation and consumption in response to changes in the light-dark cycle [42]. What is needed is the elucidation of the detailed structure of the molecular regulatory networks and signalling pathways of these processes, by utilization and integration of transcriptomic, proteomic and metabolic concentration profiles that become increasingly available from international research collaborations like AGRON-OMICS[1] and TiMet-Consortium [138]. The inference of molecular regulatory networks from post-genomic data has been a central topic in computational systems biology for over a decade. A variety of methods have been proposed and several procedures have been pursued to objectively assess the network reconstruction accuracy [80, 149, 148]. The objective of the present Chapter is to complement these studies in six important respects. Firstly, I have taken a particular focus on circadian regulation. To this end, I have taken the central circadian clock network in *A. thaliana*, as published by Guerriero et al. [62], as a ground truth for evaluation, and closely followed recent experimental protocols for data generation, including the entrainment of seedlings to different light-dark cycles, and the knock-out of various key regulatory genes. To make the data generated from this network as realistic as possible, I have modelled gene and protein interactions as a Markov jump process (MJP) based on Michaelis-Menten kinetics. This is to be preferred over mechanistic models based on ordinary differential equations [used e.g. by 113], as MJPs capture the intrinsic stochasticity of molecular interactions. MJPs

[1] https://www.agronomics.ethz.ch/

also avoid artefacts that result from the dynamics of ordinary differential equations converging to stable limit cycles with completely regular oscillations, which are never observed in actual experiments [62]. Secondly, I have assessed the impact of missing values on the reconstruction task. Protein-gene interactions affect transcription rates, but both these rates as well as protein concentrations might not be available from the wet lab assays. In such situations, mRNA concentrations have to be taken as proxy for protein concentrations, and rates have to be approximated by finite difference quotients. For both approximations, I have quantified the ensuing deterioration in network reconstruction accuracy. Thirdly, I have investigated the dependence of the network reconstruction accuracy on the degree of connectivity and recurrency in the network topology. The central circadian clock network is densely connected with several tight feedback loops. However, I expect the regulatory network, via which the clock acts on carbon metabolism, to be sparser and with more feed-forward structures. In my study I have therefore quantified how the network reconstruction depends on the degree of recurrency, and how the performance varies as critical feedback cycles are pruned. Fourthly, I have investigated the effect of non-linear transformations of the data, suggested by the underlying chemical kinetic equations (Michaelis-Menten), and I have proposed a novel combination of hierarchical Bayesian models with multiple change point processes. Fifthly, I have included a substantial range of different state-of-the-art models, which to my knowledge has not been attempted before. This includes mutual information based methods, graphical Gaussian models, sparse regression methods, automatic relevance determination, hierarchical Bayesian regression, change-point processes, Gaussian mixture models, Bayesian networks, Bayesian spline autoregression models, state space models, and Gaussian processes. I have carried out a systematic comparative model evaluation with an ANOVA scheme to distinguish genuine differences in model performance from exogenous factors and confounding effects. Finally, my study includes a performance evaluation on novel qRT-PCR gene expression time series from *A.thaliana*, which was provided by the TiMet project [138, 43].

4.2 Regression model

The starting point of my study is the mathematical formulation of transcriptional regulation introduced by Barenco et al. [10],

$$y_{n,m} = \frac{dx_{n,m}}{dm} = \alpha_n + f_n(\mathbf{x}_{\boldsymbol{\pi}_n,m}) - \lambda_n x_{n,m} \tag{4.1}$$

where $x_{n,m}$ is the mRNA concentration of gene n at time[2] m, α_n is the basal transcription rate for gene n, λ_n is the mRNA degradation rate for gene n, $f_n(.)$ is an unknown regulation function, and $\mathbf{x}_{\boldsymbol{\pi}_n,m}$ is the set of gene expression values of the putative regulators $\boldsymbol{\pi}_n$ of gene n at time m, as explained above. This fundamental equation provides the basis for learning and inference in systems biology, as e.g. described by Lawrence et al. [88]. A common approach is to approximate the time derivative on the left-hand side by a finite difference quotient:

$$\frac{dx_{n,m}}{dm} \approx \frac{x_{n,m+\Delta m} - x_{n,m}}{\Delta m} \tag{4.2}$$

which for a unit time delay $\Delta m = 1$ leads to

$$x_{n,m+1} = x_{n,m} + \alpha_n + f_n(\mathbf{x}_{\boldsymbol{\pi}_n}, m) - \lambda_n x_{n,m} = h(x_{n,m}, \mathbf{x}_{\boldsymbol{\pi}_n,m}) \tag{4.3}$$

for some function $h(x_{n,t}, \mathbf{x}_{\boldsymbol{\pi}_n,t})$. This equation provides the basis for a variety of 'dynamic' algorithms, including dynamic Bayesian networks [80], time-delay mutual information methods [156] and time-shifted regression methods [106]. However, as I demonstrate in more detail in Section 4.6.5, the finite difference approximation of Equation (4.2) is not particularly good, and I therefore work with the explicit representation of Equation (4.1). This might look like a 'static' method, as no time-shift operation is needed, but the dynamics are explicitly represented by the time derivative $y_{n,m} = \frac{dx_{n,m}}{dm}$. The data for my study consists of mRNA concentration profiles $\{x_{1,m}, \ldots, x_{N,m}\}_{m=1,\ldots,M}$ and associated protein concentration profiles $\{x_{p,1,m}, \ldots, x_{p,N^\star,m}\}_{m=1,\ldots,M}$. For each individual gene $n = 1, \ldots, N$ I use its observed concentrations $x_{n,1}, \ldots, x_{n,M}$ to compute the gradients $y_{n,m}$ $(m = 1, \ldots, M)$, and I then consider the gradients $y_{n,1}, \ldots, y_{n,M}$ as realizations of a target variable y_n, which monitors the transcription rates of gene n over time. Henceforth, I refer to the variable y_n and its realizations $y_{n,m}$ $(m = 1, \ldots, M)$ as the mRNA concentration time derivatives, gradients, or transcription rates synonymously. Mathematically, my goal is to find the regulators of each target variable y_m $(m = 1, \ldots, N)$, i.e. to identify variables with an effect on the transcription rates y_n of gene n. I distinguish two scenarios: In the *incomplete* data scenario the potential regulators for target variable y_n are the observed

[2] I use the symbol m instead of t for the time to follow the general convention for a "sample" or "observation" as used throughout this thesis.

mRNA concentrations of the genes $\{x_{1,m}, \ldots, x_{N,m}\}_{m=1,\ldots,M}$, including the concentrations $\{x_{n,m}\}_{m=1,\ldots,M}$ of the target gene n themselves. In the *complete* data scenario I consider the protein rather than the mRNA concentrations as potential regulators. To be consistent with the fundamental equation of transcription, Equation (4.1), $x_{n,m}$ will always be included in either scenario; I won't mention that explicitly in the text.

For consistency with the fundamental equation of transcription, Equation (4.1), I will enforce that each regulator set $\boldsymbol{\pi}_n$ for y_n contains the concentration x_n of n, symbolically $x_n \in \boldsymbol{\pi}_n$. Thereby — as the transcription rate y_n of gene n will certainly depend on its mRNA concentration x_n — I add the mRNA concentrations of gene n to the protein profiles. The potential regulators for y_n are then given by $\{x_{n,m}, x_{p,1,1} \ldots, x_{p,N^\star,m}\}_{m=1,\ldots,M}$. However, I ignore this distinction in the methodological definitions, and use the term regulators generically for both scenarios.

4.3 Method Extensions

The methods that take part in this study have been previously described in Chapter 2. An exception are the following modifications to the hierarchical Bayesian regression (HBR) method that take the genetic data set into special consideration. The time-varying dynamic of the plant data is under the major influence of light and darkness that are expressed typically over 24 hours. I exploit to change-process of the HBR to model the light and dark phase as explained in Section 4.3.1 with HBR-light. Furthermore I modify the HBR in such a way that change-points are applied to the amplitude of mRNA response gradients (Section 4.3.2, HBR-cps). I anticipate a substantial improvement on the approximation of Michaelis-Menten dynamics with this approach. A simple, yet effective, method is the expansion of the explanatory data with product and non-linear terms as described in 4.3.3. Although, this does not modify the HBR method itself, I will refer to this expansion as HBR-nl.

4.3.1 Fixed change-point induced by the external light condition (HBR-light)

Since light may have a substantial effect on the regulatory relationships of the circadian clock, I divide the observations of the target variables into two segments according to a binary light phase indicator: $h = 1$ (light) versus $h = 2$ (darkness). This reflects the nature of the laboratory experiments, where *A.thaliana* seedlings are grown in an artificial light chamber whose light is switched on or off. It is straightforward to

generalize this approach to more than two segments to allow for extended dawn and dusk periods in natural light. Given that the light phase is known, I consider the segmentation as fixed, and I refer to the model as the hierarchical Bayesian regression (HBR) model with two light-induced components (HBR-light). Since I also assume that light has a substantial influence, I do not penalize any differences between the interaction parameters associated with the two light phases and apply the *uncoupled* non-homogeneous Bayesian regression model, shown in the right panel of Figure 2.1.

4.3.2 Change-points in the amplitude of the target variable (HBR-cps)

To approximate the non-linear dynamics of the Michaelis-Menten kinetics, I sort the realizations $y_{n,1}, \ldots, y_{n,M}$ of each target variable, y_n, in increasing order to obtain the order statistics $y_{n,(1)} \leq \ldots \leq y_{n,(M)}$.[3] Applying the non-homogeneous Bayesian regression models to the ordered realizations, $y_{n,(1)}, \ldots, y_{n,(M)}$, then effectively yields a segmentation of the realizations, $y_{n,1}, \ldots, y_{n,M}$, with respect to the amplitude of the target variable y_n. To infer the number of change-points and the change-point locations, I again follow Grzegorczyk and Husmeier [60] and use a point process prior, where the distance between two successive change-points, $M_{n,h} = \tau_{n,h+1} - \tau_{n,h}$, is assumed to have a negative binomial distribution with hyper-parameters $p \in [0,1]$ and $k = 1$, symbolically $M_{n,h} \sim NBIN(p,1)$. I apply both variants of the non-homogeneous Bayesian regression model. The uncoupled variant is shown in the right panel of Figure 2.1, and I set $\mathbf{m}_{n,h} = \mathbf{0}$ for all $h \geq 0$ in Equation (2.11). In the coupled variant the regression parameter vectors, $\mathbf{w}_{n,h}$ $(h = 1, \ldots, H_n)$, are sequentially coupled via Equations (2.11-2.12). I refer to these hierarchical Bayesian regression models as the change-point-divided hierarchical Bayesian regression models (HBR-cps).

4.3.3 HBR with additional non-linear terms

A straightforward extension of the HBR method is to include non-linear terms in the design matrix $\mathbf{X}_{\boldsymbol{\pi}_n}$. In my study I tested, as an alternative to the HBR model just described, the inclusion of quadratic and inverse terms. So for a set of regulators $\boldsymbol{\pi}_n = \{A, B\}$, the columns of design matrix $\mathbf{X}_{\boldsymbol{\pi}_n}$, $[1, x_A(m), x_B(m)]'$ are replaced by $[1, x_A(m), x_B(m), x_A(m), x_B(m), 1/x_A(m), 1/x_B(m)]'$, where the inverse terms are included for a better approximation of the Michaelis-Menten kinetics, and the mixed

[3] For each y_n I apply exactly the same permutation to order the realizations of the explanatory variables (covariates) and thereby ensure that the segment-specific design matrices are built properly.

term is included for a better modelling of heterodimer effects. I refer to this extension of the HBR model as the non-linear HBR (HBR-nl) model.

4.3.4 Marginal interaction posterior probabilities

For the previously described hierarchical Bayesian regression models (HBR, HBR-nl, HBR-light, and HBR-cps) MCMC simulation techniques are employed to generate samples from the posterior distributions. Keeping only the sampled regulator sets, $\boldsymbol{\pi}_n^{(1)}, \ldots, \boldsymbol{\pi}_n^{(S)}$, where $(1, \ldots, S)$ indexes the generated MCMC samples, corresponds to a marginalization over all other sampled parameters. An estimator of the marginal posterior probability of a regulatory interaction between the regulator x_i^n and the target variable y_n, symbolically $x_i^n \rightarrow y_n$, is then given by the fraction of regulator sets that contain x_i^n:

$$P(x_i^n \rightarrow y_n) = \frac{1}{S} \sum_{s=1}^{S} I(x_i^n \in \boldsymbol{\pi}_n^{(s)}) \tag{4.4}$$

where $I(x_i^n \in \boldsymbol{\pi}_n^{(s)})$ is an indicator function, which is 1 if x_i^n is in the set of regulators $\boldsymbol{\pi}_n^{(s)}$, and zero otherwise. For the hierarchical Bayesian regression models I use the marginal interaction posterior probabilities to score the interactions with respect to their strengths.

4.4 Data

This section describes the data used for a critical comparative assessment of the method performance. I use a combination of *real* laboratory data and *realistic* simulated data.

Real data have the advantage that they were obtained from real organisms using real assays. In my case, these are transcriptional concentration time courses from *A.thaliana* seedlings obtained with quantitative reverse transcription polymerase chain reaction (qRT-PCR). The use of real data mimics the actual application a biologist is interested in. A disadvantage, however, is the absence of a ground truth, making it difficult to evaluate the prediction from the different methods.

Realistic data are simulated from a mathematical model of the molecular interactions occurring in the signalling pathways and regulatory networks. Since the data have been synthetically generated, the ground truth is known and can be used for an objective performance evaluation. A disadvantage is that the data generation process might

Elementary Molecular Reaction	
$X_{DNA} + X_{protein} \xrightarrow{k_1} X_{DNA} + X_{mRNA} + X_{protein}$	Transcription
$X_{mRNA} \xrightarrow{k_2} X_{mRNA} + X_{protein}$	Translation
$X_{mRNA} \xrightarrow{k_3} \emptyset, X_{protein} \xrightarrow{k_4} \emptyset$	Degradation
$2X_{protein} \xrightarrow{k_5} X_{dimer}$	Dimerisation

Table 4.1: Illustration of elementary molecular reactions with discrete stochastic kinetics. The letter "X" represents a single molecule of the type indicated by the subscript, the symbol $\emptyset$ indicates the disappearance of a molecule. Arrows indicate reactions, i.e. the transformation of the products on the left to the products on the right. The lower case letters above the arrows denote chemical kinetic parameters. The reactions are modelled mathematically with a Markov jump process. Reactions occur stochastically according to a Poisson process, whose intensity is the sum of the kinetic parameters; here: $\lambda = k_1 + \ldots + k_5$. The propensity of a reaction is proportional to its kinetic parameter, i.e. given that a reaction has occurred, the probability that the nature of this reaction is of type i is k_i/λ.

make simplifying assumptions that render the data insufficiently representative of real biological systems studied in the laboratory. The challenge, hence, is to make the data generation process as realistic as possible, and I describe below how I have accomplished this objective.

4.4.1 Generation of realistic data

Various mathematical models have been developed to describe the molecular interactions and signal transduction processes in the central circadian clock of *A.thaliana* [93, 113, 115]. They are based on systems of ordinary differential equations (ODEs) that describe the chemical kinetics of transcription initiation, translation, and post-translational modification, using mass action kinetics and/or Michaelis-Menten kinetics. In principle, I could use these mathematical models together with the published values of the kinetic rate parameters to generate synthetic transcription profiles from the circadian regulatory networks published by Locke et al. [93] and Pokhilko et al. [113, 115], then use the latter as a gold standard for my method evaluation.

However, this approach would not generate data that are sufficiently biologically realistic. The solutions of ODEs typically converge to limit cycles with regular oscillations and constant amplitude, which fail to capture the stochastic amplitude variation observed in real qRT-PCR experiments. In addition, the damping of oscillations exper-

Chemical Kinetics Described by Ordinary Differential Equations (ODEs)

mRNA Concentration Change

$$\frac{dPRR9_{mRNA}}{dt} = q_3 \cdot light \cdot P_{protein} + n_7 \cdot \frac{g_8^h}{g_8^h + TOC1_{protein}^h} \cdot \frac{LHY_{protein}^i}{LHY_{protein}^i + g_9^i} - m_{12} \cdot PRR9_{mRNA}$$

Protein Concentration Change

$$\frac{dPRR9_{protein}}{dt} = p_8 \cdot PRR9_{mRNA} - (m_{13} \cdot light + m_{22} \cdot dark) \cdot PRR9_{protein}$$

Discrete Stochastic Kinetics of Molecular Reactions

mRNA Count Update

$$PRR9_{mRNA} = PRR9_{transcr} \uparrow + PRR9_{mRNA.degrad} \downarrow$$

$$PRR9_{transcr} = \Omega \cdot \left(\frac{q_3}{\Omega} \cdot light \cdot P_{protein} + \frac{(g_8 \cdot \Omega)^h}{(g_8 \cdot \Omega)^h + TOC1_{protein}^h} \cdot \left(n_4 + n_7 \cdot \frac{LHY_{protein}^i}{LHY_{protein}^i + (g_9 \cdot \Omega)^i} \right) \right)$$

$$PRR9_{mRNA.degrad} = m_{12} \cdot PRR9_{mRNA}$$

Protein Count Update

$$PRR9_{protein} = PRR9_{translate} \uparrow + PRR9_{protein.degrad} \downarrow$$

$$PRR9_{translate} = p_8 \cdot PRR9_{mRNA}$$

$$PRR9_{protein.degrad} = (m_{13} \cdot light + m_{22} \cdot dark) \cdot PRR9_{protein}$$

Table 4.2: Ordinary differential equations (ODEs) and corresponding discrete molecular reaction kinetics for the morning gene PRR9. The symbol "$PRR9_x$" denotes the concentration of a molecular species of the morning gene PRR9, specified by the index "x". For instance, $PRR9_{mRNA}$ is the concentration of mRNA transcribed from PPR9, $PRR9_{protein}$ is the concentration of PRR9 protein, etc.. The symbol *light* is a binary indicator for the status of light (1=light, 0=darkness), *dark* = 1-*light*, lower case letters indicate kinetic parameters, and Ω is a volume parameter. *Top panel*: ODE description of chemical kinetics, with non-linear Michaelis-Menten kinetics for mRNA concentration change, and linear mass action kinetics for protein concentration change. *Bottom panel:* The corresponding discrete kinetic reactions, which in the limit $\Omega \to \infty$ converge to the ODE solutions. An upper arrow $\uparrow$ on the right indicates an amount by which the quantity on the left is increased, a down arrow $\downarrow$ on the right indicates an amount by which the quantity on the left is decreased. The reactions occur stochastically, with propensities determined by the reaction rates. Mathematical details can be found from Wilkinson [153]. The complete set of equations for all genes in the central circadian clock of *A.thaliana* is available from Guerriero et al. [62].

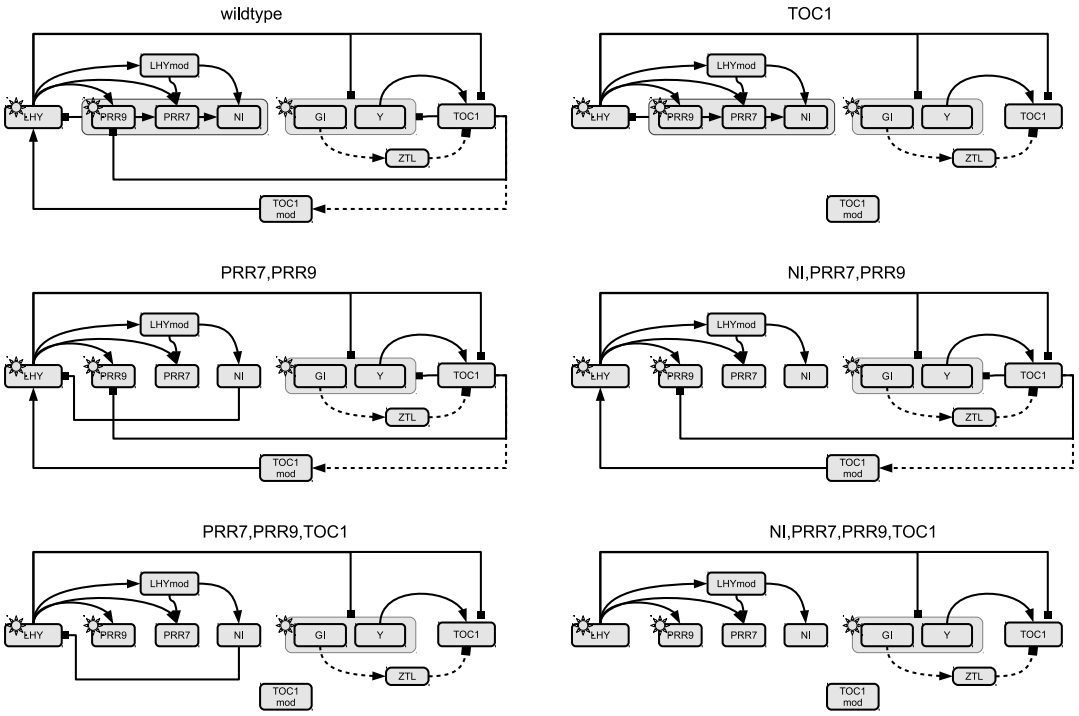


Figure 4.1: Model network of the circadian clock in *A.thaliana* and network modifications. Each graph shows interactions among core circadian clock genes with different degrees of interconnectedness. Solid lines show protein-gene interactions; dashed lines show protein modifications; and the regulatory influence of light is symbolized by a sun symbol. The top left panel ('wildtype') shows the network structure published by Pokhilko et al. [114]. The remaining panels show modified network structures, corresponding to subsequent pruning of the wildtype network. This is realized by artificially disabling certain proteins (displayed in the panel title) to act as transcription factor and thus loosing their regulatory function on mRNA transcription that existed in the wildtype network. The expression of the associated mRNA of these proteins is not affected. Grey boxes group sets of regulators or regulated components. Arrows symbolize activations and bars inhibitions.

imentally observed in constant light conditions is not correctly modelled. The problem of ODEs is that the intrinsic fluctuations of molecular processes in the cell are ignored, thereby not allowing for molecular noise that may have a significant impact on the behaviour of the system [62, 152].

For a more realistic approach, I model the individual molecular processes of transcription, translation, degradation, dimerisation etc. as individual discrete events, as shown in Tables 4.1 and 4.2. Statistical mechanics arguments then lead to a Markov jump process in continuous time whose instantaneous reaction rates are directly proportional to the number of molecules of each reacting component [152, 153]. Such dynamics can be simulated exactly using standard discrete-event simulation techniques, as illustrated in Table 4.1. For my study, I followed Guerriero et al. [62] and adopted the Bio-PEPA framework from Ciocchetta and Hillston [29] to simulate gene expression profiles for the core circadian clock of *A.thaliana*, using the Bio-PEPA Eclipse Plug-in[4]. This framework is built on a stochastic process algebra implementation of chemical kinetics, and the stochastic simulations are run with the Gillespie algorithm [54]. Figure 4.2 illustrates such stochastically generated mRNA time series data using Bio-PEPA and the corresponding real data from qRT-PCR measurements for two components of the circadian clock.

In order to correctly quantify stochastic fluctuations, concentrations are represented as numbers of molecules per unit volume. This requires the unit volume size Ω to be defined, which scales the molecule amounts and kinetic laws such that a unit concentration in an ODE representation becomes a molecule count close to Ω; see Guerriero et al. [62] for more details. The size of Ω has a strong influence on the stochasticity of the system. Since larger volumes entail a more pronounced averaging effect, the stochasticity decreases with increasing values of Ω, and the solutions from the equivalent deterministic ODEs are subsumed as a limiting case for $\Omega \to \infty$. Conversely, decreasing values of Ω increase the stochasticity. Guerriero et al. [62] showed that replacing the continuous deterministic dynamics of ODEs by the discrete stochastic dynamics with an appropriate choice of Ω leads to a more accurate matching of the experimental data, including the damping of oscillations experimentally observed in constant light, better entrainment to light in several light patterns, better entrainment to changes in photo period, and the correct modelling of secondary peaks experimentally observed for certain photo periods.

I simulated mRNA and protein concentration profiles over time from the circadian clock regulatory network published in Guerriero et al. [62] and Pokhilko et al. [114], shown in Figure 4.1 (top left, network 'wildtype') and Figure 4.17 (middle left, network 'P2010'). This involves genetic regulatory reactions for mRNA transcription, protein translation, and mRNA and protein degradation for 7 genes. Figure 4.3 shows the

[4]http://www.biopepa.org

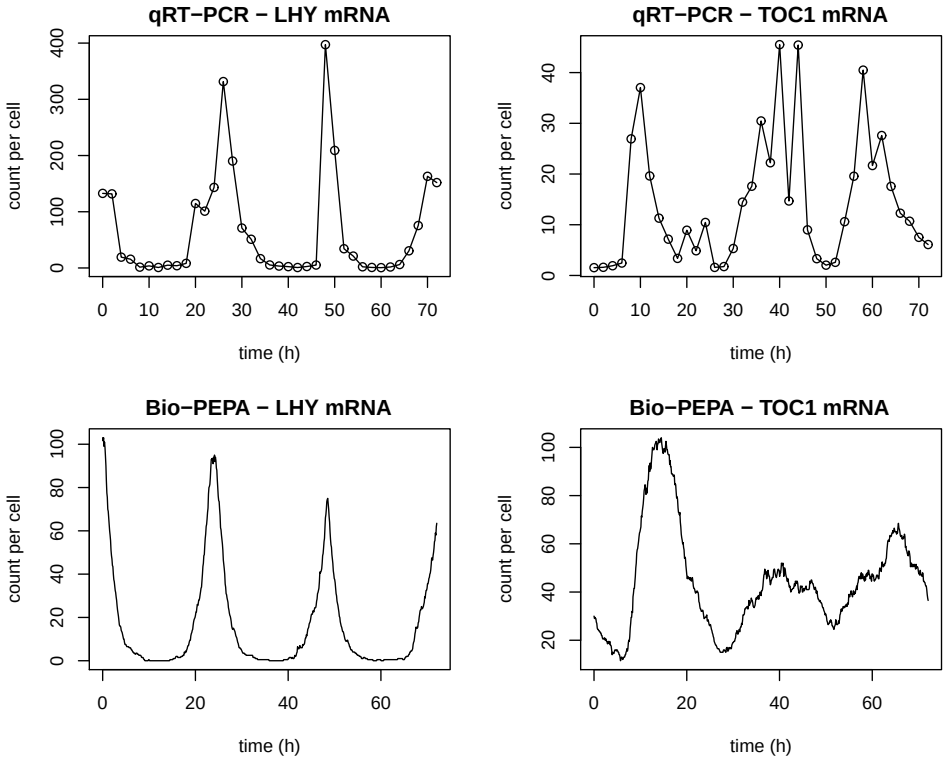


Figure 4.2: Real mRNA data in comparison to generated data. The top plot shows the qRT-PCR time series data for the LHY and TOC1 mRNA of *A.thaliana* with two hour measurement intervals (derived from Section 4.4.2, 'TiMet' [138]). The bottom panels show the corresponding synthetic measurements for the stochastically simulated data described in Section 4.4.1 with a unit volume of $\Omega = 100$.

trajectories of the mRNA and protein measurements for 6 of the 7 components of the clock (mRNA/protein for hypothetical Y is not displayed) for a regular day with 12 hour light and 12 night. Table 4.2 shows the underlying chemical kinetic reactions for a single component in this network (PRR9), as an illustration. A full list of reactions and their corresponding mathematical descriptions is available from the supplementary material from Guerriero et al. [62].

An additional advantage of this procedure is that it is straightforward to assess the effect of network structure modification on the performance of the network reconstruction methods. This can easily be effected by inactivating certain reactions in the gold standard network, by setting the respective reaction rates to zero. Figure 4.1 shows the complete circadian regulatory network in *A.thaliana*, as published by Guerriero

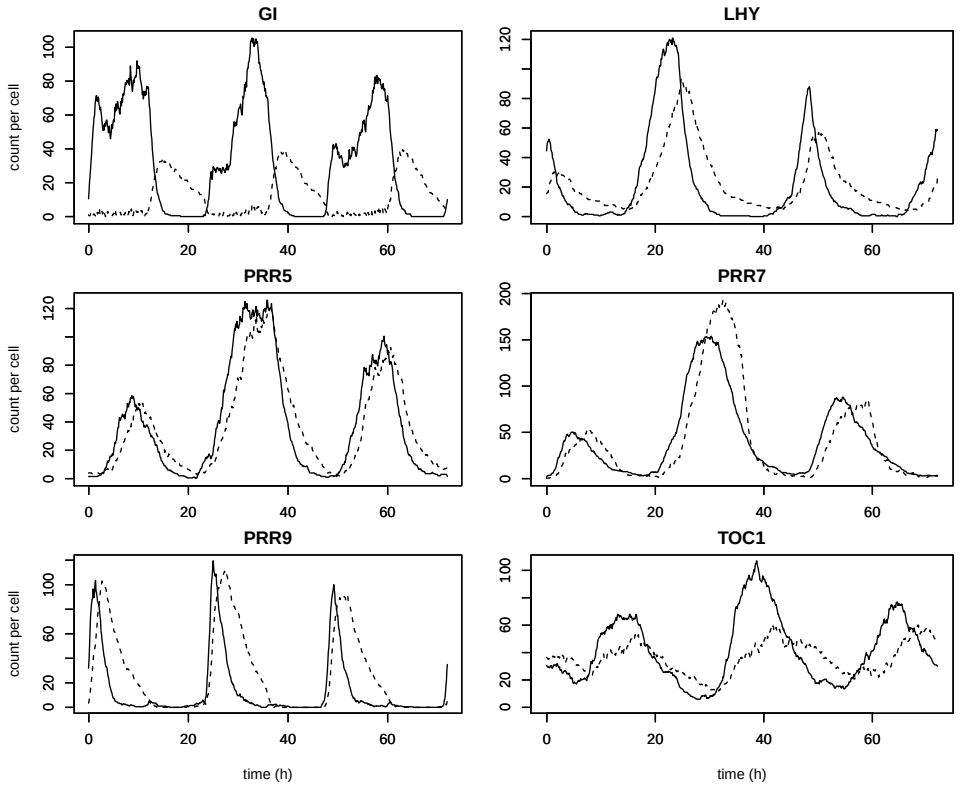


Figure 4.3: Synthetically generated mRNA and protein time series using Marcov Jump Processes (MJP) with Bio-PEPA. Each panel shows the mRNA (solid line) and corresponding protein profiles (dashed line) for different components of the circadian clock of *A.thaliana* as described in Section 4.4.1. The light conditions in this particular data set is a regular day with 12 hour light and 12 hour darkness, without any knock-outs, and a unit volume of $\Omega = 100$. Note the long time-delay between the mRNA and protein concentration of 'GI' (top left panel). This is because the formation of this protein depends on the protein Zeitlupe (ZTL, not shown), which exhibits a substantial phase shift compared to the 'GI' mRNA expression.

et al. [62] and Pokhilko et al. [114] ('wildtype'), and several modified sparser structures, which are used throughout my study. The exact setup of the data generation process is described in detail in Section 4.5.1.

4.4.2 Real data

In addition to the realistic data simulated from a mathematical description of the molecular interaction processes, as described above, I used real transcription profiles

for the key circadian regulatory genes in the model plant *A.thaliana*. The objective is to infer putative gene regulatory networks with the various statistical methods described in Section 4.2, and then to compare these predictions with network models of the circadian clock from the biological literature [92, 86, 75, 114, 113, 115]. It is important to note that, as opposed to the realistic data described in the previous subsection, no proper ground truth exists. Besides the fact that these models show noticeable variations, they were not obtained on the basis of proper statistical model selection, as described e.g. by Vyshemirsky and Girolami [145]. Nevertheless, a qualitative comparison will reveal to what extent the postulated interaction features and structural network characteristics from the literature are consistent with those inferred from the data.

The data used in my study come from the EU project 'TiMet' [138], whose objective is the elucidation of the interaction between circadian regulation and metabolism in plants. The data consist of transcription profiles for the core clock genes from the leaves of various genetic variants of *A.thaliana*, measured with qRT-PCR. The study involves two wildtypes of the strains Columbia (Col-0) and Wasilewski (WS) and 5 clock mutants, namely a double knock-out 'LHY/CCA1' in the WS strain, a single knock-out of 'GI' and 'TOC1' in the strain Col-0, a double-knock-out 'PRR7/PRR9' in strain Col-0, and a single knock-out of 'ELF3'. The plants were grown in the following 3 light conditions: a diurnal cycle with 12 hours light and 12 hour darkness (12L/12D), an extended night with full darkness for 24 hours (DD), and an extended light with constant light (LL) for 24 hours. An exception is the 'ELF3' mutant, which was grown only in 12L/12D condition. Samples were taken every 2 hours to measure mRNA concentrations. Further information on the data and the experimental protocols is available from Flis et al. [43]. For my study, I recorded the transcription profiles of the core clock genes that are included in the models from the literature [62, 114]: LHY, CCA1, NI (PRR5), PRR7, PRR9, TOC1, ELF3, ELF4, LUX, and GI.

4.5 Methodological details

4.5.1 Preparation of realistic data

I used the Bio-PEPA framework [29] to generate mRNA and protein concentration profiles with Markov jump processes. As discussed in Section 4.4.1, these profiles are sensitive to the choice of the unit volume parameter Ω. For values of $\Omega < 10$, the concentration profiles are dominated by stochasticity, whereas for $\Omega > 1000$ they become indistinguishable from the deterministic solutions of ODEs. In my study, I used a value

of $\Omega = 100$, as suggested by Guerriero et al. [62], which gives the best match to the experimental qRT-PCR data, in particular with respect to the fluctuations of the qRT-PCR amplitudes. I simulated mRNA and protein concentration time series from the circadian regulatory network from Guerriero et al. [62] and Pokhilko et al. [114], shown in the top left panel of Figure 4.1, named 'wildtype'. In addition, I simulated mRNA and protein concentration time series from a series of modified network structures, in which various feedback loops and recurrent interactions had been removed[5]; these networks are shown in the remaining panels of Figure 4.1. For each of these network types I created 11 interventions, in emulation of the biological protocols from Flis et al. [43] and Edwards et al. [38]. These interventions include knock-outs of the genes GI, LHY, TOC1, and the double knock-out PRR7/PRR9. The knock-outs were simulated by down-regulating the transcription rates of the targeted genes, and replacing them by random noise, drawn from a truncated normal distribution (to ensure non-negativity of the concentrations). Again, in emulation of the biological protocols of Flis et al. [43] and Edwards et al. [38], I simulated varying photo-periods of 4, 6, 8, 12, and 18 hours as well as a full dark (DD) and a full light (LL) cycle, each following a 12h-12h light-dark cycle entrainment phase over 5 days. For each type of intervention, concentration time series were generated to encompass a simulated epoch of 6 days, of which the first 5 days were used for entrainment. After entrainment, molecule counts of mRNA and proteins were recorded in 2 hour intervals of simulated time, for 24 hours, giving a total of 13 'observations'. Combining these 13 observations for each intervention type yields 143 observations in total for each of the regulatory network structures shown in Figure 4.1. For each intervention type and sampling interval length, five independent data sets were generated; this corresponds to five independent laboratory experiments. For the results reported in this study, the data was not log transformed. However, I compared the learning accuracy obtained from the original and the log transformed data. The results, presented in Appendix A.2, suggest that the log transformation is counter-productive. This is consistent with the fact that a log transformation leads to more complicated expressions in Equation (4.1), as a consequence of the chain rule of differential calculus, which renders the learning task harder. To standardize the data, I followed the widely established procedure to rescale all molecule concentrations to zero mean and unit standard deviation. Two different data types were used in my evaluation procedures: complete data, which include both the mRNA and the protein concentrations, and incomplete data, in which the protein concentrations are

[5] I turned off the translation of those proteins contributing to interactions I like to suppress.

missing and regulatory network structures have to be inferred on the basis of mRNA concentrations alone.

In summary, I generated data for six different network structures, shown in Figure 4.1, repeating each data generation 5 times independently (i.e. starting from different random number generator seeds), and using complete observations (mRNAs and proteins) versus incomplete observations (mRNA only).

4.5.2 Preparation of real data

The mRNA profiles for the genes LHY, CCA1, NI, PRR7, PRR9, TOC1, ELF3, ELF4, LUX, and GI were extracted from the TiMet data [43], yielding a total of 266 samples per gene. I used the mean copy number of mRNA per cell and applied a gene wise Z-score transformation for data standardization. I did not log transform the data following the analysis in Appendix A.2. An additional binary light indicator variable with 0 for darkness and 1 for light was included to indicate the status of the experimentally controlled light condition.

4.5.3 Rate estimation

Motivated by the fundamental equation of transcriptional regulation, Equation (4.1), the machine learning and statistical models applied in my study aim to predict the rate of gene transcription from the concentrations of the putative regulators. With de novo mRNA profiling assays, the rate of transcription could in principle be measured, but these data are often not available. I therefore applied two numerical procedures to obtain the transcription rate. Appreciating that the transcription rate is just the time derivative of the mRNA concentration $x(m)$, the first approach is to approximate it by a difference quotient:

$$\frac{dx}{dm} \approx \frac{x(m+\delta m) - x(m-\delta m)}{2\delta m} \tag{4.5}$$

This is a straightforward procedure, and two different values for δm were used in my study: $\delta m = 2$ hours, henceforth referred to as the coarse gradient, and $\delta m = 24$ minutes, henceforth referred to as the fine gradient. However, it is well known that differencing noisy time series leads to noise amplification. As an alternative procedure, I therefore used an approach based on smooth interpolation with Gaussian processes. I followed Solak et al. [133] and exploited the fact that the derivative of a Gaussian process is a Gaussian process again; hence analytic expressions for the mean and the

standard deviation of the derivative are available [133]. For the covariance of the Gaussian process, I used the squared exponential kernel, which is the standard default setting in the R package `gptk` [82]. The length and variance parameter setting that govern smoothing were optimized from the initial values of 1 using the scaled gradient descent method (SGD).

Note that motivated by Equation (4.1), all methods included in my comparative evaluation study aim to predict the time derivative of a target gene's mRNA concentrations from either the protein (complete data) or the mRNA (incomplete data) concentrations of the putative regulators. Where a method has not been originally designed for this purpose, a few trivial modifications have to be implemented; e.g., for a dynamic method that aims to predict time-shifted target mRNA concentrations at time point $m+1$ from mRNA concentrations at time point m, the time shift has to be undone, and the target mRNA concentration has to be replaced by its time derivative. Motivated by Equation (4.1), a forced link from a target gene's mRNA concentration to its time derivative is built into all regression methods to allow for mRNA degradation (represented by the linear decay term in Equation (4.1)); this is a natural implementation of biological prior knowledge about the nature of transcriptional regulation.

4.5.4 Regulatory effect of the light

I note that the entity P was introduced in the circadian clock model from Guerriero et al. [62] to model the regulatory effect of the light appropriately. In [62] P was referred to as the "light-induced *protein*", though it was de facto employed to represent a biologically unknown light-stimulated component of the circadian clock. As the model in [62] does not generate mRNA concentrations for P, I use the ("protein") concentration of P as potential regulator for both data scenarios. That is, in the complete data scenario I follow [62] and think of P as a protein, while I think of P as a gene in the incomplete data scenario. Moreover, to be consistent with the model in [62] I set the values of P to zero in the absence of light.[6] Whenever I infer P to be a regulator for a target gene n, I conclude that the transcription rate of n, symbolically y_n, depends on the light condition.

[6] In the model equations defined by Guerriero et al. [62] the entity P only appears as product term $P \cdot L$, where the light variable L is equal to zero in the absence of light. Thus, the external quantity which reflects the regulatory effect of the light is effectively given by $P \cdot L$ rather than P.

4.5.5 Gene knock-outs and mutagenesis

Both the real and the realistic data contain mutagenesis experiments with loss-of-function mutants; see Sections 4.4. Genes that have been knocked out have to be excluded as target variables, since their values result from external interventions and cannot be predicted *per se* from the expression status of their regulators. By treating the entire regulatory network as a union of bi-partite graphs - one target gene against all putative regulators - this exclusion becomes straightforward: each bi-partite network is inferred from only those experiments in which the target gene was *not* knocked out. Below, I provide further details on how the methods included in my comparative evaluation were applied specifically.

4.5.6 Method Setup

As motivated in Section 4.5.3 I implemented the network inference algorithms, described in Section 4.2, with two general modifications to account for (i) mRNA degradation and (ii) mutagenesis experiments. (i) With regard to the mRNA degradation I implemented all inference methods that explicitly select regulator sets $\boldsymbol{\pi}_n$ for the target variables y_n such that the target gene's mRNA concentration, x_n, is permanently included as a member of $\boldsymbol{\pi}_n$.[7] This corresponds to the rightmost term in Equation (4.1), and does not contribute to the target node's fan-in. (ii) For mutagenesis experiments I excluded all data points corresponding to experiments where the target gene y_n was knocked out. I note that this yields varying numbers of data points $M_n \leq M$ for each target variable y_n $(n = 1, \ldots, N)$.

4.5.6.1 Graphical Gaussian models (GGM)

I used the original code of the GGM method from Schäfer and Strimmer [127], which is implemented in the R package `GeneNet` and available from the CRAN R archive. I obtained the partial correlation matrices using function `ggm.estimate.pcor` with the `static` method and default parameter settings. From these matrices only those partial correlations were extracted that involved the target gradient response y_n. To obtain the partial correlations for the complete system, including partial correlations for all gradient responses y_n, $\forall n$, the GGM learning algorithm had to be applied repeatedly for

[7] For the Bayesian methods this can be enforced by setting the prior $P(\boldsymbol{\pi}_n)$ to zero for all $\boldsymbol{\pi}_n$ with $x_n \notin \boldsymbol{\pi}_n$.

each individual gradient response variable y_n. I treated the absolute partial correlation values as indicator for the interaction ranks.

4.5.6.2 Lasso, Elastic Net and Tesla

For Lasso and the Elastic Net I used the R software package `glmnet`, described by Friedman et al. [48]. I optimized the regression parameters with cyclical coordinate descent, as implemented in the `glmnet` package. The regularization parameters were selected so as to minimize the mean square cross-validation error, using a 10-fold cross-validation scheme. This was done automatically with the function `cv.glmnet()`. Absolute values of non-zero regression coefficients were recorded and used for ranking molecular interactions.

Tesla was run with a linear regression implementation in Matlab. The regression parameters were optimized with convex programming, using the `CVX MATLAB` package[8]. A 10-fold cross-validation scheme was applied to optimize the regularization parameters, minimizing the mean square cross-validation error. Tesla requires the prior specification of permissible change-points. I selected light as the primary segmentation criterion, and grouped measurements obtained under the same light condition (light versus darkness) together. This gives, for each gene, two different segments with potentially different regression parameters. The absolute values of the non-zero regression coefficients were recorded for both segments, and their averages were used for ranking the molecular interactions.

4.5.6.3 Hierarchical Bayesian regression (HBR)

The MCMC simulations for the Bayesian regression methods, with and without multiple change-points, as described in Sections 2.2 and 2.3, were run for 20,000 iterations each, with a burn-in period of 10,000 iterations discarded. This choice gave satisfactory convergence diagnostics, based on correlation scatter plots and Gelman-Rubin potential scale reduction factors [20, 52]. Marginal posterior probabilities of molecular interactions were obtained from the MCMC trajectories, estimated from the relative frequency of inclusion of the corresponding edges in the sampled models.

[8]Matlab software for *Disciplined Convex Programming:* http://cvxr.com/cvx/

4.5.6.4 Sparse Bayesian regression with automatic relevance determination (ARD-SBR)

For the sparse Bayesian regression approach with automatic relevance determination (SBR-ARD) I used the MATLAB implementation from the supplementary material in [121]. I used the default settings both for the hyper-parameters and the maximal number of iterations for the marginal likelihood maximization. I note that the method from Rogers and Girolami [121] is a slightly modified version of the fast marginal likelihood algorithm from Tipping and Faul [140]; for the technical details I refer the reader to the supplementary material in [121].

4.5.6.5 Bayesian splines autoregression (BSA)

I used the MATLAB programs provided with the supplementary material in [106], with the following modification: for the target genes, I replaced the future gene expression values by the estimate of the time derivatives, y_n, as discussed at the end of Section 2.8. This implementation is particularly straightforward for the gene-specific hyper-parameters, corresponding to Equations (2.8-2.9) in [106], which I elected to use, as no difference between gene-specific and global hyper-parameters was found in [106]. For the other model options, including the order of the splines, the number of knots, and the hyper-parameters of the Bayesian model, I used the default settings in the MATLAB programs; note that they had been applied in [106] to data from a very similar model (also related to circadian regulation in *A. thaliana*). For the MCMC simulations, I proceeded in the same way as for the Bayesian methods, applying standard convergence diagnostics based on potential scale reduction factors.

4.5.6.6 State-space models (SSM)

In its multivariate formulation, the SSM methods described in Section 2.9 can neither deal with target-specific potential regulator sets nor with the required target-specific exclusion of certain data in relation with mutagenesis experiments (note that mRNA concentrations of knock-out genes are the result of external interventions and cannot be predicted from within the system). However, this can be easily rectified by implementing a separate SSM for each target variable y_n, which ensures a fair comparison with the other methods. For approximate inference with the variational Bayesian EM-algorithm, I used the Matlab implementation from Beal [11]. I used the default parameter settings and varied the number of hidden nodes (i.e. the dimensionality of

the vector $\mathbf{h}$) from $d = 1$ to $d = 8$. Note that the maximal number of hidden nodes d is restricted by the number of regulators, N_n. In my simulation study I analysed various data sets, and I employed the lowest N_n as an upper bound on the number of hidden nodes d. I trained two target-specific SSMs for each $d = 1, \ldots, 8$, starting from two different random initializations, i.e. $7 \cdot 2 \cdot 8 = 112$ target-specific SSMs in total. Except for low values of d ($d \leq 2$), where I observed slightly deteriorated AUROC values for the incomplete data, I obtained very stable network predictions in terms of the posterior expectation of the interaction matrix elements $(\mathbf{CB} + \mathbf{D})_{n,i}$ ($n = 1, \ldots, N$ and $i = 1, \ldots, N_n$). Throughout this chapter I therefore only report the network reconstruction results that I obtained with $d = 8$ hidden nodes, noting that almost identical results could have been obtained for $d = 3, \ldots, 7$.

4.5.6.7 Gaussian Process (GP)

For the Gaussian process approach described in Section 2.10, I used the implementation in the `GP4GRN` software package, developed by Äijö and Lähdesmäki [6]. This software computes, for each target gene, the posterior probabilities of all potential sets of regulators. The posterior probabilities for individual molecular interactions are then obtained by marginalization, summing the posterior probabilities of all configurations of regulators that include the molecular interaction in question, as shown in Equation (2.36). The hyper-parameters $\boldsymbol{\theta}, \sigma^2$ in Equation (2.36) were optimized with the Polack-Ribiere conjugate gradient method [119] to maximize the marginal likelihood of Equation (2.34). Following Äijö and Lähdesmäki [6], the hyper-parameters $\mathbf{b}, \sigma_b^2$ were set fixed. I chose the same values as suggested by Äijö and Lähdesmäki [6]. In addition, I tried a selection of randomly perturbed values, and computed the average performance. I then selected whichever of these two alternatives achieved the higher AUROC score.

4.5.6.8 Mutual information methods (ARACNE)

The application of the mutual information approach was conducted with the ARACNE method. I used the R package `minet` [102] from the Bioconductor package, which includes a function to build a mutual information matrix (`build.mim`) together with the actual ARACNE implementation (`ARACNE`). I used the default settings with the Spearman's correlation estimator, and no discretisation for building the mutual information matrix. This matrix is passed to function `ARACNE`, which in turn produces a weighted

adjacency matrix by removing the weakest links given a triplet of links subject to a threshold, which I kept at the default value. The relevant links that involve the target gradient response y_n were extracted from the adjacency matrix and directly used as indicator for the interaction ranking. To construct the full network, the whole procedure was repeated for each target gene n.

4.5.6.9 Mixture Bayesian network models (MBN)

For the mixture Bayesian network (MBN) approach I applied the implementation of the EM-algorithm for Gaussian mixture models from the "Pattern Analysis Toolbox" by I.T. Nabney; this Matlab toolbox has been made available as supplementary material in [109]. As the EM-algorithm is a greedy optimization technique that converges to the nearest (local) maximum of the likelihood, I repeated the application 10 times, starting from different initializations.[9] This yields $S = 10$ regulator sets $\boldsymbol{\pi}_n^{(1)}, \ldots, \boldsymbol{\pi}_n^{(10)}$ for each target variable y_n $(n = 1, \ldots, N)$. In imitation of the Bayesian approach I use the fraction of regulator sets that obtain the regulator x_j^n to rank the regulatory interactions $x_j^n \rightarrow y_n$ $(n = 1, \ldots, N$ and $j = 1, \ldots, N_n)$.

4.5.6.10 Gaussian Bayesian networks (BGe)

For the Gaussian Bayesian network model with the BGe scoring metric the prior distribution of the unknown parameters is assumed to be a Gaussian-Wishart distribution with hyper-parameters α, $\mathbf{T_0}$, ν, and $\boldsymbol{\mu_0}$. In the absence of any genuine prior knowledge about the regulatory interactions I set the parameter matrix of the Wishart prior to the identity matrix, symbolically $\mathbf{T_0} = \mathbf{I}$, and the mean vector of the Gaussian prior to the zero vector, symbolically: $\boldsymbol{\mu_0} = \mathbf{0}$.[10] The scalar hyper-parameters α and ν, which can be interpreted as equivalent prior sample sizes [see 49], were set to $\alpha = N_n + 4$ and $\nu = 1$. That is, I set the equivalent prior sample sizes as uniformative as possible subject to the regulatory conditions discussed in [49]. I imposed a maximal fan-in restriction of $\mathcal{F} = 3$, which renders the computation of the marginal interaction posterior

[9] In my study I initialized the EM-algorithm with allocations obtained by the k-means cluster algorithm. Thereby the initial C_n centers of the k-means algorithms were sampled from a multivariate Gaussian $N(\boldsymbol{\mu}, \mathbf{I})$ distribution, where $\mathbf{I}$ is the identity matrix and $\boldsymbol{\mu}$ is a random expectation vector with entries sampled independently from continuous uniform distributions on the interval $[-1, +1]$. To avoid that the EM-algorithm is initialized with allocations that possess unoccupied (empty) mixture components, I re-sampled the initial centers and re-ran the k-means algorithm whenever I obtained k-means outputs with empty components.

[10] Loosely speaking, this setting ($\boldsymbol{\mu_0} = \mathbf{0}$ and $\mathbf{T_0} = \mathbf{I}$) reflects the "prior belief" that all domain variables, i.e. the potential regulators and the target variable, are i.i.d. standard normally distributed.

probabilities in Equation (2.50) computationally tractable.

4.5.7 Evaluation

To assess the network reconstruction accuracies of the previously mentioned methods I employ the area under the Receiver Operating Characteristic (AUROC) curve as defined in Section 3.4.1. There have been suggestions that precision-recall curves indicate differences in network reconstruction performance more clearly than ROC curves [33]. While this is true for large, genome-wide networks, a study in [61] has indicated that networks with a low number of nodes, as with the studied networks in Fig. 4.17, the difference between the two scoring schemes should be negligible. I therefore evaluate the performance of the method in this chapter using AUROC scores, due to their more straightforward statistical interpretation [65].

4.5.7.1 ANOVA

For my evaluation, I was running hundreds of simulations for a variety of different settings, related to the observation status of the molecular components (mRNA only versus mRNAs and proteins), the method for derivative (rate) estimation (described in Section 4.5.3), the regulatory network structure (shown in Figure 4.1), and the method applied for learning this structure from data (reviewed in Section 4.2). The results, depicted e.g. in Figures 4.8 and 4.16, are complex and elude clearly discernible patterns and trends. In order to disentangle the different factors, and in particular distinguish the effect of the model from the other confounding factors, I adopted the DELVE evaluation procedure for comparative assessment of classification and regression methods in Machine Learning [117, 118] and set up a multi-way analysis of variance (ANOVA) scheme [e.g. 18].

Let Y_{ognmk} denote the AUROC score obtained for observability status o, gradient computation g, network topology n, network reconstruction method m, and data instantiation k. The range of these index parameters is as follows: $o \in \{0, 1\}$, where $o = 0$ indicates partial (mRNAs only) and $o = 1$ complete (mRNAs and proteins) observation; $g \in \{0, 1, 2\}$, where $g = 0$ denotes coarse gradient, $g = 1$ fine gradient, and $g = 2$ gradient from a smooth interpolant; $m \in \{0, 1, 2, 3, 4, 5\}$, where $m = 0$ represents 'wildtype' (the published network topology), and $m \neq 0$ the five network modifications shown in Figure 4.1; $n \in \{0, 1, 2, \ldots, 14\}$, for the 15 network reconstruction methods discussed in Section 4.2 (and shown below in Figure 4.11), and $k \in \{0, 1, 2, 3, 4\}$ for five

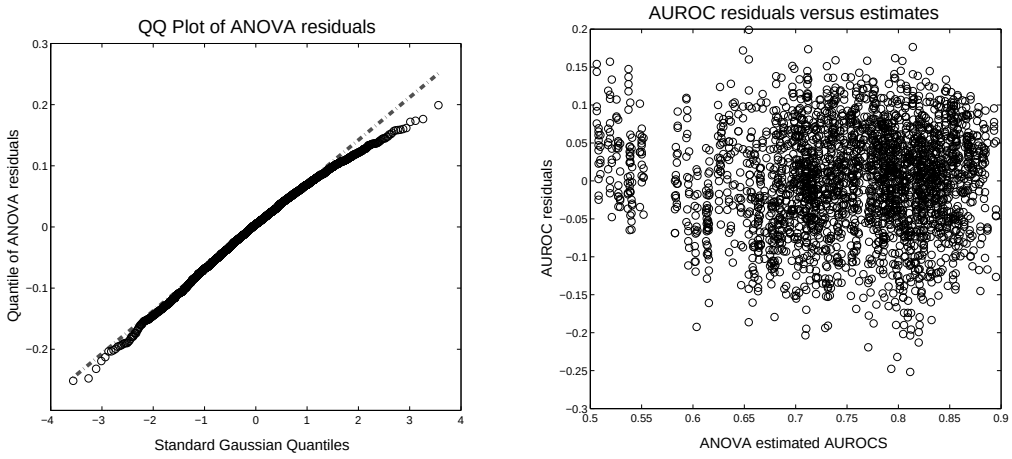


Figure 4.4: Residual diagnostic for the ANOVA model. *Left panel: QQ-plot.* The figure shows a Quantile-Quantile (QQ) plot of the residuals for the ANOVA model, described in Section 4.5.7.1, Equation (4.6). The actual quantiles (vertical axis) are plotted against the quantiles of the Gaussian distribution (horizontal axis). The linear relation indicates good agreement with the Gaussian distribution; the deviations for very low and high values point to slightly longer tails. *Right panel: Scatter plot diagnostic.* The figure shows a scatter plot of the residuals (vertical axis) against the AUROC values fitted with the ANOVA model of Section 4.5.7.1, Equation (4.6) (horizontal axis).

different data instantiations. I model the AUROC scores with the following ANOVA approach:

$$y_{ognmk} \;=\; O_o + G_g + N_n + M_m + \varepsilon_{ognmk} \tag{4.6}$$

where $\varepsilon_{ognmk} \sim N(0, \sigma^2)$ is zero-mean white additive Gaussian noise, and O_o, G_g, N_n, and M_m are main effects associated with observation status, gradient computation, network topology, and network reconstruction method, respectively.

To ascertain that the underlying assumptions of the ANOVA model are satisfied, I carried out a standard residual analysis. The objective is to test whether the residuals are independent and identically (i.i.d) normally distributed. A violation of this assumption would indicate that some structure in the data has not been captured by the decomposition of Equation (4.6), and that e.g. higher-order interaction terms would have to be included.

Figure 4.4, left panel, shows a quantile-quantile (QQ) plot of the residuals to test the assumption of a normal distribution. The straight line confirms that there is good agreement with this assumption overall, with only minor deviations for the lowest and

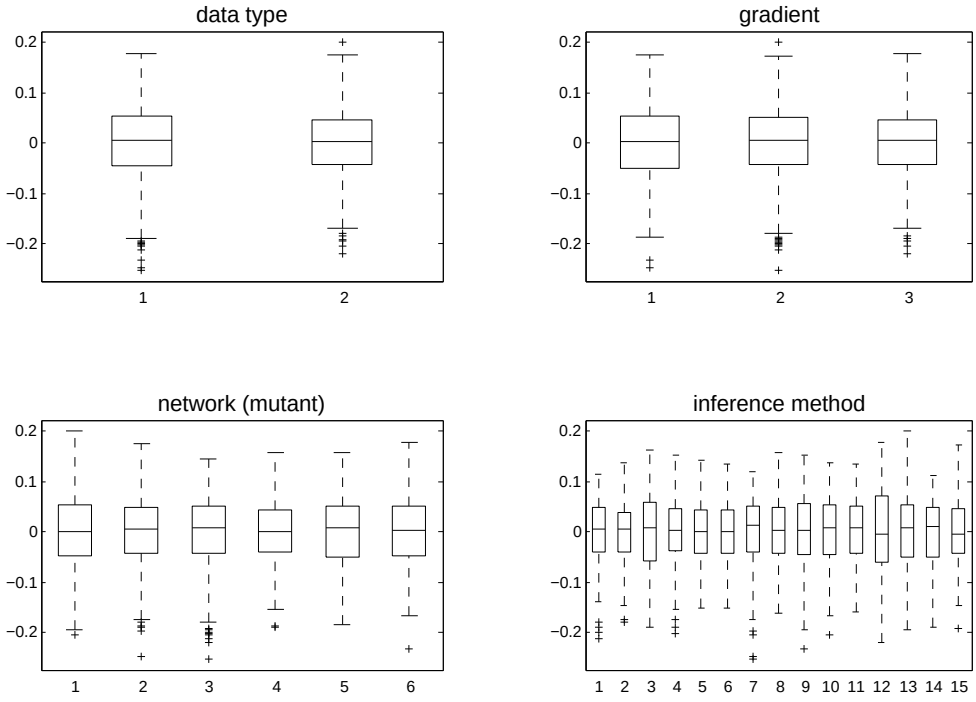


Figure 4.5: Residual diagnostic for different factors of the ANOVA model. The figure is arranged as a 2-by-2 matrix, whose four panels correspond to the four main effects of the ANOVA model of Section 4.5.7.1, Equation (4.6). Each panel shows a boxplot representation of the distribution of the residuals for all possible values of the corresponding main effects.

highest quantiles, suggesting that the residual distribution is slightly heavier-tailed.

Figure 4.4, right panel, shows a scatter plot of all residuals against the corresponding values fitted with the ANOVA model of Equation (4.6). For low values, the spread of the residuals seems to become slightly tighter, but this effect is weak, and overall there is no clearly discernible pattern of any dependence between the residual distribution and the fitted value.

Figure 4.5 shows histograms of the residuals for all possible values of the four main effects in Equation (4.6). There are no obvious deviations from a uniform pattern, and the results are consistent with the assumption that the distributions of the residuals are identical and independent of the main effects.

These diagnostics thus do not indicate any clear violation of the model assumptions and suggest that the ANOVA model provides an adequate mechanism for extracting trends and patterns from my simulations studies.

4.6 Results

The Results section can be divided into three parts. In the first part, which covers Sections 4.6.1-4.6.3, I address questions related to the application of the models: how to set the regularization parameters for the sparse regression methods, and how to set the parameter and structure prior for the Bayesian regression models. In the second part, I address the main questions of my study: How do the different methods compare with respect to the accuracy of the network reconstruction? What is the effect of missing protein concentrations? What is the effect of the network topology? What is the best way to compute the transcription rates? What is the effect of change-points, both for the light phase and the rates? This part covers Sections 4.6.4–4.6.9 and, like the previous part, is based on the realistic data described in Section 4.4.1. The final part, Section 4.6.10, describes the application to the real data from Section 4.4.2.

4.6.1 Comparison of different methods for setting the Lasso penalty parameter

The sparse regression methods Lasso and Elastic Nets require the selection of the regularization parameter λ, which trades off the strength of the ℓ_1 or ℓ_1/ℓ_2 penalty term against the data misfit term. I have compared three different procedures: 10-fold cross-validation, 10-fold cross-validation with the correction suggested by Hastie et al. [70], and BIC. The objective of 10-fold cross-validation is to select the regularization parameter that minimizes the average signal reconstruction error on held-out data. Hastie et al. [70] suggested using a larger value as follows: plot the cross-validation error as a function of λ, then select the largest value of λ for which the cross-validation error is within 1 standard deviation of the minimum cross-validation error [70]. The rationale is that Lasso is biased [108] and that the optimal value of λ chosen by cross-validation is optimal in terms of predictive (signal reconstruction) rather than explanatory (network connectivity) performance. Hastie et al. [70] suggested this correction as a heuristic scheme to improve explanatory performance. The motivation for using BIC is to avoid the computational costs of a cross-validation scheme. I compared the Lasso with these three procedures on my simulated data described in Section 4.4.1, which includes several network types (as shown in Figure 4.1), incomplete (mRNA only) and complete (mRNA and protein) data, and coarse and fine gradients (Section 4.5.3). Figure 4.6 shows a Bland-Altman plot, where the difference between the AUROC scores are plotted against the mean AUROC scores. A visual inspection suggests that standard minimum

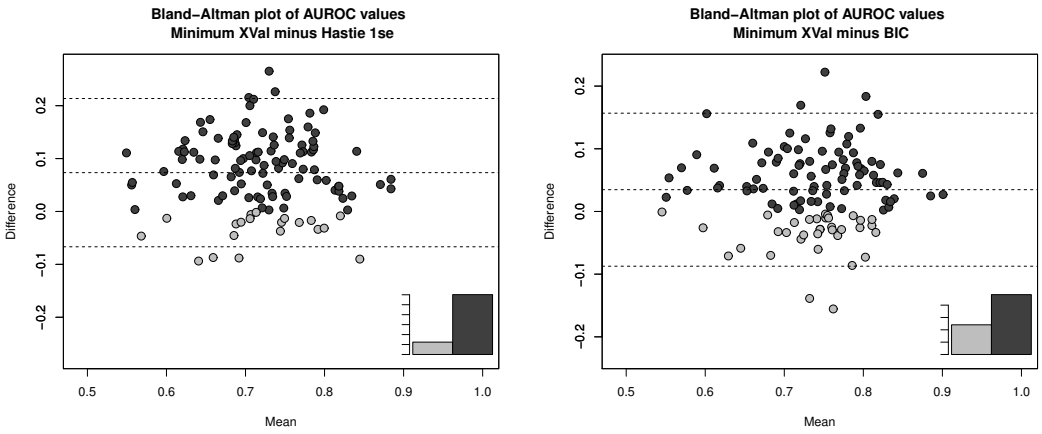


Figure 4.6: Bland-Altman plot of AUROC scores comparing different selection procedures for the regularization parameter of Lasso: The plots print the difference between the AUROC scores (vertical axis) against their mean (horizontal axis). *Left panel:* Comparison between standard minimum cross-validation and the procedure proposed by Hastie et al. [70]. *Right panel:* Comparison between minimum cross-validation and BIC. For details, see Section 4.6.1. Positive values (dark grey) indicate higher AUROC scores for the standard cross-validation procedure. Negative values (light grey) indicate higher scores for the alternative procedure (Hastie et al. or BIC). The inlet histograms in the bottom right corner show the relative frequencies of positive (dark grey) and negative (light grey) scores.

cross-validation achieves slightly higher AUROC scores on average than the other two methods. A paired t-test confirms that standard cross-validation performs significantly better than the procedure proposed in Hastie et al. [70] (p-value of 0.0004), and weakly outperforms BIC (p-value of 0.10). The standard minimum cross-validation approach thus performs overall best and was used for the further investigations.

4.6.2 Influence of the structure prior for hierarchical Bayesian regression models

I tested the Bayesian regression models, described in Sections 2.2 and 2.3, with two different prior distributions on the network structure: a uniform distribution and a truncated Poisson distribution for $P(\boldsymbol{\pi}_n)$. The Poisson prior has mean κ and a maximal cardinality matching that of the uniform prior, i.e. $|\boldsymbol{\pi}_n| \leq 3$: $P(\boldsymbol{\pi}_n|\kappa) \propto \frac{\kappa^{|\boldsymbol{\pi}_n|}}{|\boldsymbol{\pi}_n|!} I(|\boldsymbol{\pi}_n| \leq 3)$, where κ is sampled from a vague conjugate prior with a Gamma distribution $P(\kappa) = \mathcal{G}a(0.5, 1)$, following [89]. I tested the Bayesian regression model with both priors on

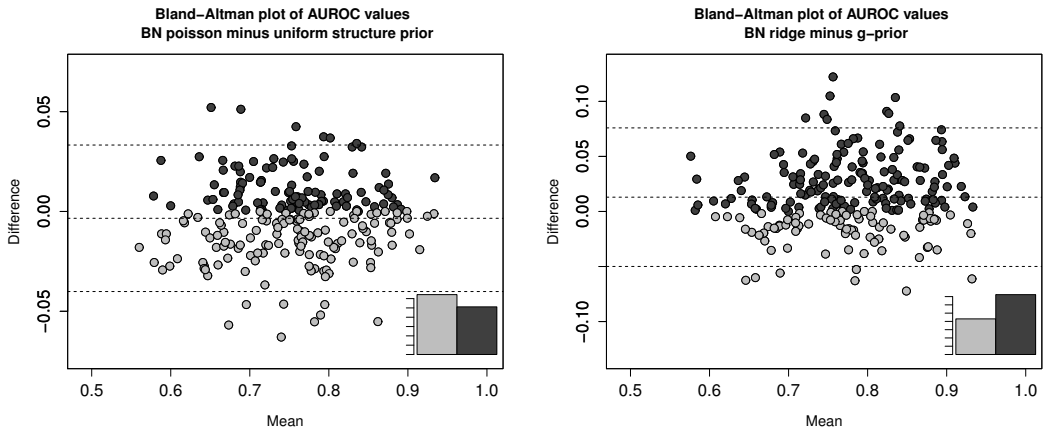


Figure 4.7: Dependence of Bayesian regression on the structure (left panel) and parameter (right panel) prior. The graphs show Bland-Altman plots, which plot the difference between two AUROC scores (vertical axis) against their mean (horizontal axis). *Left panel:* Difference between the uniform and the Poisson structure prior. Positive values (dark grey dots) indicate AUROC scores in favour of the uniform prior, negative values (light grey dots) indicate AUROC scores in favour of the truncated Poisson prior. *Right panel:* Difference between ridge regression prior and g-prior. Positive values (dark grey dots) indicate better performance of the ridge regression prior, negative values (light grey dots) indicate better performance of the g-prior. The inlet histograms in the bottom right corner show the relative frequencies of positive (dark grey) and negative (light grey) scores.

the simulated data, described in Section 4.4.1, and recorded the AUROC scores. The left panel in Figure 4.7 shows a Bland-Altman plot with the pair-wise differences of the AUROC values (vertical axis) plotted against the mean AUROC score (horizontal axis). The plot shows no noticeable difference between the two priors, and a paired t-test with a p-value of 0.17 indicates no significant difference. I decided to use the uniform prior for all further investigations.

4.6.3 Influence of the parameter prior for hierarchical Bayesian regression models

I compared two different priors on the regression parameters of the Bayesian regression model described in Sections 2.2 and 2.3: the so-called ridge regression prior of Equation (2.11), and the g-prior. The latter is widely used in the statistics literature, see e.g. Andrieu and Doucet [7] and Marin and Robert [98], and effectively replaces the di-

agonal matrix in Equation (2.11) by an outer product of the design matrix; see [98] for details. I carried out a comparative evaluation on the realistic data from Section 4.4.1. The right panel in Figure 4.7 shows a Bland-Altman plot of the pairwise differences in the AUROC scores. There is a slight shift to positive values, indicating that, overall, the ridge regression prior achieves a better performance. This difference was found to be significant, with a paired t-test giving a p-value of $2.6e-19$. I therefore used the ridge regression prior of Equation (2.11) throughout my study.

4.6.4 Comparison between the methods

A main objective of my study is a systematic comparative performance evaluation of the models reviewed in Section 4.2. These models were applied to the different data described in Section 4.4, different observabilities (proteins and mRNAs versus mRNAs only), different gradient computations (Section 4.5.3), and different network topologies (as shown in Figure 4.1). Figure 4.8 shows the distributions of AUROC scores obtained in my study. The scores vary considerably, depending on the different factors, and consistent trends and clear patterns are not easily discernible. To enable a clearer interpretation I adopted the ANOVA method described in Section 4.5.7.1. The quantity of interest is M_m - the main effect of the network reconstruction method, which is plotted in Figure 4.11.

My study suggests that with the exception of MBN and ARACNE, which show a significantly worse performance, all methods achieve a performance in the range of AUROC scores between 0.7 and 0.8. This is significantly better than random expectation, but considerably worse than perfect network reconstruction. The best performance is achieved with Gaussian Bayesian networks (BGe) and hierarchical Bayesian regression models (HBR). A somewhat surprising finding is that within the group of Bayesian regression models, no performance improvement is achieved by including change-points to indicate the light phase (HBR-light), change-points in the amplitude to model Michaelis-Menten non-linearities (HBR-cps), or non-linear (inverse and quadratic) terms (HBR-nl). In fact, the simple linear Bayesian regression model with no change-points (HBR) achieves the best performance of all the methods included in the comparison. This seems counter-intuitive, given that light has a clear influence on circadian regulation, and the processes of the underlying Michaelis-Menten kinetics are intrinsically non-linear. I discuss the reason for this behaviour in Section 4.7, where I also provide explanations for the poorer performance of some of the alternative models.

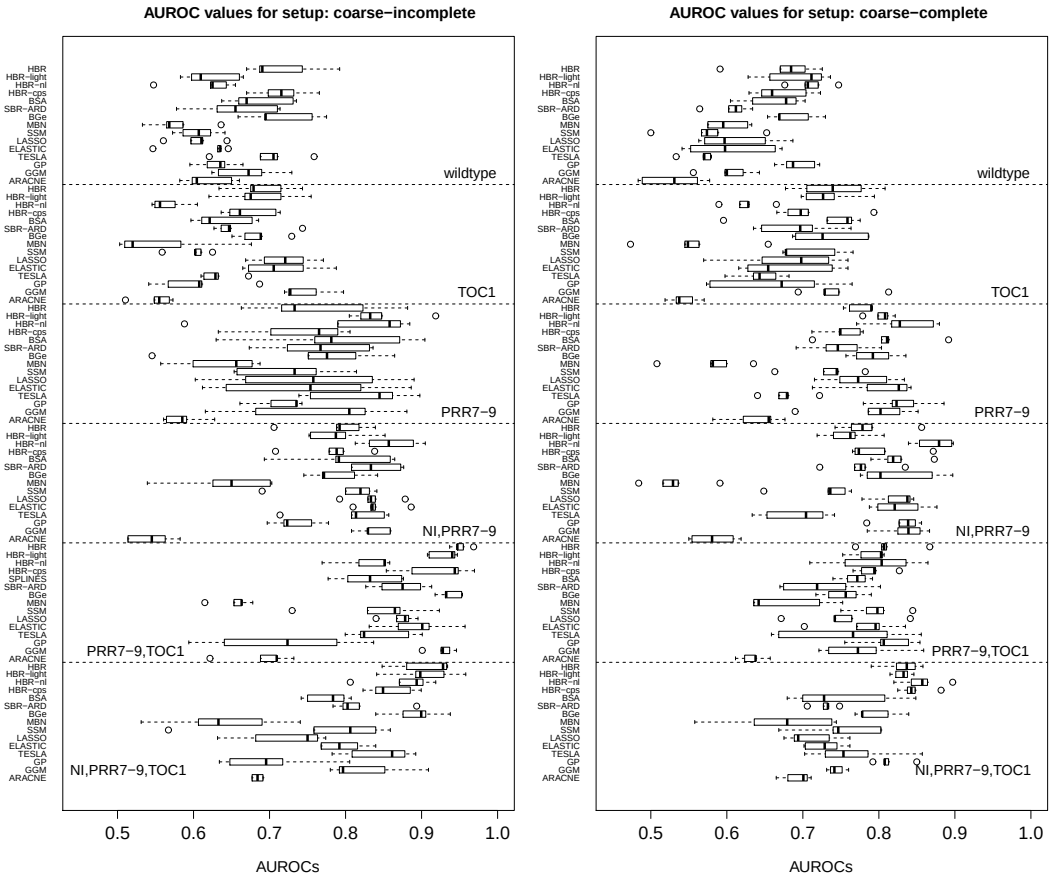


Figure 4.8: AUROC scores obtained for different reconstruction methods, different network structures, and different experimental settings. The figures show standard boxplot representations for the distributions of AUROC scores obtained in the study for the coarse response gradients (computed from Equation (4.5) with 2-hour intervals); the scores for the fine and interpolated (Gaussian process) gradient are displayed in Figure 4.9 and 4.10 respectively. *Left panel:* Incomplete data, with mRNA but no protein concentrations. *Right panel:* Complete data that include both protein and mRNA concentrations. Each panel contains six sub-panels, representing the six different network topologies shown in Figure 4.1. Each line/box represents five AUROC scores that come from the five independent data realizations. Note that the detailed results in Figures 4.8–4.10 are complex and difficult to interprete. To facilitate the identification of putative trends I apply the ANOVA method with results shown in Figures 4.11–4.14.

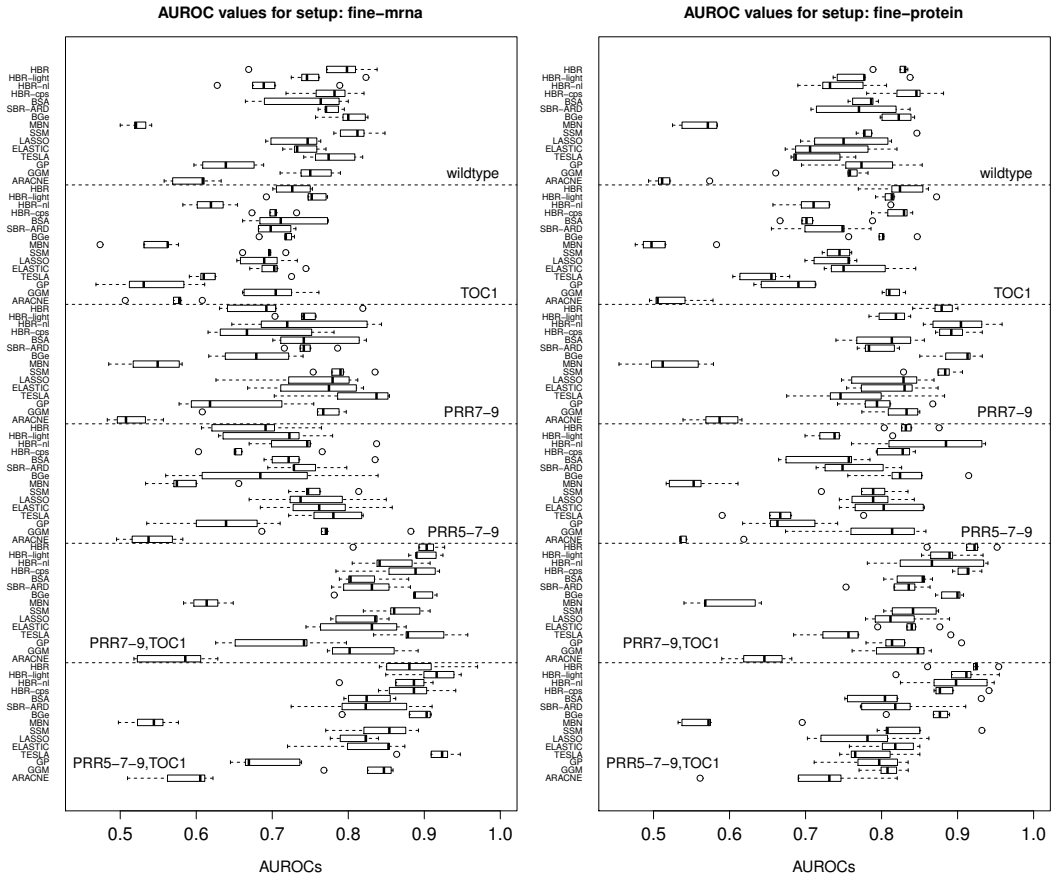


Figure 4.9: AUROC scores for fine response gradients (continued from Figure 4.8). The fine gradient is computed with Equation (4.5) using a 24 minute interval. *Left panel:* Incomplete data, with mRNA but no protein concentrations. *Right panel:* Complete data that include both protein and mRNA concentrations.

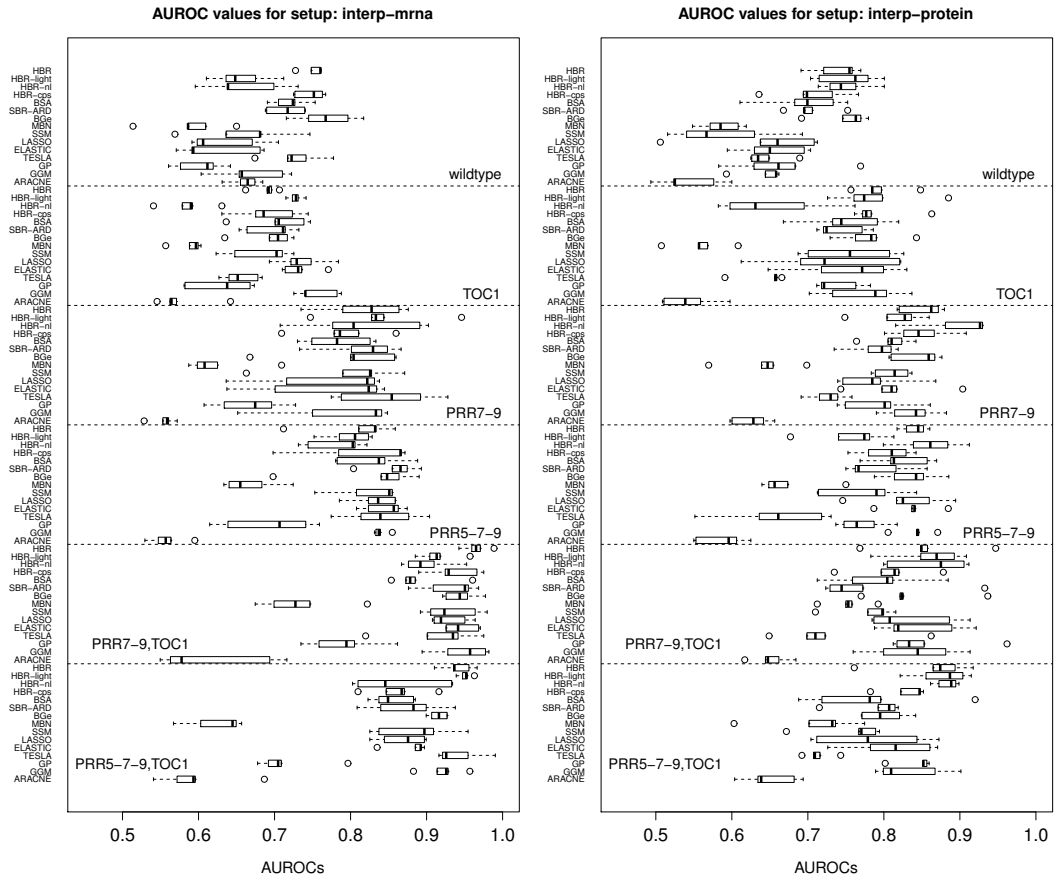


Figure 4.10: AUROC scores for Gaussian process response gradients (continued from Figure 4.8). The gradients are derived from smooth interpolation with a Gaussian process as described in Section 4.5.3. *Left panel:* Incomplete data, with mRNA but no protein concentrations. *Right panel:* Complete data that include both protein and mRNA concentrations.

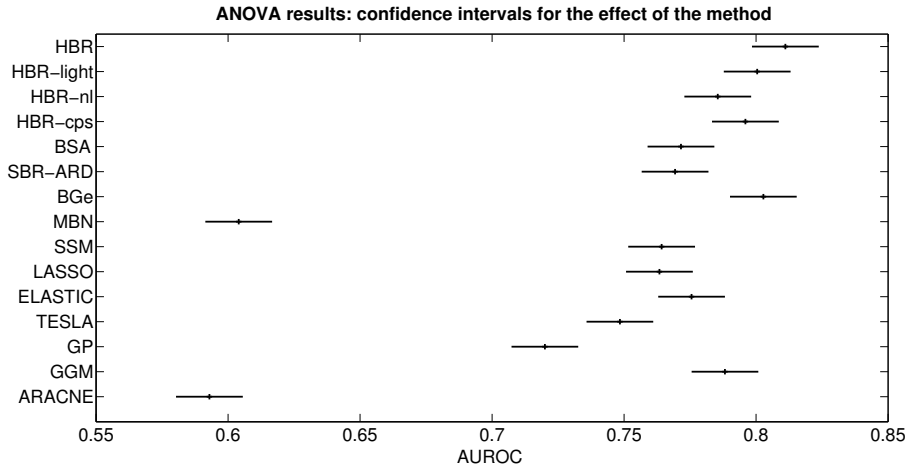


Figure 4.11: Comparison between different network reconstruction methods. The figure shows confidence intervals for the group means associated with the main effect for the network reconstruction method from the ANOVA analysis in Equation (4.6). The horizontal axis shows the AUROC score. The vertical axis represents the different methods described in Section 4.2. The labels on the vertical axis refer to the methods described in Section 4.2, using the same abbreviations as in the subtitle headers.

4.6.5 Influence of rate estimation

The mathematical formulation of chemical kinetics, e.g. based on mass action or Michaelis-Menten kinetics, as in the present study, predicts the rate of mRNA transcription as a function of the concentrations of the regulating proteins. Ideally, this rate would be measured, which could in principle be effected with *de novo* mRNA assays. These assays are not always available, though; so in the present study, I estimated the time derivatives of mRNA concentrations directly from the mRNA concentration time courses as a proxy. I compared three different approaches. In the first study, I approximated the time derivatives by finite difference quotients from the low frequency time series, where observations were taken every 2 hours. This corresponds to Equation (4.5) with $\delta m = 2h$, and I refer to it as the coarse gradient. In the second study, I repeated the same procedure on high-frequency data, where measurements were taken every 24 minutes. This corresponds to Equation (4.5) with $\delta m = 24\text{min}$, and I refer to this as the fine gradient. High frequency data with such short time intervals are rarely available in practice, though. So as an alternative, I applied a Gaussian process smoothing

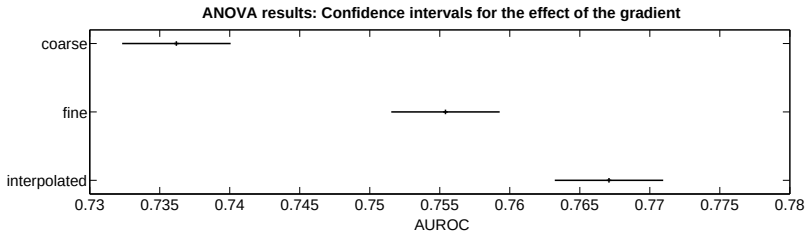


Figure 4.12: Influence of rate estimation. The figure shows confidence intervals for the group means associated with the main effect for the dependence of the performance (AUROC score) on the rate estimation method, based on the ANOVA analysis of Equation (4.6). The horizontal axis shows the AUROC score. The vertical axis represents the three rate estimation methods, as described in Section 4.5.3: coarse gradient (top), fine gradient (middle), and gradient from smooth interpolant (bottom). The confidence intervals have a width of 0.075.

approach described in Section 4.5.3. The results are shown in Figure 4.12. It can be observed that the fine gradient achieves an improvement on the coarse gradient, which is consistent with expectation. However, my study also allows a quantification of this improvement, which is in the order of Δ AUROC $= 0.02$ on average. Interestingly, my study suggests that gradient computation in combination with smooth interpolation using Gaussian processes achieves an even more substantial improvement of about Δ AUROC $= 0.03$. This indicates that intelligent data preprocessing leads to a better boost in predictive performance than blindly carrying out additional experiments.

4.6.6 Influence of missing protein concentrations

I have carried out the simulations for two types of data: complete observation, where both protein and mRNA concentrations are available, and incomplete observation, where protein concentrations are missing. The results are shown in Figure 4.13. The network reconstruction accuracy based on complete observations is slightly better than that from incomplete observations. The important new contribution of my study is to objectively assess the difference in performance, profiled over different network topologies, different ways of preprocessing the data, and different statistics and machine learning methods. This has been effected with the ANOVA approach described in Section 4.5.7.1, which quantifies the effect of missing protein concentrations as leading to a deterioration of only $\Delta AUROC = 0.002 \pm 0.003$. Hence, my study leads to the counter-intuitive finding that the difference in performance is *not* significant. I provide

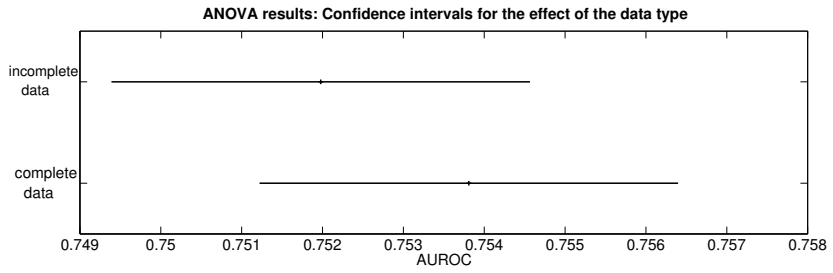


Figure 4.13: Influence of incomplete observations. The figure shows confidence intervals for the group means associated with the main effect for the observation status based on the ANOVA analysis of Equation (4.6), comparing complete data observations of both protein and mRNA concentrations versus the incomplete data that includes mRNA observations only. The horizontal axis represents AUROC scores.

a discussion in Section 4.7.

4.6.7 Influence of network topology and feedback loops

An important aspect of my study is the investigation of how the network reconstruction accuracy depends on the connectivity of the network topology and the proportion of recurrent connections. To this end I have successively pruned feedback interactions, as shown in Figure 4.1. Figure 4.14 suggests that there is a noticeable pattern, with less recurrent and sparser network structures appearing to be easier to learn and leading to higher AUROC scores. While this confirms a known and intuitively plausible trend, my study allows an objective quantification of the difference in performance, which has been found to amount to $\Delta AUROC = 0.14$ between the most and least recurrent structures.

4.6.8 Influence of change-points to indicate the light phase

I tested whether a segmentation of the data into day and night phase affects the learning performance, motivated by the hypothesis that regulation in light and dark may differ and should be modelled with two separate sets of regression parameters. To this end, light information in my realistic studies described in Section 4.4.1 was used to assign each sample in time to a light or dark phase. I extended the range of methods to include a Lasso variant that supports change-points (Tesla, described in Section 2.5), and a non-homogeneous hierarchical Bayesian regression model, described in Section 4.3.1. Simu-

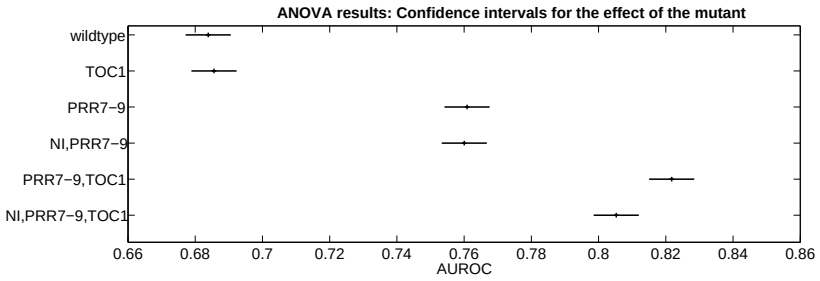


Figure 4.14: Influence of network structure. The figure shows confidence intervals for the group means associated with the main effect for the network structure, based on the ANOVA analysis of Equation (4.6), for the wildtype network published by Pokhilko et al. [114], and the five modified structures shown in Figure 4.1 (using the same labels on the vertical axis as used in Figure 4.1). As one descends from the top to the bottom on the vertical axis, the network structures become sparser, with feedback loops increasingly being pruned. The horizontal axis represents AUROC scores.

lation experiments were conducted for incomplete (mRNA only) and complete (mRNA and proteins) data, as well as for coarse and fine response gradients, as described in Section 4.5.3. Figure 4.15 shows the distribution of pairwise differences between a method without change-points and the corresponding change-point method (Lasso versus Tesla and homogeneous Bayesian regression versus non-homogeneous Bayesian regression). The somewhat counter-intuitive finding is that for complete data (protein and mRNA concentrations, in grey boxes), the inclusion of change-points leads to a deterioration of the AUROC scores for most of the network structures. I will discuss this observation in Section 4.7.

4.6.9 Effect of change-points on the response variable

I studied the effect of segmenting the domain of the response variable (i.e. the rate, that is the time derivative of the mRNA concentration) with multiple change-points. The objective is to approximate the non-linearity of the Michaelis-Menten response with a piece-wise linear model. To this end, I applied the non-homogeneous hierarchical Bayesian regression model, described in Section 4.3.2, with different settings of the maximum number of change-points. The evaluation was extended over all network topologies (shown in Figure 4.1), incomplete (mRNA only) and complete (mRNA and protein concentrations) data, and different gradient resolutions (coarse versus fine;

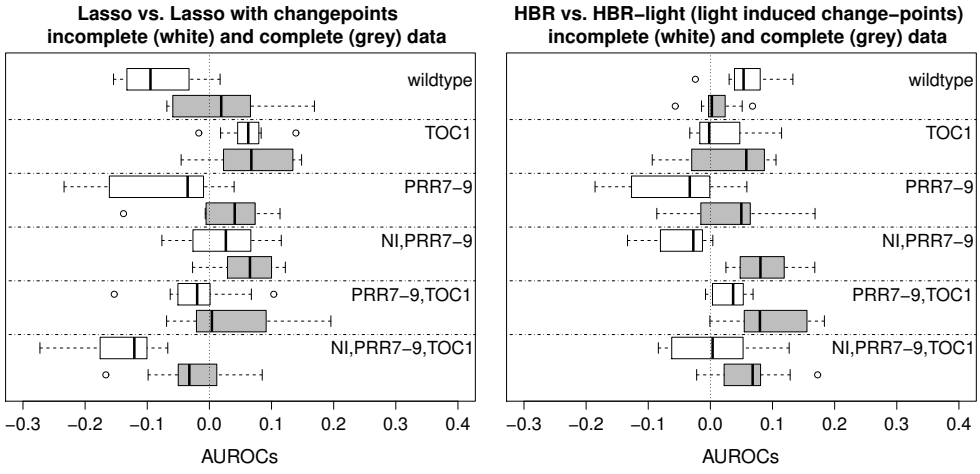


Figure 4.15: Dependence of the network reconstruction on light/dark phase segmentation for Lasso and Bayesian regression. The figure shows the distribution of pairwise AUROC differences for Lasso versus Tesla ($AUROC_{Lasso}$ - $AUROC_{Tesla}$; left panel) and homogeneous Bayesian regression versus non-homogeneous Bayesian regression with light induced change-points ($AUROC_{withoutcps} - AUROC_{withcps}$). The distributions are over all network topologies (as shown in Figure 4.1), numerical replications, and coarse and fine gradients, as described in Section 4.5.3. Grey shading: complete data with protein concentrations as predictor for the target mRNA gradients. White shading: incomplete data with mRNA concentrations as predictor for the target mRNA gradients.

see Section 4.5.3). The results are shown in Figure 4.16. The somewhat counter-intuitive finding is that the network reconstruction performance tends to deteriorate as more change-points are allowed, suggesting that despite the non-linear nature of the underlying Michaelis-Menten kinetics, imbuing the model the non-linear modelling flexibility is counter-productive. This trend is slightly stronger for complete (mRNAs and proteins) than for incomplete data (mRNAs only). I provide an explanation in Section 4.7.

4.6.10 Circadian regulation network in *Arabidopsis thaliana*

Figure 4.17 shows the network learned from the TiMet data, and six hypothetical networks published by Locke et al. [93], Kolmos et al. [86], Herrero et al. [75], and Pokhilko et al. [114, 113, 115]. Solid lines show transcriptional regulation, dashed lines represent protein complex formation. The latter cannot be learned from transcriptional data

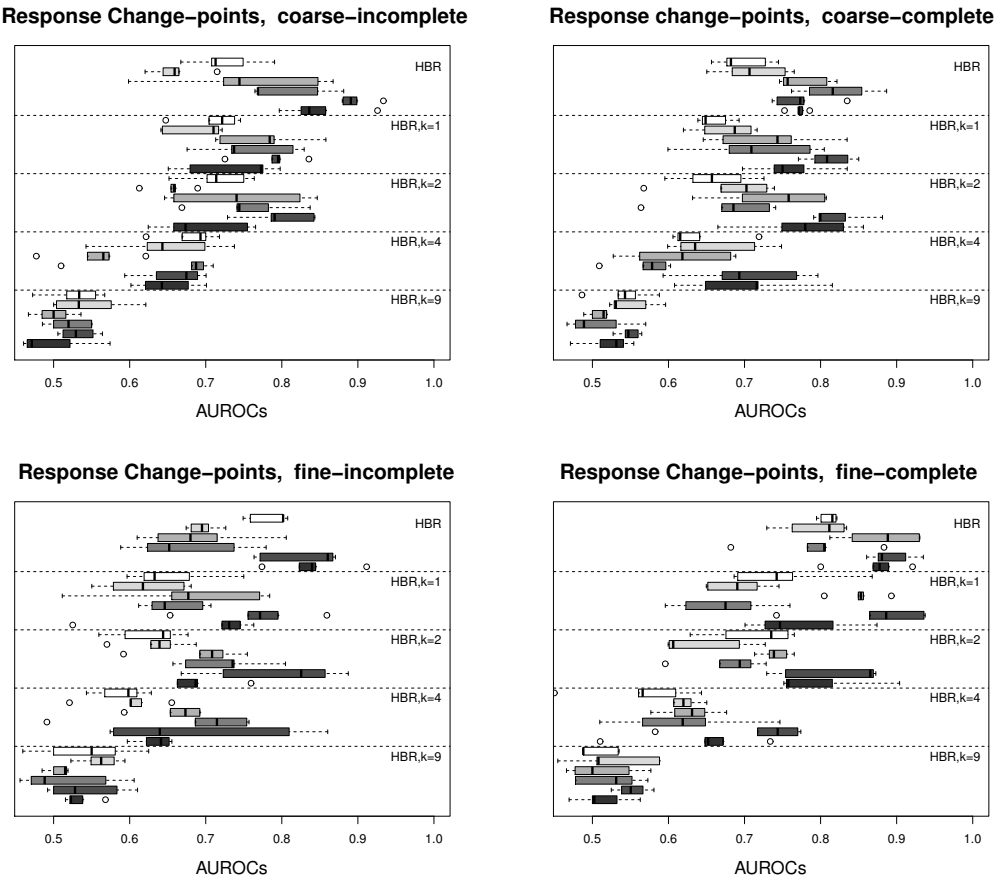


Figure 4.16: Dependence of the network reconstruction accuracy on the change-points for response segmentation. The figure contains four panels. *Left panels:* Incomplete data, which only include mRNA concentrations. *Right panels:* Complete data, which include mRNA and protein concentrations. Two different gradient computations were applied, as described in Section 4.5.3. *Top panels:* coarse gradients. *Bottom panels:* fine gradients. Each panel contains five sub-panels for five different variants of the hierarchical Bayesian regression model, described in Sections 2.2–2.3: homogeneous Bayesian regression model without change-points, and non-homogeneous Bayesian regression model with $k = 1, 2, 4$ and 9 change-points. Each sub-panel shows the distribution of AUROC scores for the six different network topologies in Figure 4.1, with increasing network sparsity from top to bottom.

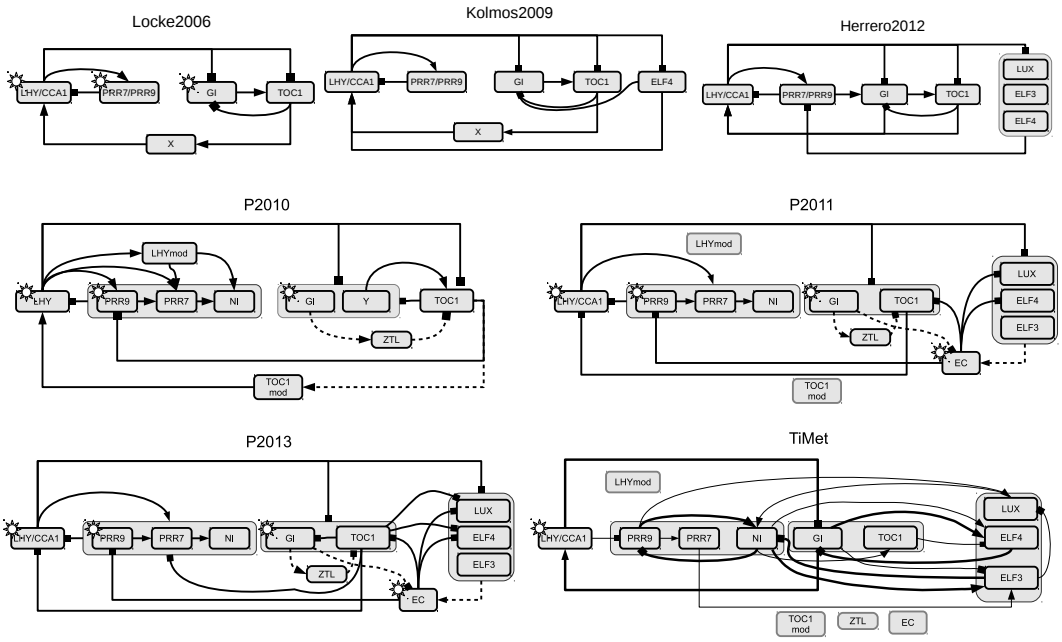


Figure 4.17: Hypothetical circadian clock networks from the literature, and inferred from the TiMet gene expression data. All panels except for the bottom right show hypothetical networks from the literature: Locke2006 [93], Kolmos2009 [86], Herrero2012 [75], P2010 [114], P2011 [113], and P2013 [115]. The bottom right panel (TiMet) displays the reconstructed network from the TiMet data, described in Section 4.4.2, using the hierarchical Bayesian regression model from Section 2.2. Gene interactions are shown by black lines; protein interactions are shown by dashed lines; an arrow head symbolizes activation and a bar head inhibition; regulation by light is represented by a sun symbol. The interactions in the reconstructed network were obtained from their estimated posterior probabilities. Those above the selected threshold of 0.95 were included in the interaction network; for the light influence see the main text.

and are thus systematically missing. This explains, for instance, why ZTL and EC are detached from the remaining network. The same applies to the modified proteins TOC1-mod and LHY-mod. Various features of the published networks are reproduced, though, like the acute light response in the transcription of LHY and CCA1, the activation of PRR7 by PRR9, the inhibition of GI by LHY/CCA1, and the inhibition of ELF4 by TOC1, which can be found in the network P2013. Various features are similar to the published networks. In the reconstructed network, NI is directly activated

by PRR9, while in the published networks, the activation is indirect, via PRR7. The positive feedback loop from the so-called evening genes to the morning genes consists of an activation of LHY/CCA1 by GI. The nature of this feedback loop (activation) is consistent with [93, 86, 114]. In these publications, the regulatory influence is caused by TOC1 rather than GI, but these two genes are "neighbours" in the published networks (meaning: regulating each other, and exhibiting similar expression profiles). One of the morning loop genes (NI) is predicted to be inhibited by ELF3. This is consistent with [113, 115], although in these publications, the interaction is indirect (via EC) and affects a neighbouring target gene (PRR9). As mentioned above, it is intrinsically infeasible to learn post-transcriptional processes, like protein complex formation, from transcriptional data alone; so it is no surprise to see that the protein complex EC is detached from the remaining network. It is particularly interesting to note that a key network motif repeatedly found in the reconstructed network concurs with the published networks. This is the two-node feedback motif in which a gene is the activator of its own inhibitor. This structure is particularly clearly seen in [93], where it occurs three times: within the group of morning genes (LHY/CCA1 activating PRR7/PRR9, PRR7/PRR9 inhibiting LHY/CCA1), within the group of evening genes (GI activating TOC1, TOC1 inhibiting GI), and between the morning and evening genes (LHY/CCA1 inhibiting TOC1, TOC1 activating LHY/CCA1). These three feedback mechanisms exist in the reconstructed network also and are highlighted with thick lines (see TiMet network in Figure 4.17), involving neighbouring nodes in the same three gene groups: morning genes (PRR9 activating NI, NI inhibiting PRR9), evening genes (GI activating ELF4, ELF4 inhibiting GI), and between morning and evening genes (GI activating LHY/CCA1, LHY/CCA1 inhibiting GI, NI activating ELF3, ELF3 inhibiting NI). This suggests that, despite deviations in the detailed mechanisms, the key topological features of the published networks have been successfully reconstructed. Finally, I attempted to learn the light influence marked with a sun symbol in the TiMet network in Figure 4.17 by allowing light as an additional variable. I correctly recovered a high probability (0.83) link to LHY/CCA1 but failed to observe any other significant occurrences. It was noted in Section 4.5.1 that the light influence on mRNA transcription is typically modulated by light sensitive proteins. Since the TiMet data lack any such protein observations, I have to assume that the light is not learned efficiently.

4.7 Discussion

The previous section has presented the results from my comparative evaluation study. Most of the patterns that I have found are clear and intuitive; the value of my study consists in the objective quantification of these trends. There are a few findings that are peculiar, though. Figure 4.11 suggests that it is counter-productive to include non-linear terms in the Bayesian regression model. Given that the true underlying dynamics are, in fact, non-linear, why does the inclusion of these effects deteriorate the model performance? Figure 4.16 suggests that an increasing number of change-points for the response segmentation in the non-homogeneous Bayesian regression model deteriorates the network reconstruction. However, more change-points give more non-linear modelling flexibility. Given that the true underlying dynamics are non-linear, why is that a disadvantage? How can we understand that the network reconstruction accuracy does not improve significantly when including protein concentrations in addition to just mRNA concentrations, as suggested by Figure 4.13? Figure 4.15 shows the effect of segmenting the data into a light and a dark phase. How can we understand that this segmentation deteriorates the network reconstruction accuracy for complete (mRNA plus protein) data? Finally, Gaussian processes are widely appreciated as a powerful modelling paradigm. So how can we explain their comparatively poor performance (see Figure 4.11)? In what follows, I will provide an explanation of these effects.

4.7.1 The effect of change-points and non-linear regressors

To investigate the effect of change-points and non-linear regressors, I devised a synthetic toy example, sketched in Figure 4.18. Consider $N = 8$ random variables, where $X_1, \ldots, X_5$ are i.i.d. standard Gaussian distributed. In the first model (Figure 4.18, top panel), the variables $X_6, \ldots, X_8$ depend on X_5 through a sigmoidal transfer function:

$$X_i = \begin{cases} \epsilon_i & , i = 1, \ldots, 5 \\ \frac{2}{1+e^{\frac{-X_5}{\theta}}} + \epsilon_i & , i = 6, 7, 8 \end{cases} \tag{4.7}$$

The random noise variables ϵ_i $(i = 1, \ldots, 8)$ are i.i.d. Gaussian $N(0, \sigma^2)$ distributed. In the second model (Figure 4.18, bottom panel), the variables $X_6, \ldots, X_8$ depend on the product term $X_4 X_5$ through a sigmoidal function. For $i = 6, 7, 8$ I have:

$$X_i = \frac{2}{1 + e^{\frac{-X_4 X_5}{\theta}}} + \epsilon_i \tag{4.8}$$

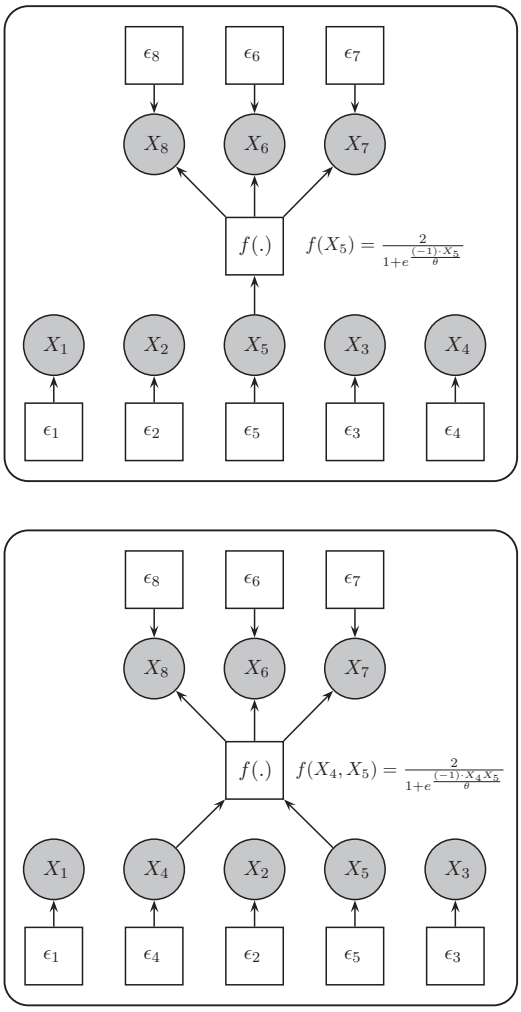


Figure 4.18: Regulatory network for synthetic data. The figure shows a graphical representation of the regulatory interactions among the eight variables of the synthetic data described in Section 4.7.1. In both panels the observed variables, $X_1, \ldots, X_8$, are represented as grey circles, while the (unobserved) random perturbations, $\epsilon_1, \ldots, \epsilon_8$, as well as the non-linear transformation $f(.)$ are represented by white squares. *Top panel:* The three variables X_6, X_7, and X_8 obtain the same deterministic input, $f(X_5)$, where $f(.)$ is a sigmoidal function. The deterministic signal is perturbed by additive i.i.d. Gaussian noise: ϵ_i $(i = 6, 7, 8)$. See main text for further details. *Bottom panel:* This graph is similar to the top panel, except that the three response variables X_6, X_7, and X_8 obtain the deterministic input $f(X_4, X_5)$, where $f(.)$ is a sigmoidal function of the product $X_4 X_5$. See main text for further details.

where the noise variables ϵ_i $(i = 6, 7, 8)$ are i.i.d. Gaussian $N(0, \sigma^2)$ distributed. For overall consistency, all variables were standardized to a standard deviation of 1, and subsequently shifted such that the minimum was equal to zero. For both toy scenarios, I generated 25 independent data instantiations with $M = 100$ data points each, from 30 different combinations of the parameters $\sigma^2 \in \{0.0001, 0.001, 0.01, 0.1, 1, 10\}$ and $\theta \in \{1, 0.8, 0.6, 0.4, 0.2\}$.

I first applied the non-homogeneous hierarchical Bayesian regression model from Section 4.3.2 to the synthetic data generated from the model in Figure 4.18, top panel. The results are shown in the left panel of Figure 4.19. For low noise levels, $\sigma^2 \leq 0.01$, the network reconstruction accuracy tends to increase with increasing numbers of change-points. Interestingly, the opposite trend is observed for high noise levels, $\sigma^2 \geq 0.1$. This behaviour has the following explanation. A target node, say X_8, depends on the true regressor, X_5, through the non-linear transfer function $f_\theta(.)$ of Equation (4.7); the deviation from linearity increases with decreasing values of θ. On the other hand, there are two covariates, X_6 and X_7, which for low noise levels σ^2 will show a strong linear correlation with the target node X_8. Consider, without loss of generality, node $X_6 = f_\theta(X_5) + \epsilon_6$, which has a linear correlation with the target node, $X_8 = f_\theta(X_5) + \epsilon_8 = X_6 + \epsilon_8 - \epsilon_6 = X_6 + \tilde{\epsilon}$, but subject to double the amount of noise: $\tilde{\epsilon} = \epsilon_8 - \epsilon_6$ implies that $\mathrm{var}(\tilde{\epsilon}) = \mathrm{var}(\epsilon_8) + \mathrm{var}(\epsilon_6) = 2\sigma^2$. Hence, if the transfer function $f_\theta(.)$ is linear, then the true regressor, X_5, is preferred over the spurious one, X_6. However, if the transfer function $f_\theta(.)$ is non-linear, then the model used for network reconstruction needs sufficient non-linear modelling capability to capture the dependence between X_5 and X_8. Otherwise, the spurious variable X_6 will be learned, despite the noise amplification. Now, a non-homogeneous Bayesian regression model with change-points implements effectively a piece-wise linear function and can thus, in principle, approximate the sigmoidal function $f_\theta(.)$. The results depend on the combination of the noise level, σ^2, and the amount of non-linearity, θ. If the noise σ^2 is low, then the effect of the noise amplification, by which the spurious variables are suppressed, is weak, and non-linear modelling capability is critical for good performance, especially as the degree of true underlying non-linearity increases. In that case, more change-points are advantageous and improve the network reconstruction accuracy, as seen from the top rows of Figure 4.19, left panel. However, piece-wise linear regression models are very flexible and can potentially over-fit the data. This tendency towards overfitting gets stronger as the noise level σ^2 increases. In addition, higher noise levels intrinsically suppress spurious variables via the effect of noise amplification,

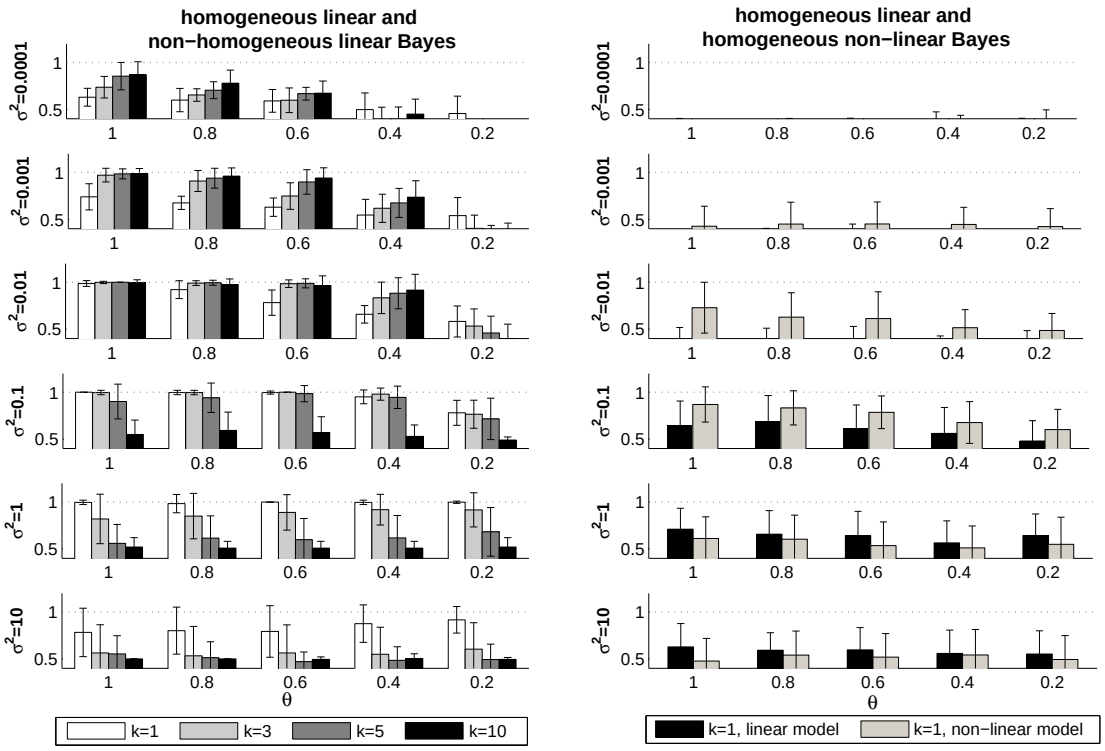


Figure 4.19: Network reconstruction accuracy on the synthetic data for non-homogeneous and non-linear Bayesian regression models. Synthetic network data were generated as described in Section 4.7.1. Left panel: Equation (4.7). Right panel: Equation (4.8). Different parameter combinations of σ^2 (the noise variance) and θ (the interaction strength) were used. The Bayesian regression models described in Sections 2.2–2.3 were applied to network reconstruction. Average AUROC scores were computed from 20 independent data instantiations. Both panels in the figure are arranged as matrices, where the rows correspond to σ^2 and the columns correspond to θ. *Left panel:* Histograms of the average AUROC scores for the homogeneous Bayesian regression model (white) and three non-homogeneous Bayesian regression models that partition the data with respect to the amplitude of the response variable, with $k = 3$ (light grey), $k = 5$ (grey), and $k = 10$ (dark-grey) segments. The change-point locations were inferred from the data. *Right panel:* Histograms of the average AUROC scores for homogeneous Bayesian regression models. Black bars refer to the conventional linear Bayesian regression models. Grey bars represent non-linear Bayesian regression models that also include two non-linear transformations of the regressor variables: inverse terms, and quadratic (2nd order) interactions terms. See Section 4.7.1 for details.

discussed above, thus reducing the need for non-linear modelling capability. As a consequence, more change-points become a disadvantage and deteriorate the network reconstruction accuracy, as seen from the bottom rows of Figure 4.19, left panel.

The right panel in Figure 4.19 compares the AUROC values obtained with two versions of the homogeneous hierarchical Bayesian regression model (Section 2.2): one has only linear terms as regressors (black boxes), the other also includes non-linear (inverse and quadratic) terms (grey boxes). The models were applied to synthetic data generated from the toy network in the right panel of Figure 4.18. For very low noise levels ($\sigma^2 \leq 0.001$), the network reconstruction is poor. For medium noise levels, ($0.01 \leq \sigma^2 \leq 0.1$), the network reconstruction improves, especially when including non-linear terms as regressors. For high noise levels, the opposite trend is observed: the network reconstruction deteriorates as a consequence of including non-linear terms as regressors. This pattern has a similar explanation as before, following the same trade-off between non-linear modelling capability and noise. A target variable, say X_8, depends non-linearly on two regressors: $X_8 = f_\theta(X_4, X_5) + \epsilon_5$, where $f(.)$ was defined in Equation (4.8). Two confounding covariates, X_6 and X_7, have the same dependence on X_4 and X_5. This leads to a spurious linear association with X_8, subject to noise amplification: $X_8 = X_6 + \epsilon_8 - \epsilon_6 = X_6 + \tilde{\epsilon}$, where $\text{var}(\tilde{\epsilon}) = \text{var}(\epsilon_8) + \text{var}(\epsilon_6) = 2\sigma^2$. For very low noise levels (Figure 4.19, right panel, top two rows), weakening the spurious linear associations by noise amplification cannot compensate for the approximation errors in modelling the true non-linear interactions; hence the poor performance. For medium noise levels (Figure 4.19, right panel, rows 3–4), the spurious linear correlations are suppressed against the non-linear true associations, especially if the model has non-linear approximation power due to the inclusion of non-linear regressors. For high noise levels, (Figure 4.19, right panel, bottom two rows), noise amplification alone substantially weakens the spurious associations, and additional non-linear modelling capability is counter-productive, due to potential overfitting.

In summary, the upshot of the synthetic toy study is as follows. Even if the true underlying regulatory processes are intrinsically non-linear, additional non-linear modelling capability, in the form of change-points or the explicit inclusion of non-linear terms, is not a panacea for better performance per se. As it turns out, the difference in performance between the linear and non-linear models depends on the amount of intrinsic non-linearity and the noise level. There is a weak trend that a higher degree of intrinsic non-linearity gives the non-linear model an edge on the linear model (Figure 4.19, left panel, rows 2 and 3). However, a more substantial influence has the noise

level. It is only for the lower noise levels that non-linear modelling capability has an advantage. For higher noise levels, it is overshadowed by the susceptibility to overfitting, which leads to a net performance deterioration. This explains why, in Figure 4.11, the linear Bayesian regression model shows a better performance than its more flexible cousins with change-points or non-linear terms, and why in Figure 4.16 the performance deteriorates with increasing numbers of change-points. The particular trade-off between non-linear modelling flexibility versus susceptibility to overfitting may vary with the nature of the data generation mechanism, which explains the different trends for mRNAs and proteins in Figure 4.15.

4.7.2 The effect of missing protein concentrations

Figure 4.13 suggests that the network reconstruction accuracy does not improve significantly when including protein concentrations in addition to just mRNA. To understand this counter-intuitive finding, note that two proteins in the circadian clock network, LHY and TOC1, occur in different isoforms, with only one of them acting as transcription factor. If protein data are missing, the gene coding for a regulatory protein has to be taken as a proxy for the regulatory protein itself, both in the modelling as well as in the gold-standard regulatory network. However, the influence of a gene on another gene is indirect, and since both protein isoforms are coded by the same gene, the distinction between isoforms becomes obsolete in the gene regulatory network. If protein data are available, then the model needs to identify the correct protein isoform to obtain a true positive score in the network prediction assessment. Hence, due to the correlation between the concentration profiles of the different isoforms, this is a harder prediction task than the reconstruction of the gene regulatory network from incomplete data (mRNA concentrations only), where this distinction is obsolete. The observation in Figure 4.13 can thus be explained as a partial compensation of two conflicting tendencies: incomplete data (mRNA only) causes an information loss, which should render the network reconstruction more difficult overall, but it also renders certain aspects of the network reconstruction task easier as a consequence of not having to distinguish between different protein isoforms. The net effect is no significant difference in performance.

4.7.3 Gaussian process performance

Regarding the poor performance of the GP, I emphasize that I was using the method exactly as described by Äijö and Lähdesmäki [6], using the authors' own software. The software uses the kernel of Equation (2.29). This is a kernel from the Matérn class, which depends on a further hyper-parameter ν; see Rasmussen and Williams [119] for the explicit expression. The hyper-parameter ν defines the degree of roughness, with $\nu = 1/2$ giving a rough Ornstein-Uhlenbeck process, and $\nu \to \infty$ reducing to the smooth squared exponential kernel of Equation (2.28). The kernel defined in Equation (2.29) corresponds to $\nu = 3/2$.

The GP model from Äijö and Lähdesmäki [6] thus depends on seven hyper-parameters: the mean $\mathbf{b} = (\overline{\alpha}, \overline{\lambda})$ and covariance $\sigma_b^2\mathbf{I}$ of the prior distribution on the basal transcription and decay rates, $\boldsymbol{\beta} = (\alpha, \lambda)$, the Matérn kernel parameters l, a and ν, and the noise variance σ^2. Only three of them are inferred in a maximum likelihood type-II sense: the length scale l, the amplitude a, and the noise variance σ^2. The other four hyper-parameters are fixed; these are ν, which defines the roughness of the Matérn class kernel, as well as the parameters that define the prior distribution on the linear parameter vector, $\boldsymbol{\beta}$.

The poor performance of the GP has two possible explanations. First, fixing four of the hyper-parameters might be too restrictive, and the chosen Matérn class with $\nu = 3/2$ might not be sufficiently representative of the actual concentration profiles. This might indicate that the choice of kernel is quite critical in determining the GP's performance. A second explanation is that for each gene n, the authors choose the set of regulators that maximizes the posterior probability of Equation (2.35). This probability is conditional on the hyper-parameters. The methodologically correct approach would be to integrate the hyper-parameters out, as e.g. discussed in Section 5 by MacKay [94]. The GP method that I have applied, effectively ignores the last two terms in Equation (5.3) of that paper. If the posterior distribution over the hyper-parameters is sharply peaked, that will not matter, as integration and optimization then effectively lead to identical results. However, it will make a difference if the posterior distribution is diffuse, in which case the GP model selection is suboptimal. My results thus suggest that the method presented by Äijö and Lähdesmäki [6] could be made more powerful with a more rigorous inference scheme, the development of which is beyond the scope of this study.

4.7.4 Comparison with other methods

I briefly discuss the performance of the other methods included in my comparative evaluation. The mutual information based method ARACNE showed the poorest performance. This is not surprising, given that most of the true networks included in my study, shown in Figure 4.1, violate the premise on which the theoretical foundations of ARACNE are based. The ARACNE network reconstruction theorem states that given some further regularity conditions, ARACNE can correctly reconstruct tree-like networks, i.e. networks containing only pairwise interactions [97]. However, there is no theoretical guarantee that densely connected networks or networks containing loops can be correctly reconstructed, and my empirical study suggests that the performance of ARACNE for such networks is, in fact, rather poor.

The poor performance of the Gaussian mixture model (GMM) is presumably due to the fact that model selection is carried out with BIC. BIC is computationally cheap, but over-regularised, leading to structures that are too sparse. My results are further consistent with the findings by Neuneier et al. [110] that modelling conditional probabilities indirectly via Equation (2.40) is inferior to modelling them directly with regression-type models, e.g. of the form discussed in [79].

The observation that Tesla shows a slightly poorer performance than Lasso is consistent with the observation that the inclusion of change-points for the light phases slightly degrades the performance of the hierarchical Bayesian model, as discussed in Section 4.7.1.

It might be surprising that the Bayesian splines autoregression (BSA) method did not outperform the computationally cheaper linear sparse regression methods Lasso and Elastic Net. This is caused by an over-sparsity of the networks predicted with BSA. As discussed by Morrissey et al. [106], the inclusion of an edge in the network leads to a more substantial increase in the parameter space dimension than for a linear model, due to the fact that an edge is associated with the high-dimensional parameter vector of the splines. Recall from Section 2.8 that the strength of the interaction between two genes n and n', which is modelled with a scalar $w_{n,n'}$ in a linear model, becomes a vector in BSA, $\mathbf{w}_{n,n'}$, spanning the entire range of B-spline basis functions. Hence, the Bayesian approach per se penalises more severely against the inclusion of extra edges than for a linear model, and the non-linear modelling potential of the splines was found to insufficiently compensate for that. I noticed that the performance of BSA improved when the default Jeffreys prior on the edge inclusion probability was replaced by a more informative prior with a concentration of probability mass

above 0.5. I have not included these results, because tuning hyper-parameters based on the network reconstruction performance is methodologically incorrect (as it would be using knowledge that is not available in real applications). These findings indicate, though, that the performance of BSA can in principle be boosted by the inclusion of informative prior knowledge. However, even when exploring deviations from the Jeffreys prior, BSA never quite reached the performance of the linear HBR method. This is consistent with the observation that my own non-linear variants of HBR never outperformed the linear version; I refer the reader back to Section 4.7.1 for a discussion of this trend.

Please refer to Section 6.1 for the conclusion of this chapter.

Chapter 5

Learning Ecological Networks

The relationships among organisms and their surroundings can be of immense complexity. To describe and understand an ecosystem as an entangled bank, multiple ways of interaction and their effects have to be considered, such as predation, competition, mutualism and facilitation. Understanding the resulting interaction networks is a challenge in changing environments, e.g. to predict knock-on effects of invasive species and to understand how climate change impacts biodiversity. The elucidation of complex ecological systems with their interactions is likely to benefit enormously from the development of new machine learning tools that aim to infer the structure of interaction networks from field data.

In this Chapter, I propose two Bayesian regression models for reconstructing species interaction networks from observed species distributions: The Bayesian regression and multiple change-point model (BRAM) as published by Aderhold et al. [2] and the Bayesian regression and Mondrian process model (BRAMP) as published by Aderhold et al. [3]. Both models have been devised to allow robust inference in the presence of spatial autocorrelation and variation in the interactions across space, i.e. distributional heterogeneity. I have evaluated both models on simulated data that combines a trophic niche model with a stochastic population model on a 2-dimensional lattice, and I have compared the performance to ℓ_1-penalized sparse regression (Lasso) and non-linear Bayesian networks with the BDe scoring scheme (Banjo). In addition, I have applied these methods to plant ground coverage data from the western shore of the Outer Hebrides with the objective to infer the ecological interactions.

Colin J. Beale and Jack L. Lennon contributed to this study by giving valueable suggestions about the ecological interpretation of my findings.

5.1 Introduction

Recent endeavours in systems biology aiming to elucidate the structure of complex interaction networks have sparked off a series of novel applications and methodological innovations in machine learning and computational statistics. This has become most evident in the field of molecular systems biology, where a large variety of more advanced methods have been developed. This includes, for instance, approximate Bayesian inference for pathway ranking [145], Gaussian process models for transcriptional regulation [78], and non-stationary dynamic Bayesian networks for inferring time-varying gene interactions [89]. The latter work in particular has motivated new machine learning research, related to the combination of dynamic Bayesian networks and multiple change-point processes [120, 59].

Ecosystems are complex dynamic systems, with interconnected networks of species interactions. Unravelling these networks strains the limits of typical ecological studies, requiring intensive observation to determine trophic interactions (predator-prey interactions) in even simple ecosystems, e.g. in [100]. And trophic interactions are not the whole story, as harder-to-observe interactions such as competition and mutualism (species interacting in a way that both partners benefit) also influence ecosystem dynamics [151]. Measures of such indirect interactions have been attempted [143], but computational inference presents an alternative, and perhaps more comprehensive, route to revealing both direct and indirect interactions within ecosystems.

Ecosystem interactions will leave traces in species distribution across space, a measure relatively easily obtained and currently available for many ecosystems, e.g. by Hagemeijer and Blair [64]. Computational algorithms can make use of such observational data, as has also been done in other areas of biology, e.g. neural activity for information flow in the brain [132], to reverse engineer the ecological interactions [103, 40]. Furthermore, as the algorithms recover interactions based on their influence on species distribution, they are not limited to any one particular type of interaction (e.g., trophic, competition), and instead are capable of revealing interactions of all types simultaneously.

Computational inference in ecological systems is challenged by the fact that interactions take place in a 2-dimensional explicit environment, and that these interactions can vary across space. The main driving forces in this respect are changes in the environment and population densities that produce direct or indirect effects on species interaction dynamics, e.g. leading to the formation of ecological niches. Hence, a chal-

lenge in ecological modelling is the partitioning of space into local neighbourhoods with similar population dynamics that can be separately modelled[1]. Prediction methods using this knowledge can improve their model accuracy. In addition, it can be beneficial to learn the partitioning directly from the data and in this way gain knowledge about potential neighbourhoods. Another distinctive feature of species networks is the so-called autocorrelation effect and dispersion of species, which can impact population densities in neighbouring regions.

Here, I meet these challenges by modifying the non-homogeneous Bayesian regression method described in Section 2.3 to 2 dimensions. The model in Section 2.3 was previously inspired by Lèbre et al. [89][2], which combines the Bayesian hierarchical regression model from Andrieu and Doucet [7] with a multiple change-point process, as proposed by Punskaya et al. [116], and pursues Bayesian inference with reversible jump Markov chain Monte Carlo (RJMCMC) [56]. I extend the model with an inference mechanism that attempts to learn the local neighbourhoods from the observed population densities in two different ways. First, I expand the 1-dimensional change-point process to two dimensions, by introducing two *a priori* independent change-point processes in perpendicular directions. This model was proposed by Aderhold et al. [2] and is called Bayesian regression and multiple change-point process (BRAM), described in Section 5.2.4. Second, I replace the multiple change-point process with a Mondrian process that has been introduced in a different context by Roy and Teh [124]. Applying the Mondrian process to the spatial domain allows a more precise partitioning of 2-dimensional space. The model was introduced by Aderhold et al. [3] and is called BRAMP, defined in Section 5.2.5. In both models, I make further use of the spatially explicit nature of ecological data by correcting for spatial autocorrelation with a parent node (in Bayesian network terminology) that explicitly represents the spatial neighbourhood of a node (see Section 5.2.3).

I evaluate the performance of the models on two synthetic data sets. The first is purely simulated data that directly resembles the underlying assumption of the linear regression model of BRAM and BRAMP together with a artificial global and Mondrian-type partitioning of space. The second data set is generated from a realistic simulation, which combines a trophic niche model of Lotka-Volterra type predator-prey interactions

[1] The separation does not necessarily mean that local neighbourhoods or "segments" become completely independent from each other but one could implement mechanisms that support information sharing, e.g. of interaction strengths in adjacent neighbourhoods or similar interconnectivity.

[2] In fact I used the original definitions and model implementation by Lèbre et al. [89] to perform the modifications. The model description in Section 2.3 that is based on Grzegorczyk and Husmeier [59] descends from the latter work.

with a stochastic population model on a 2-dimensional lattice. I compare the model's performance on both of these simulated data sets with the ℓ_1-norm penalized sparse regression (Lasso) and non-linear Bayesian networks (BDe score). I then apply my models to species counts of ground cover flora and associated abiotic variables from a strip of land across an environmental gradient on the western shore of the Outer Hebrides, to assess my model's applicability and utility for real ecological data.

5.2 Model

This section describes the modelling approach for BRAM and BRAMP, which combine the Bayesian hierarchical regression model from Andrieu and Doucet [7] and Punskaya et al. [116] with a global change-point process and Mondrian change-point process [124, 147] and pursue Bayesian inference with RJMCMC [56].

Both models are a network represented as a graph $\mathcal{G}$ in which nodes represent species, and edges (i.e. connections between nodes) represent potential species interactions. The value that the nth node in the graph $\mathcal{G}$ takes on at a given location represents the abundance of the nth species in the population. This abundance is determined by various biotic and abiotic determinants, i.e. factors that influence the abundance of species n. Abiotic factors are related to the environment and include e.g. temperature, humidity, soil type etc.. Biotic factors represent the abundance of other species. Their influences are indicative of how species interact and I aim to reconstruct the network given the spatial species abundance profiles and additional abiotic factors if available[3].

The strengths of the influences are allowed to vary geographically, based on a stochastic process of spatial variation. More specifically, the conditional probability of a species abundance at a given location is a conditional Gaussian distribution, where the conditional mean is a linear weighted sum of the abundance levels of the biotic and abiotic determinants. I model this mathematically with an approach based on Bayesian regression, which intrinsically incorporates a regularization effect that discourages the prediction of spurious interactions. The weight parameters can vary between different segments of a spatial segmentation, which adds extra flexibility to the model and allows for unobserved or latent factors. The interaction weights, the variance parameters, and the number of potential determinants are given (conjugate) prior distributions in a hierarchical Bayesian model, and the spatial segmentation is modelled with two change-

[3] Both synthetic studies do not include abiotic variables but the real world data has 12 abiotic factors that are treated in the same way as the biotic factors.

points along each spatial direction (BRAM) and non-parametrically with a Mondrian process prior (BRAMP).

For inference, all quantities are sampled from the posterior distribution with RJMCMC. Note that a complete specification of all species-determinant configurations determines the structure of a regulatory network $\mathcal{G}$: each node receives incoming directed edges from each node in its set of determinants (the so-called parent set).

5.2.1 Species interaction network

Following the definition in Section 2.1 I represent the N interacting species as nodes $n \in \{1, \dots, N\}$ in a directed graph or network $\mathcal{G} = \{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_N\}$, where $\boldsymbol{\pi}_n$ denotes the parents of node n, that is the set of nodes with a directed edge pointing to n. $\mathcal{G}_n$ is the sub-network associated with target species n, which is determined by its parent set $\boldsymbol{\pi}_n$. In contrast to mRNA self-loops studied in Chapter 4, a node cannot be contained in its own parent set, $n \notin \boldsymbol{\pi}_n$, i.e. I rule out self-interactions related to e.g. cannibalism. The species are observed or surveyed at $M^1 \times M^2$ locations defined by their (orthogonal) coordinates (l_1, l_2), at which their abundance levels $x = \{x_n(l_1, l_2)\}_{1 \leq n \leq N, 1 \leq l_1 \leq M^1, 1 \leq l_2 \leq M^2}$ are determined.

5.2.2 Regression

For all species n, the random variable $x_n(l_1, l_2)$ refers to the abundance of species n at location (l_1, l_2). Hence, (l_1, l_2) replaces the previously used symbol for an observation in the 1-dimensional domain m. Instead, I reuse m to designate a Mondrian sample as defined in Section 5.2.5. For convenience I write $x_n(l_1, l_2)$ instead of $x_{n,(l_1,l_2)}$, which would follow the previous convention of $x_{n,m}$. In addition, I do not have to differentiate between response and predictor data, which has been time gradient data (response) and concentration data (predictor) in the gene regulation study. Both of these variable types in this Chapter are species densities and will be consistently symbolized by $x_n(l_1, l_2)$.

Within any segment h, the abundance of $x_n(l_1, l_2)$ depends on the abundance levels of the species in the regulator set of species n, $\boldsymbol{\pi}_n$. The regulator set $\boldsymbol{\pi}_n$ is defined to be the same in all segments $h \in \{1, \dots, H\}$ to rule out fundamental changes to the network, since network changes among segments are less likely to occur than a change in interaction strength. I model the segment-specific linear regression model with the set of parameters $\{(w^p_{n,h})_{p \in \boldsymbol{\pi}_n}, \sigma_{n,h}\}$, where $w^p_{n,h} \in \mathbb{R}$ is a regression coefficient and $\sigma^2_{n,h} > 0$ is the noise variance for each segment h and target species n. For all species

n and all locations (l_1, l_2) in segment h, the response species $x_n(l_1, l_2)$ depends on the abundance variable of the predictor species $\{x_p(l_1, l_2)\}_{p \in \boldsymbol{\pi}_n}$ according to

$$x_n(l_1, l_2) = w^0_{n,h} + \sum_{p \in \boldsymbol{\pi}_n} w^p_{n,h} x_p(l_1, l_2) + \varepsilon_n(l_1, l_2) \tag{5.1}$$

where $\varepsilon_n(l_1, l_2)$ is assumed to be white Gaussian noise with mean 0 and variance $\sigma^2_{n,h}$, $\varepsilon_n(l_1, l_2) \sim N(0, \sigma^2_{n,h})$. I define $\mathbf{w}_{n,h} = (w^0_{n,h}, \{w^p_{n,h}\}_{p \in \boldsymbol{\pi}_n})$ to denote the vector of all regression parameters of species n in segment h. This includes the parameters defining the strength of interactions with other species p, $w^p_{n,h}$, as well as a species-specific offset term, i.e. the intercept or bias $w^0_{n,h}$. Equation (5.1) is not the final equation of regression but is further extended with the spatial autocorrelation factor in the following Section.

5.2.3 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 [90, 32]. In this case, spatial autocorrelation could lead to the identification of spurious interactions as a mere consequence of two species co-occurring in similar geographical regions. To incorporate potential spatial autocorrelation into the model, I follow an approach proposed by Faisal et al. [40] and illustrated in Figure 5.1b. The idea is to connect each node in the network to an enforced parent node that represents the average population at neighbouring cells, weighted inversely proportional to the distance of the neighbours:

$$A_n(l_1, l_2) = \frac{\sum_{(\tilde{l}_1, \tilde{l}_2) \in \mathcal{N}(l_1, l_2)} d^{-1}[(l_1, l_2), (\tilde{l}_1, \tilde{l}_2)] x_n(\tilde{l}_1, \tilde{l}_2)}{\sum_{(\tilde{l}_1, \tilde{l}_2) \in \mathcal{N}(l_1, l_2)} d^{-1}[(l_1, l_2), (\tilde{l}_1, \tilde{l}_2)]} \tag{5.2}$$

where $\mathcal{N}(l_1, l_2)$ is the spatial neighbourhood of location (l_1, l_2) (e.g. the four nearest neighbours), and $d[(l_1, l_2), (\tilde{l}_1, \tilde{l}_2)]$ is the Euclidean distance between (l_1, l_2) and $(\tilde{l}_1, \tilde{l}_2)$. The value of $A_n(l_1, l_2)$, weighted by an additional regression coefficient $w^A_{n,h}$, is integrated into Equation (5.1) yielding:

$$x_n(l_1, l_2) = w^0_{n,h} + \sum_{p \in \boldsymbol{\pi}_n} w^p_{n,h} x_p(l_1, l_2) + \varepsilon_n(l_1, l_2) + w^A_{n,h} A_n(l_1, l_2) \tag{5.3}$$

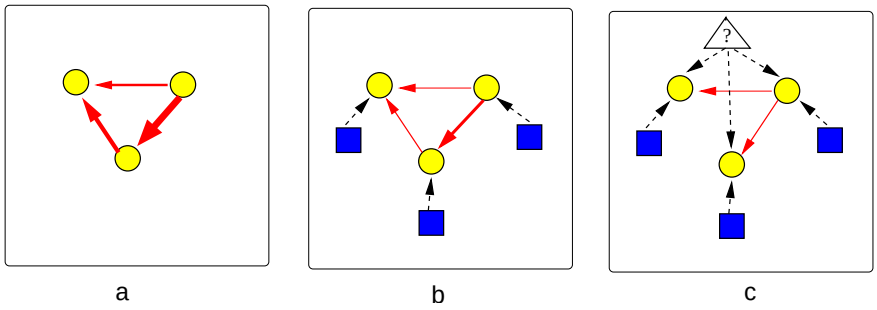


Figure 5.1: Illustration of the improved method for ecological network reconstruction. *Panel (a)* illustrates the naive approach to modelling species interaction networks. Circles represent species (nodes), and arrows present species interactions (edges). Networks inferred from species abundance or population density data alone tend to contain many spurious interactions. *Panel (b): Allowing for spatial autocorrelation.* Each node is hard-wired to an indicator node (square) that represents, via equation (5.2), the average population density in the spatial neighbourhood. *Panel (c): Allowing for missing data.* The model can be further improved by connecting all nodes to a latent node that represents unobserved effects. The observation status at a node is, in the first instance, predicted by the spatial neighbourhood and/or the latent variable. Only if the explanatory power of these correction schemes is not sufficient will there be an incentive for the inference scheme to include further edges related to species interactions. Hence the effect of these corrections is to reduce the network connectivity and filter out spurious interactions.

Thus the regression vector expands to $\mathbf{w}_{n,h} = (w^0_{n,h}, \{w^p_{n,h}\}_{p\in\boldsymbol{\pi}_n}, w^A_{n,h})$. In this way the abundance of species n at location (l_1, l_2) is, in the first instance, determined 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.

5.2.4 BRAM: Multiple Global Change-points

One major goal of the thesis is the expansion of the 1-dimensional change-point process that was described in Section 2.3 to 2 dimensions. To accomplish this each spatial dimension can be treated independently by applying a Poisson process to the change-point vectors associated to each spatial dimension. This naturally leads to a global segmentation, i.e. a change-point in either one of the dimensions will reach across the whole space as illustrated in Figure 5.2.

The regulatory relationships among the species may be influenced by latent vari-

Symbol	Short verbal description
s_n	number of parents for response node n
$\overline{s}$	fan-in restriction for parent set
Λ	mean of parent edges
i	index of orthogonal directions, horizontal ($i = 1$) or vertical ($i = 2$)
M^i	number of locations/samples in direction i
k_n^i	number of intervals demarcated by the change-points along direction i for response node n
$\overline{k^i}_n$	maximum number of change-points in direction i for response node n
λ	mean of the change-points
$\boldsymbol{\tau}_n^i$	change-point vector for response node n and direction i, $\boldsymbol{\tau}_n^i = (\tau_{n,0}^i, \ldots, \tau_{n,k^i+1}^i)$

Table 5.1: Overview of symbols related to the global multiple change-points process. Note, that some of the parameters depend on the response node n. However, in most instances I will drop the corresponding subsript in the text to avoid cluttering.

ables, which are represented by spatial change-points. I assume that latent effects in close spatial proximity are likely to be similar, but locations where spatially close areas are not similar are distinguished by change-points. They are modelled with two *a priori* independent multiple change-point processes along the two orthogonal spatial directions denoted with $i \in \{1, 2\}$ for each response node n. Note that I will omit the node index n for each change-point vector or value to improve readability, but each response node n has in fact a particular set of change-points $\boldsymbol{\tau}_n^1$ and $\boldsymbol{\tau}_n^2$ associated to it, and, thus, a particular number of change-points k_n^1 and k_n^2. I will drop the subscript n in some occasions to avoid cluttering. A change-point vector is then defined with $\boldsymbol{\tau}^i = (\tau_0^i, \ldots, \tau_{k^i+1}^i)$, where $\tau_0^i := 1, \tau_{k^i+1}^i := M^i$, and M^i is the number of locations (observations) in each direction i. The vector $\boldsymbol{\tau}^i$ thus contains an *a priori* unknown number of k^i change-points, and both vectors, $\boldsymbol{\tau}^1$ and $\boldsymbol{\tau}^2$, partition the space into $H = (k^1 + 1)(k^2 + 1)$ non-overlapping segments, demarcated by the change-points. I denote the latent variable associated with a segment by $h \in \{1, \ldots, H\}$. If two locations (l_1, l_2) and $(\tilde{l}_1, \tilde{l}_2)$ are in the same segment, $\tau_a^1 \leq l_1, \tilde{l}_1 < \tau_{a+1}^1$ and $\tau_b^2 \leq l_2, \tilde{l}_2 < \tau_{b+1}^2$, where $a \in \{0, \ldots, k^1\}$ and $b \in \{0, \ldots, k^2\}$, then they are assigned the same latent variable: $h(l_1, l_2) = h(\tilde{l}_1, \tilde{l}_2)$. Thus, the latent effect is conveyed through the assignment of coordinates that exist in close vicinity and belong to the same "neighbourhood" (segments). This neighbourhood is defined with the change-points $\boldsymbol{\tau}^i$ and the isomorphism between change-points and segments, such that segment h is demarcated by

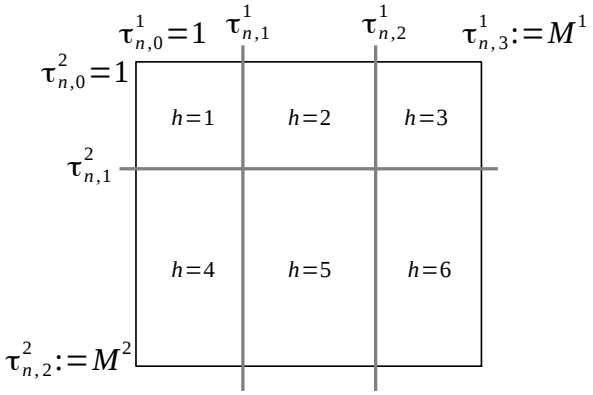


Figure 5.2: Multiple global change-point example. Partitioning with a horizontal change-point vector $\boldsymbol{\tau}_n^{i=1} = (\tau^1_{n,1}, \tau^1_{n,2})$ and vertical vector $\boldsymbol{\tau}_n^{i=2} = (\tau^2_{n,1})$. The pseudo-change-points $\tau^1_{n,0} = \tau^2_{n,0} := 1$ define the left and upper boundaries, whereas $\tau^1_{n,3} = M^1$ and $\tau^2_{n,2} = M^2$ define the right and lower boundaries, where M^1 and M^2 are the number of locations along the horizontal and vertical direction, respectively. The number of change-points is $k^1_n = 2$, $k^2_n = 1$ and the number of segments $H_n = 6$.

change-points $\{\tau^1_{[f_1(h)-1]}, \tau^1_{f_1(h)}, \tau^2_{[f_2(h)-1]}, \tau^2_{f_2(h)}\}$.

5.2.4.1 Prior probability

To encourage a sparse network structure in the sub-graph $\mathcal{G}_n \in \mathcal{G}$, I impose a truncated Poisson prior with mean Λ and maximum fan-in $\overline{s}$ on the number s_n of parents for a response node n:

$$p(s_n|\Lambda) \propto \frac{\Lambda^{s_n}}{s_n!}\mathbb{1}_{\{s_n \leq \overline{s}\}} \tag{5.4}$$

In my synthetic evaluations, small values of $\overline{s} = (3, 4, 5)$ have brought a noticeable difference in performance compared to higher settings of $\overline{s}$. Conditional on s_n, the prior for the parent set $\boldsymbol{\pi}_n$ is a uniform distribution over all parent sets with cardinality s_n: $p(\boldsymbol{\pi}_n \,|\, |\boldsymbol{\pi}_n| = s_n) = 1/\binom{N-1}{s_n}$. The overall prior on the network structure $\mathcal{G}$ is given by factorization and marginalization:

$$p(\mathcal{G}|\Lambda) = \prod_{n=1}^{N} p(\boldsymbol{\pi}_n|\Lambda);$$
$$p(\boldsymbol{\pi}_n|\Lambda) = \sum_{s_n=1}^{\overline{s}} p(\boldsymbol{\pi}_n|s_n)p(s_n|\Lambda) \tag{5.5}$$

For both spatial directions $i \in \{1,2\}$, the (k^i+1) segments are delimited by k^i change-points, where k^i is distributed a priori as a truncated Poisson random variable with mean λ and maximum number of change-points $\overline{k^i} = M^i - 1$:

$$p(k^i|\lambda) \propto \frac{\lambda^{k^i}}{k^i!}\mathbb{1}_{\{k^i \leq \overline{k^i}\}} \tag{5.6}$$

Note that for practical application, the setting of $\overline{k^i}$ is not limited to $M^i - 1$, but could be set to smaller values, and thus further restrict the estimated amount of change-point values k^i through the truncated prior in the previous Equation. Conditional on k^i change-points, the change-point position vector $\boldsymbol{\tau}^i = (\tau^i_0, \ldots, \tau^i_{k^i+1})$ takes non-overlapping integer values, which I take to be uniformly distributed a priori. There are $(M^i - 1)$ possible positions for the k_i change-points, thus vector $\boldsymbol{\tau}^i$ has the prior density $p(\boldsymbol{\tau}^i|k^i) = 1/\binom{M^i-1}{k^i}$. Conditional on the parent set $\boldsymbol{\pi}_n$ of size s_n, the $s_n + 2$ regression coefficients, denoted by $\mathbf{w}^h_n = (w^h_{n0}, w^h_{nA}, (w^h_{nm})_{m\in\boldsymbol{\pi}_n})$, are assumed zero-mean multivariate Gaussian distributed with covariance matrix $(\sigma^h_n)^2\Sigma_n$,

$$p(\mathbf{w}^h_n|\boldsymbol{\pi}_n, \sigma^h_n) \;=\; |2\pi(\sigma^h_n)^2\Sigma_{n,h}|^{-\frac{1}{2}}\exp\left(-\frac{[\mathbf{w}^h_n]^\dagger\Sigma^{-1}_{n,h}\mathbf{w}^h_n}{2(\sigma^h_n)^2}\right) \tag{5.7}$$

where the symbol $\dagger$ denotes matrix transposition, $\Sigma_{n,h} = \delta^{-2}D^\dagger_{n,h}(x)D_{n,h}(x)$ and $D_{n,h}(x)$ is the $s_{n,h} = \prod_{i=1}^{2}(\tau^i_{f_i(h)} - \tau^i_{f_i(h)-1}) \times (s_n + 2)$ matrix whose first column is a vector of 1s, for the constant in (5.1), the second column is a vector of autocorrelation variables, defined in (5.2), and the remaining columns contain the observed abundance values $x_n(l_1, l_2)$ for all species $n \in \boldsymbol{\pi}_n$ and all locations (l_1, l_2) in segment h: $\tau^i_{f_i(h)-1} \leq l_i < \tau^i_{f_i(h)}, i \in \{1,2\}$. This so-called g-prior is widely used in Bayesian statistics; see e.g. Andrieu and Doucet [7]. Finally, the conjugate prior for the variance $(\sigma^h_n)^2$ is the inverse gamma distribution, $p((\sigma^h_n)^2) = \mathcal{IG}(\upsilon_0, \gamma_0)$. Following Lèbre et al. [89], I set the hyper-hyper-parameters 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 change-points and parents, respectively, and δ^2 is the expected signal-to-noise ratio. Following Lèbre et al. [89], these hyper-parameters are drawn

from vague conjugate hyper-priors, which are in the (inverse) gamma distribution family: $p(\Lambda) = p(\lambda) = \mathcal{G}\dashv(0.5, 1)$ and $p(\delta^2) = \mathcal{IG}(2, 0.2)$.

5.2.4.2 Posterior probability

Equation (5.3) implies that the likelihood is

$$p(x_n^h|\tau^1_{f_1(h)-1}, \tau^1_{f_1(h)}, \tau^2_{f_2(h)-1}, \tau^2_{f_2(h)}, \mathcal{G}, \mathbf{w}_n^h, \sigma_n^h) = \left(\sqrt{2\pi}\sigma_n^h\right)^{-s_{n,h}} \exp\left(-\frac{(x_n^h - D_{n,h}(x)\mathbf{w}_n^h)^\dagger\ (x_n^h - D_{n,h}(x)\mathbf{w}_n^h)}{2(\sigma_n^h)^2}\right) \tag{5.8}$$

From Bayes theorem, the posterior distribution is given by the following equation, where all prior distributions have been defined above:

$$\begin{aligned} p(k_1, k_2, \boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \mathcal{G}, \mathbf{w}, \sigma^2, \lambda, \Lambda, \delta^2|x) \propto\ & p(\delta^2)p(\lambda)p(\Lambda)p(\mathcal{G}|\Lambda) \\ & \prod_{i=1}^{2} p(k_i|\lambda)p(\boldsymbol{\tau}^i|k^i) \prod_{h=1}^{H}\prod_{n=1}^{N} p([\sigma_n^h]^2)p(\mathbf{w}_n^h|\boldsymbol{\pi}_n, [\sigma_n^h]^2, \delta^2) \\ & p(x_n^h|\tau^1_{f_1(h)-1}, \tau^1_{f_1(h)}, \tau^2_{f_2(h)-1}, \tau^2_{f_2(h)}, \mathcal{G}, \mathbf{w}_n^h, \sigma_n^h) \end{aligned} \tag{5.9}$$

The signal-to-noise prior $p(\delta^2)$, the prior for the mean number of parents $p(\Lambda)$ and change-points $p(\lambda)$, and the network structure $p(\mathcal{G}|\Lambda)$ are the same for all segments of the homogeneous model. They are placed outside the product terms of Equation 5.9 together with the prior for the number of change-points $p(\lambda)$. The priors that relate to individual change-point vectors, i.e. the number of change-points $p(k^i|\lambda)$, and the change-point vector probability $p(\boldsymbol{\tau}^i|k^i)$ are multiplied for each spatial dimension $i = 1$ and $i = 2$. Finally, the probabilities that can change for each segment and individual response include the noise variance prior $p([\sigma_n^h]^2)$, the parameter prior $p(\mathbf{w}_n^h)$, and the likelihood $p(x_n^h)$. They are placed inside the products of Equation 5.9 that iterate over all segments $h = (1, .., H)$ and response nodes $n = (1, .., N)$.

5.2.4.3 Inference

An attractive feature of the chosen model is that the marginalization over the parameters $\mathbf{w} = \{\mathbf{w}_n^h, 1 \leq n \leq N, 1 \leq h \leq H\}$ and $\sigma^2 = \{(\sigma_n^h)^2, 1 \leq n \leq N, 1 \leq h \leq H\}$ in the posterior distribution of (5.9) is analytically tractable [89, 7]:

Symbol	Short verbal description
i	vertical ($i = 1$) or horizontal ($i = 2$) direction
Θ_i	spatial domain, normalized to interval $[0, 1]$
λ	budget of a Mondrian sample, $\lambda' = \lambda - E$
τ_i	vertical or horizontal size of a Mondrian sample, $\tau_i = \lvert\Theta_i\rvert$
τ	half-perimeter of sample $\tau = \lvert\Theta_1\rvert + \lvert\Theta_2\rvert$
K	total number of samples (nodes) in the Mondrian tree
k	index of node in the Mondrian tree corresponds to Mondrian sample
E_k	cost of a cut of sample k, $E \sim \exp(\tau)$
χ_k	cut position in sample k
m	Mondrian sample $m = \langle i, \chi, \lambda', m_<, m_> \rangle$
m_k	sample corresponding to the node with index k in the Mondrian tree
$m_<, m_>$	left/right or top/bottom descendent of a sample m
H	total number of (uncut) segments defined by the Mondrian process
$h(k)$	segment derived from Mondrian sample m_k
ϕ	Mondrian tree leaf siblings, i.e. adjacent samples $(m_<, m_>)$ that can be merged
$\boldsymbol{\zeta}$	parameter vector containing all costs E and cuts χ

Table 5.2: Overview of symbols related to the Mondrian process. See Figure 5.3 for an illustration.

$$p(k^1, k^2, \boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \mathcal{G}, \lambda, \Lambda, \delta^2 | x) \;=\; \int p(k^1, k^2, \boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \mathcal{G}, \mathbf{w}, \sigma^2, \lambda, \Lambda, \delta^2 | x) d\mathbf{w} d\sigma^2 \qquad (5.10)$$

The number of change-points and their location, $k^1, k^2, \boldsymbol{\tau}^1, \boldsymbol{\tau}^2$, the network structure $\mathcal{G}$ and the hyper-parameters $\lambda, \Lambda, \delta^2$ can be sampled from the posterior distribution $p(k^1, k^2, \boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \mathcal{G}, \lambda, \Lambda, \delta^2 | x)$ with RJMCMC [56], following the scheme in [89, 7]. By marginalization and under the assumption of convergence, this gives me a sample of networks from the posterior distribution $p(\mathcal{G}|x)$. By further marginalization, I get the posterior probabilities of all species interactions $p(n \to \tilde{n}|x)$, which defines a ranking of the interactions in terms of posterior confidence. For the synthetic data for which the true network structure is known, this ranking allows the computation of the areas under the ROC (AUROC) and precision-recall (AUPREC) curves as discussed in Section 3.4.1.

5.2.5 BRAMP: Mondrian Process Change-points

The global change-point process described in the previous section lacks the capability to create segmentations with spatially varying length scales and different local fineness and coarseness characteristics. In fact, introducing global change-points that might improve segmentation in one region can introduce artefacts in the form of undesired partitioning in other regions. In order to provide varying levels of fineness of the segments and thereby account for spatial alterations of the regulatory relationships among species on a local scale, I adapt a local partitioning approach called the Mondrian process, introduced in [124] and described in detail in [123]. The Mondrian process can be expressed as a recursive generative process that randomly executes axis-aligned cuts, partitioning the underlying space in a hierarchical fashion akin to decision trees or kd-trees (Figure 5.3). The distinguishing feature of this recursive stochastic process is that it assigns probabilities to the various events in such a way that it is consistent (in a sense I make precise later). The implication of consistency is that the Mondrian process can be extended to infinite spaces and used as a non-parametric prior in multiscale modelling. It can also be regarded as a n-dimensional generalization of the Poisson process, and it has the same self-consistency property.

Here I will introduce the Mondrian process into the framework of a Bayesian regression model and partitioning a 2-dimensional domain $\Theta_1 \times \Theta_2$ (longitude times latitude) inhabited by the species profiles as published in [3]. The so-called "budget" is a hyper-parameter λ that determines the average number of cuts in the partition. At each stage of the recursion, a Mondrian sample can either define a trivial partition $\Theta_1 \times \Theta_2$, i.e. a segment, or a cut that creates two sub-processes $m_<$ and $m_>$: $m = \langle i, \chi, \lambda', m_<, m_> \rangle$, where i is the horizontal or vertical direction and χ the position of the cut. The direction i and position χ are drawn from a binomial and uniform distribution, respectively, both depending on Θ_1 and Θ_2, as shown in line 5 of Algorithm 1. The process of cutting a segment is limited by the budget λ associated to each segment and the cost E of a cut. Conditional on half-perimeter $\tau = |\Theta_1| + |\Theta_2|$, a cut is introduced yielding $m_<$ and $m_>$ if the cost $E \sim \exp(\tau)$ does not exceed the budget λ, i.e. satisfies $\lambda' = \lambda - E > 0$. The process is recursively repeated on $m_<$ and $m_>$ until the budgets are exhausted, as shown in Algorithm 1. This creates a binary tree with the initial Mondrian sample $m_{k=1}$ as the root node spanning the unit square $[0;1]^2$ and sub-nodes representing Mondrian samples $m_{1<k\leq K}$, $k \in \{1, \ldots, K\}$ where K is the total number of nodes in the tree, e.g. $K = 15$ in Figure 5.3. The leaf nodes present non-overlapping segments and are associated each with a latent variable $h(k) \in \{1, \ldots, H\}$ labelled with

$m^{h(k)}$ in the tree (right panel in Figure 5.3), where H constitutes the total number of leafs associated to the number of uncut segments in the 2-dimensional domain, e.g. $H = 8$ in the left panel of Figure 5.3. Hence, the variable $h(k)$ is an index to the segments or 'spatial neighbourhoods' in space and can be understood as a latent effect that determines the different interactions among species, as described in the regression model in Section 5.2.2.

I apply the same regression model as defined Section 5.2.2, with the sole difference that the segment indices $h \in \{1, \dots, H\}$ are replaced with the uncut Mondrian partition indices $h(k) \in \{1, \dots, H\}$.

Algorithm 1 MCMC Mondrian cut: Note, the Mondrian generative process corresponds to lines 1-4 and 7, i.e. the MCMC move extends it by considering the acceptance probability in lines 5-6.

1: Input: m, λ
2: $h(k) \leftarrow \mathcal{U}(1, Z)$ ▷ uniformly select uncut segment $h(k)$
3: $\lambda^{'} \leftarrow \lambda - E$ with $E \sim \exp(|\Theta_1^{h(k)}| + |\Theta_2^{h(k)}|)$
4: **if** $\lambda^{'} \geq 0$ **then** ▷ if budget sufficient draw direction $d \in \{1, 2\}$,
5: ▷ where $i = 1$ is vertical and $i = 2$ is horizontal
6: $i \sim \mathcal{B}(|\Theta_1^{h(k)}|/(|\Theta_1^{h(k)}| + |\Theta_2^{h(k)}|))$
7: $\chi | d \sim \mathcal{U}(\Theta_i^{h(k)})$ ▷ draw cut position χ
8: $\alpha \leftarrow min\{1, r\}$ ▷ acceptance probability, equation 5.13
9: **if** $\alpha > u \sim \mathcal{U}(0, 1)$ **then** ▷ accept with sub-trees $m_< \; m_>$
10: $m^{h(k)} \leftarrow \langle i, \chi, \lambda^{'}, m_<, m_> \rangle$
11: **end if**
12: **end if**

5.2.5.1 Prior probabilities

The priors for the parameter $\mathbf{w}_{n,h(k)}$, regulator set $\boldsymbol{\pi}_n$, variance σ_n^2, and signal-to-noise hyper-parameter δ_n^2 are defined in the same way as for BRAM in Section 5.2.4.1. However, the notation of the segment changes from h to $h(k)$ so that I can identify the partition of a Mondrian sample m_k with node index k. In addition, the size of matrix $D_{n,h(k)}$ below Equation 5.7 becomes $s_{n,h(k)} = |\widehat{\Theta}_1^{h(k)}||\widehat{\Theta}_2^{h(k)}| \times (p_n + 2)$, where $|\widehat{\Theta}_i^{h(k)}|_{i \in \{1,2\}}$ denotes the size of a Mondrian sample in direction i.

The prior distribution of the Mondrian process depends on the hyper-parameter λ and is defined via the generative process described in Algorithm 1. However, for the RJMCMC scheme described below all that is needed is the prior ratio, which is given by (5.14).

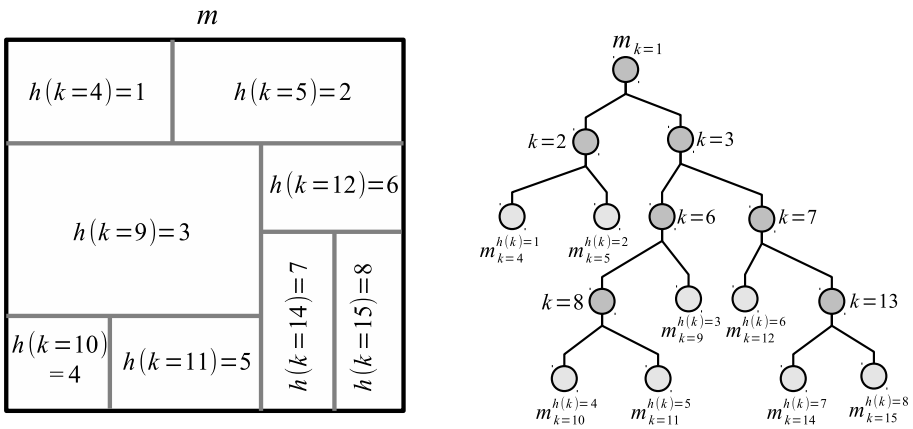


Figure 5.3: Mondrian process example. The left panel shows an example partitioning with a Mondrian process. The right panel displays the associated tree with labels of the latent variable $h(k)$ identifying each non-overlapping segment with leaf nodes (light grey) designated as $m_k^{h(k)}$, where k indexes all tree nodes.

5.2.5.2 Posterior probability

The likelihood follows from Equation (5.1) and closely resembles the previously defined likelihood of BRAM in Equation (5.8):

$$\mathcal{L}(x_n^{h(k)}|\mathcal{G}_n, \mathbf{w}_n^{h(k)}, \sigma_n^{h(k)}) = \left(\sqrt{2\pi}\sigma_n^{h(k)}\right)^{-s_{n,h(k)}} \times \exp\left(-\frac{(x_n^{h(k)} - D_{n,h(k)}(x)\mathbf{w}_n^{h(k)})^\dagger \ (x_n^{h(k)} - D_{n,h(k)}(x)\mathbf{w}_n^{h(k)})}{2(\sigma_n^{h(k)})^2}\right)$$

An attractive feature of the chosen model is that the marginalization over the parameters $\mathbf{w} = \{\mathbf{w}_n^{h(k)}, 1 \leq n \leq N, 1 \leq h(k) \leq H\}$ and $\sigma^2 = \{(\sigma_n^{h(k)})^2, 1 \leq n \leq N, 1 \leq h(k) \leq H\}$ is analytically tractable [89, 7], and I obtain a closed-form expression for the marginal likelihood:

$$\mathcal{L}(x_n^{h(k)}|\mathcal{G}_n, \delta^2) = \int \mathcal{L}(x_n^{h(k)}|\mathcal{G}_n, \mathbf{w}_n^{h(k)}, \sigma_n^{h(k)}) \\ p([\sigma_n^{h(k)}]^2)p(\mathbf{w}_n^{h(k)}|\boldsymbol{\pi}_n, [\sigma_n^{h(k)}]^2, \delta^2)d\mathbf{w}_n^{h(k)}d[\sigma_n^{h(k)}]^2 \tag{5.11}$$

The objective of Bayesian inference is to sample from the posterior distribution given by

$$p(m, \mathcal{G}, \kappa, \delta^2|x) \propto \mathcal{L}(x_n^{h(k)}|\mathcal{G}_n, \delta^2)p(\delta^2)p(E)p(\mathcal{G}|\kappa)p(m|\lambda) \tag{5.12}$$

where all prior distributions have been defined above. To this end, I pursue a Gibbs sampling like strategy, where I iteratively sample new hyper-parameters from $p(\kappa,\delta^2|\mathcal{G}, m, x)$, a new network structure from $p(\mathcal{G}|\kappa,\delta^2, m, x)$, and a new Mondrian process segmentation of the spatial domain from $p(m|\mathcal{G},y, \lambda)$. The first distribution is of standard form due to conjugacy of the prior, and the hyper-parameters can be sampled directly. However, direct sampling from the other two distributions is intractable, and I therefore apply RJMCMC [56]. To sample new network structures $\mathcal{G}$, I follow the scheme described by Lèbre et al. [89], which is based on edge birth and death moves. To sample new Mondrian process partitions, I adopt the method proposed by Wang et al. [147], which I will briefly outline in the next Section. The scheme could be extended to infer λ, but that has not been done yet, and I assume this hyper-parameter to be fixed and hence have not made it explicit on the left-hand side of equation (5.12).

I'm primarily interested in a sample of network structures from the posterior distribution $p(\mathcal{G}|x)$, which I obtain by marginalizing over the hyper-parameters and Mondrian process partitionings. I get the posterior probabilities of all species interactions $p(n \rightarrow n'|x)$ by further marginalization, which define the ranking of interactions in terms of posterior confidence.

5.2.5.3 Inference

As described above, an essential step of the inference procedure is to sample a new Mondrian process segment m from $p(m|\mathcal{G}, y, \lambda)$. The current state of the Mondrian process m is represented by a structure tree as illustrated in Figure 5.3 and a model parameter vector $\boldsymbol{\zeta}$, which contains all previous costs E_k and cut locations χ_k. Note that all budgets and domains can be computed from that recursively. When a cut move is proposed (marked with $+$), the current parameter values are augmented by supplementary random variates u_1 and u_2 in such a way that the dimensions in the higher and lower dimensional parameter spaces are matched. I uniformly sample a spatial segment $h(k)$ draw u_1 and u_2 from the density $q(u_1, u_2)$ and set $\boldsymbol{\zeta} \rightarrow \boldsymbol{\zeta}^+ = \langle \boldsymbol{\zeta}, E^{h(k)} = u_1, \chi^{h(k)} = u_2 \rangle$. If $E^{h(k)}$ does not exceed the budget $\lambda^{h(k)}$ the cut move

proceeds as shown in Algorithm 1, where $\chi^{h(k)}$ defines the position proportional to the sample domain size, which follows a Bernoulli distribution $\mathcal{B}$. The proposed new Mondrian process state m^+ is accepted with probability $\alpha = min\{1, r\}$,

$$r = \frac{P(m^+|\lambda)}{P(m|\lambda)} \times \frac{q(m|m^+)}{q(m^+|m)} \times \frac{\mathcal{L}(x_<^{h(k)}|\mathcal{G}, \delta^2)\mathcal{L}(x_>^{h(k)}|\mathcal{G}, \delta^2)}{\mathcal{L}(x^{h(k)}|\mathcal{G}, \delta^2)} \times J \tag{5.13}$$

$$\frac{q(m|m^+)}{q(m^+|m)} = \frac{Z}{\phi(m^+)q(E^{h(k)}, \chi^{h(k)})}, \tag{5.14}$$

$$\frac{P(m^+|\lambda)}{P(m|\lambda)} = \frac{\omega_<^{h(k)}\omega_>^{h(k)}p(E^{h(k)})p(\chi^{h(k)})}{\omega^{h(k)}} \tag{5.15}$$

Here, the subscripts $>$ and $<$ refer to the two new spatial segments associated with the cut, $x_>^{h(k)}$ and $x_<^{h(k)}$ are the corresponding subsets of $x^{h(k)}$, and $x^{h(k)}$ denotes the species abundance data associated with segment/leaf node $h(k)$. Following the standard RJMCMC scheme [56], the four terms in (5.13) are the prior ratio, inverse proposal ratio, marginal likelihood ratio and Jacobian. The latter is one, $J = 1$, the marginal likelihood is given by (5.11), and the prior and proposal ratios are given by (5.14), where ϕ denotes the number of Mondrian leaf siblings, i.e. adjacent segments that can be merged in order to restore m, and $\omega^{h(k)} = \int_\lambda^\infty \tau^{h(k)} \exp(-\tau^{h(k)}e)de = \exp(-\tau^{h(k)}\lambda^{h(k)})$ denotes the probability of no further cut. By setting $q(E_k, \chi_k) = p(E_k)p(\chi_k)$, the expression naturally simplifies. The state m is replaced by the proposal m^+ in the case the move is accepted. The probability of removing a cut is given by the inverse of (5.13). A shift move replaces the direction i and position χ of a cut, which separates the adjacent segments $h(k_1)$ and $h(k_2)$ yielding the proposal segments $h(k_1)^+$ and $h(k_2)^+$. The acceptance probability is $\alpha = min\{1, \mathcal{L}(x^{h(k_1)^+})\mathcal{L}(x^{h(k_2)^+})/\mathcal{L}(x^{h(k_1)})\mathcal{L}(x^{h(k_2)})\}$ after cancelling the proposal and prior ratios because budget, cost and number of Mondrian samples remain invariant. Whenever a segment is cut or merged, the affected regression coefficients are sampled from the posterior.

5.3 Data

5.3.1 Synthetic Data

For an objective evaluation of network inference, I test the ability of the previously described methods to recover the true network structure from synthetic test data generated from a piece-wise linear regression model following equation (5.1). I generated

two types of data sets: One that uses a global change-point process and thus follows the model assumption of BRAM. In this data set I inserted regular change-points at location 5 and 10 along the horizontal and vertical axis.The second set resembles a Mondrian process type partitioning and was used in [3] to evaluate the performance of BRAMP. In this set I iteratively subdivide the data grid into local segments, e.g. as shown in the left panel of Figure 5.3. I refer to these data set as *Synth-BRAM* for the former and *Synth-BRAMP* for the latter set.

The number of observations along each axis was selected to be $M^1 = M^2 = 15$ for both data sets. The number of nodes n was set to 10 and the number of regulators for each node was sampled from a Poisson distribution. The regression coefficients $\mathbf{w}_{n,h}$ together with the intercept $w^0_{n,h}$ of each segment h or $h(k)$ were sampled from a uniform distribution in the interval of $[-1; -0.5]$ and $[0.5, 1.0]$. The noise ε_n was sampled from a normal distribution. Nodes without an incoming edge were initialised to a Gaussian random number. The values of the remaining nodes were calculated at each grid cell following equation (5.1).

5.3.2 Simulated Population Dynamics

For a realistic evaluation, I followed [40] and generated data from an ecological simulation that combines a niche model [154] with a stochastic population model [87] in a 2-dimensional lattice.

5.3.2.1 Niche model and species interactions

The niche model defines the structure of the trophic network and has two parameters: the number of species N and the connectivity (or network density) defined as L/N^2 where L is the number of interactions (edges) in the network. Each species n is assigned a niche value x_n, drawn uniformly from $[0, 1]$. This gives an ordering of the species, where higher values mean that species are higher up in the food chain. For each species a niche range R_n is drawn from a beta distribution with expected value $2C$ (where C is the desired connectivity), and species n consumes all species falling in a range R_n that is placed by uniformly drawing the centre c_n of the range from $[R_n/2, x_n]$ as illustrated in Figure 5.5 and introduced in [154]. Despite its simplicity, it was shown there that the resulting networks share many characteristics with real food webs.

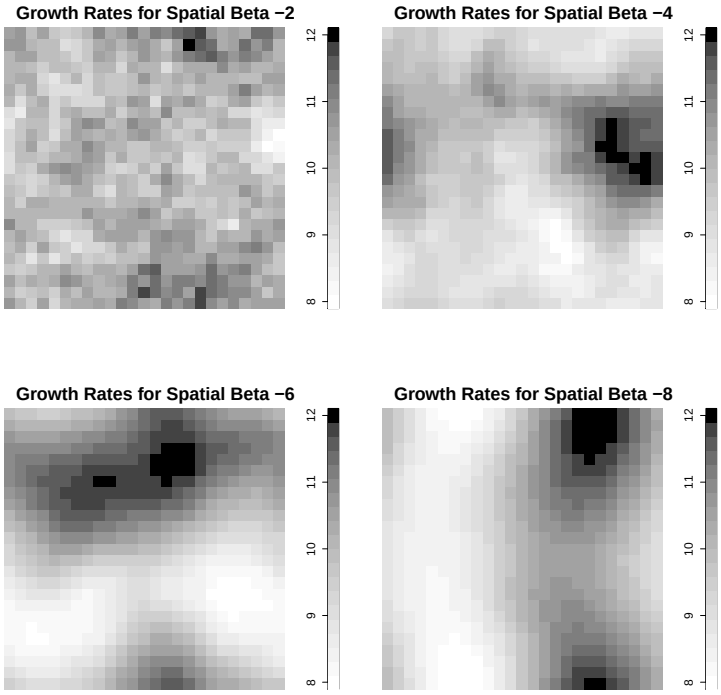


Figure 5.4: Spatial distribution. Shown are the spatial distributions of growth rates r_n entering equation (5.16) as the spatial β parameter (Section 5.2.2) decreases from -2 to -8. A value of 0 corresponds to uniformly random noise, and -2 is Brownian noise.

5.3.2.2 Stochastic population dynamics

The population model is defined by a stochastic differential equation where the dynamics of the log abundance $X_n(t)$ of species n at time t can be expressed as:

$$\frac{dX_n(t)}{dt} = r_n + \frac{\sigma_d}{\sqrt{e^{X_n(t)}}}\frac{dA_n(t)}{dt} + \sigma_e\frac{dB_n(t)}{dt} - \gamma X_n(t) - \Omega(X) + \sigma_E\frac{dE(t)}{dt} \tag{5.16}$$

where X is the set of all $X_N(t)$, r_n is the growth rate of species n, σ_d is the standard deviation of the demographic effect, $A_n(t)$ is the species-specific demographic effect, σ_e is the standard deviation of the species-specific environmental effect, $B_n(t)$ is the species-specific environmental effect, 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 environment. The growth rates r_n are location dependent (depending on the cell of a rectangular grid), with a spatial pattern that is generated by noise with spectral density f^β (with $\beta < 0$, and f

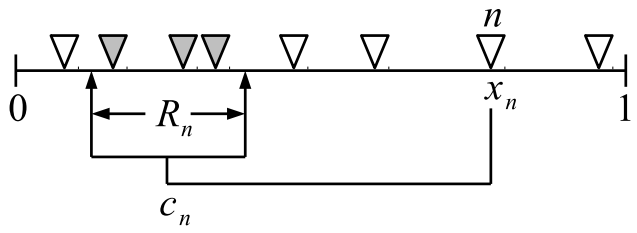


Figure 5.5: Diagram of the niche model. Species are indicated with a triangle. A species n is placed with a niche value x_n into the interval $[0, 1]$. A value c_n is uniformly drawn that defines the centre of the range R_n. All species with a value x inside this interval, i.e. $c_n - \frac{R_n}{2} \leq x \leq c_n + \frac{R_n}{2}$, as indicated by grey triangles, are consumed ('eaten') by species n. Diagram adapted from Williams and Martinez [154].

denoting the spatial frequency at which the noise is measured). An illustration is given in Figure 5.4. To model species migration, I included an exponential dispersal model, where the probability of a species moving from one location to another is determined by the Euclidean distance between the locations.

5.3.2.3 Interactions and Simulation

To incorporate the niche model, I modified the term Ω in (5.16) to include predator-prey interactions in the Lotka-Volterra form. I explored two versions: one where predatory interactions had a relatively strong negative effect on prey (strong predation) and one where the impact of predation was less severe (weak predation). Strong predation is more akin to traditional predator-eat-prey interactions, whereas weak predation is more akin to partially destructive predation (e.g., grazing) or aggression.

I applied this model to 10 species living in a 25-by-25 rectangular grid. I simulated the dynamics of this model for 3000 steps and then recorded species abundance levels in all grid cells at the final step; this corresponds to an ecological survey carried out at a fixed moment in time. For each grid cell I counted the number of species that went extinct. These counts were added up over all cells, yielding a total number of extinctions. A simulation was rejected if these extinctions exceeded the value 50. For each of the spatial β parameters displayed in Figure 5.14, i.e. $\beta = (-2, -4, -6, -8)$, 30 surveys were collected by running the simulation repeatedly with different networks and parameter initialisations.

5.3.3 Real World Plant Data

I have applied BRAM and BRAMP real-world data from Lennon et al. [91], including 106 vascular plants and 12 environmental variables (abiotic factors) collected from a 200m x 2162m Machair vegetation land stripe at the western shore of the Outer Hebrides. Samples were taken at 217 locations, each 1m x 1m in size, equally distributed with a 50m spacing. Plant samples were measured as ground coverage in percentage and physical samples as absolute values (such as moisture, pH value, organic matter and slope). The data was log-normal transformed after observing substantial skewness in the distributions. Each sample point was mapped into a 2D grid ignoring locations with no sample data available. The spatial autocorrelation value for each plant and location was calculated from neighbours inside a radius of 70m. Since I'm interested only in plant interactions, I defined each plant to have all 12 physical soil variables as fixed input, i.e., permanent predictor variables.

5.4 Comparative Evaluation

To evaluate the network reconstruction accuracy for the simulated data, where the true network structure is known, I calculate AUROC and AUPREC values as described in Section 3.4.1. I start with the evaluation of BRAM on the synthetic data *Synth-BRAM* in Section 5.5.2.1, followed by BRAMP on the synthetic data *Synth-BRAMP* in Section 5.5.2.2. Both methods are run for 20000 iterations, a burn-in of 15000 iterations and a thin-out of 10 iterations, i.e. a sample of the edge indicator is taken every 10 iterations. The iteration size for the simulated population data (Section 5.5.3) is increased to 50000 and for the real world data (Section 5.5.4) to 100000 iterations with the first 3/4 of the chain to be considered the burn-in phase. For BRAMP, I followed [147] and set the hyper-parameter of the Mondrian process to the fixed value $\lambda = 1$ for all my simulations. I included a comparison with ℓ_1-regularized linear regression (Lasso, Section 2.4), using the optimization algorithm proposed by Grandvalet [55] and implemented in the R package *glmnet*. Besides being widely applied in molecular systems biology [142], Lasso has been recommended to be used more widely in ecology [31], and was found to outperform all competing methods by Faisal et al. [40]. To construct the design matrix, I mapped the sample space from two-dimensions to 1 dimension by simply reading the data along the x-axis and then down along the y-axis, i.e. line by line and without considering multiple change-points. The regularization parameter λ that controls the network sparsity was inferred with 10-fold cross-validation using the

function *cv.glmnet*(), which led to better results than optimizing the BIC score. The resulting optimized λ parameter was applied for the final regression. This yielded regression coefficients that can be interpreted as edge weights that indicate the strength and sign of interactions among species. For obtaining the ROC and precision-recall curves, I ranked the potential interactions based on the absolute values of the non-zero interaction parameters. I further included a comparison with a non-linear Bayesian network, as implemented in the software package BANJO[4], described in Section 2.14. I discretised the data with Hartemink's pairwise mutual information method described by Hartemink [66] (implemented in *R* package *bnlearn*), because this method yielded a better performance than quantile discretisation. The number of discretisation levels was chosen to be 3 based on empirical tests carried out by Yu et al. [155]. Search was done using simulated annealing with random walk proposals. Simulated annealing was run on each dataset until convergence (typically 7 hours of CPU time). Using the top 100 high-scoring (BDe score) networks I computed edge probabilities for ranking.

Finally, I applied BRAM and BRAMP to the real world data, revealing putative plant interactions.

5.5 Results and Discussion

5.5.1 MCMC convergence

The previously defined MCMC iteration lengths are dependent on the following convergence study that have been performed for each of the data types. For BRAM and BRAMP I have to identify if the methods are statistically sound and convergence occurs at all and at what point convergence is typically established given a certain data set. Naturally, larger data sets that include more observations (a greater area or finer sample resolution) and larger number of covariates (more species) have a longer burn-in phase. I use the methods described in Section 3.4 to detect the length of the necessary burn-in phase by running multiple chains for the same data and monitor the edge indication vectors. The edge indicators allow the calculation of edge posterior probabilities given several samples of a MCMC chain. They can be displayed in a scatter plot for two independent chains as illustrated in the left panel of Figure 5.6. The similarity of the probability values for both chains that is expressed as dots positioned close to the diagonal line indicates good agreement of the chains and possible convergence of the

[4]The dynamic time-varying nature of the Banjo was adopted to the static case thus I effectively only exploit the BDe scoring scheme.

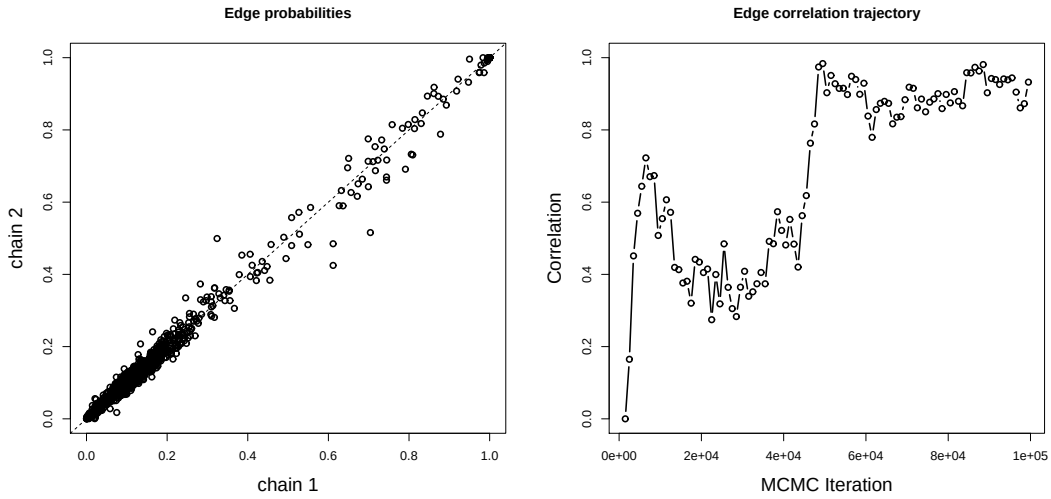


Figure 5.6: Interaction posterior probabilities of two MCMC runs. The left scatter plot shows a snapshot of the edge probability samples from two independent MCMC chains. The right plot displays the trajectory of correlation values given the edge probability samples from two chains. Both plots can be used to determine the length of the burn-in phase of a MCMC chain.

MCMC at the point of sampling.

However, this only constitutes a snap shot of two chains and an overview of associated correlation values over the course of an MCMC simulation is more convenient to detect changes in convergence. A trajectory of edge correlation values over the course of the MCMC simulations is displayed in the right panel of Figure 5.6. The plot shows a stabilization of correlation values > 0.75 after about 50000 iterations so the burn-in phase can be considered to end approximately at this point. Note that these trajectories are not always as coherent as displayed in Figure 5.6. In the cases when the predictor variables have a poor descriptive power it is usually difficult to observe a clear pattern in the trajectory. In addition, a complex posterior landscape that has multiple local optimal solutions (posterior distribution) can lead to temporal shifts in the predictive pattern. In this case sampling will not only vary between different MCMC chains but also show variations in the same MCMC chain whenever the chain jumps from one local to another local solution state.

This can better be observed with the trajectories of the Potential Scale Reduction Factor (PSRF), which has been described in Section 3.4.2. The PSRF measures the variations in and between MCMC chains with low values indicating a relative stabi-

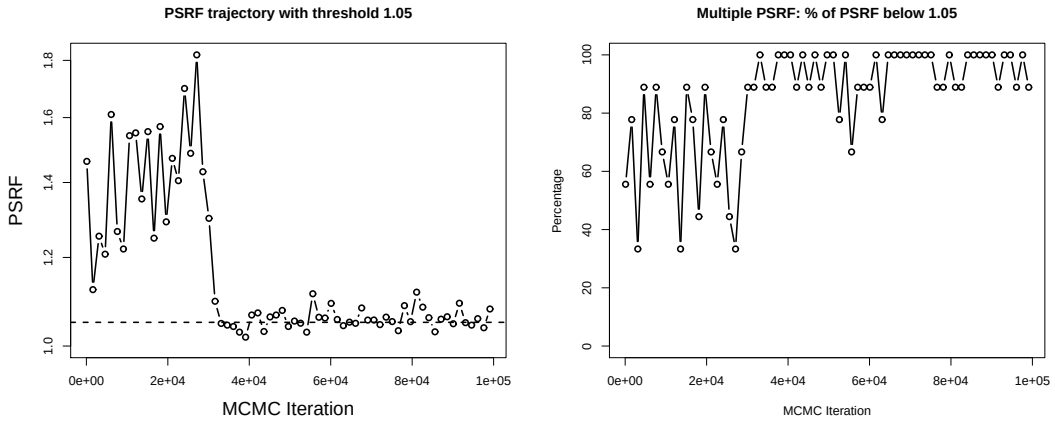


Figure 5.7: Potential scale reduction factor (PSRF) of MCMC chains. The left plot shows a sample PSRF trajectory involving 3 independent MCMC chains for the same data. The PSRF drops below a threshold value of 1.05 after about 40000 iteration, which indicates convergence of the chains at this point. The right plot summarizes multiple PSRF values derived from several scalar estimates of the same chains. Overall convergence is indicated by higher percentage values. The plots are samples from the ecological application in Section 5.5.3.

lization of the MCMC down to a critical value of 1.05, which is considered a good indicator for chain convergence. The left panel in Figure 5.7 shows an example trajectory of PSRF values over time. A disadvantage of the PSRF is that its calculation is based on a single scalar value. However, in my case there are several edge indicator scalars in a single chain and thus I have to consider the maximum, mean, or median PSRF value of a chain. The median value seems to best reflect the overall trend of convergence because complete un-converged outliers often distort mean and maximum values. The alternative is to summarize all PSRF values for multiple scalars by calculating the percentage of PSRF values below a critical threshold, e.g. 1.05, as displayed in the right panel of the same Figure.

5.5.2 Synthetic Data

The results in Section 5.5.2.1 are derived from the earlier publication in [2] and only include BRAM, HBR, and Banjo. The aim was to quantify in which extend BRAM is able to outperform the homogeneous Bayesian regression method (HBR) that lacks a change-point process. The subsequent publication of BRAMP in [3] led to the modifica-

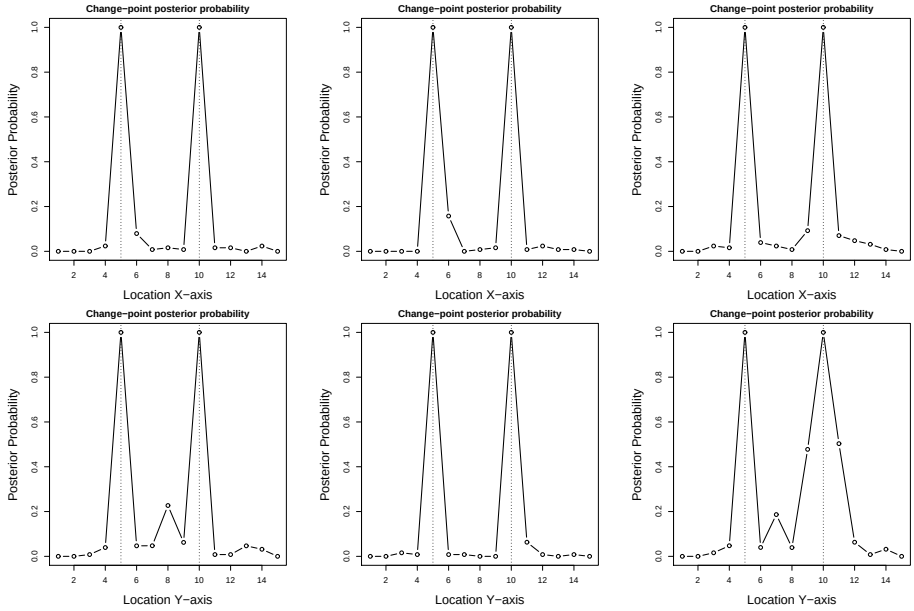


Figure 5.8: Change-point posterior probabilities inferred with BRAM on three data sets of the synthetic data *Synth-*. The upper row shows the change-points in the vertical direction (x-axis) and the lower row in the horizontal direction (y-axis). Each column represents one data set. The number of locations is 15 in each direction. The dashed lines mark the real change-points (at 5 and 10 on both directions) with peaks at these locations indicating correctly inferred change-points.

tion of the synthetic data to match the model assumption of BRAMP but also includes BRAM (Section 5.5.2.2). To provide a fair comparison between the methods spatial autocorrelation variables are disabled for all methods on the synthetic data because no dispersion effect was simulated with this data model (Section 5.3.1).

5.5.2.1 Global change-points (*Synth-BRAM*)

The main purpose of the *Synth-BRAM* data set was to test the ability of BRAM to recover certain predefined change-points that have been inserted on a 15×15 location grid at the locations 5 and 10 in each direction. The data was generated as described in Section 5.3.1 and BRAM applied to several independent instantiations of this data. The inferred change-point vectors $\boldsymbol{\tau}_1$ (x-axis) and $\boldsymbol{\tau}_2$ (y-axis) were recorded after the established burn-in phase with a thin-out of 10 iterations. Given these samples the overall posterior probability of change-point occurrence was calculated for each location in both directions. The resulting probabilities for three sample data sets are displayed in

Figure 5.8. The x-axis of each plot correspond to the 15 locations in each direction and the y-axis to the change-point posterior probability. Dashed lines at 5 and 10 indicate the true change-points of the simulated data and lined dots indicate the change-point posterior probabilities. It can be observed that the inferred change-points recover the true change-points with high confidence of close to 1.0 and thus proving that the change-point process of BRAM works as expected.

Following up the successful recovery of the true change-points I applied the HBR, Lasso, and Banjo to the same data set for a comparison of the learning accuracy of the true network. The corresponding AUROC and AUPREC scores for this evaluation are presented in Figure 5.9. BRAM produces the highest scores and thus outperforms all competing schemes. This is not surprising, in that the data have been generated from a process that is consistent with the modelling assumptions of BRAM. However, it is reassuring both that the MCMC inference scheme can successfully deal with the increased model complexity, and that it leads to an improvement over the competing models in terms of actual network reconstruction accuracy. The large difference of scores obtained with BRAM and HBR also underlines the importance of the change-point process.

5.5.2.2 Mondrian change-points (*Synth-BRAMP*)

This data set was used to study the performance of the Bayesian regression model with Mondrian process change-points (BRAMP). The data in the spatial domain was iteratively subdivided into smaller segments as to simulate a Mondrian process and to present a more realistic segmentation scheme. The first evaluation concerned the ability of BRAMP to learn the true reference segmentation as I have previously demonstrated with BRAM and the global change-points. I generated data following the protocol for the *Synth-BRAMP* data (see Section 5.3.1) that creates Mondrian type segments as shown in the upper panels of Figure 5.10. The lower panel shows the corresponding posterior probabilities of the segment boundaries that was recovered with BRAMP. For each location (x_1, x_2) in the grid I recorded the sampled Mondrian sample and its segment identifier $h(k)$ after the burn-in phase of the MCMC. I further observed the transitions from one segment to another by locating adjacent locations belonging to different spatial neighbourhoods (segments)[5]. This yielded a sufficient amount of boundary samples from which I calculated the posterior probability of segment bound-

[5] Although, this approach seems a little cumbersome it is easier than retrieving the segment boundaries from the Mondrian process tree, which requires to traverse the tree from the root

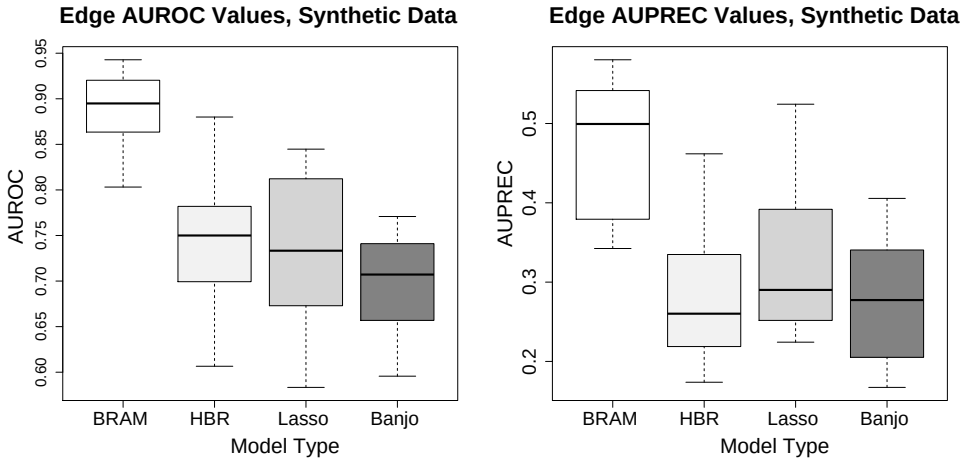


Figure 5.9: Comparison on synthetic data *Synth-BRAM*. Boxplots of AUROC (left panel) and AUPREC (right panel) scores obtained with four methods on the globally segmented synthetic data described in Section 5.3.1: a Bayesian regression model with global change-points (BRAM), a Bayesian linear regression model without change-points (HBR), sparse ℓ_1-regularized linear regression (Lasso), and a homogeneous Bayesian network with the BDe score (BANJO). The boxplots show the distributions of the scores for 30 independent data sets, where the horizontal bar shows the median, the box margins show the 25th and 75th percentiles and the whiskers indicate data within 2 times the inter-quartile range.

aries as displayed in different shades of grey in the lower panels of Figure 5.10. The left panels of Figure 5.10 show a simple segmentation with only two subsequent cuts. Both cuts are learned with high confidence by BRAMP indicated by the black colour (probability of 1.0). However, a false boundary is also learned with medium confidence at location 7 (x-axis) extending from location 11 to 15 on the y-axis. Similar artefacts can be observed in the lower right panel of Figure 5.10. Although, most of the true boundaries are learned, there is a false positive boundary at location 8 (x-axis), stretching from location 1 - 4 (y-axis) (boundary $\boldsymbol{A}$). The true vertical boundary to the right at location 12 (x-axis) is in contrast learned with lower probability (boundary $\boldsymbol{B}$).

These artefacts point to an intrinsic short-coming of the Mondrian process. Since the process is organized in a hierarchical fashion represented in a tree (see the Mondrian process tree in Figure 5.3), it is not possible to revert or manipulate segment cuts that have been applied to inner nodes, i.e. previously cut segments, of the tree. It is easy for the process to accept a segment cut in the early phase of segmentation that provides a rough approximation of the true main boundaries, such as at location 8 along the full vertical axis (boundary $\boldsymbol{C}$ on the lower right panel). However, this leads to the

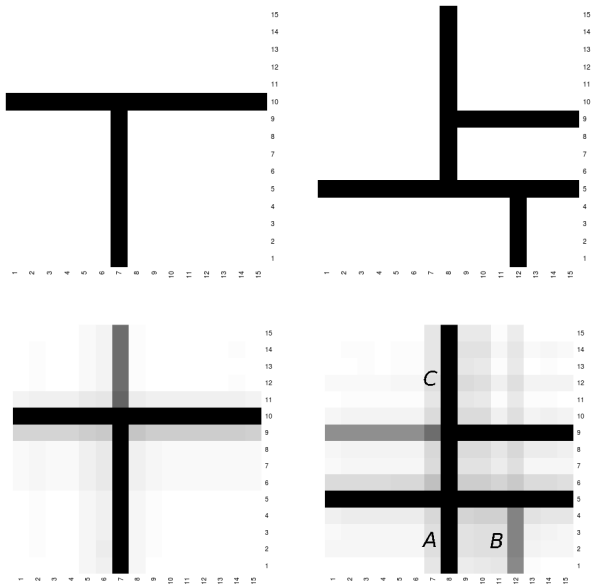


Figure 5.10: Posterior probabilities of the segment boundaries learned with BRAMP applied to data *Synth-BRAMP*. The top row displays two predefined segmentations of the two-dimensional grid assembling both a Mondrian process. The lower panels show the corresponding inferred posterior probabilities of the segment boundaries using the Bayesian regression method with Mondrian process segmentation (BRAMP). The different shades of grey illustrate the probability of a boundary with black values indicating a probability of 1.0. Most segments can be learned with high confidence, although there are spurious boundaries.

formation of solution subsets of possible segmentations with the property that i) the boundary $\boldsymbol{A}$ can not be easily removed, and ii) the true boundary $\boldsymbol{B}$ becomes difficult to recover. To accomplish i) all previous cuts that depend on boundary $\boldsymbol{C}$ have to be reversed until $\boldsymbol{C}$ can be removed again, which is unlikely to happen. For the case ii), the number of locations (samples) involved in the likelihood calculation is much smaller because the boundary $\boldsymbol{A}$ limits the space of the segment. As a consequence, the small sample size leads to a high degree of uncertainty in the likelihood such that the boundary $\boldsymbol{B}$ will often be rejected and not properly learned. This lack of flexibility could be addressed in future work by introducing non-hierarchical approaches to the segmentation.

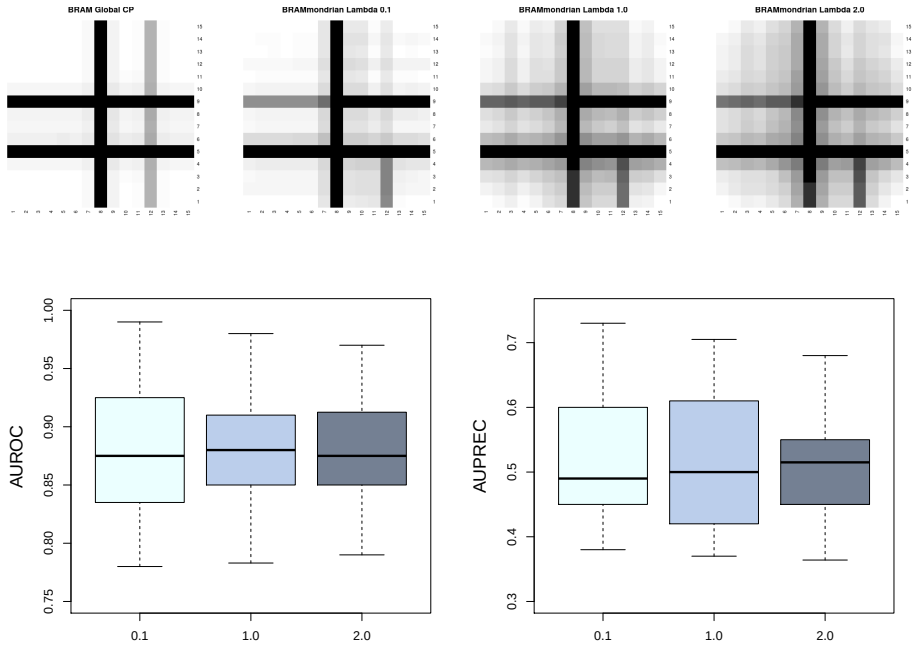


Figure 5.11: Influence of the Mondrian process start budget hyper-parameter λ on segment inference with BRAMP: Top panels show (from left to right) a Bayesian regression method with global change-points (BRAM, for reference), a Bayesian regression with Mondrian process change-points (BRAMP) with a start budget of $\lambda = 0.1$, BRAMP with $\lambda = 1$, and BRAMP with $\lambda = 2$. The true boundaries are displayed in the top right panel of Figure 5.10. The shades of grey indicate boundary posterior probabilities. The bottom row displays the corresponding AUROC and AUPREC scores obtained for different setting of λ.

Dependence on the start budget λ Another important aspect of BRAMP is the necessity to predefine a hyper-parameter λ that controls the strength, i.e. the depth of segmentation. This parameter is called the start budget of the initial un-cut Mondrian sample (space) and is inherited in a decremental fashion whenever a split occurs. The start budget controls the acceptance of a segment split if the cost of the split does not exceed the budget of the segment. I did not attempt to learn this parameter because it would implicate a change of the whole Mondrian tree structure whenever the start budget is altered. The effect of different settings of $\lambda = (0.1, 1, 2)$ is shown for the synthetic data *Synth-BRAMP* in Figure 5.11. The top panels show that an increase of λ intensifies the acceptance of segment cuts, which is indicated with an increase

of grey shaded false boundaries. Corresponding AUROC and AUPREC scores are displayed in the lower panels of Figure 5.11. The scores are quite similar to each other despite the alteration of the initial λ setting. In fact, it is somehow surprising since one would expect a decrease of inference performance given an increased number of spurious segments that arise from a higher setting of, e.g., $\lambda = 2.0$ as can be seen on the top right plot of Figure 5.11. It is possible, however, that the small size of the grid with only 15×15 locations remains largely unaffected by these changes as long as λ is large enough so that inference of the major boundaries is possible. The proper setting of λ is likely to effect large grids with a very fine segmentation. In these cases λ can be considered a prior knowledge about the coarseness of segmentation and should be set accordingly. Based on the above observations and the standard setting in [147], I decided to use $\lambda = 1$ for all simulations.

Finally, the results of the comparison to the competing methods BRAM, HBR, Lasso and Banjo is presented in Figure 5.12. Again, this is of little surprise since the data *Synth-BRAMP* have been generated from a process that is consistent with the modelling assumption of BRAMP. Notable is the improvement of BRAMP over BRAM, which indicates that BRAMP is more flexible in terms of identifying local segments. It is reassuring that both non-homogeneous Bayesian regression schemes (BRAMP and BRAM) can handle the increased model complexity, and improve network reconstruction accuracy compared to the competing methods HBR, Lasso, and Banjo.

5.5.3 Simulated Population Dynamics

5.5.3.1 Effect of spatial autocorrelation

The simulated population dynamics data includes dispersion effects that suggests the use of the spatial autocorrelation variable that was introduced in Equation 5.2. I study the effect of including this variable on the accuracy of network inference of BRAM and HBR by comparison to the same methods without the variable. Figure 5.13 demonstrates the improvement of AUROC and AUPREC scores for different setting of the spatial β parameter (Figure 5.4) defined in Section 5.3.2. The spatial β controls the heterogeneity of species distribution and thus affects the degrees of dispersion. The displayed values are pairwise differences of BRAM and HBR without spatial autocorrelation variable minus BRAM and HBR with the variable. Hence, the negative values indicate an improvement for both of the setting of β.

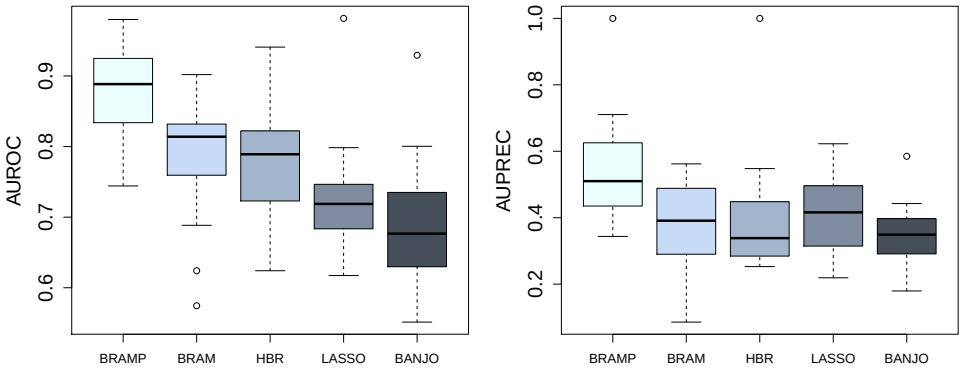


Figure 5.12: Comparison on Synthetic Data *Synth-BRAMP*. Boxplots of AUROC (left panel) and AUPREC (right panel) scores obtained with three methods on the synthetic data described in Section 5.3.1: the Bayesian regression model with Mondrian process change-points (BRAMP), the Bayesian regression model with global change-points (BRAM), a Bayesian linear regression model without change-points (HBR), ℓ_1-penalized sparse regression (Lasso), and a homogeneous Bayesian network with the BDe score (Banjo). Each boxplot shows the distribution of scores of 30 independent data sets.

5.5.3.2 Comparison to HBR, Lasso, and Banjo

I compared the performance of BRAMP, BRAM, HBR, Lasso, and Banjo in terms of network recovery. For a fair comparison I added the spatial autocorrelation variables to all these methods. Recalling that lower values of β lead to the formation of clusters or "neighbourhoods" of similar species concentrations, it is more challenging to uncover underlying networks with a homogeneous method such as HBR, Lasso, or Banjo. However, for BRAM and BRAMP I would expect a better performance compared to the other methods as can be observed in Figure 5.14 and 5.15 where all AUROC and AUPREC scores are shown for varying levels of β and for the weak and strong predation data types. BRAMP (white box) has the tendency to perform better or at least as good as the other methods, although this observation is much stronger pronounced in the data set with weak predation (Figure 5.15) than in strong predation (Figure 5.14). BRAM remains in close competition to BRAMP with slightly lower scores in the strong predation data but significant lower scores in the weak predation data. The homogeneous Bayesian regression method HBR does not show as much difference to BRAM for strong predation as for weak predation. There is also clear indication that Banjo

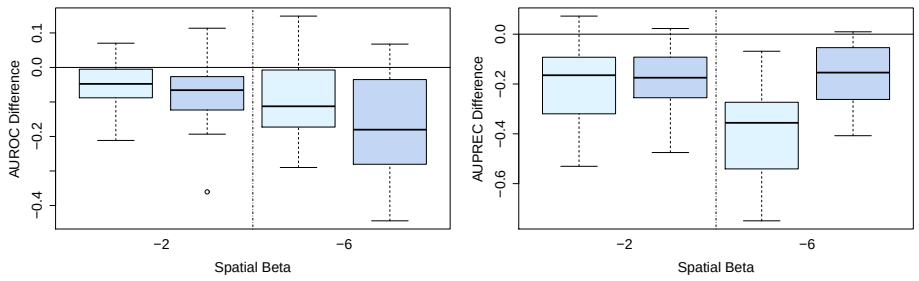


Figure 5.13: Pairwise difference plot demonstrating the effect of the spatial autocorrelation variable. The left panel displays the AUROC differences and the right panel the AUPREC differences for trophic simulated data (Section 5.3.2) with two settings of spatial $\beta = (-2, -6)$. The grey boxes correspond to BRAM and dark grey boxes to HBR. Shown are the pairwise differences of BRAM and HBR without spatial autocorrelation minus the same methods but with spatial autocorrelation variables. Negative values indicate better performance for the methods using spatial autocorrelation.

(darkest grey box) is outperformed by all other methods (Figure 5.14 and 5.15), which underlines the detrimental effect of the information loss inherent in data discretisation.

The comparison with HBR and Lasso leads to results that, on the face of it, appear less conclusive. On the weak predation data BRAMP and BRAM tend to outperform both HBR and Lasso (Figure 5.15), while the latter methods are on a par with BRAM on the strong predation data (Figure 5.14). Lasso showed, on average, the same performance as the HBR method. For weak predation, the abundance profiles showed much stronger spatial oscillations than for strong predation, or conversely: for strong predation, these abundance profiles were much flatter than for weak predation. This suggests that weak predation leads to much stronger spatial heterogeneity than strong predation. If there is little spatial heterogeneity, then there is not much benefit in using a change-point model. Hence, for strong predation with little spatial heterogeneity, BRAM does not outperform HBR, and consequently it also does not outperform Lasso. This assessment was originally published in [2] by the time when BRAMP was not available. However, the scores for BRAMP in Figure 5.14 show a slight improvement over the other methods, suggesting that the Mondrian process change-points of BRAMP are more sensitive in detecting variations in the strong predation data.

This raises the question of why strong predation leads to less spatial heterogeneity in the first place. Spatial heterogeneity implies that in some regions prey are more affected

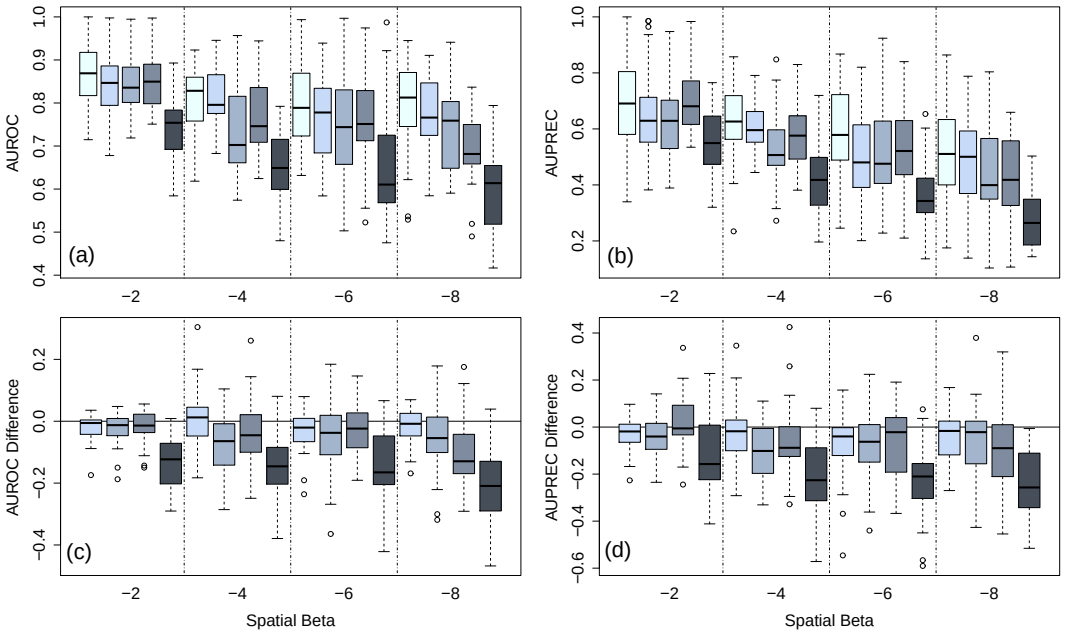


Figure 5.14: Comparative evaluation of five network reconstruction methods for the stochastic population dynamics data with strong predation. AUROC and AUPREC scores obtained on the trophic simulated data described in Section 5.3.2 for different settings of the spatial β parameter. *Top row:* absolute scores. *Bottom row:* difference scores, with Mondrian process method BRAMP taken as reference (target method score minus BRAMP score), i.e. negative values indicate better performance of BRAMP. The abscissa represents different values of the spatial β parameter, whose influence is illustrated in Figure 5.4. Panels: (a) Absolute AUROC values for BRAMP (white), BRAM (light grey), HBR (grey), Lasso (dark grey), Banjo (darkest grey); (b) Absolute AUPREC values; (c) Pairwise difference of AUROC and (d) AUPREC.

by predators than in others. For strong predation these fluctuations are stronger than for weak predation, in fact so strong that some prey are driven to extinction. However, the way I set up the simulations is such that populations with an extinction rate above a threshold are rejected. This is motivated by the limited size of the spatial area in the simulated ecological landscape. This limited size 'traps' prey in an unnatural way; high extinction rates are rejected as being ecologically unrealistic. Populations with the highest spatial heterogeneity are the ones most affected by extinction, thus the rejection mechanism favours more homogeneous populations when predation is strong,

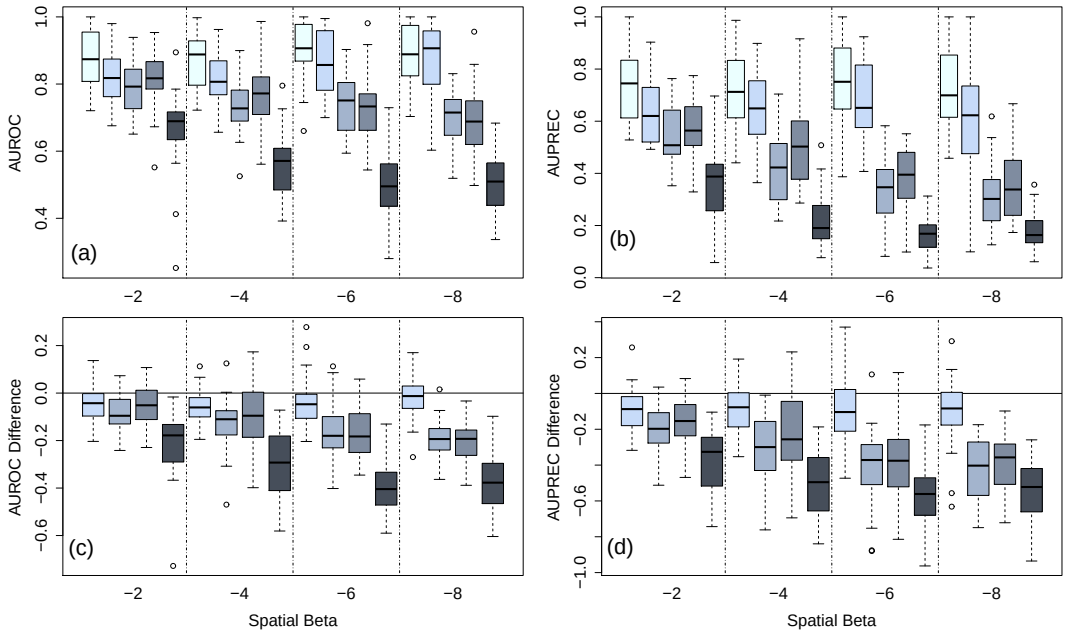


Figure 5.15: Comparative evaluation of five network reconstruction methods, weak predation. AUROC (left column) and AUPREC (right column) obtained on the trophic simulated data described in Section 5.3.2. The simulations were carried out as for Figure 5.14, but with weakened influence of the predators on the prey. See caption of Figure 5.14 for details. Panels: (a) Absolute AUROC values for BRAMP (white), BRAM (light grey), HBR (grey), Lasso (dark grey), Banjo (darkest grey); (b) Absolute AUPREC values; (c) Pairwise difference of AUROC and (d) AUPREC.

which I confirmed empirically by inspection of the spatial abundance profiles.

My simulation studies thus suggest that in the presents of very low spatial heterogeneity, when there is little room for improvement, BRAM shows the same performance as Lasso (Figure 5.15) and BRAMP slightly increases this performance. This is reassuring, given that Lasso was found to outperform all competing models in [40]. When there is genuine spatial heterogeneity, BRAMP and BRAMP outperform Lasso and all homogeneous models without change-points (Figure 5.14).

In summary, BRAMP consistently outperforms the other methods, as displayed in Figure 5.14 with the single exception of BRAM and a spatial $\beta = -8$. Table 5.3 summarises the corresponding p-values of paired Wilcoxon tests for the AUROC scores comparing BRAMP against BRAM, HBR, and Lasso. The low p-values indicate a

Table 5.3: Improvement of the Bayesian regression model with Mondrian process change-points (BRAMP) on the stochastic population dynamics data. P-values for paired one-sided Wilcoxon tests for the difference of AUROC scores between BRAMP and the competing methods (BRAM, HBR, Lasso) for several spatial β values. The alternative hypothesis states that BRAMP scores are greater than the competing methods with low p-values < 0.05 indicating significant performance gain of BRAMP.

Spatial β:	-2	-4	-6	-8
BRAM	2.2e-04	1.9e-04	6.4e-03	0.14
HBR	1.2e-06	2.9e-07	1.0e-07	1.9e-09
Lasso	6.1e-04	7.2e-04	1.3e-08	9.3e-10

significant performance gain of BRAMP suggesting that the Mondrian process better captures the spatial heterogeneity of the population dynamics. In fact, both non-homogeneous Bayesian regression models, BRAMP and BRAM, achieve high AUROC scores for the data simulated with low spatial β values and weak predation, i.e. high data heterogeneity. The performance of HBR, Lasso, and Banjo deteriorates as expected with an increase of data heterogeneity (i.e., lower spatial β).

5.5.4 Real World Plant Data

I have applied BRAMP[6] to the plant abundance data from the ecological survey described in Section 5.3.3. I sampled interaction network structures from the posterior distribution with MCMC and computed the marginal posterior probabilities of the individual potential species interactions. I kept all species interactions with a marginal posterior probability above 0.1, resulting in 39 out of 106 species with relevant interactions in the reconstructed network shown in Figure 5.16. Interactions with a probability higher than 0.5 are displayed in thick lines. Negative interactions are displayed as dashed lines and positive interactions as full lines. They were derived as mean edge weights over all segments and multiple samples from the MCMC chain.

Since I had defined the 12 soil attributes as fixed predictors to each plant, the interactions in this network represent plant-plant interactions not mediated by similar soil preferences. This network can lead to the formation of new ecological hypotheses. For

[6]Only the network inferred with BRAMP is shown here. However, BRAM is used for the study on clustering of the spatial segmentation as presented in Figure 5.18 and 5.17 and in [2]. I did not repeat this particular study for BRAMP

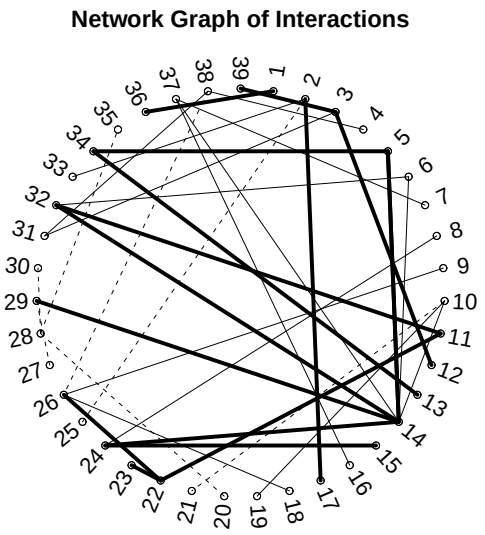


Figure 5.16: Species interaction network. Species interactions as inferred with BRAMP (Section 5.2.5), with an inferred marginal posterior probability of 0.5 (thick lines) and 0.1 (thin lines). Solid lines are positive (e.g. mutualism, facilitation) and dashed are negative interactions (e.g. resource competition). Species are represented by numbers and have been ordered phylogenetically as displayed in Table 5.4.

instance, *Ranunculus bulbosus* (species 14) is densely connected with five interspecific links above the threshold. Can that be related to its tolerance for nutrient-poor soil and its preferred occurrence in species-rich patches? There is a noticeable imbalance between positive and negative interactions. The dominance of positive interactions in the Machair vegetation is surprising given that much research in ecology has emphasised the role of competition within communities, though this is now changing as the potentially important role of facilitation is recognised (e.g. [21]). Ecologists also suggest that positive interactions may be more characteristic for harsh environments (e.g. in [19]) as is found in the Machair vegetation from where the data comes from. It is worth remembering however that the interactions observed in these data occur between species at the same trophic level and as such are but one horizontal slice of a much more complex hierarchical food web involving plant pathogens, insect and mammalian herbivores and their predators. Nonetheless, the relative lack of negative interactions is intriguing in that it suggests that interspecific competition does not dominate this grassland system.

Figure 5.17 shows, for a selected plant species, the marginal posterior probability of

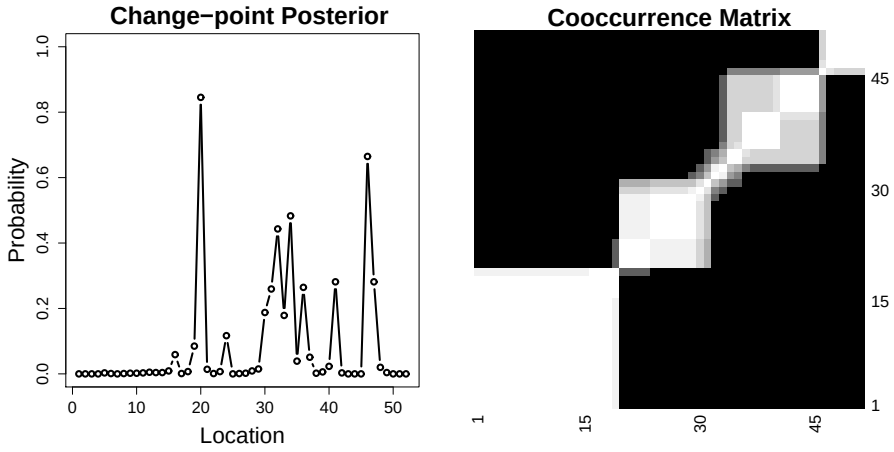


Figure 5.17: Inferred spatial segmentation for a selected plant species, *Carex pulicaris* using BRAM. Left panel: Marginal posterior probability of a change-point occurring along the longitudinal direction in arbitrary units (corresponding to the plot location ID number in the ecological survey). **Right panel:** Co-occurrence matrix for the selected plant species. The axes represent the position along the longitudinal direction, as before. The grey shading indicates the posterior probability of two longitudinal positions being assigned to the same spatial segment, i.e. of not being separated by a change-point, ranging from 0 (black) to 1 (white).

a change-point along the longitudinal direction as well as the posterior co-occurrence matrix, as introduced by Grzegorczyk and Husmeier [58]. I clustered plant species on the basis of these co-occurrence matrices, using a simple clustering algorithm (K-means with restarts) combined with the gap statistic for deciding on the number of clusters (see Section 3.3 and [137, 69]). The results are shown in Figure 5.18. Ecologists could make use of clusters like these to, e.g., identify species which share similar ecological sensitivities. These results demonstrate that the proposed method provides a useful tool for explorative data analysis in ecology with respect to both species interactions and spatial heterogeneity.

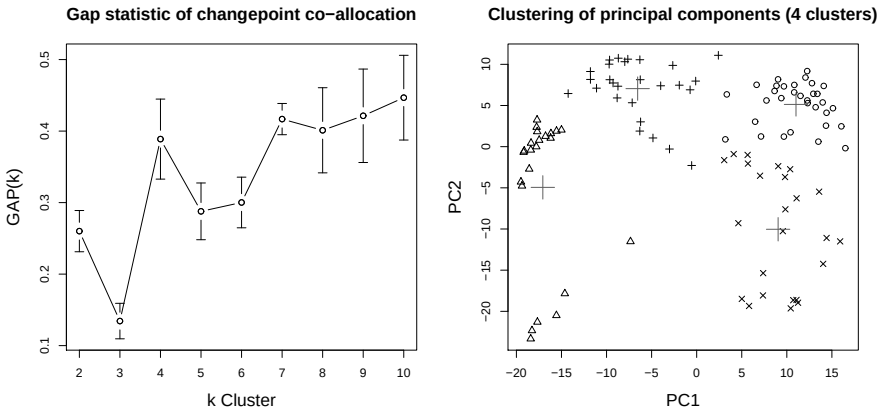


Figure 5.18: Clustering of plant species based on their inferred spatial segmentation. The plant species included in the ecological survey described in Section 5.3.3 were clustered on the basis of the inferred co-occurrence matrices, shown in Figure 5.17. **Left panel:** The gap statistic described in Section 3.3 suggests that $k = 2$ and $k = 4$ are reliable cluster numbers because the gap difference to the subsequent cluster, $GAP(k) - GAP(k+1)$, is greater than the standard error at $GAP(k)$. This indicates that the increase of the sum of pairwise distances from k to $k+1$ is significant and, hence, that k is a reasonable cluster number. **Right panel:** A plot of the plant species in the space spanned by the first principal components that were computed from the inferred co-occurrence matrices. The symbols indicate cluster membership and the large crosses the center of each cluster.

Table 5.4: Indices with full scientific names as appearing in Figure 5.16. These plants can be assigned to four taxonomies of forbs (1-19), grasses (20-29), rushes (30-33) and sedges (34-39).

ID	Name	ID	Name
1	*Anagallis tenella*	21	*Aira praecox*
2	*Calluna vulgaris*	22	*Anthoxanthum odoratum*
3	*Drosera rotundifolia*	23	*Cynosurus cristatus*
4	*Epilobium palustre*	24	*Festuca rubra*
5	*Galium verum*	25	*Festuca vivipara*
6	*Hypochaeris radicata*	26	*Holcus lanatus*
7	*Leontodon autumnalis*	27	*Koeleria macrantha*
8	*Lychnis flos-cuculi*	28	*Molinia caerulea*
9	*Odontites verna*	29	*Poa pratensis*
10	*Plantago lanceolata*	30	*Juncus effusus*
11	*Potentilla erecta*	31	*Juncus kochii*
12	*Potentilla palustris*	32	*Luzula campestris*
13	*Prunella vulgaris*	33	*Luzula pilosa*
14	*Ranunculus bulbosus*	34	*Carex arenaria*
15	*Ranunculus repens*	35	*Carex demissa*
16	*Sagina procumbens*	36	*Carex dioica*
17	*Succia pratensis*	37	*Carex flacca*
18	*Trifolum repens*	38	*Carex nigra*
19	*Viola riviniana*	39	*Eriophorum angustifolum*
20	*Agrostis capillaris*		

Chapter 6

Conclusion and future work

The aim of this thesis was to use machine learning techniques to learn regulatory networks in ecology and molecular biology. To this end, various methods for network inference were applied to the gene regulatory system of a circadian clock in plants. Different scenarios of the experimental setup were evaluated in respect to the performance of inference. The best performing method, namely the hierarchical Bayesian Network, was successfully modified in two different ways to infer species interactions in ecology. The following sections summarize the findings from the two previous chapters. Section 6.3 gives an outlook of future work in both fields of biology.

6.1 Gene Regulation

In Chapter 4 I have carried out a comparative evaluation of 15 state-of-the-art statistics/machine learning methods for regulatory network reconstruction, using the central gene regulatory network of the circadian clock in the model plant *A.thaliana* and a series of network modifications. To evaluate the network reconstruction performance objectively from a proper gold standard, I simulated mRNA and protein concentration time series from a published regulatory network structure. The simulations were based on a mathematical description of the individual molecular reactions, modelled with Markov jump processes to capture the intrinsic stochasticity of these events. The data generation process also emulated various experimental interventions carried out in the laboratory, including the knock-out of certain target genes, and the exposure of plants to different artificial light-dark cycles.

I have investigated the effects of various model choices and inference settings: the

estimation of the optimal regularization parameters in sparse regression (Lasso, Elastic Net, and Tesla), and the choice of both the structure and the parameter priors in hierarchical Bayesian regression. For estimating the regularization parameters, I have shown that cross-validation is slightly preferable to BIC, and that the heuristic modification suggested by Hastie et al. [70] is counter-productive. For the structure prior of the hierarchical Bayesian regression model, I have found that there is no significant advantage in using a truncated Poisson distribution on the cardinalities of the sets of regulators over a uniform distribution, subject to the same fan-in restriction. For the parameter prior of the hierarchical Bayesian regression model, I have found that the ridge regression prior significantly outperforms the g-prior.

In the main part of my study, I have applied the competing network reconstruction methods to a large variety of data, generated from different network structures, with different status of observation (mRNA only versus mRNA and proteins), and different methods for estimating de novo mRNA transcription rates. I have systematically disentangled the different effects with an ANOVA scheme. My results confirm various intuitively plausible trends, e.g. that the difficulty of network reconstruction increases with increasing network connectivity, and that for estimating de novo mRNA transcription rates, data smoothing has a beneficial effect. The novel contribution of my study consists in objectively quantifying these effects, in terms of average AUROC score differences associated with the respective main effects in the ANOVA scheme. For the model comparison, I have shown that hierarchical Bayesian regression outperforms all other methods, again objectively quantifying the performance gain.

My study has also revealed various surprising trends. Since the mechanisms of transcriptional regulation are based on non-linear Michaelis-Menten kinetics, explicitly imbuing the network reconstruction method with non-linear modelling capability via change-points in the response variable or the inclusion of inverse and quadratic terms should generally benefit the network reconstruction performance. My study has refuted this conjecture. I have carried out further synthetic toy studies to shed light on these effects. My study suggests that the results vary substantially with the amount of non-linearity and the noise variance, indicating the regimes where explicit non-linear modelling capability is beneficial, or counter-productive.

I have finally applied the best network reconstruction method from the comparative assessment to the mRNA concentration profiles from the TiMet project. The reconstructed network contains several topological features that are consistent with recently published regulatory networks of the circadian clock in *A.thaliana*. However, the de-

tailed structure clearly differs. This difference is a consequence of the different nature of the methods. For the networks published in the literature, the processes of transcriptional regulation were modelled with ordinary differential equations. The network structures were not selected with rigorous statistical inference; doing that e.g. with the procedure proposed by Vyshemirsky and Girolami [145] is computationally prohibitive. The consequence is a considerable degree of reliance on intuition and biological prior knowledge, as evidenced by repeated network modifications in the literature (see Figure 4.17). The methods applied in this thesis are based on more abstract models of molecular regulatory interactions, which render objective statistical inference computationally viable. Hence, our understanding of circadian regulation at the molecular level will potentially improve as a consequence of a synthesis of both approaches, which will suggest novel avenues for model adjustment. The proposed network reconstruction methods are particularly useful for linking circadian regulation in plants to metabolism, due to the current absence of detailed hypotheses and reliable mechanistic models.

6.2 Ecological Species Networks

In Chapter 5 I have addressed the problem of reconstructing species interaction networks from species abundance data. To this end, I have proposed two Bayesian models that combine Bayesian piecewise linear regression with a multiple change-point processes (BRAM) and Bayesian piecewise linear regression with a Mondrian process (BRAMP). These models are both motivated by a model that has been proposed in the molecular systems biology literature by Lèbre et al. [89], but has been adapted from the temporal domain, i.e. gene expression time series, to the spatial one, i.e. snapshot of species distributions in space, typical of ecological surveys.

I have introduced and tested three essential modifications, illustrated and motivated in Figure 5.1. First, I extended the 1-dimensional change-point process from Lèbre et al. [89] to a 2-dimensional one, which corresponds to a richer latent variable structure that allows modelling unobserved effects with smooth geographical variation. Second, I replaced the global change-point in the 2 dimensions with a Mondrian process following Roy and Teh [124], which improves the versatility of the spatial segmentation by providing a more fine grain approach. Third, I explicitly introduced an additional enforced parent node for each species, which represents the average species abundance from the spatial neighbourhood of the current location and thereby allows a correction for spatial autocorrelation.

I first tested my models on simple simulated data based on a linear regression model that resembled the model assumption of the change-points in BRAM and BRAMP. The results for this data confirmed the validity of both change-point processes and the proper functioning of the MCMC sampling scheme. The true change-points were consistently recovered with high confidence. In comparison to HBR, Lasso, and Banjo both models outperformed the competition as expected. BRAM and BRAMP also showed high AUROC scores of approximately 0.9 (a score of 1 is perfect recovery and 0.5 random recovery) for the recovery of the true network which indicates that the spatial segmentation facilitates the learning of the structure.

I further tested my models on data from trophic simulations, which combine spatial species dispersal with demographic and environmental effects and predator-prey interactions of the Lotka-Volterra form defined by a trophic network obtained from a niche model. The data simulations were conducted with the assumption of weak and strong predatory interactions that proved to lead to pronounced spatial heterogeneity in the case of strong predation and a more homogeneous data set in the case of weak predation. A first evaluation revealed that it is indeed important to consider the inclusion of a spatial autocorrelation variable into the predicting models in order to reflect the effects of species dispersion. This was observed for the Bayesian regression method with global change-points (BRAM) and without change-point (HBR) with significant improvements in network recovery. I further compared BRAMP, BRAM, HBR, Lasso and Banjo on this data set and observed the following differences. In the absence of pronounced spatial heterogeneity (strong predation), when there appears to be little room for improvement over the homogeneous models, i.e. HBR, Lasso, and Banjo, the performance of BRAM is on a par with Lasso and HBR. The Bayesian regression with the Mondrian process (BRAMP) can slightly improve over all methods on this data set, which is likely due the its ability to model the spatial segmentation on a finer scale. Although, the strong predation data has little to no homogeneity. there nevertheless seems to be enough variability in the data that BRAMP is able to exploit. In the presence of spatial heterogeneity (weak predation), BRAMP and BRAM clearly outperform all competing models (Figure 5.15). BRAMP dominates the performance over BRAM as I expected with the improved Mondrian change-point process.

An application to plant species abundance data from a ecological survey has demonstrated how the proposed methods can be used as a tool for hypothesis generation with respect to species interactions and spatial distribution patterns. The main problem with real data analysis is the 'objective' evaluation. In ecology, we currently lack

any gold standard, and the situation is more difficult than in molecular systems biology, where several databases about molecular functions and interactions exist. A more thorough evaluation of my models on real data, which is the objective of ongoing work, needs to be done in close collaboration with ecologists and will ultimately be based on somewhat circumstantial evidence. For the purpose of method assessment I will therefore pursue, in parallel, more extensive studies based on simulated data, with the objective to make the underlying models increasingly ecologically realistic.

6.3 Future work

Gene Regulation The study in Chapter 4 has demonstrated an extensive evaluation of various methods and setups related to circadian clock regulation. From a biological point of view, it would be valuable to extend this study to the scale that involves metabolite reactions and enzyme activities. This is essential in two ways: First, enzyme activities play a major role in the formation and modification of proteins. This can also involve the processing of transcription factors that in turn effect gene regulation. In so far, my approach of predicting gene activity through gene and protein concentrations alone is a simplification of the real underlying biological processes. Second, it is an open question if gene regulatory circuits are involved in the active control of metabolic functions, i.e. whether gene networks can steer metabolism to optimal states [15]. The circadian clock that I studied in Chapter 4 is hypothesised to be essential in plant development and growth by causing rhythmic expression of a multitude of genes. Is it possible that this influence extends to the level of metabolites which control various functions such as signalling, growth, and stress responses? A multiscale approach could introduce more realism by accounting for processes that take place on different levels of organization. For instance, a model of catalytic activity of enzymes and known metabolite pathways with links to the activity of the circadian clock, or links that connect these molecular traits to physiological characteristics and processes such as growth, flowering, or nutrient consumption as demonstrated in a broad multiscale model of *A. thaliana* in the Ph.D. thesis from Chew [26].

From a methodological point of view, the circadian network that I studied was a rather small network with around ten components. However, the number of components can be much larger as it is usually the case in genomic studies where thousands of genes act as putative prediction variables. Thus it is a major challenge at this scale to find the proper subset of features that explain a specific target gene. The pre-selection

of a feature ensemble can facilitate and speed up network inference by significantly decreasing network complexity [125]. I could think of mechanisms such as the Lasso and Elastic Net that serve as pre-selection step[1] for network inference in the likely case that I expand the search for putative features.

The use of biological prior knowledge about suspected binding behaviour, such as from cis-regulatory sites that match to transcription factor binding motifs, is another approach to feature selection. In contrast to feature pre-selection, however, known binding information could carefully guide feature selection by favouring specific predictors. To avoid deterministic choices based on this prior knowledge, one could model the selection in a probabilistic fashion using a Bayesian network with information sharing, e.g. by following Dondelinger et al. [35]. Another Bayesian network approach that could be considered is the encoding of the prior distribution of networks as an energy function that quantifies the difference of energy to a prior knowledge network as proposed by Werhli and Husmeier [150]. The Bayesian framework by Mukherjee and Speed [107] could also serve as a starting point to model multiple informative prior beliefs or so called "concordance functions". This study demonstrated how prior knowledge about different network features, such as individual edges or edges between classes of vertices, can be incorporated into a Bayesian framework.

Recently, there has been mounting evidence that so called model ensembles can increase the robustness of model prediction in systems biology [96]. The scale of the presented study in this thesis with 15 involved methods naturally invites the construction of such ensembles and would also allow the evaluation of different strategies for ensemble formation. Model ensembles are likely to improve model predictions and should be thoroughly investigated.

Ecological Application The study in Chapter 5 has shown a clear advantage of the Mondrian process (BRAMP) over a global change-point model (BRAM): namely, that it adapts the number of segments locally and therefore can deal with ecosystems that change rapidly in some areas, but slowly in others. However, the Mondrian process and the global change-point are intrinsically based on two distinguished perpendicular directions. This may be more appropriate for some applications than for others. For the application in my study, the plant ecosystem on the island of Uist, these two distinguished perpendicular directions exist. The island's ecogeography, with the open

[1] E.g. by dismissing putative parents that were found to have regression coefficients of zero, i.e. do not influence the regression model under a sparsity constraint.

sea in the west, and abutting land in the other directions, implies that the east-west soil profile (longitudinal coordinate) differs systematically from the north-south profile (latitudinal coordinate); see [91]. Similar patterns can be found on many other coastal islands, where for principal directions that do not coincide with latitude and longitudinal, the Mondrian process can be formulated in terms of a local, rotated coordinate system.

However, the Mondrian process will not always be the most appropriate model. For instance, for applications with rotational invariance other models, e.g. based on a Voronoi tesselation [111], might be be better suited. Voronoi tesselation only requires the location coordinates of the samples so that any shape of landscape can theoretically be modelled. In particular, centroidal Voronoi tesselation (Voronoi diagram) has shown its capability to approximate many patterns in nature and has been previously applied to model ecological dynamics, e.g. in [101, 122]. Future research with Voronoi diagrams could follow along the lines of landscape genetics as presented in [63]. Alternatively, a Pitman-Yor processes [134] (i.e. a distant dependent Dirichlet process), in analogy with image segmentation, could be attempted.

Another potential improvement concerns the parameter prior. For the current prior on the regression model the coefficients are assumed to be distributed according to a zero-mean multivariate Gaussian with a covariance drawn from an inverse gamma distribution. This prior is symmetric around 0 and hence does not discourage sign changes. A justification can, in fact, be given based on various recent ecology publications, which discuss how the nature of interactions can change with varying environmental conditions (e.g. [25, 141, 95, 27]). Mutualistic interactions may become neutral or antagonistic (i.e. involve a sign change), either temporarily or over parts of the range of the interacting species, and this is not ruled out by the prior I employ. However, the scenarios described above are, overall, quite rare, and they are in particular unlikely to apply to trophic interactions. In fact, if we know that, for two interacting species A and B, A eats B in rectangle 1, we would assume that it is more likely that A also eats B in rectangle 2 than the other way round. This prior notion can be incorporated into the model by putting a species dependent prior on the mean, and drawing the mean independently from this prior for each rectangle. The implementation of this idea effectively adds an extra layer to the Bayesian hierarchy, and has been investigated by Grzegorczyk and Husmeier [59] in the context of molecular systems biology.

Outlook on Multiscale Modelling The previously mentioned work from Chew [26] has shown the potential of multiscale modelling by introducing an integrated framework that connects different levels of organization in the plant *A. thaliana*. Chew had linked together different models involving photo periodism, carbon dynamics, a photothermal model, and functional-structural plant properties. Each of these components can provide important clues about the state of a plant system, and hence improve the prediction accuracy for methods that eventually infer hidden dependencies under study, such as how the circadian clock affects the growth of a plant. However, plants like any other species are also affected by exterior biotic factors that are typically organized on a larger scale compared to the biomolecular processes. This can include interspecies communication, e.g. through airborne signals in plants [73], or symbiotic relationship with other species such as fungi [17] or bacteria in soil [81]. The integration of such information requires knowledge about the ecological system and niche that forms the exterior environment of the individual under study. Thus, it seems to be beneficial to integrate molecular knowledge with ecological knowledge because ultimately these systems are interconnected.

Other examples that illustrate the benefits of multiscale modelling include the model of blood cell mechanics in malaria as proposed by Fedosov et al. [41] or the migration of cancer cells [83]. The latter study illustrates how principles on higher levels of organization in ecology can be found in the lower level of cell organization. The previously mentioned landscape genetics [63] demonstrates how molecular information from population genetics is influenced by the environment and landscape. In fact, one can also learn something about the environment itself, e.g. spatial neighbourhoods and niches from the spatially varying samples of genetic information. For instance, ordinary species density data could be augmented with genetic data that would provide additional information, e.g. about past migration patterns from conserved genes or evolutionary dependencies that could impact the prior knowledge.

The amount of available options for designing such integrated frameworks seems staggering and is limited only by the available data and the quality of a model to imitate the real natural processes. To deal with this increased complexity, it is conceptually straightforward to apply the models presented in my thesis to multiple transcription time-series from different locations of the plant. It is well known that transcription profiles can differ, e.g. in root, shoots, and the leaves of plants but currently this has not been integrated, although the data exists. These different locations can be treated in the same way as different locations in an ecological setting and allowing

for information sharing would provide means of signalling between plant components. Furthermore, inference could be extend to different phases of development of the plant by partitioning multiple time-series in different stages of development. This data is also available and can be found as transcription profiles from seeds, young plants, and mature plants. The inclusion of additional environmental factors such as various soil and air attributes was already demonstrated in the ecological study in Chapter 5 and could be included into the genetic regulatory inference. Integrating higher levels of organization such as the influence of other species in the vicinity of the model organism could provide additional information, although laboratory conditions are usually quite uniform. Quantification of the ground coverage of the plant, the amount and kind of mycelium in the soil, and the dominating type of bacteria are additional influencing factors that might be beneficial.

Appendix A

Gene Regulation: Comparison between Biopepa and qRT-PCR profiles, and assessing the effect of the log transformation

The gene regulation study in Chapter 4 uses a synthetic but realistic data set and a real world data set. Both seem to be similar in terms of the magnitude of the mRNA expression profiles. Here I study to what extend the data is similar and whether it has to be log-transformed or not.

The right panel of Figure A.1 shows a QQ-plot to compare the distribution of mRNA concentrations between the realistic data (Section 4.4.1) and the qRT-PCR profiles from the Timet project (Section 4.4.2). There is only a mild deviation from an overall linear dependence, which suggests that the specific technical aspects of qRT-PCR measurements, described e.g. in [14], do not require a major modification of my stochastic-process model of transcriptional regulation, as reviewed in Table 4.2 and implemented in Biopepa. This further suggests that the patterns and trends observed in the comparative evaluation based on my realistic data are indicative of results for real qRT-PCR data, and can be used for providing estimates of expected prediction accuracy and guiding decisions on model choice.

This in particular concerns the decision of whether or not to log-transform the data. Inserting log-transformed concentrations, $\tilde{x}_{g,t} = \log(x_{g,t})$, into the fundamental equation of transcriptional regulation, Equation (4.1), and applying the chain rule of differential calculus yields:

$$\frac{d\tilde{x}_{g,t}}{dt} = \left[\alpha_g + f_g(\exp(\tilde{\mathbf{x}}_{\boldsymbol{\pi}_n,t}) - \lambda_g \exp(\tilde{x}_{g,t})\right] \exp(-\tilde{x}_{g,t}) \tag{A.1}$$

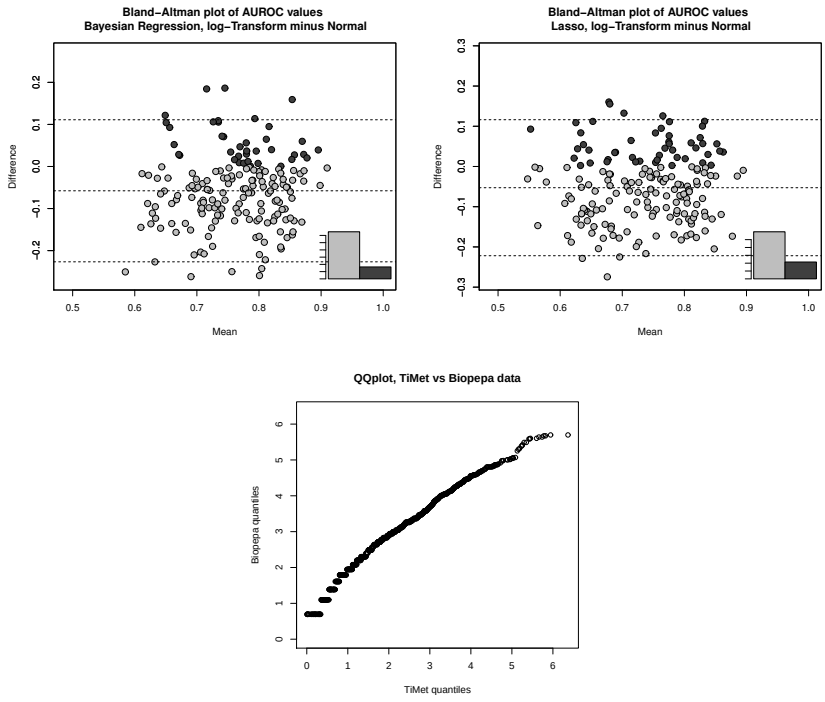


Figure A.1: Blant-Altman plot comparing network reconstruction accuracies between log-transformed and original data, and QQ-plot for comparing Biopepa and qRT-PCR data. The AUROC scores obtained from the original data are compared to those obtained from log-transformed data (the y-axis displays the difference, i.e. log-transformed minus original data). The left panel shows the results when applying the HBR method (avg. difference -0.058) and the centre panel for the Lasso method (avg. difference -0.053). For both methods, a majority of negative values can be observed (indicated by the grey box in the embedded histogram), i.e. log-transforming the data is detrimental to the learning accuracy. The right panel displays a QQ-plot comparing the distribution of realistic (Biopepa) and real (qRT-PCR) mRNA concentrations.

It is seen that in comparison with Equation (4.1), the log transformation has led to a more complicated functional dependence, not only by including an extra multiplicative factor $\exp(-\tilde{x}_{g,t})$ on the right-hand side, but also by making f_g a function of $\exp\left(\tilde{\mathbf{x}}_{\boldsymbol{\pi}_n,t}\right)$, which increases the amount of non-linearity in the system. This suggests that for network reconstruction, a log-transformation of the data will be counter productive.

To test this conjecture, I have repeated the network reconstruction on the realistic data after subjecting them to a log transformation. The results are summarised in

Figure A.1, which displays the differences in the form of Blant-Altman plots for the Lasso (centre panel) and HBR (left panel) methods. The average AUROC score difference is 0.06 in favour of the original, non-log transformed data. The distribution of paired differences shows that the proportion of negative differences, where the network reconstruction has deteriorated as a consequence of the log transformation, is significantly higher than the proportion of positive differences. This confirms my conjecture that log-transforming the mRNA concentrations is counter productive. Due to the reasoning in the second paragraph, that patterns observed for the realistic data are indicative of results to be expected for real qRT-PCR data, I have therefore elected *not* to log-transform the TiMet data.

Appendix B

Discrepancies of Area under the curve calculation (AUPREC)

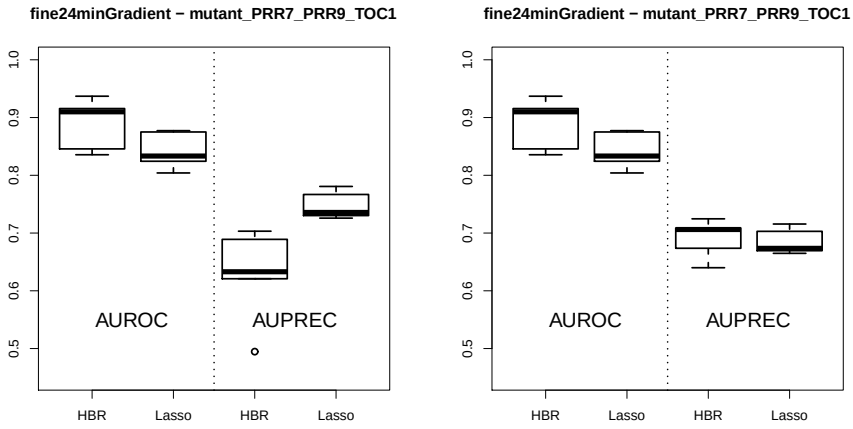


Figure B.1: Discrepancy of AUROC and AUPREC scores. Example comparison of AUROC and AUPREC distributions for the HBR and Lasso method with missing PREC extrapolation (left panel) and with PREC extrapolation (right panel). *L*eft panel: HBR shows equal and slightly better AUROC scores than Lasso but significantly worse corresponding AUPREC scores. *R*ight panel: The same data and AUROC scores but the AUPREC scores are calculated based on the extrapolation of missing TP and FP values following Davis and Goadrich [34]. This improves the AUPREC scores of HBR over Lasso and is consistent with the AUROC scores, see claim 1 in [34]: "If a curve dominates in ROC space then it dominates in PR space".

The AUPREC score was described in Section 3.4.1 as a tool for assessing the quality of network inference given a true known network configuration. While comparing this score for synthetic data with the corresponding AUROC scores, I sometimes encountered a discrepancy in the distribution of both scores for specific methods. Figure B.1 shows such a discrepancy with AUROC and AUPREC scores for the methods HBR and Lasso given a data set from the synthetic study in Chapter 4. As can be seen, the AUPREC scores of HBR drop significantly compared to the score of Lasso, although, both methods produce similar AUROC scores. The difference in the calculated amplitude of the curve can only be attributed to the missing "true negative" counts of the precision value of the PREC curve (see the Precision/Recall terms below). However, it seems unlikely that a lack of these counts would cause such a large deviation since they are usually recovered in abundant quantities. Thus I looked into the way the PREC curve is constructed. A possible faulty interpolation of missing intermediate "true positive" (TP), "false negative" (FN), and "false positive" (FP) was ruled out since the procedure followed exactly the one described in [34]. To recall: The variables that define the PREC curve are the Recall=$TP/(TP+FN)$ and Precision=$TP/(TP+FP)$. The values for TP, FN, and FP are derived by varying a threshold in the range of $[0, 1]$ and marking edges that have an edge indicator [1] that is greater than the threshold as "positive" counts and edges that are smaller as "negative" counts. True and false counts are then determined by referencing to a true edge structure.

A sample of a typical PREC curve is displayed in the top panel of Figure B.2. It shows that the range of Recall values spans the $(0, 1]$ interval, although this is not necessarily always the case. For instance, the HBR method applied to the synthetic molecular data set (Section 4.4.1 and as example in Figure B.1) regularly recovered large numbers of TP for the highest threshold of 1 ($TP_{thres=1}$), i.e. a substantial number of true positive counts would match with inferred edges of the highest confidence of 1. Given the above term for the Recall and observing that $(TP+FN)$ is equal to the total number of edges T_{total} in the true network, a significant gap of Recall values can appear if T_{total} is not substantially larger than $TP_{thres=1}$. This is illustrated in the lower left plot of Figure B.2, where missing values of the Recall are in the interval $(0, 0.42]$. In this example, the first TP encountered for the highest threshold was $TP_{thres=1} = 8$. Given that $T_{total} = 20$, the lowest Recall value is $TP_{thres=1}/T_{total} = 0.42$ with a corresponding low AUPREC value of 0.31.

[1] An edge indicator reflects the confidence in a learned edge and is expressed in the interval $[0, 1]$, where 1 is the greatest confidence. This can be derived from an edge posterior probability in the context of Bayesian regression, or .e.g an absolute correlation value.

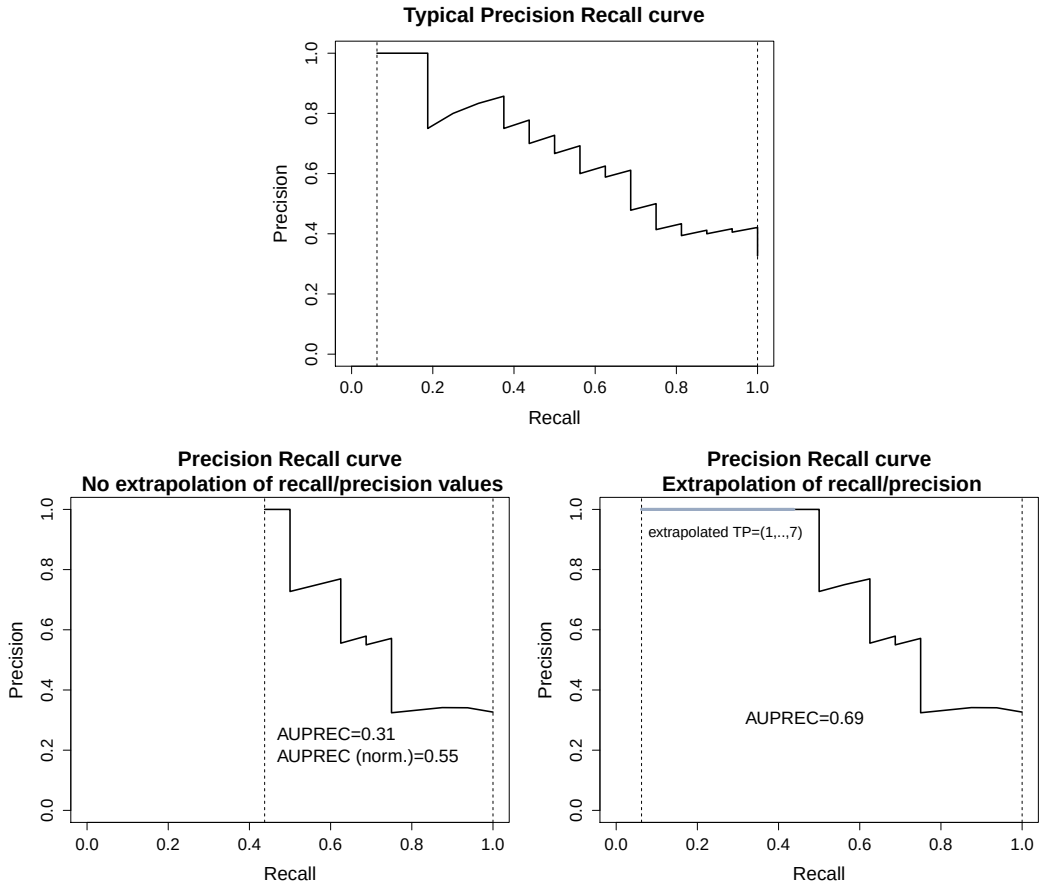


Figure B.2: Precision-Recall Curves and Recall extrapolation. The top panel shows a typical PREC curve that spans almost the whole recall interval of $[0, 1]$. The lower left plot shows a PREC curve that is based on true positive (TP) values that lack low counts, in this case $TP = (1, .., 7)$ and thus the curve covers only parts of the recall interval. Normalization into $[0, 1]$ ("AUPREC (norm.)") poorly reflects the full potential of the underlying data, which is shown in the lower right plot where TP values are extrapolated in the missing TP range.

The interpolation procedure of missing intermediary TP and FP values from Davis and Goadrich [34] can be used to fill the gap of missing small Recall values down to $TP = 0$. However, interpolation to $TP = 0$ is not always save since the Precision value can become undefined whenever $TP = 0$ and $FP = 0$, and thus rendering the denominator of the Precision to zero. My previous solution to this problem was to normalize the area under the curve to the whole interval by dividing the AUPREC

by the difference of maximum and minimum Recall values, thus stretching the Recall interval to $[0, 1]$. This is also the procedure used in the left panel of Figure B.1. This consistently increases the AUPREC scores (0.55 in the example of Figure B.2) but also leads to two additional observations:

Firstly, the AUPREC scores in some cases exhibited strong variations given similar data. It appears that the specific shape of the curve under a short Recall interval can affect the normalized AUPREC score much stronger than expected. For instance, a steep decline of Precision values occurring right after a short plateau of high Precision values would lower the score dramatically when compared to a curve that exhibited a wider plateau at low Recall values. Hence, the normalization would exaggerate small differences in the TP change.

Secondly, the results always underestimated AUPREC scores obtained from extrapolating TP values down to $TP = 1$ using the same interpolation procedure from Davis and Goadrich [34]. This procedure takes into account that the Precision does not change linear with the Recall, but is affected by a local skew caused by the change of FP values. For all missing TP values, the corresponding FP values are calculated and the Recall-Precision points are interpolated with the local skew and the amount of increase of the TP value (see Table 1 in [34]). In the case low TP values are missing one can artificially introduce $TP = 1$ and extrapolate to this value. This approach proved to be more robust in light of minor changes to the shape of the curve compared to the normalization procedure above. The lower right plot in Figure B.2 illustrates the extrapolation, which is the straight horizontal grey line constructed with $TP = (1, .., 7)$. The higher AUPREC score of 0.69 reflects the method's performance better when compared to scores of similar performing methods that show PREC curves that typically cover most of the Recall interval $(0, 1]$. In conclusion: Extrapolation of small missing TP values, starting with $TP = 1$, to fill a gap of Recall values will increase the robustness of the AUPREC score and also better reflects the underlying method's performance. This applies particularly to methods that tend to recover large amounts of TP with the highest confidence.

Bibliography

[1] Aderhold, A., Husmeier, D., and Grzegorczyk, M. (2014). Statistical inference of regulatory networks for circadian regulation. *Statistical applications in genetics and molecular biology*, 13(3):227–273.

[2] Aderhold, A., Husmeier, D., Lennon, J. J., Beale, C. M., and Smith, V. A. (2012). Hierarchical Bayesian models in ecology: reconstructing species interaction networks from non-homogeneous species abundance data. *Ecological Informatics*, 11:55–64.

[3] Aderhold, A., Husmeier, D., and Smith, V. A. (2013a). Reconstructing ecological networks with hierarchical Bayesian regression and Mondrian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 75–84.

[4] Aderhold, A., Husmeier, D., Smith, V. A., Millar, A. J., and Grzegorczyk, M. (2013b). Assessment of regression methods for inference of regulatory networks involved in circadian regulation. In *Proceedings of the Tenth International Workshop on Computational Systems Biology (WCSB)*, pages 30–42.

[5] 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.

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

[7] 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.

[8] Arbeitman, M. N., Furlong, E. E. M., Imam, F., Johnson, E., Null, B. H., Baker, B. S., Krasnow, M. A., Scott, M. P., Davis, R. W., and White, K. P. (2002). Gene expression during the life cycle of *Drosophila melanogaster*. *Science*, 297(5590):2270–2275.

[9] 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.

[10] Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006). Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biology*, 7(3). R25.

[11] Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, UK.

[12] Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., and Wild, D. L. (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3):349–356.

[13] Beisner, B. E., Haydon, D. T., and Cuddington, K. (2003). Alternative stable states in ecology. *Frontiers in Ecology and the Environment*, 1(7):376–382.

[14] Bengtsson, M., Hemberg, M., Rorsman, P., and Ståhlberg, A. (2008). Quantification of mRNA in single cells and modelling of RT-qPCR induced noise. *BMC Molecular Biology*, 9(1):63.

[15] Berkhout, J., Teusink, B., and Bruggeman, F. J. (2013). Gene network requirements for regulation of metabolic gene expression to a desired state. *Scientific reports*, 3.

[16] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer, Singapore.

[17] Bolan, N. S. (1991). A critical review on the role of mycorrhizal fungi in the uptake of phosphorus by plants. *Plant and Soil*, 134(2):189–207.

[18] Brandt, S. (1999). *Data Analysis: Statistical and Computational Methods for Scientists and Engineers*. Springer, New York, USA.

[19] Brooker, R. W. and Callaghan, T. V. (1998). The balance between positive and negative plant interactions and its relationship to environmental gradients: a model. *Oikos*, pages 196–207.

[20] Brooks, S. P. and Gelman, A. (1999). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7:434–455.

[21] Bruno, J. F., Stachowicz, J. J., and Bertness, M. D. (2003). Inclusion of facilitation into ecological theory. *Evolution*, 18(3):119–125.

[22] Buma, B. and Wessman, C. A. (2011). Disturbance interactions can impact resilience mechanisms of forests. *Ecosphere*, 2(5):64.

[23] Bustin, S. A., Benes, V., Garson, J. A., Hellemans, J., Huggett, J., Kubista, M., Mueller, R., Nolan, T., Pfaffl, M. W., Shipley, G. L., et al. (2009). The miqe guidelines: minimum information for publication of quantitative real-time pcr experiments. *Clinical chemistry*, 55(4):611–622.

[24] Butte, A. J. and Kohane, I. S. (2000). Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Pacific Symposium on Biocomputing*, volume 5, pages 418–429.

[25] Callaway, R. M. and Walker, L. R. (1997). Competition and facilitation: a synthetic approach to interactions in plant communities. *Ecology*, 78(7):1958–1965.

[26] Chew, Y. H. (2013). *Multi-scale whole-plant model of Arabidopsis growth to flowering*. PhD thesis, School of Biological Scienes, University of Edinburgh, UK.

[27] Choler, P., Michalet, R., and Callaway, R. M. (2001). Facilitation and competition on gradients in alpine plant communities. *Ecology*, 82(12):3295–3308.

[28] Chu, Y. and Corey, D. R. (2012). Rna sequencing: platform selection, experimental design, and data interpretation. *Nucleic acid therapeutics*, 22(4):271–274.

[29] Ciocchetta, F. and Hillston, J. (2009). Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33):3065–3084.

[30] Cohen, J. E. (2004). Mathematics is biology's next microscope, only better; biology is mathematics' next physics, only better. *PLoS Biology*, 2(12):e439.

[31] Dahlgren, J. P. (2010). Alternative regression methods are not considered in Murtaugh (2009) or by ecologists in general. *Ecology Letters*, 13(5):E7–9.

[32] Dale, M. R. T. and Fortin, M. J. (2002). Spatial autocorrelation and statistical tests in ecology. *Ecoscience*, 9(2):162–167.

[33] Davies, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240.

[34] Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of International Conference on Machine Learning*, pages 233–240. ACM.

[35] Dondelinger, F., Lebre, S., and Husmeier, D. (2010). Reconstructing developmental gene networks using heterogeneous dynamic Bayesian networks with information sharing. *Machine Learning in Systems Biology*, page 127.

[36] Dunne, J. A., Williams, R. J., and Martinez, N. D. (2002). Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, 5:558–567.

[37] D'haeseleer, P., Liang, S., and Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726.

[38] Edwards, K. D., Akman, O. E., Knox, K., Lumsden, P. J., Thomson, A. W., Brown, P. E., Pokhilko, A., Kozma-Bognar, L., Nagy, F., Rand, D. A., and Millar, A. J. (2010). Quantitative analysis of regulatory flexibility under changing environmental conditions. *Molecular Systems Biology*, 6(1).

[39] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

[40] Faisal, A., Dondelinger, F., Husmeier, D., and Beale, C. M. (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.

[41] Fedosov, D. A., Lei, H., Caswell, B., Suresh, S., and Karniadakis, G. E. (2011). Multiscale modeling of red blood cell mechanics and blood flow in malaria. *PLoS Computational Biology*, 7(12):e1002270.

[42] Feugier, F. G. and Satake, A. (2012). Dynamical feedback between circadian clock and sucrose availability explains adaptive response of starch metabolism to various photoperiods. *Frontiers in Plant Science*, 3. 305.

[43] Flis, A., Fernandez, P., Zielinski, T., Sulpice, R., Pokhilko, A., McWatters, H., Millar, A. J., Stitt, M., and Halliday, K. J. (2013). Biological regulation identified by sharing timeseries data outside the 'omics. *Submitted*.

[44] Fogelberg, C. and Palade, V. (2009). Machine learning and genetic regulatory networks: a review and a roadmap. In *Foundations of Computational Intelligence*, volume 1, pages 3–34. Springer.

[45] Foley, J. A., DeFries, R., Asner, G. P., Barford, C., Bonan, G., Carpenter, S. R., Chapin, F. S., Coe, M. T., Daily, G. C., Gibbs, H. K., Helkowski, J. H., Holloway, T., Howard, E. A., Kucharik, C. J., Monfreda, C., Patz, J. A., and Prentice, I. C. (2005). Global consequences of land use. *Science*, 309:570–574.

[46] Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9:432–441.

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

[48] 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.

[49] Geiger, D. and Heckerman, D. (1994). Learning gaussian networks. In *International Conference on Uncertainty in Artificial Intelligence*, pages 235–243. Morgan Kaufmann Publishers.

[50] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

[51] Gelman, A. and Rubin, D. B. (1992a). Inference from iterative simulation using multiple sequences. *Statistical Science*, pages 457–472.

[52] Gelman, A. and Rubin, D. B. (1992b). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–472.

[53] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

[54] Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361.

[55] Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In Niklasson, L., Bodén, M., and Ziemske, T., editors, *ICANN 98*, volume 1 of *Perspectives in Neural Computing*, pages 201–206. Springer.

[56] Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732.

[57] Grzegorczyk, M., Aderhold, A., Smith, V. A., and Husmeier, D. (2014). Inference of circadian regulatory networks. In *Second International Work-conference on Bioinformatics and Biomedical Engineering*, volume 2, pages 1001–1014.

[58] Grzegorczyk, M. and Husmeier, D. (2011). Non-homogeneous dynamic Bayesian networks for continuous data. *Machine Learning*, 83(3):355–419.

[59] Grzegorczyk, M. and Husmeier, D. (2012). Bayesian regularization of non-homogeneous dynamic bayesian networks by globally coupling interaction parameters. In *Journal of Machine Learning Research (JMLR) Workshop and Conference Proceedings*, volume 22, pages 467–476. Microtome Publishing.

[60] Grzegorczyk, M. and Husmeier, D. (2012a). A non-homogeneous dynamic Bayesian network with sequentially coupled interaction parameters for applications in systems and synthetic biology. *Statistical Applications in Genetics and Molecular Biology (SAGMB)*, 11(4). Article 7.

[61] Grzegorczyk, M. and Husmeier, D. (2013). Regularization of non-homogeneous dynamic Bayesian networks with global information-coupling based on hierarchical Bayesian models. *Machine Learning*, pages 1–50.

[62] Guerriero, M. L., Pokhilko, A., Fernández, A. P., Halliday, K. J., Millar, A. J., and Hillston, J. (2012). Stochastic properties of the plant circadian clock. *Journal of The Royal Society Interface*, 9(69):744–756.

[63] Guillot, G., Estoup, A., Mortier, F., and Cosson, J. F. (2005). A spatial statistical model for landscape genetics. *Genetics*, 170(3):1261–1280.

[64] Hagemeijer, W. J. M. and Blair, M. J. (1997). *The EBCC atlas of European breeding birds: their distribution and abundance*. Poyser London.

[65] Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.

[66] Hartemink, A. J. (2001). *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks*. PhD thesis, MIT.

[67] Hartemink, A. J. (2010). *Banjo: Bayesian network inference with java objects*. https://www.cs.duke.edu/~amink/software/banjo.

[68] Hartigan, J. A. and Wong, M. A. (1979). A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

[69] Hastie, T., Tibshirani, R., and Friedman, J. (2001a). *The Elements of Statistical Learning*. Springer-Verlag.

[70] Hastie, T., Tibshirani, R., and Friedman, J. J. H. (2001b). *The Elements of Statistical Learning*, volume 1. Springer New York.

[71] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

[72] Hayete, B., Gardner, T. S., and Collins, J. J. (2007). Size matters: network inference tackles the genome scale. *Molecular Systems Biology*, 3(1).

[73] Heil, M. and Karban, R. (2010). Explaining evolution of plant communication by airborne signals. *Trends in Ecology & Evolution*, 25(3):137–144.

[74] Henneman, M. L. and Memmott, J. (2001). Infiltration of a Hawaiian community by introduced biological control agents. *Science*, 293(5533):1314–1316.

[75] Herrero, E., Kolmos, E., Bujdoso, N., Yuan, Y., Wang, M., Berns, M. C., Uhlworm, H., Coupland, G., Saini, R., Jaskolski, M., Webb, A., Golçaves, J., and Davis, S. J. (2012). EARLY FLOWERING4 recruitment of EARLY FLOWERING3 in the nucleus sustains the *Arabidopsis* circadian clock. *Plant Cell Online*, 24(2):428–443.

[76] Hertel, J., Lindemeyer, M., Missal, K., Fried, C., Tanzer, A., Flamm, C., Hofacker, I. L., Stadler, P. F., et al. (2006). The expansion of the metazoan microRNA repertoire. *BMC Genomics*, 7(1):25.

[77] Holling, C. S. (1973). Resilience and stability of ecological systems. *Annual review of ecology and systematics*, pages 1–23.

[78] Honkela, A., Girardot, C., Gustafson, E. H., Liu, Y.-H., Furlong, E. E. M., Lawrence, N. D., and Rattray, M. (2010). Model-based method for transcription factor target identification with limited data. *Proceedings of the National Academy of Sciences*, 107(17):7793–7798.

[79] Husmeier, D. (1999). *Neural Networks for Conditional Probability Estimation: Forecasting Beyond Point Predictions*. Perspectives in Neural Computing. Springer, London.

[80] Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics*, 19:2271–2282.

[81] Ingham, R. E., Trofymow, J., Ingham, E. R., and Coleman, D. C. (1985). Interactions of bacteria, fungi, and their nematode grazers: effects on nutrient cycling and plant growth. *Ecological Monographs*, 55(1):119–140.

[82] Kalaitzis, A. A., Honkela, A., Gao, P., and Lawrence, N. D. (2013). *gptk: Gaussian Processes Tool-Kit*. R package version 1.06.

[83] Kareva, I. (2011). What can ecology teach us about cancer? *Translational Oncology*, 4(5):266.

[84] Ko, Y., Zhai, C., and Rodriguez-Zas, S. L. (2007). Inference of gene pathways using Gaussian mixture models. In *International Conference on Bioinformatics and Biomedicine*, pages 362–367. Fremont, CA.

[85] Ko, Y., Zhai, C., and Rodriguez-Zas, S. L. (2009). Inference of gene pathways using mixture Bayesian networks. *BMC Systems Biology*, 3. 54.

[86] Kolmos, E., Nowak, M., Werner, M., Fischer, K., Schwarz, G., Mathews, S., Schoof, H., Nagy, F., Bujnicki, J. M., and Davis, S. J. (2009). Integrating ELF4 into the circadian system through combined structural and functional studies. *HFSP Journal*, 3(5):350–366.

[87] Lande, R., Engen, S., and Saether, B. E. (2003). *Stochastic Population dynamics in Ecology and Conservation*. Oxford University Press.

[88] Lawrence, N. D., Girolami, M., Rattray, M., and Sanguinetti, G. (2010). *Learning and inference in computational systems biology*. MIT Press Cambridge.

[89] Lèbre, S., Becq, J., Devaux, F., Lelandais, G., and Stumpf, M. P. H. (2010). Statistical inference of the time-varying structure of gene-regulation networks. *BMC Systems Biology*, 4. 130.

[90] Lennon, J. J. (2000). Red-shifts and red herrings in geographical ecology. *Ecography*, 23:101–113.

[91] Lennon, J. J., Beale, C. M., Reid, C. L., Kent, M., and Pakeman, R. J. (2011). Are richness patterns of common and rare species equally well explained by environmental variables. *Ecography*, 34:529–539.

[92] 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.

[93] Locke, J. C. W., Kozma-Bognár, L., Gould, P. D., Fehér, B., Kevei, E., Nagy, F., Turner, M. S., Hall, A., and Millar, A. J. (2006). Experimental validation of a predicted feedback loop in the multi-oscillator clock of *Arabidopsis thaliana*. *Molecular Systems Biology*, 2(1). 59.

[94] MacKay, D. J. C. (1992). Bayesian interpolation. *Neural computation*, 4(3):415–447.

[95] Maestre, F. T., Callaway, R. M., Valladares, F., and Lortie, C. J. (2009). Refining the stress-gradient hypothesis for competition and facilitation in plant communities. *Journal of Ecology*, 97(2):199–205.

[96] Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., Allison, K. R., Kellis, M., Collins, J. J., Stolovitzky, G., et al. (2012). Wisdom of crowds for robust gene network inference. *Nature Methods*, 9(8):796–804.

[97] Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla-Favera, R., and Califano, A. (2006). ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(S7).

[98] Marin, J.-M. and Robert, C. P. (2007). *Bayesian core: A practical approach to computational Bayesian statistics*. Springer, New York, USA.

[99] Meier-Schellersheim, M., Fraser, I. D. C., and Klauschen, F. (2009). Multi-scale modeling for biologists. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(1):4–14.

[100] Memmott, J., Fowler, S., Paynter, Q., Sheppard, A. W., and Syrett, P. (2000). The invertebrate fauna on broom, *Cytisus scoparius*, in two native and two exotic habitats. *Acta Oecologica*, 21(3):213–222.

[101] Mercier, F. and Baujard, O. (1997). Voronoi diagrams to model forest dynamics in French Guiana. In *Proceedings of GeoComputation*, volume 97.

[102] Meyer, P. E., Lafitte, F., and Bontempi, G. (2008). MINET: An open source R/Bioconductor Package for Mutual Information based Network Inference. *BMC Bioinformatics*, 9.

[103] Milns, I., Beale, C. M., and Smith, V. A. (2010). Revealing ecological networks using Bayesian network inference algorithms. *Ecology*, 91:1892–1899.

[104] Mitchell, T. J. and Beauchamp, J. J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

[105] Mora, T. and Bialek, W. (2011). Are biological systems poised at criticality? *Journal of Statistical Physics*, 144(2):268–302.

[106] Morrissey, E. R., Juárez, M. A., Denby, K. J., and Burroughs, N. J. (2011). Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully Bayesian spline autoregression. *Biostatistics*, 12(4):682–694.

[107] Mukherjee, S. and Speed, T. P. (2008). Network inference using informative priors. *Proceedings of the National Academy of Sciences*, 105(38):14313–14318.

[108] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. The MIT Press.

[109] Nabney, I. T. (2002). *NETLAB: algorithms for pattern recognition*. Springer.

[110] Neuneier, R., Hergert, F., Finnoff, W., and Ormoneit, D. (1994). Estimation of conditional densities: a comparison of neural network approaches. In *International Conference on Artificial Neural Networks*, pages 689–692. Springer.

[111] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2009). *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons.

[112] Passos, J. F., Nelson, G., Wang, C., Richter, T., Simillion, C., Proctor, C. J., Miwa, S., Olijslagers, S., Hallinan, J., Wipat, A., Saretzki, G., Rudolph, K. L., Kirkwood, T. B. L., and von Zglinicki, T. (2010). Feedback between p21 and reactive oxygen production is necessary for cell senescence. *Molecular Systems Biology*, 6(1).

[113] Pokhilko, A., Fernández, A. P., Edwards, K. D., Southern, M. M., Halliday, K. J., and Millar, A. J. (2012). The clock gene circuit in *Arabidopsis* includes a repressilator with additional feedback loops. *Molecular Systems Biology*, 8:574.

[114] Pokhilko, A., Hodge, S. K., Stratford, K., Knox, K., Edwards, K. D., Thomson, A. W., Mizuno, T., and Millar, A. J. (2010). Data assimilation constrains new connections and components in a complex, eukaryotic circadian clock model. *Molecular Systems Biology*, 6(1).

[115] Pokhilko, A., Mas, P., and Millar, A. J. (2013). Modelling the widespread effects of TOC1 signalling on the plant circadian clock and its outputs. *BMC Systems Biology*, 7(1):1–12.

[116] Punskaya, E., Andrieu, C., Doucet, A., and Fitzgerald, W. J. (2002). Bayesian curve fitting using MCMC with applications to signal segmentation. *Signal Processing, IEEE Transactions on*, 50(3):747–758.

[117] Rasmussen, C. E. (1996). *Evaluation of Gaussian processes and other methods for non-linear regression*. PhD thesis, University of Toronto, Canada.

[118] Rasmussen, C. E., Neal, R. M., Hinton, G. E., van Camp, D., Revow, M., Ghahramani, Z., Kustra, R., and Tibshirani, R. (1996). *The DELVE manual*.

[119] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*, volume 1. MIT Press Cambridge, MA.

[120] Robinson, J. W. and Hartemink, A. J. (2010). Learning non-stationary dynamic Bayesian networks. *Journal of Machine Learning Research*, 11:3647–3680.

[121] 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.

[122] Roque, W. L. and Choset, H. (1998). The green island formation in forest fire modeling with voronoi diagrams. In *Proceedings 3rd CGC Workshop on Computational Geometry*. Citeseer.

[123] Roy, D. M. (2011). *Computability, inference and modeling in probabilistic programming*. PhD thesis, Massachusetts Institute of Technology, Cambridge, United States.

[124] Roy, D. M. and Teh, Y. W. (2008). The Mondrian process. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, volume 21.

[125] Ruyssinck, J., Huynh-Thu, V. A., Geurts, P., Dhaene, T., Demeester, P., and Saeys, Y. (2014). NIMEFI: Gene regulatory network inference using multiple ensemble feature importance algorithms. *PloS One*, 9(3):e92709.

[126] Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529.

[127] Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genomics and Molecular Biology*, 4.

[128] Scheffer, M., Carpenter, S., Foley, J. A., Folke, C., and Walker, B. (2001). Catastrophic shifts in ecosystems. *Nature*, 413(6856):591–596.

[129] Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470.

[130] Shinozaki, K., Yamaguchi-Shinozaki, K., and Seki, M. (2003). Regulatory network of gene expression in the drought and cold stress responses. *Current Opinion in Plant Biology*, 6(5):410–417.

[131] Smith, M. and Kohn, R. (1996). Nonparametric regression using Bayesian variable selection. *Journal of Econometrics*, 75:317–343.

[132] Smith, V. A., Yu, J., Smulders, T. V., Hartemink, A. J., and Jarvis, E. D. (2006). Computational inference of neural information flow networks. *PLoS Computational Biology*, 2(11):1436–1449.

[133] Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2002). Derivative observations in Gaussian process models of dynamic systems. In *Proceedings of Neural Information Processing Systems*. MIT Press.

[134] Sudderth, E. and Jordan, M. I. (2009). Shared segmentation of natural scenes using dependent Pitman-Yor processes. *Advances in Neural Information Processing Systems*, 21:1585–1592.

[135] Tait, R. C. (1999). The application of molecular biology. *Current Issues in Molecular Biology*, 1(1):1–12.

[136] Tibshirani, R. (1995). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

[137] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

[138] TiMet-Consortium (2012). *The TiMet Project - Linking the clock to metabolism: http://timing-metabolism.eu*.

[139] Tipping, M. E. (2001). Spare Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.

[140] Tipping, M. E. and Faul, A. C. (2003). Fast marginal likelihood maximisation for sparse Bayesian models. In *International Workshop on Artificial Intelligence and Statistics*, volume 1, pages 3–6.

[141] Valiente-Banuet, A. and Verdú, M. (2008). Temporal shifts from facilitation to competition occur between closely related taxa. *Journal of Ecology*, 96(3):489–494.

[142] van Someren, E. P., Vaes, B. L., Steegenga, W. T., Sijbers, A. M., Dechering, K. J., and Reinders, M. J. (2006). Least absolute regression network analysis of the murine osterblast differentiation network. *Bioinformatics*, 22(4):477–484.

[143] 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.

[144] Vitousek, P. M., Aber, J. D., Howarth, R. W., Likens, G. E., Matson, P. A., Schindler, D. W., Schlesinger, W. H., and Tilman, D. G. (1997). Human alteration

of the global nitrogen cycle: sources and consequences. *Ecological Applications*, 7(3):737–750.

[145] Vyshemirsky, V. and Girolami, M. A. (2008). Bayesian ranking of biochemical system models. *Bioinformatics*, 24:833–839.

[146] Wang, G., Zhu, X., Hood, L., and Ao, P. (2013). From Phage lambda to human cancer: endogenous molecular-cellular network hypothesis. *Quantitative Biology*, pages 1–18.

[147] Wang, P., Laskey, K. B., Domeniconi, C., and Jordan, M. M. (2011). Nonparametric Bayesian Co-clustering Ensembles. In *Proceedings of the SIAM International Conference on Data Mining*, pages 331–342.

[148] Weirauch, M. T., Cote, A., Norel, R., Annala, M., Zhao, Y., Riley, T. R., Saez-Rodriguez, J., Cokelaer, T., Vedenko, A., Talukder, S., Consortium, D., H.J., B., Morris, Q., Bulyk, M, L., G., S., and Hughes, T. (2013). Evaluation of methods for modeling transcription factor sequence specificity. *Nature Biotechnology*.

[149] 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.

[150] Werhli, A. V. and Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, 6(1):1–47.

[151] Werner, E. E. and Peacor, S. D. (2003). A review of trait-mediated indirect interactions in ecological communities. *Ecology*, 84(5):1083–1100.

[152] Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics*, 10(2):122–133.

[153] Wilkinson, D. J. (2011). *Stochastic modelling for systems biology*, volume 44. CRC press.

[154] Williams, R. J. and Martinez, N. D. (2000). Simple rules yield complex food webs. *Nature*, 404(6774):180–183.

[155] Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., and Jarvis, E. D. (2004). Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–603.

[156] Zoppoli, P., Morganella, S., and Ceccarelli, M. (2010). TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by ab information theoretic approach. *BMC Bioinformatics*, 11(154).

[157] 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.