University of Groningen

Advanced non-homogeneous dynamic Bayesian network models for statistical analyses of time series data

Shafiee Kamalabad, Mahdi

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2019

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Shafiee Kamalabad, M. (2019). *Advanced non-homogeneous dynamic Bayesian network models for statistical analyses of time series data*. [Thesis fully internal (DIV), University of Groningen]. University of Groningen.

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 13-02-2023

Advanced non-homogeneous dynamic Bayesian network models for statistical analyses of time series data

Mahdi Shafiee Kamalabad

university of
groningen

The research for this doctoral dissertation has been carried out at the Faculty of Science and Engineering, University of Groningen, the Netherlands, within the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence.

© Copyright 2018 M. Shafiee Kamalabad
PhD Thesis, University of Groningen, the Netherlands

ISBN 978-94-034-1265-8 (printed version)
ISBN 978-94-034-1264-1 (electronic version)

This thesis was typeset using LaTeX template by Hildeberto Jordan Kojakhmetov.

Advanced non-homogeneous dynamic Bayesian network models for statistical analyses of time series data

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus prof. dr. E. Sterken,
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

14 January 2019 at 14:30 hours

by

Mahdi Shafiee Kamalabad

born on 21 September 1982
in Tehran, Iran

Supervisor
Prof. E. Wit

Co-supervisor
Dr. M. A. Grzegorczyk

Assessment committee
Prof. D. Husmeier
Prof. C.J. Albers
Prof. J. Mulder

To my parents

&

To my Mozhgan, Alicenna and Delina

Contents

1 Introduction **1**
1.1 Static and dynamic Bayesian networks 1
1.2 Network inference 2
1.3 Non-homogeneous DBNs (NH-DBNs) 2
1.4 Another conceptual problem 5
1.5 The aim of this thesis 5
1.6 Outline of thesis contribution 6

2 Partially sequentially segmentwise coupled NH-DBNs **9**
2.1 Methods 10
2.2 Data 22
2.3 Empirical results 24
2.4 Discussion and conclusions 28

3 Generalized sequentially coupled NH-DBNs **31**
3.1 Methods 31
3.2 Data 49
3.3 Empirical results 51
3.4 Discussion and conclusions 58

4 Partially edge-wise coupled NH-DBNs **61**
4.1 Methods 62
4.2 Data 76
4.3 Hyperparameter and simulation settings 79
4.4 Empirical results 79
4.5 Discussion and conclusions 85
4.6 Appendix 86

5 Partially NH-DBNs based on Bayesian regression models with partitioned design matrices **89**
5.1 Methods 90
5.2 Implementation 97
5.3 Data and empirical results 98
5.4 Discussion and conclusions 104

5.5 Appendix 105

6 Comparative evaluation of various frequentist and Bayesian non-homogeneous Poisson counting models 111

6.1 Methods 112
6.2 Validation 123
6.3 Data 125
6.4 Simulation Details 128
6.5 Comparative Evaluation Study 129
6.6 Further model diagnostics 138
6.7 Discussion and conclusions 144
6.8 Appendix 146

Summary 153

Samenvatting 157

Acknowledgments 161

Bibliography 163

Curriculum Vitae 169

Chapter 1

Introduction

Inferring network topologies of interacting units from temporal data is a statistically challenging task in many scientific disciplines. The goal is to learn the dependencies between the units from the data and to represent them in form of a network. A topical example is the field of computational system biology, where one of the major goals is to learn cellular networks, such as gene regularity transcription networks (see, e.g., [18]) and protein signaling pathways (see, e.g., [51].) Further examples include neural information flow networks [60] and ecological networks [2].

One class of models that has been widely applied to deal with this challenge, is the class of dynamic Bayesian network (DBN) models. The underling assumption is that the regulatory processes are homogeneous, so that DBNs assume the network interaction parameters to stay constant in time. For many real-world applications, this homogeneity assumption is too restrictive and can lead to wrong conclusions. To address this shortcoming, non-homogeneous dynamic Bayesian networks (NH-DBNs) have been proposed in the literature. Section 1.3 of this chapter gives an overview to different types of NH-DBNs and also discusses their advantages and disadvantages.

1.1 Static and dynamic Bayesian networks

Dynamic Bayesian networks (DBNs) are a popular class of models for learning the dependencies between random variables from temporal data.[1] Unlike in static Bayesian networks (BNs), a dependency between two random variables X and Y is typically interpreted in terms of a regulatory interaction with a time delay. A directed edge from variable X to variable Y, symbolically $X \to Y$, indicates that the value of variable Y at any time point t depends on the realisation of X at the previous time point $t-1$. Therefore, in DBNs, since all interactions are subject to a time lag the network does not have to be acyclic.

---

[1] DBNs extend standard static Bayesian networks (BNs) with the concept of time.

Typically, various variables $X_1, \dots, X_k$ have a regulatory effect on a target Y, and the relationship between $X_1, \dots, X_k$ and Y can be represented by a regression model that takes the time lag into account. E.g., if the time lag is one time point, the regression model takes the form:

$$y_t = \beta_0 + \beta_1 x_{1,t-1} + ... + \beta_k x_{k,t-1} + u_t \qquad (t = 2, \dots, T) \tag{1.1}$$

where T is the number of time points, y_t is the value of Y at time point t, $x_{i,t-1}$ is the value of covariate X_i at time point $t-1$, $\beta_0, \dots, \beta_k$ are regression coefficients, and u_t is the "unexplained" noise at time point t.

1.2 Network inference

In dynamic Bayesian network (DBN) applications there are usually N domain variables $Y_1, \dots, Y_N$ and the goal is to infer the covariates of each variable Y_i. As the covariates can be learned for each Y_i separately, DBN learning can be thought of as learning the covariates for a set of target variables $\{Y_1, \dots, Y_N\}$. There are N regression tasks, and in the i-th regression model, Y_i is the target variable and the remaining $N-1$ variables take the role of the potential covariates. The goal is to infer a covariate set $\pi_i \subset \{Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_N\}$ for each Y_i. From the covariate sets $\pi_1, \dots, \pi_N$ a network can be extracted. The network shows all regulatory interactions among the variables $Y_1, \dots, Y_N$. An edge $Y_j \to Y_i$ indicates that Y_j is a covariate of Y_i, i.e. that $Y_j \in \pi_i$. In the terminology of DBNs Y_j is then called a regulator of Y_i. All variables in π_i are regulators of Y_i $(i = 1, \dots, N)$.

1.3 Non-homogeneous DBNs (NH-DBNs)

The conventional assumption in dynamic Bayesian network models (DBNs) is that the regulatory relationships are homogeneous, so that the network parameters do not change in time. That is, the regression coefficients $\beta_0, \dots, \beta_K$ in Equation (1.1) stay constant across all time points ($t = 2, \dots, T$). Thus DBNs infer the network structure along with one single set of network parameters, and those parameters then apply to the whole time series. This homogeneity assumption is very restrictive and can lead to wrong results and conclusions. Therefore, DBNs cannot deal with non-homogeneous regularity processes, which often arise in systems biology. For example in a cellular network, the strength of the regulatory interactions are often exposed to (unobserved) external factors, such as cellular, environmental and/or experimental conditions (see, e.g., [8]), that influence the interactions. This renders the traditional DBNs inappropriate for most of the applications in systems biology. Therefore non-homogeneous dynamic Bayesian network models (NH-DBNs) have been proposed (see, e.g., [37]). NH-DBNs are a powerful statistical tool and do not make use of the homogeneity assumption.

The concept of non-homogeneity leads to time varying network parameters and/or time varying network structures. Therefore, NH-DBNs can be divided into two conceptual groups: NH-DBNs that allow only the network parameters to vary in time (see, e.g., [23]) and NH-DBNs that also allow the network structure to be time-dependent, see, e.g., [49], [38] or [14]. A statistical problem is that gene expression time series are often short so that NH-DBNs with time-dependent network structures are over-flexible and lead to inflated inference uncertainties. With regard to our biological applications throughout this thesis, we therefore focus on NH-DBNs which only allow the network parameters to change.

NH-DBNs with time-varying network parameters have been implemented with various allocation models to divide the data into disjoint data subsets:

- DBNs have been combined with free mixture models (MIX); see, e.g., [34] or [26].
- DBNs have been combined with hidden Markov models (HMM); see, e.g., [62] or [22].
- DBNs have been combined with multiple changepoint processes (CPS). see, e.g., [38] or [23].

The models infer the data segmentation, the joint network structure and the segment- or component-specific interaction parameters altogether from the data. In this thesis we focus on changepoint-divided (CPS) NH-DBNs, which have become the most widely applied NH-DBNs.

1.3.1 Changepoint-divided NH-DBNs

Changepoint-divided (CPS) non homogeneous dynamic Bayesian networks (NH-DBNs) models infer changepoints, which divide the data into disjunct segments. The data within each segment are modeled with linear regression models. There is a shared network structure among segments, and the segment-specific network parameters are learned for each segment separately. In typical applications in systems biology these NH-DBNs divide a short time series into even shorter segments, containing only a few data points. Learning the network parameters for each segment separately (conventional 'uncoupled' NH-DBN models) see, e.g. [38], then inevitably leads to over flexibility and inflated inference uncertainties. Moreover, they do not incorporate the reasonable prior assumption that neighbouring segments are often more likely to have similar network interaction parameters than distant segments.

To address these bottlenecks, more realistic models which allow for gradual adaptations of the network interaction parameters, have been proposed. E.g., the frequentistic models, proposed by [3], [36] and [35]. Those models make use of L1-regularized regression models ('LASSO') for the network parameter inference, and they employ a second L1 regularization term to penalize dissimilarities between network parameters of neighbouring segments. In those frequentistic models inference is based on penalized maximum likelihood approaches, and the fixed regularization parameter has to be optimized by cross-validation or in terms of the Bayesian Information Criterion ("BIC"). Bayesian models with coupling mechanisms between the segment-specific parameters have also been proposed. In [25] it was proposed to globally couple the segment-specific parameters. The key idea is to treat the segments as interchangeable units and to impose a shared hyperprior onto the prior expectations of the segment-specific parameters. In a complementary work ([24]) it was proposed to sequentially couple the parameters. The fully (sequentially) coupled model was developed to keep the network parameters of each segment similar to those of the previous segment. Here the parameters within segment h obtain as prior expectations their posterior expectations from the preceding segment $h-1$, and the coupling strength i.e., the variance of the network parameter priors (the similarity of the regression coefficients), is regulated by a coupling hyperparameter λ. This model can thus be seen as a Bayesian counterpart of the frequentistic models, mentioned above. The Bayesian models are inferred with Reversible Jump Markov Chain Monte Carlo (RJMCMC) simulations [21], and a comparative evaluation study of network reconstruction methods in [1] showed that the Bayesian models tend to reach higher network reconstruction accuracies than the frequentistic models.

1.3.2 The concept of parameter coupling

Parameter coupling can lead to significantly improved network reconstruction accuracies when the segment-specific parameters are similar, as shown in [24] and [25]. However, recently we have found that coupling can become counter-productive when the segment-specific parameters are dissimilar. The reason for that is that neither the sequential nor the global coupling scheme has an effective mechanism for uncoupling. When the segment-specific parameters are dissimilar, coupled NH-DBNs can only reduce the coupling strengths by making the parameter priors vague. This renders them significantly inferior to NH-DBNs without any coupling mechanism. Moreover, the fully coupled model suffers from another serious bottlenecks: The model couples all neighbouring segments $(h-1, h)$ with the same coupling strength. That is, it possesses only one single coupling hyperparameter λ which is shared among all segments $h > 1$ and all covariates.[2] To shed more light onto this, we note that both coupling mechanisms have been designed such that if a node A is regulated by a set of other nodes, e.g. $B \rightarrow A \leftarrow C$, then both edges have to be coupled with the same strength across all segments.[3] For many real-world applications this is unrealistic. E.g., the regulatory effect of B on A (i.e., the parameter associated with $B \rightarrow A$) can stay similar, while the regulatory effect of C on A can be subject to major changes. To re-use a traffic flow analogy from [48]: The traffic flow on the roads is different during rush hours and off-peak times. But rush hours usually do not affect the traffic flow on all roads. Typically there are susceptible roads with tailbacks during rush hours, while the traffic demand on other roads might stay constant.

---

[2] The models from [3], [36] and [35] suffer from the same drawbacks. Those models also possess only one single regularization ('tuning') parameter which determines the similarity of the network parameters among segments. The coupling strength between segments can neither vary over time nor is there any mechanism for uncoupling segments.

[3] We will therefore also refer to these models as *fully* coupled NH-DBNs.

1.4 Another conceptual problem

In many applications in systems biology, we encounter data that are collected under different experimental conditions. Instead of one single (long) time series, which can be divided into segments with natural temporal order, there are K (short) time series. These individual time series $k = 1, \dots, K$ have no natural order and are exchangeable units. That is, the available data are then automatically divided into K unordered components (=the individual time series), and there is no need for inferring the segmentation. In this situation it is often unclear a priori whether the network parameters are actually component-specific or whether they are constant across components. If the parameters stay constant, all data could be merged and be analyzed altogether with one single homogeneous DBN model. If there are component-specific parameters, then the data should not be merged and it would be better to analyze each time series separately. In the latter case, it can be useful to adapt the global parameter coupling scheme from [25], so as to encourage the network parameters to stay at least similar among components. The bottleneck of both approaches is that either the parameters are assumed to stay constant or that the parameters are assumed to be component-specific. In real-world applications there can be both types of parameters. E.g., if a variable Y is regulated by two other variables, symbolically $X_1 \rightarrow Y \leftarrow X_2$, then the regulatory interactions $X_1 \rightarrow Y$ might not be affected by the experimental conditions, while the regulatory $X_2 \rightarrow Y$ might be influenced by the condition, e.g. for $K = 2$ in terms of a linear regression model, one might have:

$$E[Y|X_1 = x_1, X_2 = x_2] = \begin{cases} \alpha x_1 + \beta x_2 & \text{if } k = 1 \\ \alpha x_1 + \gamma x_2 & \text{if } k = 2 \end{cases} \tag{1.2}$$

A homogeneous model is then inappropriate, since it would ignore that the regression coefficients β and γ are different. A non-homogeneous model comes with the drawback that the same regression coefficient α has to be learned two times separately. This is disadvantageous when the data within each component ($k = 1, 2$) are sparse and uninformative.

1.5 The aim of this thesis

To summarize what has been discussed in the previous sections, Figure 1.1 shows a graphical overview of the various NH-DBN models. In this thesis, we put our focus on the sequential and global coupling scheme and show how the coupled models can be improved, so as to address the above-mentioned drawbacks. We propose four novel non-homogeneous dynamic Bayesian network (DBN) models, which are more flexible and thus have the potential to capture the underlying interactions more accurately than the earlier proposed models.

1.6 Outline of thesis contribution

This thesis is organized as follows:

In **chapter 2**, we propose two new NH-DBN models to fix the deficits of the fully (sequentially) coupled NH-DBN model from [24]. The partially segment-wise coupled model can be seen as a consensus model between the uncoupled model and the fully coupled model. It has the uncoupled and a sequentially coupled NH-DBN models as limiting cases: If it couples all segments, it effectively becomes the fully coupled model. If it uncouples all segments, it effectively becomes the conventional uncoupled model. Moreover, we propose the generalized coupled model, which is a generalization of the fully sequentially coupled model. Like the fully sequentially coupled model, the new model does not have any option to uncouple, but it possesses segment-specific coupling parameters and allows for different coupling strengths between segments. We will demonstrate that the partially segment-wise coupled model can lead to significantly improved network reconstruction accuracies, while we do not see any significant improvements for the generalized coupled model. In chapter 3 we therefore have a closer look the generalized coupled model and refine it.

In **chapter 3**, we refine the generalized fully coupled model. In particular, we impose a hyperprior onto the second hyperparameter of the coupling parameter prior to allow for more information-exchange among the segment-specific coupling strengths.

In **chapter 4**, we present a novel partially edge-wise coupled model. Unlike the partially coupled model from chapter 2, this model infers for each individual edge whether the associated parameters should be coupled or stay uncoupled across the segments.

In **chapter 5**, we introduce another consensus model, which we refer to as a partially NH-DBN model. This model has been developed for the situation described in Section 1.4. The new model aims to infer the best trade-off between a homogeneous model (with constant parameters) and a non-homogeneous model (with component-specific parameters). In this chapter we also propose a Gaussian process based approach to deal with non-equidistant measurements. The (non-homogeneous) dynamic Bayesian network models assume that the domain variables have been measured at equidistant time points. For applications where this assumption is violated, we propose to employ a Gaussian process to predict the values at equidistant data points.

Chapter 6 presents a study, which is independent to those presented in the previous chapters. In chapter 6 we perform a comparative evaluation study

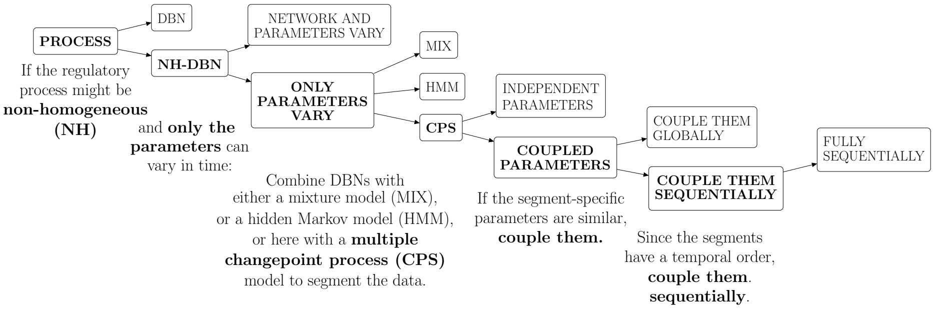

PROCESS
DBN
NH-DBN
If the regulatory process might be **non-homogeneous (NH)**
and **only the parameters** can vary in time:
NETWORK AND PARAMETERS VARY
ONLY PARAMETERS VARY
MIX
HMM
CPS
Combine DBNs with either a mixture model (MIX), or a hidden Markov model (HMM), or here with a **multiple changepoint process (CPS)** model to segment the data.
INDEPENDENT PARAMETERS
COUPLED PARAMETERS
If the segment-specific parameters are similar, **couple them.**
COUPLE THEM GLOBALLY
COUPLE THEM SEQUENTIALLY
Since the segments have a temporal order, **couple them. sequentially**.
FULLY SEQUENTIALLY

Figure 1.1: Overview of non-homogeneous dynamic Bayesian networks (NH-DBNs). We consider NH-DBNs whose parameters vary in time, and we use a multiple changepoint process (CPS) to segment the data into segments.

on popular non-homogeneous Poisson models for count data. For this study the standard homogeneous Poisson model (HOM) and three non-homogeneous variants, namely a Poisson changepoint model (CPS), a Poisson free mixture model (MIX), and a Poisson hidden Markov model (HMM) are implemented in both conceptual frameworks: a frequentist and a Bayesian framework. This yields 8 models in total, and the goal of this chapter is to shed some light onto their relative merits and shortcomings. The first major objective is to cross-compare the performances of the four models (HOM, CPS, MIX and HMM) independently for both modelling frameworks (Bayesian and frequentist). Subsequently, a pairwise comparison between the four Bayesian and the four frequentist models is performed to elucidate to which extent the results of the two paradigms ('Bayesian versus frequentist') differ. The evaluation study is performed on various synthetic Poisson data sets as well as on real-world taxi pick-up counts, extracted from the recently published New York City Taxi (NYCT) database.

Several parts of this thesis have previously been published in form of two journal articles, one in press, and four conference papers. One more paper has been submitted. The references are:

- Shafiee Kamalabad, M., Heberle A.M., Thedieck K. and Grzegorczyk, M. (2018) (accepted and in press):
  Partially non-homogeneous dynamic bayesian networks based on Bayesian regression models with partitioned design matrices. Bioinformatics, `http://dx.doi.org/10.1093/bioinformatics/bty917`, (chapter 5, see [59]).

- Shafiee Kamalabad, M. and Grzegorczyk, M. (2018):
  Hierarchical Bayesian piecewise regression model with partially edge-wise coupled parameters. Submitted to Journal of Computational and Graphical Statistics (chapter 4).

- Shafiee Kamalabad, M. and Grzegorczyk, M. (2018):
  Improving nonhomogeneous dynamic Bayesian networks with sequentially

coupled parameters. Statistica Neerlandica, 72 (3), 281-305 (chapter 3, see [58]).

- Shafiee Kamalabad, M. and Grzegorczyk, M. (2018):
  Non-homogeneous dynamic Bayesian networks with edge-wise coupled parameters. Proceedings of the International Workshop on Statistical Modelling, vol. 1, 270-275, Bristol, England (chapter 4, see [57]).

- Shafiee Kamalabad, M. and Grzegorczyk, M. (2018):
  A new partially Coupled Piece-Wise linear Regression Model for statistical network Structure Inference. Proceedings of the International Computational Intelligence methods for Bioinformatics and Biostatistics, page 30, Caparica, Portugal (chapter 2, see [56]).

- Shafiee Kamalabad, M. and Grzegorczyk, M. (2017):
  A sequentially coupled non-homogeneous dynamic Bayesian network model with segment-specific coupling strengths. Proceedings of the International Workshop on Statistical Modelling, vol. 1, 173-178, Groningen, Netherlands (chapter 3, see [55]).

- Shafiee Kamalabad, M. and Grzegorczyk, M. (2016): A non-homogeneous dynamic Bayesian network model with partially sequentially coupled network parameters. Proceedings of the International Workshop on Statistical Modelling, vol. 1, 139-144, Rennes, France (chapter 2, see [54]).

- Grzegorczyk, M. and Shafiee Kamalabad, M. (2016):
  Comparative evaluation of various frequentist and Bayesian non-homogeneous Poisson counting models. Computational Statistics, 32 (1), 1-33. (chapter 6, see [28]).

Chapter 2

Partially sequentially segmentwise coupled NH-DBNs

A common Bayesian approach is to employ a multiple changepoint process to divide the time series into disjunct segments with segment-specific network interaction parameters and to model the data in each segment by linear Bayesian regression models. The conventional uncoupled models infer the network interaction parameters for each segment separately. There is no information-sharing among segments. The uncoupled models are very flexible, but for short time series they can be subject to inflated inference uncertainties. It was therefore proposed to couple the network interaction parameters sequentially among segments. The key idea is to enforce the parameters of any segment to stay similar to those of the previous by using the posterior expectation of the network parameters as prior expectation for the consecutive segment. Node-specific coupling parameters regulate the variance of the parameter priors and so the strength of coupling. However, the proposed models are based on coupling mechanisms which can become disadvantageous: First, the models enforce coupling for all segments without any option to uncouple, and second, they couple all pairs of neighbouring segments with the same coupling strength. In this chapter we propose two improved hierarchical Bayesian models to fix those deficits. The partially segment-wise coupled model infers for each segment whether it is coupled to (or uncoupled from) the previous segment. The generalized coupled model introduces segment-specific coupling strengths, so as to allow for a greater model flexibility.

The partially segment-wise coupled model (M3) is a consensus model between the uncoupled model and the fully coupled model. There is a discrete binary variable δ_h for each segment h indicating whether segment h is coupled to the previous segment ($\delta_h = 1$) or uncoupled from the previous segment ($\delta_h = 0$).

Along with the network structure, the changepoints, the segment-specific network interaction parameters and the values of those indicator variables are inferred from the data. The new partially coupled M3 model has the original models as limiting cases: If it couples all segments, it effectively becomes the fully coupled model. If it uncouples all segments, it effectively becomes the conventional uncoupled model. The second model, which we propose here, is the generalised (fully) coupled model, which we refer to as the M4 model. The M4 model generalises the fully coupled model by introducing segment-specific coupling strength hyperparameters. Alternatively, the M4 model can also be thought of as a continuous version of the partially segment-wise coupled model (M3). The M4 model replaces the segment-specific binary indicator variables $\delta_h \in \{0, 1\}$ (uncoupled vs. coupled) by continuous coupling hyperparameters $\lambda_h \in \mathbb{R}^+$. For each pair of neighbouring segments $(h-1, h)$ there is then a segment-specific continuous coupling strength hyperparameter λ_h.

The work presented in this chapter, is still work in progress. Some parts of this chapter have also been appeared in proceedings of the International conference on Computational Intelligence methods for Bioinformatics and Biostatistics (2018)(see [56]) and proceedings of the International Workshop on Statistical Modelling (2016)(see [54]).

2.1 Methods

2.1.1 Learning dynamic networks with time-varying parameters

Consider a network domain with N random variables $Z_1, \ldots, Z_N$ being the nodes. Let $\mathbf{D}$ denote a data matrix whose N rows correspond to the variables and whose $T+1$ columns correspond to equidistant time points $t = 1, \ldots, T+1$. The element in the i-th row and t-th column, $\mathbf{D}_{i,t}$, is the value of Z_i at time point t. Temporal data are usually modelled with dynamic Bayesian networks, where all interactions are subject to a time lag: An edge $Z_i \to Z_j$ indicates that $\mathbf{D}_{j,t+1}$ (Z_j at $t+1$) depends on $\mathbf{D}_{i,t}$ (Z_i at t). Z_i is then called a parent (node) of Z_j.

Because of the time lag, there is no acyclicity constraint in dynamic Bayesian networks, and the parent nodes of each node Z_j can be learned separately. A common approach is to use a regression model where $Y := Z_j$ is the response and the other variables $\{Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_N\} =: \{X_1, \ldots, X_n\}$ are the $n := N-1$ covariates. Each data point $\mathcal{D}_t$ $(t \in \{1, \ldots, T\})$ contains a response value $Y = \mathbf{D}_{j,t+1}$ and the shifted covariate values: $X_1 = \mathbf{D}_{1,t}, \ldots, X_{j-1} = \mathbf{D}_{j-1,t}, X_j = \mathbf{D}_{j+1,t}, \ldots, X_n = \mathbf{D}_{N,t}$, where $n = N-1$. Having a covariate set $\boldsymbol{\pi}_j$ for each Z_j, a network can be built by merging the covariate sets: $\mathcal{G} := \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N\}$. There is the edge $Z_i \to Z_j$ if and only if $X_i \in \boldsymbol{\pi}_j$. As the same regression model is used for each Z_j separately, we describe the models (M1-M4) using the general terminology: Y is the response and $X_1, \ldots, X_n$ are the covariates.

To allow for time-dependent regression coefficients, a piece-wise linear regression model can be used. A set of changepoints $\boldsymbol{\tau} := \{\tau_1, \dots, \tau_{H-1}\}$ with $1 \leq \tau_h < T$ divides the data points $\mathcal{D}_1, \dots, \mathcal{D}_T$ into disjunct segments $h = 1, \dots, H$ covering $T_1, \dots, T_H$ consecutive data points, where $\sum T_h = T$. Data point $\mathcal{D}_t$ ($1 \leq t \leq T$) belongs to segment h if $\tau_{h-1} < t \leq \tau_h$, where $\tau_0 := 1$ and $\tau_H := T$.
We assume all covariate sets $\boldsymbol{\pi} \subset \{X_1, \dots, X_n\}$ with up to $\mathcal{F} = 3$ covariates to be equally likely a priori, $p(\boldsymbol{\pi}) = c$, while parent sets with more than $\mathcal{F}$ covariates get a zero prior probability ('fan-in restriction').[1] Further we assume that the distance between changepoints is geometrically distributed with parameter $p \in (0, 1)$, so that

$$p(\boldsymbol{\tau}) = \left(\prod_{h=1}^{H-1} (1-p)^{\tau_h - \tau_{h-1} - 1} \cdot p \right) \cdot (1-p)^{\tau_H - \tau_{H-1} - 1} = (1-p)^{(T-1)-(H-1)} \cdot p^{H-1}$$

With $\mathbf{y} = \mathbf{y}_{\boldsymbol{\tau}} := \{\mathbf{y}_1, \dots, \mathbf{y}_H\}$ being the set of segment-specific response vectors, implied by $\boldsymbol{\tau}$, the posterior distribution takes the form:

$$p(\boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\theta} | \mathbf{y}) \propto p(\boldsymbol{\pi}) \cdot p(\boldsymbol{\tau}) \cdot p(\boldsymbol{\theta} | \boldsymbol{\pi}, \boldsymbol{\tau}) \cdot p(\mathbf{y} | \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\theta}) \tag{2.1}$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\pi}, \boldsymbol{\tau})$ denotes the set of all model parameters, including the segment-specific parameters and those parameters which are shared among segments.
In subsections 2.1.3 to 2.1.6 we assume $\boldsymbol{\pi} \subset \{X_1, \dots, X_n\}$ and the segmentation $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_H\}$ induced by $\boldsymbol{\tau}$ to be fixed, and we do not make $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ explicit anymore. Without loss of generality, we assume further that $\boldsymbol{\pi}$ contains the first k covariates: $\boldsymbol{\pi} := \{X_1, \dots, X_k\}$. Focusing only on the model-parameters $\boldsymbol{\theta}$, Equation (2.1) reduces to:

$$p(\boldsymbol{\theta} | \mathbf{y}) \propto p(\boldsymbol{\theta}) \cdot p(\mathbf{y} | \boldsymbol{\theta})$$

How to infer the covariate set $\boldsymbol{\pi}$ and changepoint set $\boldsymbol{\tau}$ from the data is subsequently described in Subsection 2.1.7.

2.1.2 A generic Bayesian piece-wise linear regression model

Consider a Bayesian regression model where Y is the response and $X_1, \dots, X_k$ are the covariates. We assume that T data points $\mathcal{D}_1, \dots, \mathcal{D}_T$ have been measured at equidistant time points and that the data can be subdivided into disjunct segments $h \in \{1, \dots, H\}$, where segment h contains T_h data points and has the segment-specific regression coefficient vector $\boldsymbol{\beta}_h$. Let $\mathbf{y}_h$ be the response vector and $\mathbf{X}_h$ be the design matrix for segment h, where each $\mathbf{X}_h$ includes a first column of 1's for the intercept. For each segment $h = 1, \dots, H$ we assume a Gaussian likelihood:

$$\mathbf{y}_h | (\boldsymbol{\beta}_h, \sigma^2) \sim \mathcal{N}(\mathbf{X}_h \boldsymbol{\beta}_h, \sigma^2 \mathbf{I}) \tag{2.2}$$

---

[1] The fan-in restriction is biologically motivated, as it is known that genes are rarely regulated by more than 2-3 other regulator genes [31].

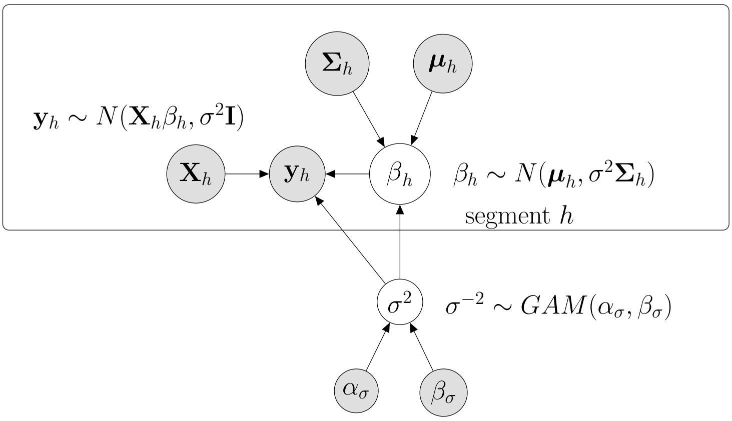

$\mathbf{\Sigma}_h$
$\boldsymbol{\mu}_h$
$\mathbf{y}_h \sim N(\mathbf{X}_h\beta_h, \sigma^2\mathbf{I})$
$\mathbf{X}_h$
$\mathbf{y}_h$
β_h
$\beta_h \sim N(\boldsymbol{\mu}_h, \sigma^2\mathbf{\Sigma}_h)$
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 2.1: Graphical representation of the generic model. Parameters that have to be inferred are represented by white circles. The data and the fixed hyperparameters are represented by grey circles. Circles within the plate are specific for segment h.

where $\mathbf{I}$ is the identity matrix, and σ^2 is the noise variance parameter, which is shared among segments. We impose an inverse Gamma prior on σ^2, $\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$, and we assume that the $\boldsymbol{\beta}_h$'s have Gaussian prior distributions:

$$\boldsymbol{\beta}_h|(\boldsymbol{\mu}_h, \mathbf{\Sigma}_h, \sigma^2) \sim \mathcal{N}(\boldsymbol{\mu}_h, \sigma^2\mathbf{\Sigma}_h) \tag{2.3}$$

Re-using the parameter σ^2 in Equation (2.3), yields a fully-conjugate prior in both $\boldsymbol{\beta}_h$ and σ^2 (see, e.g., Sections 3.3 and 3.4 in [19]). Figure 2.1 shows a graphical model representation of this generic model. For notational convenience we define:

$$\boldsymbol{\theta} := \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_H; \mathbf{\Sigma}_1, \dots, \mathbf{\Sigma}_H\}$$

The full conditional distribution of $\boldsymbol{\beta}_h$ is (cf. Section 3.3 in [5]):

$$\boldsymbol{\beta}_h|(\mathbf{y}_h, \sigma^2, \boldsymbol{\theta}) \sim \mathcal{N}([\mathbf{\Sigma}_h^{-1} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h]^{-1}(\mathbf{\Sigma}_h^{-1}\boldsymbol{\mu}_h + \mathbf{X}_h^\mathsf{T}\mathbf{y}_h), \sigma^2(\mathbf{\Sigma}_h^{-1} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h)^{-1}) \tag{2.4}$$

and the segment-specific marginal likelihoods with $\boldsymbol{\beta}_h$ integrated out are:

$$\mathbf{y}_h|(\sigma^2, \boldsymbol{\theta}) \sim \mathcal{N}(\mathbf{X}_h\boldsymbol{\mu}_h, \sigma^2\mathbf{C}_h(\boldsymbol{\theta})) \tag{2.5}$$

where $\mathbf{C}_h(\boldsymbol{\theta}) := \mathbf{I} + \mathbf{X}_h\mathbf{\Sigma}_h\mathbf{X}_h^\mathsf{T}$ (cf. Section 3.3 in [5]). From Equation (2.5) we get:

$$p(\sigma^2|\mathbf{y}, \boldsymbol{\theta}) \propto p(\sigma^2) \cdot \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2, \boldsymbol{\theta}) = (\sigma^{-2})^{a_\sigma + \frac{1}{2}\cdot T - 1} e^{-\sigma^{-2}\left(b_\sigma + \frac{1}{2}\cdot\Delta^2(\boldsymbol{\theta})\right)}$$

where $\mathbf{y} := \{\mathbf{y}_1, \dots, \mathbf{y}_H\}$ and $\Delta^2(\boldsymbol{\theta}) := \sum_{h=1}^{H}(\mathbf{y}_h - \mathbf{X}_h\boldsymbol{\mu}_h)^\mathsf{T}\mathbf{C}_h(\boldsymbol{\theta})^{-1}(\mathbf{y}_h - \mathbf{X}_h\boldsymbol{\mu}_h)$. The shape of $p(\sigma^2|\mathbf{y}, \boldsymbol{\theta})$ implies:

$$\sigma^{-2}|(\mathbf{y}, \boldsymbol{\theta}) \sim GAM\left(\alpha_\sigma + \frac{1}{2}\cdot T, \beta_\sigma + \frac{1}{2}\cdot\Delta^2(\boldsymbol{\theta})\right) \tag{2.6}$$

For the marginal likelihood, with $\boldsymbol{\beta}_h$ ($h = 1, \ldots, H$) and σ^2 integrated out, we apply the rule from Section 2.3.7 in [5]:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{\Gamma(\frac{T}{2} + a_\sigma)}{\Gamma(a_\sigma)} \cdot \frac{\pi^{-T/2} \cdot (2b_\sigma)^{a_\sigma}}{\left(\prod_{h=1}^{H} \det(\mathbf{C}_h(\boldsymbol{\theta}))\right)^{1/2}} \cdot \left(2b_\sigma + \Delta^2(\boldsymbol{\theta})\right)^{-(\frac{T}{2}+a_\sigma)} \tag{2.7}$$

When all parameters in $\boldsymbol{\theta}$ are fixed, the marginal likelihood of the piecewise linear regression model can be computed in closed form. In real-world applications there is normally no prior knowledge about the hyperparameters in $\boldsymbol{\theta}$.

In typical models the (hyper-)hyperparameters in $\boldsymbol{\theta}$ do not have all degrees of freedom, but depend on some free hyperparameters with their own hyperprior distributions. From now on, we will only include the free hyperparameters in $\boldsymbol{\theta}$. In the following subsections we describe four concrete model instantiations: the uncoupled model (M1), the fully coupled model (M2), the partially segment-wise coupled model (M3) and the generalised coupled model (M4), where the latter two are proposed in this chapter.

2.1.3 Model M1: The uncoupled model

A standard approach, akin to the models of [38] and [14], is to set $\boldsymbol{\mu}_h = \mathbf{0}$ and to assume that the matrices $\boldsymbol{\Sigma}_h$ are diagonal matrices $\boldsymbol{\Sigma}_h = \lambda_u \mathbf{I}$, where the parameter $\lambda_u \in \mathbf{R}^+$ is shared among segments and assumed to be inverse Gamma distributed, $\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u)$. Figure 2.2 shows the uncoupled model (M1) graphically. Using the notation of the generic model, we have:

$$\boldsymbol{\theta} = \{\lambda_u\}, \quad \mathbf{C}_h(\lambda_u) = \mathbf{I} + \lambda_u \mathbf{X}_h \mathbf{X}_h^{\mathsf{T}}, \quad \Delta^2(\lambda_u) := \sum_{h=1}^{H} \mathbf{y}_h^{\mathsf{T}} \mathbf{C}_h(\lambda_u)^{-1} \mathbf{y}_h \tag{2.8}$$

For the posterior distribution of the uncoupled model we have:

$$p(\boldsymbol{\beta}, \sigma^2, \lambda_u|\mathbf{y}) \propto p(\sigma^2) \cdot p(\lambda_u) \cdot \prod_{h=1}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_u) \cdot \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2, \boldsymbol{\beta}_h) \tag{2.9}$$

where $\boldsymbol{\beta} := \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_H\}$. From Equation (2.9) it follows for the full conditional distribution of λ_u:

$$\begin{aligned} p(\lambda_u|\mathbf{y}, \boldsymbol{\beta}, \sigma^2) &\propto p(\lambda_u) \cdot \prod_{h=1}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_u) \\ &\propto (\lambda_u^{-1})^{a_u + \frac{H\cdot(k+1)}{2}} \cdot \exp\{-\lambda_u^{-1}(b_u + \frac{1}{2}\sigma^{-2} \sum_{h=1}^{H} \boldsymbol{\beta}_h^{\mathsf{T}} \boldsymbol{\beta}_h)\} \end{aligned}$$

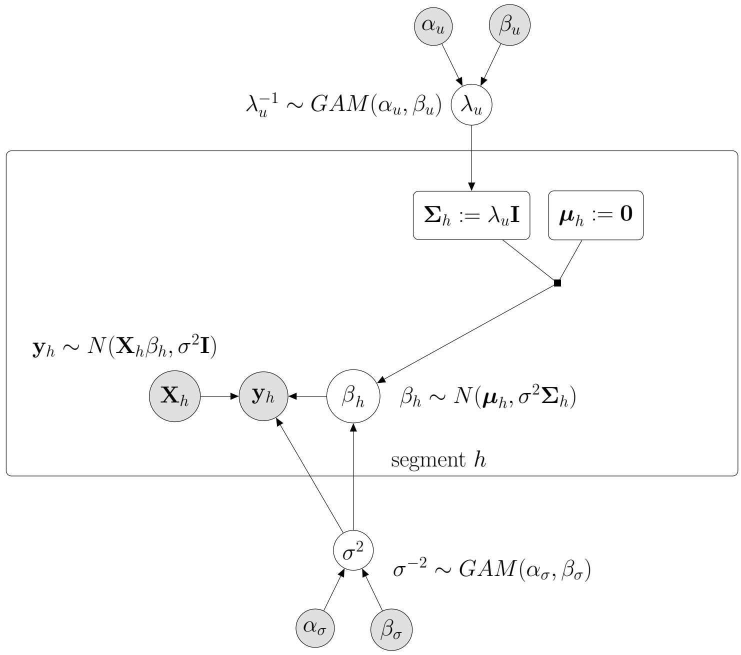

α_u
β_u
$\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u)$
λ_u
$\mathbf{\Sigma}_h := \lambda_u \mathbf{I}$
$\boldsymbol{\mu}_h := \mathbf{0}$
$\mathbf{y}_h \sim N(\mathbf{X}_h \beta_h, \sigma^2 \mathbf{I})$
$\mathbf{X}_h$
$\mathbf{y}_h$
β_h
$\beta_h \sim N(\boldsymbol{\mu}_h, \sigma^2 \mathbf{\Sigma}_h)$
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 2.2: Graphical representation of the uncoupled model (M1). Parameters that have to be inferred are represented by white circles. The data and the fixed hyperparameters are represented by grey circles. The two rectangles indicate definitions, which deterministically depend on the parent nodes. Circles and definitions within the plate are segment-specific.

and the shape of the latter density implies:

$$\lambda_u^{-1} | (\mathbf{y}, \boldsymbol{\beta}, \sigma^2) \sim GAM \left(\alpha_u + \frac{H \cdot (k+1)}{2}, \beta_u + \frac{1}{2} \sigma^{-2} \sum_{h=1}^{H} \boldsymbol{\beta}_h^{\mathsf{T}} \boldsymbol{\beta}_h \right) \tag{2.10}$$

Since the full conditional distribution of λ_u depends on σ^2 and $\boldsymbol{\beta}$, those parameters have to be sampled first. From Equation (2.6) an instantiation of σ^2 can be sampled via a collapsed Gibbs-sampling step, with the $\boldsymbol{\beta}_h$'s being integrated out. Subsequently, given σ^2, Equation (2.4) can be used to sample the $\boldsymbol{\beta}_h$'s. Finally, for each λ_u sampled from Equation (2.10) the marginal likelihood, $p(\mathbf{y}|\lambda_u)$, can be computed by plugging in the expressions from Equation (2.8) into Equation (2.7).

2.1.4 Model M2: The fully coupled model

The (fully) coupled model, proposed by [24], uses the posterior expectation of $\boldsymbol{\beta}_{h-1}$ as prior expectation for $\boldsymbol{\beta}_h$. Only the first segment $h = 1$ has an uninformative prior:

$$\boldsymbol{\beta}_h \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma^2\lambda_u\mathbf{I}) & \text{if } h = 1 \\ \mathcal{N}(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_c\mathbf{I}) & \text{if } h > 1 \end{cases} \tag{2.11}$$

where $\tilde{\boldsymbol{\beta}}_{h-1}$ is the posterior expectation of $\boldsymbol{\beta}_{h-1}$ (cf. Equation (2.4)):[2]

$$\tilde{\boldsymbol{\beta}}_{h-1} := \begin{cases} [\boldsymbol{\Sigma}_1^{-1} + \mathbf{X}_1^{\mathsf{T}}\mathbf{X}_1]^{-1}(\mathbf{X}_1^{\mathsf{T}}\mathbf{y}_1) & \text{if } h = 2 \\ [\boldsymbol{\Sigma}_{h-1}^{-1} + \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{X}_{h-1}]^{-1}(\lambda_c^{-1}\tilde{\boldsymbol{\beta}}_{h-2} + \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{y}_{h-1}) & \text{if } h > 2 \end{cases}$$

The parameter λ_c has been called the *'coupling parameter'* onto which also an inverse Gamma distribution can be imposed, $\lambda_c^{-1} \sim GAM(\alpha_c, \beta_c)$. Using the notation from the generic model (see Figure 2.1), we note that Equation (2.11) corresponds to:

$$\boldsymbol{\mu}_h = \begin{cases} \mathbf{0} & \text{if } h = 1 \\ \tilde{\boldsymbol{\beta}}_{h-1} & \text{if } h > 1 \end{cases}, \quad \boldsymbol{\Sigma}_h = \begin{cases} \lambda_u\mathbf{I} & \text{if } h = 1 \\ \lambda_c\mathbf{I} & \text{if } h > 1 \end{cases}, \quad \mathbf{C}_h(\boldsymbol{\theta}) = \begin{cases} \mathbf{I} + \lambda_u\mathbf{X}_h\mathbf{X}_h^{\mathsf{T}} & \text{if } h = 1 \\ \mathbf{I} + \lambda_c\mathbf{X}_h\mathbf{X}_h^{\mathsf{T}} & \text{if } h > 1 \end{cases}$$

$$\boldsymbol{\theta} = \{\lambda_u, \lambda_c\}, \text{ and } \Delta^2(\boldsymbol{\theta}) = \sum_{h=1}^{H}(\mathbf{y}_h - \mathbf{X}_h\tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}}\mathbf{C}_h(\boldsymbol{\theta})^{-1}(\mathbf{y}_h - \mathbf{X}_h\tilde{\boldsymbol{\beta}}_{h-1})$$

where $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$, $\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u)$ and $\lambda_c^{-1} \sim GAM(\alpha_c, \beta_c)$. As $\tilde{\boldsymbol{\beta}}_{h-1}$ is treated like a fixed hyperparameter when used as input for segment h, we exclude the parameters $\tilde{\boldsymbol{\beta}}_1, \ldots, \tilde{\boldsymbol{\beta}}_{H-1}$ from $\boldsymbol{\theta}$. Figure 2.3 shows a graphical representation of the coupled model.

For the posterior we have:

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2, \lambda_u, \lambda_c|\mathbf{y}) \quad \propto \quad & p(\sigma^2) \cdot p(\lambda_u) \cdot p(\lambda_c) \cdot p(\boldsymbol{\beta}_1|\sigma^2, \lambda_u) \\ & \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_c) \cdot \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2, \boldsymbol{\beta}_h) \end{aligned} \tag{2.12}$$

In analogy to the derivations in the previous subsection one can derive (cf. [24]):

$$\lambda_u^{-1}|(\mathbf{y}, \boldsymbol{\beta}, \sigma^2, \lambda_c) \quad \sim \quad GAM\left(\alpha_u + \frac{1 \cdot (k+1)}{2}, \beta_u + \frac{1}{2}\sigma^{-2}D_u^2\right) \tag{2.13}$$

$$\lambda_c^{-1}|(\mathbf{y}, \boldsymbol{\beta}, \sigma^2, \lambda_u) \quad \sim \quad GAM\left(\alpha_c + \frac{(H-1) \cdot (k+1)}{2}, \beta_c + \frac{1}{2}\sigma^{-2}D_c^2\right) \tag{2.14}$$

where $D_u^2 := \boldsymbol{\beta}_1^{\mathsf{T}}\boldsymbol{\beta}_1$ and $D_c^2 := \sum_{h=2}^{H}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})$.
For each $\boldsymbol{\theta} = \{\lambda_u, \lambda_c\}$ the marginal likelihood, $p(\mathbf{y}|\lambda_u, \lambda_c)$, can be computed by plugging the expressions $\mathbf{C}_h(\boldsymbol{\theta})$ and $\Delta^2(\boldsymbol{\theta})$ into Equation (2.7).

2.1.5 Model M3: The partially segment-wise coupled model, proposed here

The first new model, which we propose here, allows each segment $h > 1$ to uncouple from the previous one $h - 1$. We use an uninformative prior for segment $h = 1$, and for all

---

[2] Note: $\tilde{\boldsymbol{\beta}}_{h-1}$ in Equation (2.4) is the posterior expectation of $\boldsymbol{\beta}_{h-1}$ given σ^2 and $\boldsymbol{\theta} = \{\lambda_u, \lambda_c\}$.

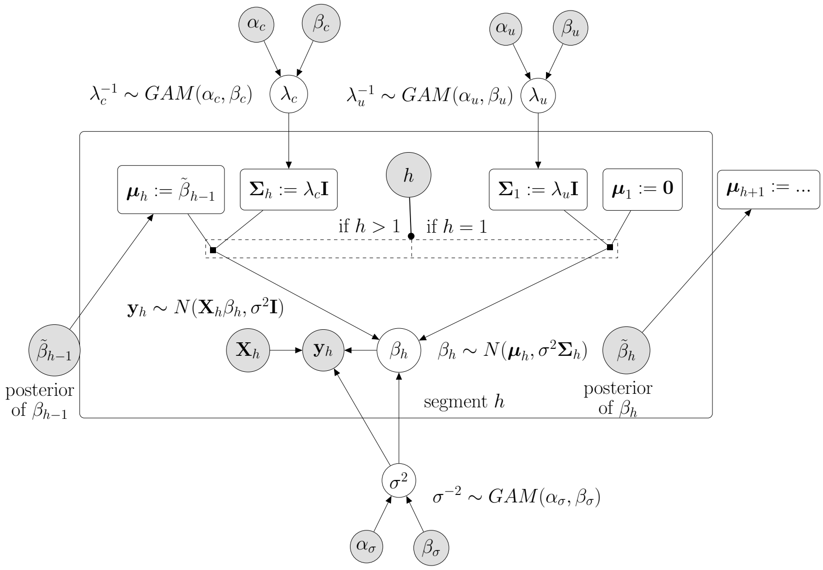

α_c
β_c
α_u
β_u
$\lambda_c^{-1} \sim GAM(\alpha_c, \beta_c)$
λ_c
$\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u)$
λ_u
$\boldsymbol{\mu}_h := \tilde{\beta}_{h-1}$
$\mathbf{\Sigma}_h := \lambda_c \mathbf{I}$
h
$\mathbf{\Sigma}_1 := \lambda_u \mathbf{I}$
$\boldsymbol{\mu}_1 := \mathbf{0}$
$\boldsymbol{\mu}_{h+1} := ...$
if $h > 1$
if $h = 1$
$\mathbf{y}_h \sim N(\mathbf{X}_h \beta_h, \sigma^2 \mathbf{I})$
$\tilde{\beta}_{h-1}$
$\mathbf{X}_h$
$\mathbf{y}_h$
β_h
$\beta_h \sim N(\boldsymbol{\mu}_h, \sigma^2 \mathbf{\Sigma}_h)$
$\tilde{\beta}_h$
posterior of β_{h-1}
segment h
posterior of β_h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 2.3: Graphical representation of the fully coupled model (M2). See caption of Figure 2.2 for the terminology. Each posterior expectation $\tilde{\boldsymbol{\beta}}_h$ is treated like a fixed parameter vector when used as input for segment $h+1$. The prior for $\boldsymbol{\beta}_h$ depends on h.

segments $h > 1$ we introduce a binary variable δ_h which indicates whether segment h is coupled to ($\delta_h = 1$) or uncoupled from ($\delta_h = 0$) the preceding segment $h-1$:

$$\boldsymbol{\beta}_h \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma^2 \lambda_u \mathbf{I}) & \text{if } h = 1 \\ \mathcal{N}(\delta_h \cdot \tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2 \lambda_c^{\delta_h} \lambda_u^{1-\delta_h} \mathbf{I}) & \text{if } h > 1 \end{cases} \tag{2.15}$$

where $\tilde{\boldsymbol{\beta}}_{h-1}$ is again the posterior expectation of $\boldsymbol{\beta}_{h-1}$. The new priors from Equation (2.15) yield for $h \geq 2$ the following posterior expectations (cf. Equation (2.4)):

$$\tilde{\boldsymbol{\beta}}_{h-1} = \left(\lambda_c^{-\delta_{h-1}} \lambda_u^{-(1-\delta_{h-1})} \mathbf{I} + \mathbf{X}_{h-1}^{\mathsf{T}} \mathbf{X}_{h-1} \right)^{-1} \left(\delta_{h-1} \lambda_c^{-1} \tilde{\boldsymbol{\beta}}_{h-2} + \mathbf{X}_{h-1}^{\mathsf{T}} \mathbf{y}_{h-1} \right)$$

With $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$, $\delta_1 := 0$, we have in the generic model notation:

$$\boldsymbol{\mu}_h = \delta_h \tilde{\boldsymbol{\beta}}_{h-1}, \quad \mathbf{\Sigma}_h = \lambda_c^{\delta_h} \lambda_u^{1-\delta_h} \mathbf{I}, \quad \boldsymbol{\theta} = \{\lambda_u, \lambda_c, \{\delta_h\}_{h \geq 2}\}, \quad \mathbf{C}_h(\boldsymbol{\theta}) = \mathbf{I} + \lambda_c^{\delta_h} \lambda_u^{1-\delta_h} \mathbf{X}_h \mathbf{X}_h^{\mathsf{T}}$$

We assume the binary variables $\delta_2, \ldots, \delta_H$ to be Bernoulli distributed, $\delta_h \sim BER(\mathrm{p})$, with $\mathrm{p} \in [0,1]$ having a Beta distribution, $\mathrm{p} \sim BETA(a, b)$.

- $\delta_h = 0$ ($h \geq 2$) gives model M1 with $P(\boldsymbol{\beta}_h) = \mathcal{N}(\mathbf{0}, \lambda_u \sigma^2 \mathbf{I})$ for all h
- $\delta_h = 1$ ($h \geq 2$) gives model M2 with $P(\boldsymbol{\beta}_h) = \mathcal{N}(\tilde{\boldsymbol{\beta}}_{h-1}, \lambda_c \sigma^2 \mathbf{I})$ for $h \geq 2$.
- The new partially segment-wise coupled model infers the variables δ_h ($h \geq 2$) from the data, so as to find the best trade-off between model M1 and model M2.

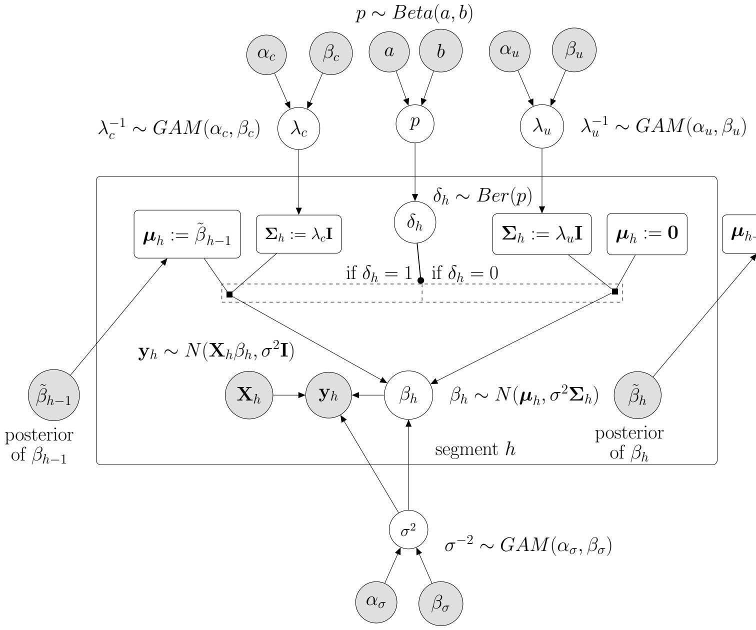

$p \sim Beta(a,b)$
α_c
β_c
a
b
α_u
β_u
$\lambda_c^{-1} \sim GAM(\alpha_c,\beta_c)$
λ_c
p
λ_u
$\lambda_u^{-1} \sim GAM(\alpha_u,\beta_u)$
$\delta_h \sim Ber(p)$
δ_h
$\boldsymbol{\mu}_h := \tilde{\beta}_{h-1}$
$\mathbf{\Sigma}_h := \lambda_c\mathbf{I}$
$\mathbf{\Sigma}_h := \lambda_u\mathbf{I}$
$\boldsymbol{\mu}_h := \mathbf{0}$
$\boldsymbol{\mu}_{h+1} := ...$
if $\delta_h = 1$
if $\delta_h = 0$
$\mathbf{y}_h \sim N(\mathbf{X}_h\beta_h, \sigma^2\mathbf{I})$
$\tilde{\beta}_{h-1}$
posterior of β_{h-1}
$\mathbf{X}_h$
$\mathbf{y}_h$
β_h
$\beta_h \sim N(\boldsymbol{\mu}_h, \sigma^2\mathbf{\Sigma}_h)$
$\tilde{\beta}_h$
posterior of β_h
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma,\beta_\sigma)$
α_σ
β_σ

Figure 2.4: Graphical representation of the partially coupled model (M3), proposed here. See caption of Figure 2.3 for the terminology. For each segment h it is inferred from the data whether the prior for $\boldsymbol{\beta}_h$ should be coupled to ($\delta_h = 1$) or uncoupled from ($\delta_h = 0$) the preceding segment $h-1$.

A graphical model presentation of the partially coupled model is shown in Figure 2.4. For $\delta_h \sim BER(\mathrm{p})$ with $\mathrm{p} \sim BETA(a,b)$ the joint marginal density of $\{\delta_h\}_{h\geq 2}$ is:

$$p(\{\delta_h\}_{h\geq 2}) = \int p(\mathrm{p}) \prod_{h=2}^{H} p(\delta_h|\mathrm{p})\, d\mathrm{p} = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \cdot \frac{\Gamma(a+\sum\limits_{h=2}^{H}\delta_h)\Gamma(b+\sum\limits_{h=2}^{H}(1-\delta_h))}{\Gamma(a+b+(H-1))}$$

For the posterior distribution of the partially segment-wise coupled model we get:

$$\begin{aligned} p(\boldsymbol{\beta},\sigma^2,\lambda_u,\lambda_c,\{\delta_h\}_{h\geq 2}|\mathbf{y}) \quad &\propto \quad p(\sigma^2)\cdot p(\lambda_u)\cdot p(\lambda_c)\cdot p(\{\delta_h\}_{h\geq 2})\cdot p(\boldsymbol{\beta}_1|\sigma^2,\lambda_u) \\ &\cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2,\lambda_u,\lambda_c,\delta_h)\cdot \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2,\boldsymbol{\beta}_h) \end{aligned}$$

For the full conditional distributions of λ_u and λ_c we have:

$$\begin{aligned} p(\lambda_u|\mathbf{y},\boldsymbol{\beta},\sigma^2,\lambda_c,\{\delta_h\}_{h\geq 2}) \quad &\propto \quad p(\lambda_u)\cdot \prod_{h:\delta_h=0} p(\boldsymbol{\beta}_h|\sigma^2,\lambda_u) \\ p(\lambda_c|\mathbf{y},\boldsymbol{\beta},\sigma^2,\lambda_u,\{\delta_h\}_{h\geq 2}) \quad &\propto \quad p(\lambda_c)\cdot \prod_{h:\delta_h=1} p(\boldsymbol{\beta}_h|\sigma^2,\lambda_c) \end{aligned}$$

where $\delta_1 := 0$ fixed. And it follows from the shapes of the densities:

$$\begin{aligned}\lambda_u^{-1}|(\mathbf{y}, \boldsymbol{\beta}, \sigma^2, \lambda_c, \{\delta_h\}_{h\geq 2}) &\sim GAM\left(\alpha_u + \frac{H_u \cdot (k+1)}{2}, \beta_u + \frac{1}{2}\sigma^{-2}D_u^2\right)\\ \lambda_c^{-1}|(\mathbf{y}, \boldsymbol{\beta}, \sigma^2, \lambda_u, \{\delta_h\}_{h\geq 2}) &\sim GAM\left(\alpha_c + \frac{H_c \cdot (k+1)}{2}, \beta_c + \frac{1}{2}\sigma^{-2}D_c^2\right)\end{aligned}$$

where $H_c = \sum_h \delta_h$ is the number of coupled segments, $H_u = \sum_h (1-\delta_h)$ is the number of uncoupled segments, so that $H_c + H_u = H$, and

$$D_u^2 := \sum_{h:\delta_h=0} \boldsymbol{\beta}_h^{\mathsf{T}}\boldsymbol{\beta}_h, \quad D_c^2 := \sum_{h:\delta_h=1} (\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1}) \tag{2.16}$$

For each parameter instantiation $\boldsymbol{\theta} = \{\lambda_u, \lambda_c, \{\delta_h\}_{h\geq 2}\}$ the marginal likelihood, $p(\mathbf{y}|\boldsymbol{\theta})$, can be computed with Equation (2.7), where $\mathbf{C}_h(\boldsymbol{\theta})$ was defined above, and

$$\Delta^2(\boldsymbol{\theta}) = \sum_{h=1}^{H} (\mathbf{y}_h - \delta_h \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}} \left[\mathbf{I} + \lambda_c^{\delta_h}\lambda_u^{1-\delta_h}\mathbf{X}_h\mathbf{X}_h^{\mathsf{T}}\right]^{-1} (\mathbf{y}_h - \delta_h \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})$$

Moreover, we have for each binary variable δ_k ($k = 2, \dots, H$):

$$p(\delta_k = 1|\lambda_u, \lambda_c, \{\delta_h\}_{h\neq k}, \mathbf{y}) \propto p(\mathbf{y}|\lambda_u, \lambda_c, \{\delta_h\}_{h\neq k}, \delta_k = 1) \cdot p(\{\boldsymbol{\delta}_h\}_{h\neq k}, \delta_k = 1)$$

so that its full conditional distribution is:

$$\delta_k|(\lambda_u, \lambda_c, \{\delta_h\}_{h\neq k}, \mathbf{y}) \sim BER\left(\frac{p(\mathbf{y}|\lambda_u, \lambda_c, \{\delta_h\}_{h\neq k}, \delta_k = 1) \cdot p(\{\boldsymbol{\delta}_h\}_{h\neq k}, \delta_k = 1)}{\sum_{j=0}^{1} p(\mathbf{y}|\lambda_u, \lambda_c, \{\delta_h\}_{h\neq k}, \delta_k = j) \cdot p(\{\boldsymbol{\delta}_h\}_{h\neq k}, \delta_k = j)}\right)$$

Thus, each δ_k ($k > 1$) can be sampled with a collapsed Gibbs sampling step where $\{\boldsymbol{\beta}_h\}$, σ^2 and p have been integrated out.

2.1.6 Model M4: The generalised coupled model, proposed here

We generalise the fully coupled model M2 by introducing segment-specific coupling parameters λ_h for the segments $h = 2, \dots, H$. This yields:

$$\boldsymbol{\beta}_h \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma^2\lambda_u\mathbf{I}) & \text{if } h = 1\\ \mathcal{N}(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_h\mathbf{I}) & \text{if } h > 1\end{cases} \tag{2.17}$$

where $\tilde{\boldsymbol{\beta}}_{h-1}$ is again the posterior expectation of $\boldsymbol{\beta}_{h-1}$. For the parameters λ_h we assume that they are inverse Gamma distributed, $\lambda_h^{-1} \sim GAM(\alpha_c, \beta_c)$, with hyperparameters α_c and β_c. Figure 2.5 gives a graphical model representation. Recalling the generic notation and setting $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$ and $\lambda_1 := \lambda_u$, Equation (2.17) gives:

$$\boldsymbol{\mu}_h = \tilde{\boldsymbol{\beta}}_{h-1}, \quad \boldsymbol{\Sigma}_h = \lambda_h\mathbf{I}, \quad \mathbf{C}_h(\boldsymbol{\theta}) = \mathbf{I} + \lambda_h\mathbf{X}_h\mathbf{X}_h^{\mathsf{T}}, \quad \boldsymbol{\theta} = \{\lambda_u, \{\lambda_h\}_{h\geq 2}\},$$

$$\text{and } \Delta^2(\boldsymbol{\theta}) = \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h\tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}}\mathbf{C}_h(\boldsymbol{\theta})^{-1}(\mathbf{y}_h - \mathbf{X}_h\tilde{\boldsymbol{\beta}}_{h-1})$$

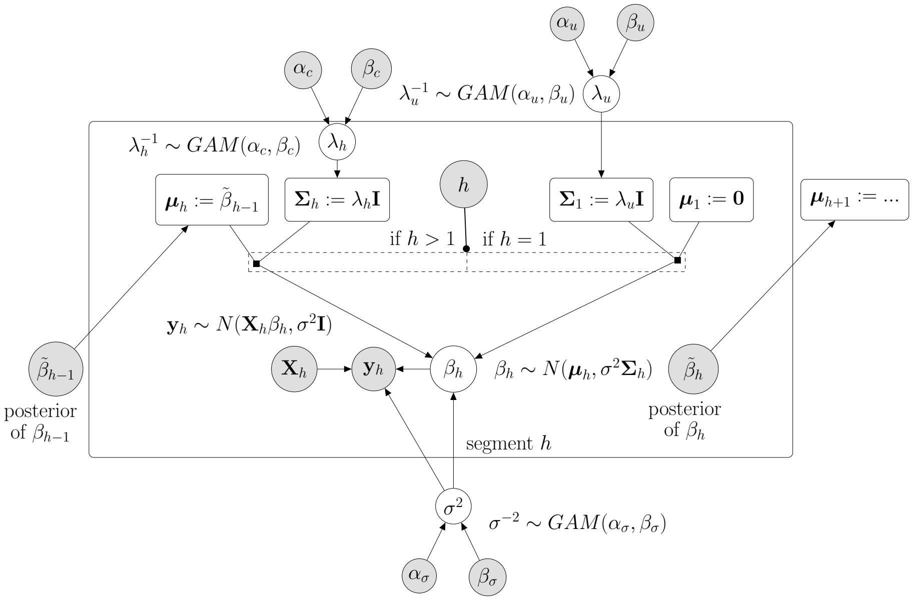

α_u
β_u
α_c
β_c
$\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u)$
λ_u
$\lambda_h^{-1} \sim GAM(\alpha_c, \beta_c)$
λ_h
$\boldsymbol{\mu}_h := \tilde{\beta}_{h-1}$
$\boldsymbol{\Sigma}_h := \lambda_h \mathbf{I}$
h
$\boldsymbol{\Sigma}_1 := \lambda_u \mathbf{I}$
$\boldsymbol{\mu}_1 := \mathbf{0}$
$\boldsymbol{\mu}_{h+1} := ...$
if $h > 1$
if $h = 1$
$\mathbf{y}_h \sim N(\mathbf{X}_h \beta_h, \sigma^2 \mathbf{I})$
$\tilde{\beta}_{h-1}$
$\mathbf{X}_h$
$\mathbf{y}_h$
β_h
$\beta_h \sim N(\boldsymbol{\mu}_h, \sigma^2 \boldsymbol{\Sigma}_h)$
$\tilde{\beta}_h$
posterior of β_{h-1}
posterior of β_h
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 2.5: Graphical representation of the generalised coupled model (M4), proposed here. See caption of Figures 2.2-2.3 for the terminology. Unlike the M2 model, this new model has segment-specific coupling parameters λ_h $(h > 1)$.

For the posterior we have:

$$\begin{aligned} p(\boldsymbol{\beta}, \sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}|\mathbf{y}) \quad &\propto \quad p(\sigma^2) \cdot p(\lambda_u) \cdot \left(\prod_{h=2}^{H} p(\lambda_h) \right) \\ &\cdot p(\boldsymbol{\beta}_1|\sigma^2, \lambda_u) \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_h) \cdot \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2, \boldsymbol{\beta}_h) \end{aligned} \tag{2.18}$$

Like for the coupled model M2, it can be derived for $k = 2, \dots, H$:

$$\begin{aligned} \lambda_k^{-1}|(\mathbf{y}, \boldsymbol{\beta}, \sigma^2, \lambda_u, \{\lambda_h\}_{h\neq k}) \quad &\sim \quad GAM\left(\alpha_c + \frac{(k+1)}{2}, \beta_c + \frac{1}{2}\sigma^{-2} D_k^2 \right) \\ \text{and} \quad \lambda_u^{-1}|(\mathbf{y}, \boldsymbol{\beta}, \sigma^2, \{\lambda_h\}_{h\geq 2}) \quad &\sim \quad GAM\left(\alpha_u + \frac{(k+1)}{2}, \beta_u + \frac{1}{2}\sigma^{-2} D_u^2 \right) \end{aligned}$$

where $D_u^2 := \boldsymbol{\beta}_1^{\mathsf{T}} \boldsymbol{\beta}_1$ and $D_k^2 := (\boldsymbol{\beta}_k - \tilde{\boldsymbol{\beta}}_{k-1})^{\mathsf{T}} (\boldsymbol{\beta}_k - \tilde{\boldsymbol{\beta}}_{k-1})$.
For each $\boldsymbol{\theta} = \{\lambda_u, \{\lambda_h\}_{h\geq 2}\}$ the marginal likelihood, $p(\mathbf{y}|\{\lambda_u, \{\lambda_h\}_{h\geq 2}\})$, can be computed with Equation (2.7); using the expressions $\mathbf{C}_h(\boldsymbol{\theta})$ and $\Delta^2(\boldsymbol{\theta})$ defined above.

2.1.7 Reversible Jump Markov Chain Monte Carlo inference

We use Reversible Jump Markov Chain Monte Carlo simulations [21] to generate posterior samples $\{\boldsymbol{\pi}^{(w)}, \boldsymbol{\tau}^{(w)}, \boldsymbol{\theta}^{(w)}\}_{w=1,\dots,W}$. In each iteration we re-sample the parameters in $\boldsymbol{\theta}$

from their full conditional distributions (Gibbs sampling), and we perform two Metropolis-Hastings moves; one on the covariate set $\boldsymbol{\pi}$ and one on the changepoint set $\boldsymbol{\tau}$. For the four models Equation (2.1) takes the form:[3]

$$p(\boldsymbol{\pi},\boldsymbol{\tau},\boldsymbol{\theta}|\mathbf{y}) \propto \begin{cases} p(\boldsymbol{\pi})\cdot p(\boldsymbol{\tau})\cdot p(\lambda_u)\cdot p(\mathbf{y}|\boldsymbol{\pi},\boldsymbol{\tau},\lambda_u) & \text{(M1)}\\ p(\boldsymbol{\pi})\cdot p(\boldsymbol{\tau})\cdot p(\lambda_u)\cdot p(\lambda_c)\cdot p(\mathbf{y}|\boldsymbol{\pi},\boldsymbol{\tau},\lambda_u,\lambda_c) & \text{(M2)}\\ p(\boldsymbol{\pi})\cdot p(\boldsymbol{\tau})\cdot p(\lambda_u)\cdot p(\lambda_c)\cdot p(\{\delta_h\}_{h\geq 2})\cdot p(\mathbf{y}|\boldsymbol{\pi},\boldsymbol{\tau},\lambda_u,\lambda_c,\{\delta_h\}_{h\geq 2}) & \text{(M3)}\\ p(\boldsymbol{\pi})\cdot p(\boldsymbol{\tau})\cdot p(\lambda_u)\cdot \left(\prod_{h=2}^{H} p(\lambda_h)\right)\cdot p(\mathbf{y}|\boldsymbol{\pi},\boldsymbol{\tau},\lambda_u,\{\lambda_h\}_{h\geq 2}) & \text{(M4)} \end{cases}$$

For the models M1-M2 the dimension of $\boldsymbol{\theta}$ does not depend on $\boldsymbol{\tau}$. For the new models M3-M4 the dimension of $\boldsymbol{\theta}$ depends on $\boldsymbol{\tau}$. Model M3 has a discrete parameter $\delta_h \in \{0,1\}$ for each segment $h > 1$. Model M4 has a continuous parameter $\lambda_h \in \mathbb{R}^+$ for each segment $h > 1$. That is, we have three standard cases of RJMCMC:

M1: $\boldsymbol{\theta} = \{\lambda_u\}$. All segment-specific parameters can be integrated out.

M2: $\boldsymbol{\theta} = \{\lambda_u, \lambda_c\}$. All segment-specific parameters can be integrated out.

M3: $\boldsymbol{\theta} = \{\lambda_u, \lambda_c, \{\delta_h\}_{h\geq 2}\}$. All segment-specific parameters can be integrated out, except for a set of discrete parameters $\{\delta_h\}_{h\geq 2}$ whose cardinality depends on H.

M4: $\boldsymbol{\theta} = \{\lambda_u, \{\lambda_h\}_{h\geq 2}\}$ All segment-specific parameters can be integrated out, except for a set of continuous parameters $\{\lambda_h\}_{h\geq 2}$ whose cardinality depends on H.

The model-specific full conditional distributions for the Gibbs sampling steps have already been provided in subsections 2.1.3 to 2.1.6. For sampling covariate sets $\boldsymbol{\pi}$ we implement 3 moves: the covariate 'removal (R)', 'addition (A)', and 'exchange (E)' move. Each move proposes to replace $\boldsymbol{\pi}$ by a new covariate set $\boldsymbol{\pi}^*$ having one covariate more (A) or less (R) or exchanged (E). When randomly selecting the move type and the involved covariate(s), we get for all models the acceptance probability:

$$A(\boldsymbol{\pi} \to \boldsymbol{\pi}^*) = \min\left\{1, \frac{p(\mathbf{y}|\boldsymbol{\pi}^*,\ldots)}{p(\mathbf{y}|\boldsymbol{\pi},\ldots)}\cdot\frac{p(\boldsymbol{\pi}^*)}{p(\boldsymbol{\pi})}\cdot HR_{\boldsymbol{\pi}}\right\}$$

with the Hastings Ratios: $HR_{\boldsymbol{\pi},R} = \dfrac{|\boldsymbol{\pi}|}{n-|\boldsymbol{\pi}^*|}$, $HR_{\boldsymbol{\pi},A} = \dfrac{n-|\boldsymbol{\pi}|}{|\boldsymbol{\pi}^*|}$, $HR_{\boldsymbol{\pi},E} = 1$

For sampling changepoint sets $\boldsymbol{\tau}$ we also implement 3 move types: the changepoint 'birth (B)', 'death (D)', and 're-allocation (R)' move. Each move proposes to replace $\boldsymbol{\tau}$ by a new changepoint set $\boldsymbol{\tau}^*$ having one changepoint added (B) or deleted (D) or re-allocated (R). When randomly selecting the move type, the involved changepoint and the new changepoint location, we get for the models M1 and M2:

$$A(\boldsymbol{\tau} \to \boldsymbol{\tau}^*) = \min\left\{1, \frac{p(\mathbf{y}|\boldsymbol{\tau}^*,\ldots)}{p(\mathbf{y}|\boldsymbol{\tau},\ldots)}\cdot\frac{p(\boldsymbol{\tau}^*)}{p(\boldsymbol{\tau})}\cdot HR_{\boldsymbol{\tau}}\right\}$$

with the Hasting Ratios: $HR_{\boldsymbol{\tau},B} = \dfrac{T-1-|\boldsymbol{\tau}^*|}{|\boldsymbol{\tau}|}$, $HR_{\boldsymbol{\tau},D} = \dfrac{|\boldsymbol{\tau}^*|}{T-1-|\boldsymbol{\tau}|}$, $HR_{\boldsymbol{\tau},R} = 1$

For the new models M3 and M4 the changepoint moves also affect the numbers of parameters in $\{\delta_h\}_{h\geq 2}$ and $\{\lambda_h\}_{h\geq 2}$, respectively. For segments that stay identical we keep the

---

[3] The likelihoods in the posterior distributions are marginalized over the regression coefficient vectors $\{\boldsymbol{\beta}_h\}$ and the variance σ^2. In M3, where $\delta_h \sim BER(\text{p})$, the parameter p is also integrated out.

parameters unchanged. For altering segments we re-sample the corresponding parameters. For M3 we flip coins to get candidates for the involved δ_h's. This yields:

$$A([\boldsymbol{\tau}, \{\delta_h\}] \rightarrow [\boldsymbol{\tau}^*, \{\delta_h\}^*]) = \min\left\{1, \frac{p(\mathbf{y}|\boldsymbol{\tau}^*, \{\delta_h\}^*, \ldots)}{p(\mathbf{y}|\boldsymbol{\tau}, \{\delta_h\}, \ldots)} \frac{p(\boldsymbol{\tau}^*)}{p(\boldsymbol{\tau})} \frac{p(\{\delta_h\}^*)}{p(\{\delta_h\})} \cdot HR_{\boldsymbol{\mathcal{T}}} \cdot c_{\boldsymbol{\mathcal{T}}}\right\}$$

where $c_{\boldsymbol{\mathcal{T}},B} = 1/2$ for birth, $c_{\boldsymbol{\mathcal{T}},D} = 2$ for death, and $c_{\boldsymbol{\mathcal{T}},R} = 1$ for re-allocation moves. For M4 we re-sample the involved λ_h's from their priors $p(\lambda_h)$. We obtain:

$$A([\boldsymbol{\tau}, \{\lambda_h\}] \rightarrow [\boldsymbol{\tau}^*, \{\lambda_h\}^*]) = \min\left\{1, \frac{p(\mathbf{y}|\boldsymbol{\tau}^*, \{\lambda_h\}^*, \ldots)}{p(\mathbf{y}|\boldsymbol{\tau}, \{\lambda_h\}, \ldots)} \cdot \frac{p(\boldsymbol{\tau}^*)}{p(\boldsymbol{\tau})} \cdot HR_{\boldsymbol{\mathcal{T}}}\right\}$$

since the new ratio in $HR_{\boldsymbol{\mathcal{T}}}$ cancels with the prior ratio $\frac{p(\{\lambda_h\}^*)}{p(\{\lambda_h\})}$.

2.1.8 Edge scores and areas under precision-recall curves (AUC)

For a network with N variables $Z_1, \ldots, Z_N$ we infer N separate regression models. For each Z_i we get a sample $\{\boldsymbol{\pi}^{(w)}, \boldsymbol{\tau}^{(w)}, \boldsymbol{\theta}^{(w)}\}_{w=1,\ldots,W}$ from the i-th posterior distribution. From the covariate sets we form a sample of graphs $G^{(w)} = \{\boldsymbol{\pi}_1^{(w)}, \ldots, \boldsymbol{\pi}_N^{(w)}\}_{w=1,\ldots,W}$. The marginal posterior probability that the network has the edge $Z_i \rightarrow Z_j$ is:

$$\hat{e}_{i,j} = \frac{1}{W}\sum_{w=1}^{W} I_{i\rightarrow j}(\mathcal{G}^{(w)}) \text{ where } I_{i\rightarrow j}(\mathcal{G}^{(w)}) = \begin{cases} 1 & \text{if } X_i \in \boldsymbol{\pi}_j^{(w)} \\ 0 & \text{if } X_i \notin \boldsymbol{\pi}_j^{(w)} \end{cases}$$

We refer to $\hat{e}_{i,j}$ as the 'score' of the edge $Z_i \rightarrow Z_j$.

If the true network is known and has M edges, we evaluate the network reconstruction accuracy as follows: For each threshold $\xi \in [0, 1]$ we extract the n_ξ edges whose scores $\hat{e}_{i,j}$ exceed ξ, and we count the number of true positives T_ξ among them. Plotting the precisions $P_\xi := T_\xi/n_\xi$ against the recalls $R_\xi := T_\xi/M$, gives the precision-recall curve. Precision-recall curves have advantages over the traditional Receiver-Operator-Characteristic (ROC) curves (for ROC curves see, e.g., [32]). The advantages were for example shown in [11]. In this thesis we refer to the area under the precision-recall curve as AUC ('area under curve') value.

2.1.9 Hyperparameter settings and simulation details

For the models M1, M2 and M4 we re-use the hyperparameters from the earlier works by [38] and [24]: $\sigma^{-2} \sim GAM(\alpha_\sigma = \nu, \beta_\sigma = \nu)$ with $\nu = 0.005$, $\lambda_u^{-1} \sim GAM(\alpha_u = 2, \beta_u = 0.2)$, and $\lambda_c^{-1} \sim GAM(\alpha_c = 3, \beta_c = 3)$. For M3 we use the same setting with the extension: $\delta_h \sim BER(\text{p})$ with $\text{p} \sim BETA(a = 1, b = 1)$. For the new models M3-M4 we also experimented with alternative hyperparameter settings, which we took from [24]. In agreement with the results reported in [24], we found that the models are rather robust w.r.t. the hyperparameter values. The results were only slightly affected by the hyperparameter values.

For all models we run each Reversible Jump Markov Chain Monte Carlo (RJMCMC) simulation for $V = 100,000$ iterations. Setting the burn-in phase to $0.5V$ (50%) and thinning out by the factor 10 during the sampling phase, yields $W = 0.5V/10 = 5000$ samples from each posterior. To check for convergence, we compared the samples of independent simulations, using the conventional diagnostics, based on potential scale reduction factors [20] as well as scatter plots of the estimated edge scores. The latter type of diagnostic has become very common for Bayesian networks (see, e.g., the work by [17]). For most of the data sets, analysed here, the scatter plot diagnostics indicated almost perfect convergence already after $V = 10,000$ iterations; see Figure 2.9(a) for an example. For $V = 100,000$ iterations we consistently observed perfect convergence for all data sets.

2.2 Data

2.2.1 Synthetic network data

For model comparisons we generated various synthetic network data sets. We report here on two studies with realistic network topologies, shown in Figure 2.6. In both studies we assumed the data segmentation to be known. Hence, we kept the changepoints in $\boldsymbol{\tau}$ fixed at their right locations and did not perform reversible jump Markov chain Monte Carlo moves on $\boldsymbol{\tau}$.

Study 1: For the RAF pathway [50] with $N = 11$ nodes and $M = 20$ edges, shown in Figure 2.6, we generated data with $H = 4$ segments having $m = 10$ data points each. For each node Z_i and its parent nodes in $\boldsymbol{\pi}_i$ we sampled the regression coefficients for $h = 1$ from standard Gaussian distributions and collected them in a vector $\mathbf{w}_1^i$ which we normalised to Euclidean norm 1, $\mathbf{w}_1^i \leftarrow \mathbf{w}_1^i / |\mathbf{w}_1^i|$. For the segments $h = 2, 3, 4$ we use: $\mathbf{w}_h^i = \mathbf{w}_{h-1}^i$ ($\delta_h = 1$, coupled) or $\mathbf{w}_h^i = -\mathbf{w}_{h-1}^i$ ($\delta_h = 0$, uncoupled). The design matrices $\mathbf{X}_h^i$ contain a first column of 1's for the intercept and the segment-specific values of the parent nodes, shifted by one time point. To the segment-specific values of Z_i: $\mathbf{z}_h^i = \mathbf{X}_h^i \mathbf{w}_h^i$ we element-wise added Gaussian noise with standard deviation $\sigma = 0.05$. For all coupling scenarios $(\delta_2, \delta_3, \delta_4) \in \{0, 1\}^3$, we generated 25 data sets having different regression coefficients.

Study 2: This study is similar to the first one with three changes: (i) We used the yeast network [8] with $N = 5$ nodes and $M = 8$ edges, shown in Figure 2.6, (ii) again we generated data with $H = 4$ segments, but we varied the number of time points per segment $m \in \{2, 3, \ldots, 12\}$. (iii) We focused on one scenario: For each node Z_i and its parent nodes in $\boldsymbol{\pi}_i$ we generated two vectors $\mathbf{w}_\diamond^i$ and $\mathbf{w}_\star^i$ with standard Gaussian distributed entries. We re-normalised the first vector to Euclidean norm 1, $\mathbf{w}_\diamond^i \leftarrow \mathbf{w}_\diamond^i / |\mathbf{w}_\diamond^i|$, and the 2nd vector to norm 0.5, $\mathbf{w}_\star^i \leftarrow 0.5 \cdot \mathbf{w}_\star^i / |\mathbf{w}_\star^i|$. We set $\mathbf{w}_1^i = \mathbf{w}_2^i = \mathbf{w}_\diamond^i$ so that the segments $h = 1$ and $h = 2$ are coupled, and $\mathbf{w}_3^i = \mathbf{w}_4^i = (\mathbf{w}_\diamond^i + \mathbf{w}_\star^i)/(|\mathbf{w}_\diamond^i + \mathbf{w}_\star^i|)$, so that the segments $h = 3$ and $h = 4$ are coupled, while the coupling between $h = 3$ and $h = 2$ is 'moderate'. For each m we generated 25 data matrices with different regression coefficients.

2.2.2 Yeast gene expression data

[8] synthetically designed a network in *S. cerevisiae* (yeast) with $N = 5$ genes, and measured gene expression data under galactose- and glucose-metabolism:

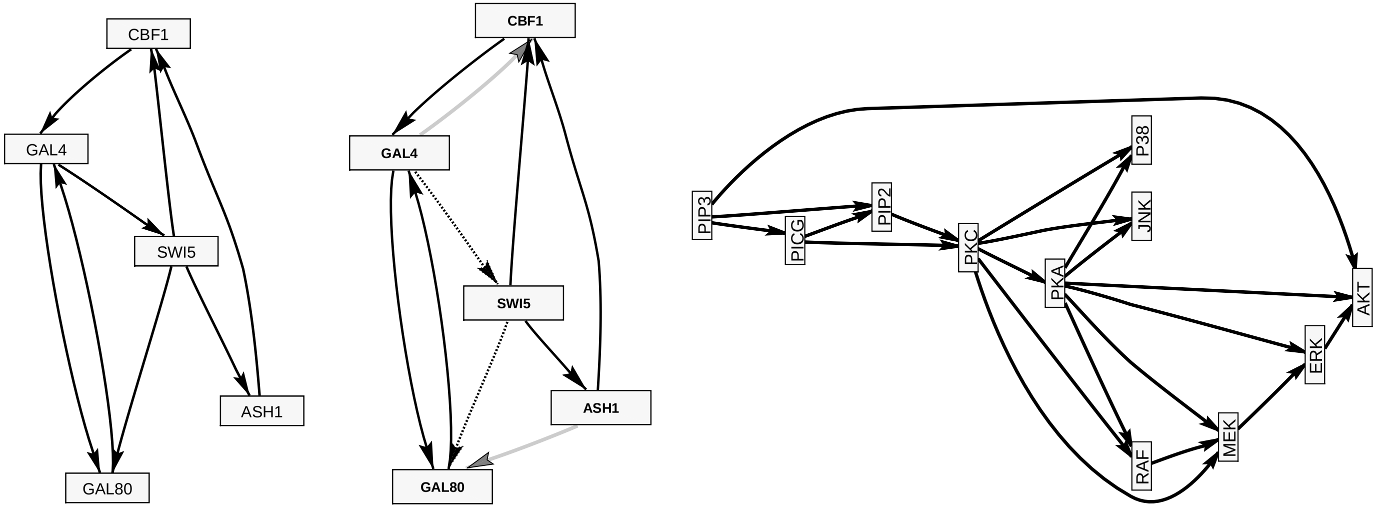

CBF1
GAL4
SWI5
ASH1
GAL80
CBF1
GAL4
SWI5
ASH1
GAL80
PIP3
PICG
PIP2
PKC
PKA
P38
JNK
RAF
MEK
ERK
AKT

Figure 2.6: Network structures. Left: The true yeast network with $N = 5$ nodes and $M = 8$ edges. Centre: Network prediction obtained with model M3. The grey (dotted) edges correspond to false positives (negatives). Right: RAF pathway with $N = 11$ nodes and $M = 20$ edges.

16 measurements were taken in galactose and 21 measurements were taken in glucose, with 20 minutes intervals in between measurements. Although the network is small, it is an ideal benchmark data set: The network structure is known, so that network reconstruction methods can be cross-compared on real wet-lab data. We pre-process the data as described in [24]. The true network structure is shown in Figure 2.6 (left panel). As an example, a network prediction obtained with the partially coupled model (M3) is shown in the centre panel. For the prediction we extracted the 8 edges with the highest scores.

2.2.3 Arabidopsis gene expression data

The circadian clock in *Arabidopsis thaliana* optimizes the gene regulatory processes with respect to the daily dark:light cycles (photo periods). In four experiments *A. thaliana* plants were entrained in different dark:light cycles, before gene expression data were measured under constant light condition over 24- and 48-hour time intervals. We follow [24] and merge the four time series to one single data set with $T = 47$ data points and focus our attention on the $N = 9$ core genes: LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3.

2.3 Empirical results

We found that the proposed partially coupled model (M3), performed, overall, better than the other models. Thus, we decided to use M3 as reference model.

2.3.1 Results for synthetic network data

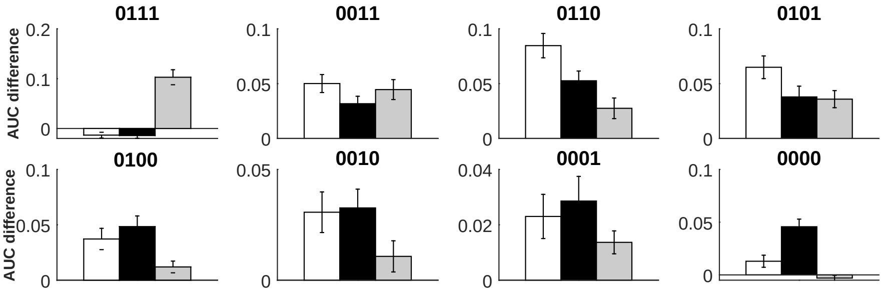
We start with the RAF-pathway for which we generated network data for 8 different coupling scenarios. Figure 2.7(a) compares the network reconstruction accuracies in terms of average AUC value differences. For 6 out of 8 scenarios the three AUC differences are clearly in favour of M3. Not surprisingly, for the two extreme scenarios, where all segments $h \geq 2$ are either coupled ('0111') or uncoupled ('0000'), M3 performs slightly worse than the fully coupled models (M2 and M4) or the uncoupled model (M1), respectively. But unlike the uncoupled model (M1) for coupled data ('0111'), and unlike the coupled models (M2 and M4) for uncoupled data ('0000'), the partially coupled model (M3) never performs significantly worse than the respective 'gold-standard' model. For the partially coupled model, Figure 2.7(b) shows the posterior probabilities that the segments $h = 2, 3, 4$ are coupled. The trends are in good agreement with the true coupling mechanism. Model M3 correctly infers whether the regression coefficients stay similar (identical) or change (substantially). The generalised coupled model (M4) can only adjust the segment-specific coupling strengths, but has no option to uncouple. Like the coupled model (M2), it fails when the parameters are subject to drastic changes. When comparing the coupled model (M2) with the generalised coupled model (M4), we see that M2 performs better when only one segment is coupled, while the new M4 model is superior to M2 if two segments are coupled, see the scenarios '0011', '0110', and '0101'.

For the yeast network we generated data corresponding to a '0101' coupling scheme and the change of the parameters (from the 2nd to the 3rd segment) is less drastic than for the RAF pathway data. Figure 2.8 shows how the AUC differences vary with the number of time points T, where $T = 4m$ and m is the number of data points per segment. For sufficiently many data points the effect of the prior diminishes and all models yield high AUC values (see bottom right panel). There are then no substantial differences between the AUC values anymore. However, for the lower sample sizes again the partially coupled model (M3) performs clearly best. For $12 \leq m \leq 28$ model M3 is clearly superior to all other models and for $30 \leq T \leq 40$ it still outperforms the uncoupled (M1) as well as the coupled (M2) model. The performance of the generalised model (M4) is comparable to the performance of the uncoupled model. For moderate sample sizes ($12 \leq T \leq 44$) model M4 is superior to the fully coupled model (M2).

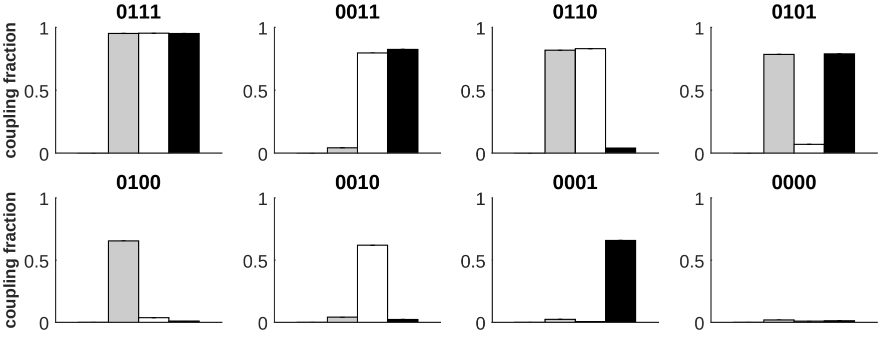
2.3.2 Results for yeast gene expression data

For the yeast gene expression data we assume the changepoint(s) to be unknown and we infer the segmentation from the data. Figure 2.9(a) shows convergence diagnostics for the partially coupled model (M3). It can be seen from the scatter plots that $V = 10,000$ RJMCMC iterations yield already almost perfect convergence. The edge scores of 15 independent MCMC runs are almost identical to each other.

The average AUC scores of the models M1-M4 are shown in Figure 2.9(b). Since the number of inferred changepoints grows with the hyperparameter p of the geometric distribution on the distance between changepoints, we implemented the models with different p's. The uncoupled model is superior to the coupled model for the lowest p ($p = 0.02$) only, but becomes more and more inferior to the coupled model, as p increases. This result is consistent with the finding in [24], where it was argued that parameter coupling gets more important when the number of segments H increases so that the individual segments become shorter. The new partially coupled model (M3) performs consistently better than the uncoupled and the coupled model (M1-M2). The only exemption occurs for $p = 0.1$ where the coupled model (M2) appears to perform slightly (but not statistically significantly) better than M3. For p's up to $p = 0.05$ the fully coupled (M2) and the generalised fully coupled model (M4) perform approximately equally well. However, for the three highest p's the new model M4 performs better than the coupled model (M2) and even outperforms the partially coupled model (M3). While the performances of the models M1-M3 decrease with the number of changepoints, the performance of model M4 stays robust. Subsequently, we re-analysed the yeast data with $K = 1, \ldots, 5$ fixed changepoints. For each K we used the first changepoint to separate the two parts of the time series (galactose vs. glucose metabolism). Successively we located the next changepoint in the middle of the longest segment to divide it into 2 segments, until K changepoints



(a) AUC differences, M3 **versus** M2, M4 and M1



(b) Coupling probabilities for model M3

Figure 2.7: Results for synthetic RAF pathway data. We distinguish 8 coupling scenarios $(\delta_1 = 0, \delta_2, \delta_3, \delta_4)$. **(a)**: Each histogram has three bars for the average AUC differences between the partially coupled model (M3) and the other models: 'M3 vs. M1 [=Uncoupled]' (gray), 'M3 vs. M4 [=Generalised]' (black), and 'M3 vs. M2 [=Coupled]' (white). The error bars indicate 95% t-test confidence intervals. **(b)**: Diagnostic for the partially coupled model (M3): The bars give the posterior probabilities $p(\delta_h = 1|\mathcal{D})$ that segment h is coupled to $h-1$ $(h = 2, 3, 4)$.

were set. Figure 2.10(a-b) show the average AUC scores and the AUC score differences in favour of the partially coupled model (M3). Panel (a) reveals that the partially coupled model (M3) reaches again the highest network reconstruction accuracies. Panel (b) shows that the superiority of M3 is statistically significant, with only one exemption: For $K = 1$ the uncoupled model M1 performs as good as the partially coupled model (M3). Subsequently, we investigated the segment-specific coupling posterior probabilities $p(\delta_h = 1|\mathcal{D})$ $(h = 2, \ldots, H = K + 1)$ for the partially coupled model (M3) and the posterior distributions of the coupling parameters $\lambda_u, \lambda_2, \ldots, \lambda_{K+1}$ for the generalized model (M4), but we could not find clear trends for any gene. As an example, we provide the results for gene ASH1 in Figure 2.10(c-d). Panel (c) shows that the coupling posterior probabilities of model M3 do not have a clear pattern. However, it becomes obvious that the partially coupled model makes use of segment-wise switches between the

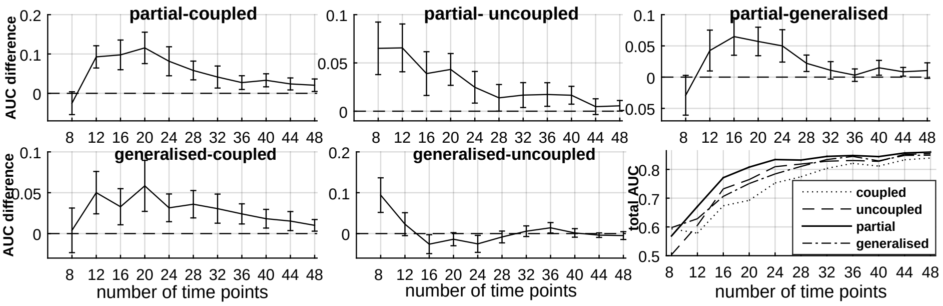
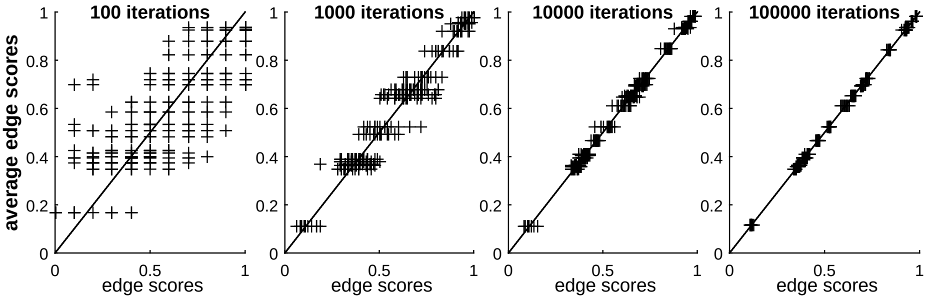
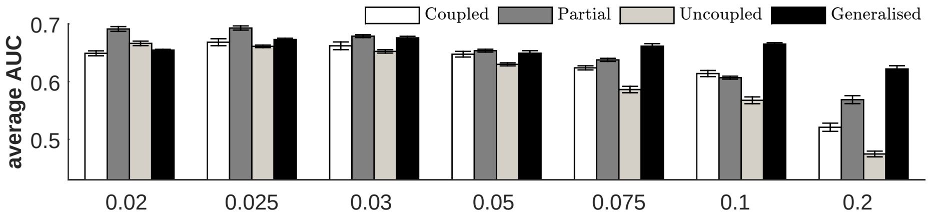


Figure 2.8: Results for synthetic yeast data. Five panels show the average AUC differences plotted against the numbers of data points T. The error bars indicate 95% t-test confidence intervals. The bottom right panel shows the model-specific average AUC values.



(a) **Edge score scatter plots for the proposed partially coupled model (M3).**



(b) **Average AUC results for real yeast data with changepoint inference.**

Figure 2.9: Analysis of the real yeast data. Panel (a): For each run length, $V \in \{100, 1000, 10000, 100000\}$ we performed 15 RJMCMC simulations with the partially coupled model (M3). We used the hyperparameter $p = 0.05$ for the changepoint prior. For each V there is a scatter plot where the simulation-specific edge scores (vertical axis) are plotted against the average scores for that V (horizontal axis). **Panel (b)**: We implemented the models M1-M4 with different hyperparameters p of the geometric distribution for the distance between changepoints. For each p the bars show the model-specific average AUC scores. The error bars indicate standard deviations.

uncoupled and the coupled approach. Panel (d) shows that the distributions of the coupling parameters, $\lambda_2, \ldots, \lambda_{K+1}$, of model M4 do not substantially differ among segments. This might be the reason why the generalised coupled model (M4) is not superior to the fully coupled model (M2).

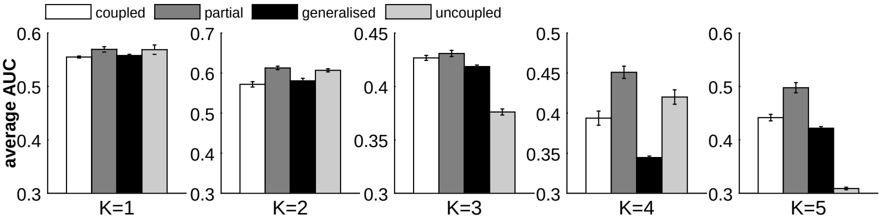
2.3.3 Application to Arabidopsis gene expression data

For the *Arabidopsis thaliana* gene expression data we cannot objectively compare the network reconstruction accuracies of the four models, since the true circadian clock network is not known. We therefore only applied the partially coupled model (M3), which we had found to be the best model in our earlier studies. Figure 2.11 shows the *Arabidopsis thaliana* network, which was reconstructed using the hyperparameter $p = 0.1$ for the geometric distribution on the distance between changepoints. To obtain a network prediction, we extracted the 20 edges with the highest edge scores. Although a proper evaluation of the network prediction is beyond the scope of this paper, we note that several features of the network are consistent with the plant biology literature. E.g. the feedback loop between LHY and $TOC1$ is the most important key feature of the circadian clock network (see, e.g., the work by [41]). Many of the other predicted edges have been reported in more recent works. E.g. the edges $LHY \rightarrow ELF3$, $LHY \rightarrow ELF4$, $GI \rightarrow TOC1$, $ELF3 \rightarrow PRR3$ and $ELF4 \rightarrow PRR9$ can all be found in the circadian clock network (hypothesis) of [29].

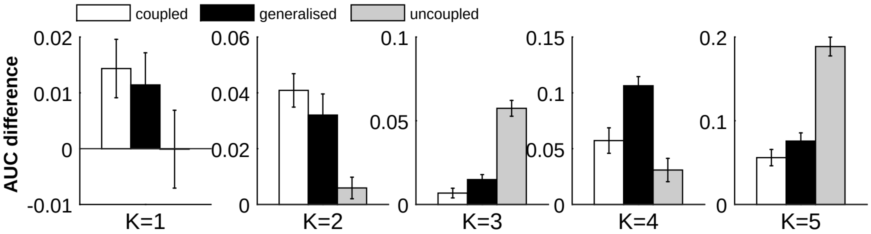
2.4 Discussion and conclusions

We have proposed two new Bayesian piece-wise linear regression models (M3 and M4) for reconstructing regulatory networks from gene expression time series. The partially coupled model (M3), shown in Figure 2.4, is a consensus model between the conventional uncoupled model (M1) and the fully sequentially coupled model (M2), proposed by [24]. In the uncoupled model (M1) the segment-specific regression coefficients have to be learned for each segment separately. In the fully coupled model (M2) each segment is compelled to be coupled to the previous one. The new partially coupled model (M3) combines features of the uncoupled and the fully coupled model. It infers for each segment whether it is coupled to (or uncoupled from) the preceding segment. The generalised coupled model (M4) is a generalisation of the coupled model (M2). Like the fully coupled model (M2), it does not have any option to uncouple, but it possesses segment-specific coupling parameters and allows for different coupling strengths between segments. Figure 2.5 shows a graphical model representation of the generalised model.

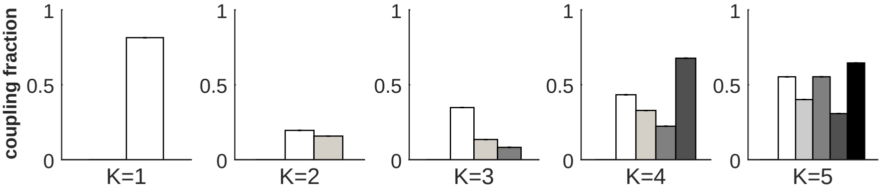
We have cross-compared the two newly proposed models (M3-M4) with the two established models (M1-M2); see Figures 2.2-2.3 for their graphical model representations. Our empirical results show that the generalised coupled model (M4) does not yield consistent improvements over the coupled model (M2). In



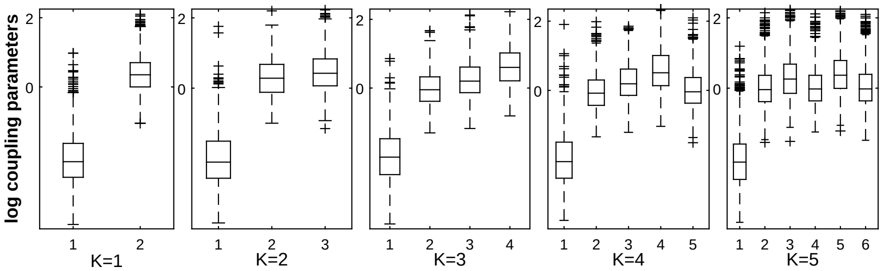
(a) **AUCs of M2 (coupled), M3 (partially), M4 (generalised) and M1 (uncoupled).**



(b) **Partially (M3) vs. M2 (coupled), M4 (generalised) and M1 (uncoupled).**



(c) **M3: Average segment-specific coupling probabilities for target gene ASH1.**



(d) **M4: Segment-specific distributions of the coupling parameters for ASH1.**

Figure 2.10: Results for real yeast data with fixed changepoints. We imposed $K \in \{1, \dots, 5\}$ fixed changepoints. See Subsection 2.3.2 for more detail. **Panel (a)** show the model-specific average total AUC scores with error bars indicating standard deviations. **Panel (b)** shows the AUC score differences with error bars indicating 95% two-sided t-test confidence intervals. **Panel (c)**: Diagnostic for the partially coupled model (M3): The bars give the posterior probabilities $p(\delta_h = 1|\mathcal{D})$ that segment h is coupled to $h-1$ ($h = 2, \dots, K+1$) for gene ASH1. **Panel (d)**: Diagnostic for the generalized coupled model (M4): In each panel there is a boxplot for each segment showing the distributions of the logarithmic coupling parameters λ_h for gene ASH1.

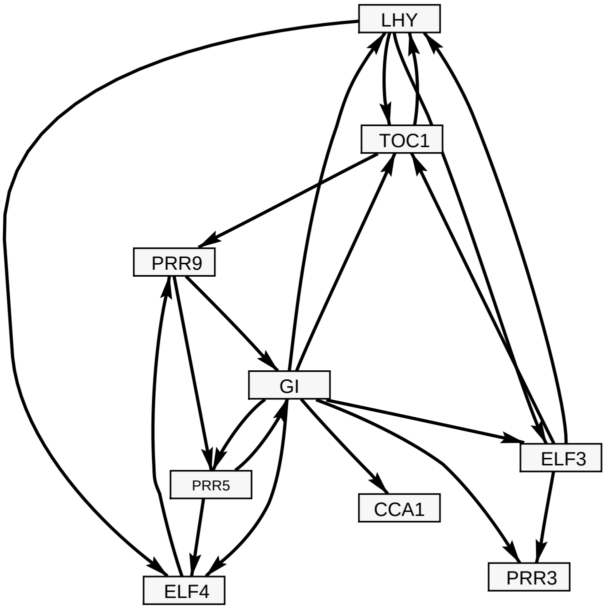


Figure 2.11: Prediction of the circadian clock network in *Arabidopsis thaliana*. The prediction was obtained with the proposed partially coupled model (M3), using the hyperparameter $p = 0.1$ for the geometric distribution on the distance between changepoints. The network shows the 20 edges with the highest edge scores.

next chapter, we will try to figure out why the generalised fully coupled model did not perform better than the fully coupled model and we will try to refine this model.

The partially coupled model (M3), on the other hand, led consistently to improved network reconstruction accuracies. It has the desirable feature that it comprises the uncoupled (M1) and the fully coupled (M2) model as limiting cases. As the new partially segment-wise coupled model (M3) infers the best trade-off between M1 and M2 from the data, we would argue that this model should have precedence over the models M1 and M2 with regard to future applications.

Chapter 3

Generalized sequentially coupled NH-DBNs

A drawback of the fully sequentially coupled model, as discussed in previous chapters, is that all pairs of neighboring segments $(h-1, h)$ are coupled with the same coupling parameter λ_c and thus with the same strength. For addressing this pitfall we introduced generalized sequentially coupled model with segment-specific coupling parameters in chapter 2. We showed that, this generalized model does not perform consistently better reconstruction accuracy than the coupled model in term of area under precision recall curve (AUC). In this chapter, we investigate this problem and try to refine it. Therefore, we introduce an improved version of generalized coupled model, proposed in chapter 2. Unlike the original model, that is, sequentially coupled model, our novel model possesses segment-specific coupling parameters, so that the coupling strengths between parameters can vary over time. Thereby, to avoid model over-flexibility and to allow for some information exchange among time segments, we globally couple the segment-specific coupling (strength) parameters by introducing a hyperprior onto the second hyperparameter of the coupling parameter prior which refine the generalized coupled model introduced in previous chapter. Our empirical results on synthetic as well as on real biological network data show that the new model yields better network reconstruction accuracies than the original model.

The work, presented in this chapter, has been published in Statistica Neerlandica (2018) (see [58]). Some parts of it also have been appeared in Proceedings of the International Workshop on Statistical Modelling (2017) (see [55]).

3.1 Methods

We consider piecewise-linear regression models where the random variable Y is the target and the random variables $X_1, \ldots, X_k$ are the covariates. We assume that T data points $\mathcal{D}_1, \ldots, \mathcal{D}_T$ are available and that the subscript index $t \in \{1, \ldots, T\}$

refers to T equidistant time points. Each data point $\mathcal{D}_t$ contains a value of the target Y and the corresponding values of the k covariates. We assume further that the T data points are allocated to H disjunct segments, $h \in \{1, \dots, H\}$. Segment h contains T_h consecutive data points with $\sum_{h=1}^{H} T_h = T$. Within each individual segment h we apply a Bayesian linear regression model with a segment-specific regression coefficient vector $\boldsymbol{\beta}_h = (\beta_{h,0}, \dots, \beta_{h,k})^{\mathsf{T}}$. Let $\mathbf{y}_h$ be the target vector of length T_h and let $\mathbf{X}_h$ be the T_h-by-$(k+1)$ design matrix for segment h, which includes a first column of 1′s for the intercept. For each segment $h = 1, \dots, H$ we assume a Gaussian likelihood:

$$\mathbf{y}_h | (\boldsymbol{\beta}_h, \sigma^2) \sim \mathcal{N}(\mathbf{X}_h \boldsymbol{\beta}_h, \sigma^2 \mathbf{I}) \tag{3.1}$$

where $\mathbf{I}$ denotes the T_h-by-T_h identity matrix, and σ^2 is the noise variance parameter, which is shared among segments. We impose an inverse Gamma prior on σ^2, $\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$. In the forthcoming subsections we will present different model instantiations with different prior distributions for the segment-specific regression coefficient vectors $\boldsymbol{\beta}_h$ $(h = 1, \dots, H)$.

3.1.1 The original sequential coupling scheme

In the sequentially coupled piecewise linear regression model, proposed by [24], it is assumed that the regression coefficient vectors $\boldsymbol{\beta}_h$ have the following Gaussian prior distributions:

$$\boldsymbol{\beta}_h | (\sigma^2, \lambda_u, \lambda_c) \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma^2 \lambda_u \mathbf{I}) & \text{if } h = 1 \\ \mathcal{N}(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2 \lambda_c \mathbf{I}) & \text{if } h > 1 \end{cases} \qquad (h = 1, \dots, H) \tag{3.2}$$

where $\mathbf{0}$ is the zero vector of length $k+1$, $\mathbf{I}$ denotes the $(k+1)$-by-$(k+1)$ identity matrix, and $\tilde{\boldsymbol{\beta}}_{h-1}$ is the posterior expectation of $\boldsymbol{\beta}_{h-1}$. That is, only the first segment $h = 1$ gets an uninformative prior expectation, namely the zero vector, while the subsequent segments $h > 1$ obtain informative prior expectations, stemming from the preceding segment $h - 1$. We follow [24] and refer to $\lambda_u \in \mathbb{R}^+$ as the signal-to-noise ratio (SNR) parameter and to $\lambda_c \in \mathbb{R}^+$ as the coupling parameter. A low (high) SNR parameter λ_u yields a peaked (vague) prior for $h = 1$ in Equation (3.2), and thus that the distribution of $\boldsymbol{\beta}_1$ is peaked (diffuse) around the zero vector. A low (high) coupling parameter λ_c yields a peaked (vague) prior for $h > 1$ in Equation (3.2), and thus a strong (weak) coupling of $\boldsymbol{\beta}_h$ to the posterior expectation $\tilde{\boldsymbol{\beta}}_{h-1}$ from the preceding segment. We note that re-employing the variance parameter σ^2 in Equation (3.2), yields a fully-conjugate prior in both groups of parameters $\boldsymbol{\beta}_h$ $(h = 1, \dots, H)$ and σ^2 (see, e.g., Sections 3.3 and 3.4 in [19]) with the marginal likelihood given below in Equation (3.10). The posterior distribution of $\boldsymbol{\beta}_h$ $(h = 1, \dots, H)$ can be computed in closed form [24]:

$$\begin{aligned} &\boldsymbol{\beta}_h | (\mathbf{y}_h, \sigma^2, \lambda_u, \lambda_c) \\ &\quad \sim \begin{cases} \mathcal{N}\left([\lambda_u^{-1}\mathbf{I} + \mathbf{X}_1^{\mathsf{T}}\mathbf{X}_1]^{-1}\mathbf{X}_1^{\mathsf{T}}\mathbf{y}_1, \sigma^2(\lambda_u^{-1}\mathbf{I} + \mathbf{X}_1^{\mathsf{T}}\mathbf{X}_1)^{-1}\right) & \text{if } h = 1 \\ \mathcal{N}\left([\lambda_c^{-1}\mathbf{I} + \mathbf{X}_h^{\mathsf{T}}\mathbf{X}_h]^{-1}(\lambda_c^{-1}\tilde{\boldsymbol{\beta}}_{h-1} + \mathbf{X}_h^{\mathsf{T}}\mathbf{y}_h), \sigma^2(\lambda_c^{-1}\mathbf{I} + \mathbf{X}_h^{\mathsf{T}}\mathbf{X}_h)^{-1}\right) & \text{if } h \geq 2 \end{cases} \end{aligned} \tag{3.3}$$

and the posterior expectations in Equation (3.3) are the prior expectations used in Equation (3.2):

$$\tilde{\boldsymbol{\beta}}_{h-1} := \begin{cases} [\lambda_u^{-1}\mathbf{I} + \mathbf{X}_1^{\mathsf{T}}\mathbf{X}_1]^{-1}\mathbf{X}_1^{\mathsf{T}}\mathbf{y}_1 & \text{if } h = 2 \\ [\lambda_c^{-1}\mathbf{I} + \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{X}_{h-1}]^{-1}(\lambda_c^{-1}\tilde{\boldsymbol{\beta}}_{h-2} + \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{y}_{h-1}) & \text{if } h \geq 3 \end{cases} \tag{3.4}$$

[24] assigned inverse Gamma priors to the parameters λ_u and λ_c:

$$\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u) \tag{3.5}$$

$$\lambda_c^{-1} \sim GAM(\alpha_c, \beta_c) \tag{3.6}$$

The fully sequentially coupled model is then fully specified and we will refer to it as the $\mathcal{M}_{0,0}$ model. A graphical model representation for the relationships in the first segment $h = 1$ is provided in Figure 3.1, while Figure 3.2 shows a graphical model representation for the segments $h > 1$. The posterior distribution of the $\mathcal{M}_{0,0}$ model fulfills:

$$\begin{aligned} p(\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_c | \mathbf{y}_1, \dots, \mathbf{y}_H) & \\ & \propto p(\mathbf{y}_1, \dots, \mathbf{y}_H, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_c) \\ & \propto \prod_{h=1}^{H} p(\mathbf{y}_h | \sigma^2, \boldsymbol{\beta}_h) \cdot p(\boldsymbol{\beta}_1 | \sigma^2, \lambda_u) \\ & \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h | \sigma^2, \lambda_c) \cdot p(\sigma^2) \cdot p(\lambda_u) \cdot p(\lambda_c) \end{aligned} \tag{3.7}$$

Like the regression coefficient vectors $\boldsymbol{\beta}_h$, whose full conditional distributions have been provided in Equation (3.3), the parameters λ_u and λ_c can also be sampled from their full conditional distributions.

$$\begin{aligned} & \lambda_c^{-1} | (\mathbf{y}_1, \dots, \mathbf{y}_H, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_c) \\ & \sim GAM\left(\alpha_c + \frac{(H-1)\cdot(k+1)}{2}, \beta_c + \frac{1}{2}\sigma^{-2} \cdot \sum_{h=2}^{H} (\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}} (\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})\right) \\ & \lambda_u^{-1} | (\mathbf{y}_1, \dots, \mathbf{y}_H, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_c) \\ & \sim GAM\left(\alpha_u + \frac{1\cdot(k+1)}{2}, \beta_u + \frac{1}{2}\sigma^{-2} \cdot \boldsymbol{\beta}_1^{\mathsf{T}} \boldsymbol{\beta}_1\right) \end{aligned} \tag{3.8}$$

For the parameter σ^2 a collapsed Gibbs sampling step, with $\boldsymbol{\beta}_h$ $(h = 1, \dots, H)$ integrated out, can be used:

$$\begin{aligned} & \sigma^{-2} | (\mathbf{y}_1, \dots, \mathbf{y}_H, \lambda_u, \lambda_c) \\ & \sim GAM\left(\alpha_\sigma + \frac{1}{2} \cdot T, \beta_\sigma + \frac{1}{2} \cdot \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}} \mathbf{C}_h^{-1} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})\right) \end{aligned} \tag{3.9}$$

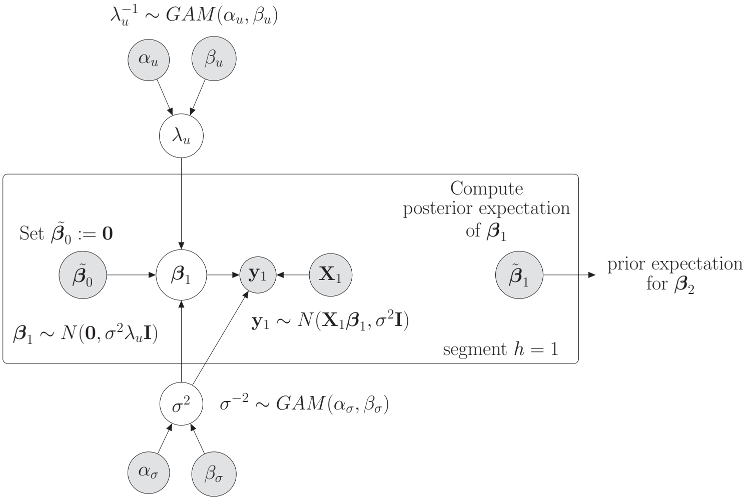

$\lambda_u^{-1} \sim GAM(\alpha_u, \beta_u)$
α_u
β_u
λ_u
Compute posterior expectation of $\boldsymbol{\beta}_1$
Set $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$
$\tilde{\boldsymbol{\beta}}_0$
$\boldsymbol{\beta}_1$
$\mathbf{y}_1$
$\mathbf{X}_1$
$\tilde{\boldsymbol{\beta}}_1$
prior expectation for $\boldsymbol{\beta}_2$
$\mathbf{y}_1 \sim N(\mathbf{X}_1\boldsymbol{\beta}_1, \sigma^2\mathbf{I})$
$\boldsymbol{\beta}_1 \sim N(\mathbf{0}, \sigma^2\lambda_u\mathbf{I})$
segment $h = 1$
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 3.1: Graphical model of the probabilistic relationships in the first segment, $h = 1$. Parameters that have to be inferred are represented by white circles. The observed data ($\mathbf{y}_1$ and $\mathbf{X}_1$) and the fixed hyperparameters are represented by grey circles. All nodes in the plate are specific for the first segment. The posterior expectation $\tilde{\boldsymbol{\beta}}_1$ is computed and then treated like a fixed hyperparameter vector when used as input for segment $h = 2$.

$$\text{where } \tilde{\boldsymbol{\beta}}_0 := \mathbf{0} \text{ and } \mathbf{C}_h := \begin{cases} \mathbf{I} + \lambda_u \mathbf{X}_h \mathbf{X}_h^\mathsf{T} & \text{if } h = 1 \\ \mathbf{I} + \lambda_c \mathbf{X}_h \mathbf{X}_h^\mathsf{T} & \text{if } h > 1 \end{cases}$$

For the marginal likelihood, with $\boldsymbol{\beta}_h$ ($h = 1, \dots, H$) and σ^2 integrated out, the marginalization rule from Section 2.3.7 of [5] can be applied:

$$\begin{aligned} p(\mathbf{y}_1, \dots, \mathbf{y}_H | \lambda_u, \lambda_c) &= \frac{\Gamma(\frac{T}{2} + a_\sigma)}{\Gamma(a_\sigma)} \cdot \frac{\pi^{-T/2} \cdot (2b_\sigma)^{a_\sigma}}{(\prod_{h=1}^{H} \det(\mathbf{C}_h))^{1/2}} \\ &\quad \cdot (2b_\sigma + \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})^\mathsf{T} \mathbf{C}_h^{-1} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1}))^{-(\frac{T}{2} + a_\sigma)} \end{aligned} \tag{3.10}$$

where the matrices $\mathbf{C}_h$ were defined below Equation (3.9). For the derivations of the full conditional distributions in Equations (3.8) and (3.9) and the marginal likelihood in Equation (3.10) we refer to [24].

3.1.2 The improved sequential coupling scheme, proposed here

We propose to generalized the sequentially coupled model from Subsection 3.1.1 by introducing segment-specific coupling parameters λ_h ($h = 2, \dots, H$) and a new hyperprior onto the second hyperparameter of the coupling parameter prior.

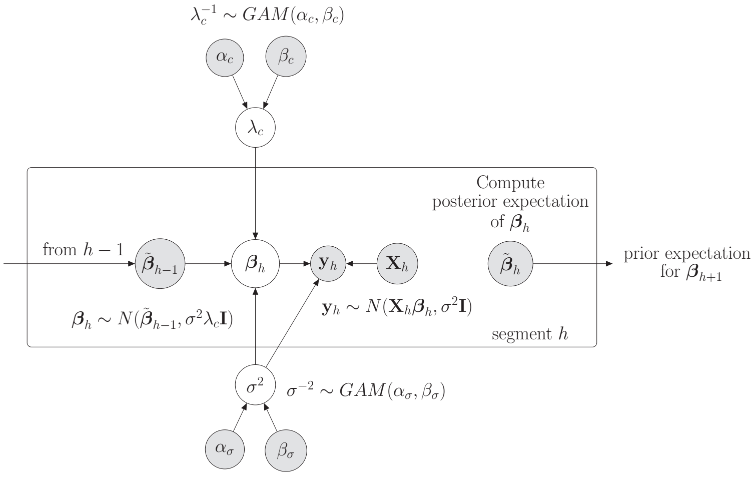

$\lambda_c^{-1} \sim GAM(\alpha_c, \beta_c)$
α_c
β_c
λ_c
Compute posterior expectation of $\boldsymbol{\beta}_h$
from $h-1$
$\tilde{\boldsymbol{\beta}}_{h-1}$
$\boldsymbol{\beta}_h$
$\mathbf{y}_h$
$\mathbf{X}_h$
$\tilde{\boldsymbol{\beta}}_h$
prior expectation for $\boldsymbol{\beta}_{h+1}$
$\boldsymbol{\beta}_h \sim N(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_c\mathbf{I})$
$\mathbf{y}_h \sim N(\mathbf{X}_h\boldsymbol{\beta}_h, \sigma^2\mathbf{I})$
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 3.2: Graphical model of the probabilistic relationships within and between segments $h > 1$ for the $\mathcal{M}_{0,0}$ model from [24]. Parameters that have to be inferred are represented by white circles. The observed data and the fixed hyperparameters are represented by grey circles. All nodes in the plate are specific for segment h. The posterior expectation $\tilde{\boldsymbol{\beta}}_{h-1}$ of the regression coefficient vector from the previous segment $h-1$ is treated like a fixed hyperparameter vector. The posterior expectation $\tilde{\boldsymbol{\beta}}_h$ is computed and forwarded as fixed hyperparameter vector to the subsequent segment $h+1$.

This yields the new prior distributions:

$$\boldsymbol{\beta}_h|(\sigma^2, \lambda_u, \lambda_2, \ldots, \lambda_H) \sim \begin{cases} \mathcal{N}(\mathbf{0}, \sigma^2\lambda_u\mathbf{I}) & \text{if } h=1 \\ \mathcal{N}(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_h\mathbf{I}) & \text{if } h>1 \end{cases} \tag{3.11}$$

where $\tilde{\boldsymbol{\beta}}_{h-1}$ is again the posterior expectation of $\boldsymbol{\beta}_{h-1}$. For notational convenience we now introduce two new definitions, namely: $\lambda_1 := \lambda_u$ and $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$. We can then compactly write: For $h = 2, \ldots, H$

$$\tilde{\boldsymbol{\beta}}_{h-1} := [\lambda_{h-1}^{-1}\mathbf{I} + \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{X}_{h-1}]^{-1}(\lambda_{h-1}^{-1}\tilde{\boldsymbol{\beta}}_{h-2} + \mathbf{X}_{h-1}^{\mathsf{T}}\mathbf{y}_{h-1}) \tag{3.12}$$

We show in the next subsection that $\tilde{\boldsymbol{\beta}}_{h-1}$, defined in Equation (3.12), is the posterior expectation of $\boldsymbol{\beta}_{h-1}$, cf. Equation (3.14). For the parameter λ_u we re-use the inverse Gamma prior with hyperparameters α_u and β_u. For the first segment $h = 1$ we thus have the same probabilistic relationships like for the original model, compare the graphical model representation in Figure 3.1. For the parameters λ_h we assume that they are inverse Gamma distributed, $\lambda_h^{-1} \sim GAM(\alpha_c, \beta_c)$ where α_c is fixed and β_c is a free hyperparameter. A free β_c allows for information exchange among the segments $h = 2, \ldots, H$. We impose a Gamma hyperprior onto β_c, symbolically $\beta_c \sim GAM(a, b)$.

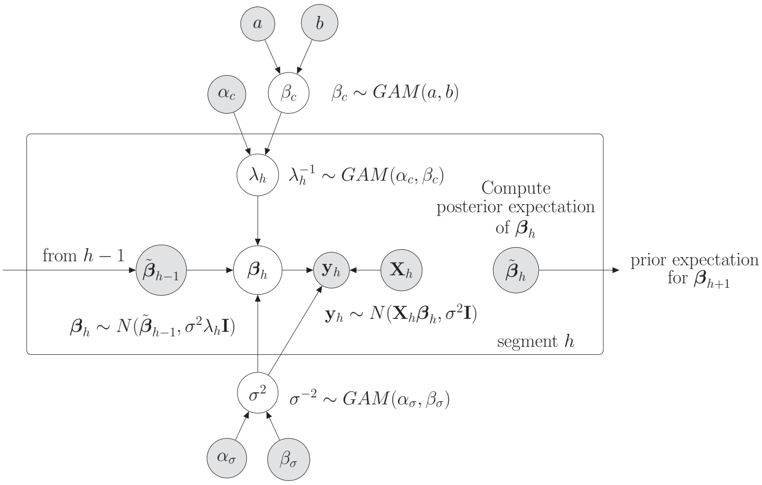

a
b
α_c
β_c
$\beta_c \sim GAM(a,b)$
λ_h
$\lambda_h^{-1} \sim GAM(\alpha_c,\beta_c)$
Compute posterior expectation of $\boldsymbol{\beta}_h$
from $h-1$
$\tilde{\boldsymbol{\beta}}_{h-1}$
$\boldsymbol{\beta}_h$
$\mathbf{y}_h$
$\mathbf{X}_h$
$\tilde{\boldsymbol{\beta}}_h$
prior expectation for $\boldsymbol{\beta}_{h+1}$
$\boldsymbol{\beta}_h \sim N(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_h\mathbf{I})$
$\mathbf{y}_h \sim N(\mathbf{X}_h\boldsymbol{\beta}_h, \sigma^2\mathbf{I})$
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma,\beta_\sigma)$
α_σ
β_σ

Figure 3.3: Graphical model of the probabilistic relationships within and between segments $h > 1$ for the proposed $\mathcal{M}_{1,1}$ model. Parameters that have to be inferred are represented by white circles. The observed data and the fixed hyperparameters are represented by grey circles. All nodes in the plate are specific for segment h. Unlike the original $\mathcal{M}_{0,0}$ model, whose graphical model is shown in Figure 3.2, the $\mathcal{M}_{1,1}$ model has a specific coupling parameter λ_h for each segment $h > 1$. Furthermore, there is a new Gamma hyperprior onto the second parameter of the Inverse Gamma prior on λ_h. The hyperprior allows for information exchange among the segment-specific coupling parameters $\lambda_2, \ldots, \lambda_H$.

We refer to the improved model as the $\mathcal{M}_{1,1}$ model. A graphical model representation of the relationships within and between segments $h > 1$ is provided in Figure 3.3. The posterior of the $\mathcal{M}_{1,1}$ model is

$$\begin{aligned}
&p(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_2, \ldots \lambda_H, \beta_c | \mathbf{y}_1, \ldots, \mathbf{y}_H) \\
&\qquad \propto p(\mathbf{y}_1, \ldots, \mathbf{y}_H, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_2, \ldots \lambda_H, \beta_c) \\
&\qquad \propto \prod_{h=1}^{H} p(\mathbf{y}_h | \sigma^2, \boldsymbol{\beta}_h) \cdot p(\boldsymbol{\beta}_1 | \sigma^2, \lambda_u) \\
&\qquad \quad \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h | \sigma^2, \lambda_h) \cdot p(\sigma^2) \cdot p(\lambda_u) \\
&\qquad \quad \cdot \prod_{h=2}^{H} p(\lambda_h | \beta_c) \cdot p(\beta_c)
\end{aligned} \tag{3.13}$$

3.1.3 Full conditional distributions of the improved generalized coupled model

In this subsection we derive the full conditional distributions for the $\mathcal{M}_{1,1}$ model, proposed in Subsection 3.1.2. For the derivations we exploit that the full conditional densities are proportional to the joint density, and thus proportional to the factorized joint density in Equation (3.13). From the shape of the densities we can conclude what the full conditional distributions are. For notational convenience, let $\{\lambda_h\}_{h\geq 2}$ denote the set of coupling parameters $\lambda_2, \dots, \lambda_H$ and let $\{\lambda_k\}_{k\neq h}$ denote the set of coupling parameters $\lambda_2, \dots, \lambda_{h-1}, \lambda_{h+1}, \dots, \lambda_H$ with parameter λ_h left out.

The full conditional distribution of $\boldsymbol{\beta}_h$ ($h = 1, \dots, H$) can be derived as follows. With $\lambda_1 := \lambda_u$ and $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$ we have:

$$\begin{aligned}
&p(\boldsymbol{\beta}_h|\mathbf{y}_1, \dots, \mathbf{y}_H, \{\boldsymbol{\beta}_k\}_{k\neq h}, \sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \\
&\quad \propto p(\mathbf{y}_1, \dots, \mathbf{y}_H, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \\
&\quad \propto p(\boldsymbol{\beta}_h|\lambda_h, \sigma^2) \cdot p(\mathbf{y}_h|\boldsymbol{\beta}_h, \sigma^2) \\
&\quad \propto e^{-\frac{1}{2}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^\mathsf{T}(\lambda_h\sigma^2\mathbf{I})^{-1}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})} \cdot e^{-\frac{1}{2}(\mathbf{y}_h - \mathbf{X}_h\boldsymbol{\beta}_h)^\mathsf{T}(\sigma^2\mathbf{I})^{-1}(\mathbf{y}_h - \mathbf{X}_h\boldsymbol{\beta}_h)} \\
&\quad \propto e^{-\frac{1}{2}\cdot\boldsymbol{\beta}_h^\mathsf{T}(\sigma^{-2}[\lambda_h^{-1}\cdot\mathbf{I} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h])\boldsymbol{\beta}_h \,+\, \boldsymbol{\beta}_h^\mathsf{T}\left(\sigma^{-2}(\lambda_h^{-1}\tilde{\boldsymbol{\beta}}_{h-1} + \mathbf{X}_h^\mathsf{T}\mathbf{y}_h)\right)}
\end{aligned}$$

and from the shape of the latter density it follows for the full conditional distribution:

$$\begin{aligned}
&\boldsymbol{\beta}_h|(\mathbf{y}_1, \dots, \mathbf{y}_H, \{\boldsymbol{\beta}_k\}_{k\neq h}, \sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \sim \\
&\quad \mathcal{N}\left([\lambda_h^{-1}\mathbf{I} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h]^{-1}(\lambda_h^{-1}\tilde{\boldsymbol{\beta}}_{h-1} + \mathbf{X}_h^\mathsf{T}\mathbf{y}_h)\ ,\ \sigma^2[\lambda_h^{-1}\mathbf{I} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h]^{-1}\right)
\end{aligned} \tag{3.14}$$

We note that the posterior expectation in Equation (3.14) is identical to the one which we used in Equation (3.12).
For the full conditional distributions of the segment-specific coupling parameters λ_h ($h = 2, \dots, H$) we get:

$$\begin{aligned}
&p(\lambda_h|\mathbf{y}_1, \dots, \mathbf{y}_H, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \{\lambda_k\}_{k\neq h}, \beta_c) \\
&\quad \propto p(\mathbf{y}_1, \dots, \mathbf{y}_H, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \\
&\quad \propto p(\lambda_h|\beta_c) \cdot p(\boldsymbol{\beta}_h|\sigma^2, \lambda_h) \\
&\quad \propto \frac{\beta_c^{\alpha_c}}{\Gamma(\alpha_c)}(\lambda_h^{-1})^{\alpha_c - 1} e^{-\beta_c\lambda_h^{-1}} \\
&\quad \cdot (2\pi)^{-\frac{k+1}{2}} \frac{1}{\sqrt{\det(\lambda_h\sigma^2\mathbf{I})}} e^{-\frac{1}{2}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^\mathsf{T}(\lambda_h\sigma^2\mathbf{I})^{-1}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})} \\
&\quad \propto (\lambda_h^{-1})^{\alpha_c + \frac{k+1}{2} - 1} \cdot e^{-\lambda_h^{-1}(\beta_c + \frac{1}{2}\sigma^{-2}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^\mathsf{T}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})}
\end{aligned}$$

and it follows from the shape of the full conditional density:

$$\begin{aligned}
&\lambda_h^{-1}|(\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\lambda_u,\{\lambda_k\}_{k\neq h},\beta_c) \\
&\qquad\sim GAM\left(\alpha_c+\frac{(k+1)}{2},\beta_c+\frac{1}{2}\sigma^{-2}(\boldsymbol{\beta}_h-\tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}}(\boldsymbol{\beta}_h-\tilde{\boldsymbol{\beta}}_{h-1})\right)
\end{aligned} \tag{3.15}$$

For the full conditional distribution of λ_u we get in a similar way:

$$\begin{aligned}
&p(\lambda_u|\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\{\lambda_h\}_{h\geq 2},\beta_c) \\
&\quad\propto p(\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\lambda_u,\{\lambda_h\}_{h\geq 2},\beta_c) \\
&\quad\propto p(\lambda_u)\cdot p(\boldsymbol{\beta}_1|\sigma^2,\lambda_u) \\
&\quad\propto (\lambda_u^{-1})^{\alpha_u-1}e^{-\beta_u\lambda_u^{-1}}\cdot\frac{1}{\sqrt{\det(\lambda_u\sigma^2\mathbf{I})}}e^{-\frac{1}{2}\boldsymbol{\beta}_1^{\mathsf{T}}(\lambda_u\sigma^2\mathbf{I})^{-1}\boldsymbol{\beta}_1} \\
&\quad\propto (\lambda_u^{-1})^{(k+1)/2+\alpha_u-1}\cdot e^{-\lambda_u^{-1}(\beta_u+0.5\sigma^{-2}\boldsymbol{\beta}_1^{\mathsf{T}}\boldsymbol{\beta}_1)}
\end{aligned}$$

The full conditional density has the shape of the inverse Gamma distribution in Equation (3.8), i.e. the full conditional distribution of λ_u stays unchanged:

$$\begin{aligned}
&\lambda_u^{-1}|(\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\{\lambda_h\}_{h\geq 2},\beta_c) \\
&\qquad\sim GAM\left(\alpha_u+\frac{1\cdot(k+1)}{2},\beta_u+\frac{1}{2}\sigma^{-2}\cdot\boldsymbol{\beta}_1^{\mathsf{T}}\boldsymbol{\beta}_1\right)
\end{aligned} \tag{3.16}$$

The new hyperparameter β_c can also be sampled from its full conditional distribution. The shape of

$$\begin{aligned}
&p(\beta_c|\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2\lambda_u,\{\lambda_h\}_{h\geq 2}) \\
&\quad\propto p(\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2\lambda_u,\{\lambda_h\}_{h\geq 2},\beta_c) \\
&\quad\propto p(\beta_c)\cdot\prod_{h=2}^{H}p(\lambda_h|\beta_c) \\
&\quad\propto \frac{b^a}{\Gamma(a)}\cdot\beta_c^{a-1}\cdot e^{-b\beta_c}\cdot\prod_{h=2}^{H}\left(\frac{\beta_c^{\alpha_c}}{\Gamma(\alpha_c)}\cdot\lambda_h^{\alpha_c-1}\cdot e^{-\beta_c\lambda_h^{-1}}\right) \\
&\quad\propto \beta_c^{a+(H-1)\alpha_c-1}\cdot e^{-(b+\sum_{h=2}^{H}\lambda_h^{-1})\beta_c}
\end{aligned}$$

implies for the full conditional distribution of β_c:

$$\begin{aligned}
&\beta_c|(\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\lambda_u,\{\lambda_h\}_{h\geq 2}) \\
&\qquad\sim GAM\left(a+(H-1)\cdot\alpha_c,b+\sum_{h=2}^{H}\lambda_h^{-1}\right)
\end{aligned} \tag{3.17}$$

For the noise variance parameter σ^2 we follow [24] and implement a collapsed Gibbs sampling step (with the $\boldsymbol{\beta}_h$'s integrated out). We have:

$$\begin{aligned} p(\mathbf{y}_h|\sigma^2, \lambda_h) = \int p(\mathbf{y}_h, \boldsymbol{\beta}_h|\sigma^2, \lambda_h) d\boldsymbol{\beta}_h &= \int p(\mathbf{y}_h|\boldsymbol{\beta}_h, \sigma^2, \lambda_h) p(\boldsymbol{\beta}_h|\sigma^2, \lambda_h) d\boldsymbol{\beta}_h \\ &= \int p(\mathbf{y}_h|\boldsymbol{\beta}_h, \sigma^2) p(\boldsymbol{\beta}_h|\sigma^2, \lambda_h) d\boldsymbol{\beta}_h \end{aligned}$$

A standard rule for Gaussian integrals (see, e.g., Section 2.3.2 in [5]) implies for the latter integral:

$$\mathbf{y}_h|(\sigma^2, \lambda_h) \sim \mathcal{N}(\mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2[\mathbf{I} + \lambda_h \mathbf{X}_h \mathbf{X}_h^{\mathsf{T}}]) \tag{3.18}$$

With $\lambda_1 := \lambda_u$, $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$, and using the marginal likelihood from Equation (3.18) we have:

$$\begin{aligned} & p(\sigma^2|\mathbf{y}_1, \dots, \mathbf{y}_H, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \\ & \quad \propto p(\sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c|\mathbf{y}_1, \dots, \mathbf{y}_H) \\ & \quad \propto p(\mathbf{y}_1, \dots, \mathbf{y}_H, \sigma^2, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \\ & \quad \propto \left(\prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2, \lambda_h) \right) \cdot p(\sigma^2) \cdot p(\lambda_u) \cdot \prod_{h=2}^{H} p(\lambda_h|\beta_c) \cdot p(\beta_c) \\ & \quad \propto \exp\{-\sigma^{-2}(\beta_\sigma + 0.5 \cdot \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}} (\mathbf{I} + \lambda_h \mathbf{X}_h \mathbf{X}_h^{\mathsf{T}})^{-1} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1}))\} \\ & \quad \cdot (\sigma^{-2})^{\alpha_\sigma + 0.5 \cdot T - 1} \end{aligned}$$

And the shape of the latter density implies the collapsed Gibbs sampling step (with the $\boldsymbol{\beta}_h$'s integrated out):

$$\begin{aligned} & \sigma^{-2}|(\mathbf{y}_1, \dots, \mathbf{y}_H, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c) \\ & \sim GAM\left(\alpha_\sigma + \frac{1}{2} \cdot T, \beta_\sigma + \frac{1}{2} \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}} \left(\mathbf{I} + \lambda_h \mathbf{X}_h \mathbf{X}_h^{\mathsf{T}} \right)^{-1} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1}) \right) \end{aligned} \tag{3.19}$$

where $\lambda_1 := \lambda_u$ and $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$.

For the marginal likelihood, with $\boldsymbol{\beta}_h$ ($h = 1, \dots, H$) and σ^2 integrated out, again the marginalization rule from Section 2.3.7 of [5] can be applied. For the improved model the marginalization rule implies:

$$\begin{aligned} & p(\mathbf{y}_1, \dots, \mathbf{y}_H|\lambda_u, \{\lambda_h\}_{h\geq 2}) \\ & \quad = \frac{\Gamma(\frac{T}{2} + a_\sigma)}{\Gamma(a_\sigma)} \cdot \frac{\pi^{-\frac{T}{2}} (2b_\sigma)^{a_\sigma}}{\left(\prod_{h=1}^{H} \det(\mathbf{C}_h) \right)^{1/2}} \\ & \quad \cdot \left(2b_\sigma + \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}} \mathbf{C}_h^{-1} (\mathbf{y}_h - \mathbf{X}_h \tilde{\boldsymbol{\beta}}_{h-1}) \right)^{-(\frac{T}{2} + a_\sigma)} \end{aligned} \tag{3.20}$$

where $\lambda_1 := \lambda_u$, $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$, and $\mathbf{C}_h := \mathbf{I} + \lambda_h \mathbf{X}_h \mathbf{X}_h^{\mathsf{T}}$,

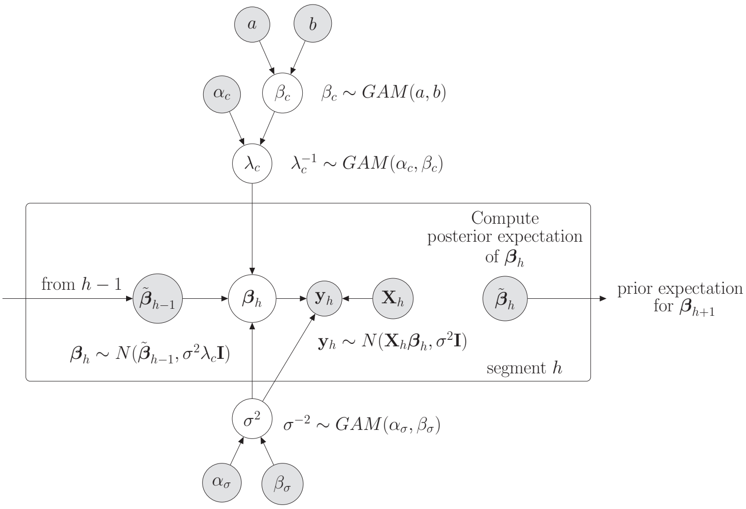

a
b
α_c
β_c
$\beta_c \sim GAM(a,b)$
λ_c
$\lambda_c^{-1} \sim GAM(\alpha_c, \beta_c)$
Compute posterior expectation of $\boldsymbol{\beta}_h$
from $h-1$
$\tilde{\boldsymbol{\beta}}_{h-1}$
$\boldsymbol{\beta}_h$
$\mathbf{y}_h$
$\mathbf{X}_h$
$\tilde{\boldsymbol{\beta}}_h$
prior expectation for $\boldsymbol{\beta}_{h+1}$
$\mathbf{y}_h \sim N(\mathbf{X}_h\boldsymbol{\beta}_h, \sigma^2\mathbf{I})$
$\boldsymbol{\beta}_h \sim N(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_c\mathbf{I})$
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 3.4: Graphical model for the 1st 'in between' model $\mathcal{M}_{1,0}$. See caption of Figure 3.2 for the terminology. The model is an extension of the original sequentially coupled model, whose graphical model is shown in Figure 3.2. Unlike the original $\mathcal{M}_{0,0}$ model, the $\mathcal{M}_{1,0}$ has a free hyperparameter, β_c, with a Gamma hyperprior.

3.1.4 Models 'in between' the original and the improved sequentially coupled model

In the last subsection we have proposed an improved sequentially coupled model, $\mathcal{M}_{1,1}$. Compared to the original model $\mathcal{M}_{0,0}$ from Subsection 3.1.1 we have proposed two modifications: (1) To replace the shared coupling parameter λ_c by segment-specific coupling parameters $\{\lambda_h\}_{h\geq 2}$, and (2) to impose a hyperprior onto the hyperparameter β_c of the inverse Gamma prior on the coupling parameters $\{\lambda_h\}_{h\geq 2}$. To shed more light onto the relative merits of the two individual modifications, we also define the two 'in between' models where only one of the two modifications is implemented.

The first 'in between' model $\mathcal{M}_{1,0}$ does not introduce segment-specific coupling parameters, but places a hyperprior onto the hyperparameter β_c of the Inverse Gamma prior on the shared coupling parameter λ_c. A graphical model representation for $\mathcal{M}_{1,0}$ is shown in Figure 3.4. The posterior distribution of the $\mathcal{M}_{1,0}$ model is an extension of Equation (3.7):

$$\begin{aligned} p(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_H, \sigma^2, \lambda_u, \lambda_c, \beta_c | \mathbf{y}_1, \ldots, \mathbf{y}_H) \quad \propto \quad & \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2, \boldsymbol{\beta}_h) \cdot p(\boldsymbol{\beta}_1|\sigma^2, \lambda_u) \\ & \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_c) \cdot p(\sigma^2) \\ & \cdot p(\lambda_u) \cdot p(\lambda_c|\beta_c) \cdot p(\beta_c) \end{aligned}$$

The modification does neither change the earlier defined full conditional distributions from Subsection 3.1.1 nor the marginal likelihood in Equation (3.10). The only difference is that β_c has become a free parameter which, thus, must now be sampled too. For the full conditional distribution of β_c in the $\mathcal{M}_{1,0}$ model we have:

$$\begin{aligned}
& p(\beta_c|\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2\lambda_u,\lambda_c) \\
& \qquad \propto p(\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2\lambda_u,\{\lambda_h\}_{h\geq 2},\beta_c|\mathbf{y}_1,\dots,\mathbf{y}_H) \\
& \qquad \propto p(\lambda_c|\beta_c)\cdot p(\beta_c) \\
& \qquad \propto \frac{\beta_c^{\alpha_c}}{\Gamma(\alpha_c)}\cdot\lambda_c^{\alpha_c-1}\cdot e^{-\beta_c\lambda_c^{-1}}\cdot\frac{b^a}{\Gamma(a)}\cdot\beta_c^{a-1}\cdot e^{-b\beta_c} \\
& \qquad \propto \beta_c^{a+\alpha_c-1}\cdot e^{-(b+\lambda_c^{-1})\beta_c}
\end{aligned}$$

This implies for the full conditional distribution of β_c:

$$\beta_c|(\mathbf{y}_1,\dots,\mathbf{y}_H,\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\lambda_u,\lambda_c)\sim GAM\left(a+\alpha_c,b+\lambda_c^{-1}\right) \tag{3.21}$$

The second 'in between' model $\mathcal{M}_{0,1}$ does make use of segment-specific coupling parameters $\{\lambda_h\}_{h\geq 2}$, but keeps the hyperparameter β_c of the Inverse Gamma priors on the parameters $\{\lambda_h\}_{h\geq 2}$ fixed. This yields that the segment-specific coupling parameters $\lambda_2,\dots,\lambda_H$ are independent a priori. A graphical model representation is shown in Figure 3.5. The posterior distribution of the 2nd 'in between' model $\mathcal{M}_{0,1}$ is a simplified version of Equation (3.13):

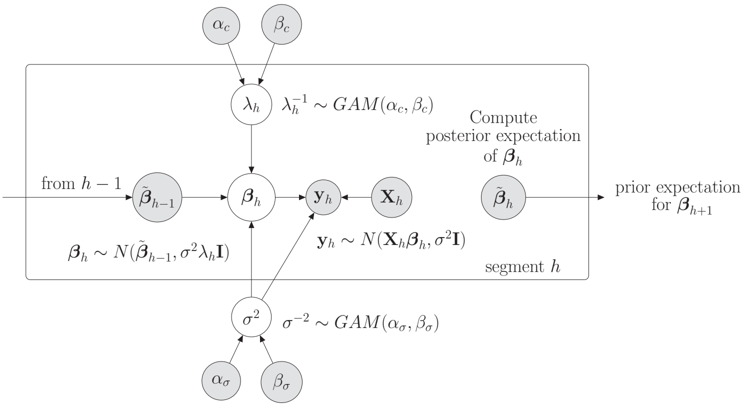
$$\begin{aligned}
p(\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_H,\sigma^2,\lambda_u,\{\lambda_h\}_{h\geq 2}|\mathbf{y}_1,\dots,\mathbf{y}_H) \quad \propto \quad & \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2,\boldsymbol{\beta}_h)\cdot p(\boldsymbol{\beta}_1|\sigma^2,\lambda_u) \\
& \cdot\prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2,\lambda_h) \\
& \cdot p(\sigma^2)\cdot p(\lambda_u)\cdot\prod_{h=2}^{H} p(\lambda_h)
\end{aligned}$$

and the modification (i.e. fixing β_c) does neither change the full conditional distributions in Equations (3.14), (3.15), (3.16) and (3.19) nor the marginal likelihood in Equation (3.20). The only difference is that β_c is kept fixed and will not be inferred from the data. The corresponding Gibbs sampling step (see Equation (3.21)) is never performed.

3.1.5 Learning the covariate set

In typical applications the covariates have to be inferred from the data. That is, there is a set of potential covariates and the subset, relevant for the target Y, has to be found.

Let $X_1,\dots,X_n$ be a set of *potential* covariates for the target variable Y, and let



α_c
β_c
λ_h
$\lambda_h^{-1} \sim GAM(\alpha_c, \beta_c)$
Compute posterior expectation of $\boldsymbol{\beta}_h$
from $h-1$
$\tilde{\boldsymbol{\beta}}_{h-1}$
$\boldsymbol{\beta}_h$
$\mathbf{y}_h$
$\mathbf{X}_h$
$\tilde{\boldsymbol{\beta}}_h$
prior expectation for $\boldsymbol{\beta}_{h+1}$
$\boldsymbol{\beta}_h \sim N(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_h\mathbf{I})$
$\mathbf{y}_h \sim N(\mathbf{X}_h\boldsymbol{\beta}_h, \sigma^2\mathbf{I})$
segment h
σ^2
$\sigma^{-2} \sim GAM(\alpha_\sigma, \beta_\sigma)$
α_σ
β_σ

Figure 3.5: Graphical model for the 2nd 'in between' model $\mathcal{M}_{0,1}$. See caption of Figure 3.3 for the terminology. The model is similar to the proposed improved sequentially coupled model, whose graphical model is shown in Figure 3.3. Unlike the proposed $\mathcal{M}_{1,1}$ model, the $\mathcal{M}_{0,1}$ model has a fixed hyperparameter β_c.

$\mathcal{D}_1, \ldots, \mathcal{D}_T$ be equidistant temporally ordered data points. Each $\mathcal{D}_t$ contains a target value y_t and the values $x_{1,t-1}, \ldots, x_{n,t-1}$ of the n potential covariates. A priori we assume that all covariate sets $\pi \subset \{X_1, \ldots, X_n\}$ with up to three covariates are equally likely, while all parent sets with more than three elements have zero prior probability.[1]

$$p(\pi) = \begin{cases} \frac{1}{c} & \text{if} |\pi| \leq 3 \\ 0 & \text{if} |\pi| > 3 \end{cases} \quad \text{where} \quad c = \sum_{i=0}^{3} \binom{n}{i}$$

We make the assumption that there cannot be more than 3 covariates ($|\pi| \leq 3$) with regard to our applications in the field of gene network inference, and we note that this assumption might be inappropriate for other applications. However, in the context of gene regulatory networks this assumption is very common. The known topologies of gene regulatory networks suggest that there are rarely genes that are regulated by more than three regulators. The assumption is thus biologically reasonable and has the advantage that it reduces the complexity of the problem and the computational costs of the Markov Chain Monte Carlo (MCMC) based model inference.

Given a fixed segmentation into the segments $h = 1, \ldots, H$, for each possible covariate set π the piecewise linear regression models can be applied. We focus our attention on the improved sequentially coupled model $\mathcal{M}_{1,1}$ from Subsection 3.1.2, but we note that the MCMC algorithm can also be used for generating samples for the competing models ($\mathcal{M}_{0,0}$, $\mathcal{M}_{1,0}$, and $\mathcal{M}_{0,1}$). Only the marginal likelihood expressions have to be replaced in the acceptance probabilities.

---

[1] To be consistent with earlier studies we assume all covariate sets containing up to three covariates to be equally likely.

Using the marginal likelihood from Equation (3.20), we obtain for the posterior of the $\mathcal{M}_{1,1}$ model:

$$\begin{aligned} p(\pi, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c | \mathcal{D}_1, \ldots, \mathcal{D}_T) \quad \propto \quad & p(\mathbf{y}_1, \ldots, \mathbf{y}_H | \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c, \pi) \\ & \cdot p(\pi) \cdot p(\lambda_u) \cdot \prod_{h=2}^{H} p(\lambda_h|\beta_c) \cdot p(\beta_c) \end{aligned} \tag{3.22}$$

Given λ_u, $\{\lambda_h\}_{h\geq 2}$, and β_c, the Metropolis-Hasting algorithm can be used to sample the covariate set π. We implement 3 moves: In the deletion move (D) we randomly select one $X_i \in \pi$ and remove it from π. In the addition move (A) we randomly select one $X_i \notin \pi$ and add it to π. In the exchange move (E) we randomly select one $X_i \in \pi$ and replace it by a randomly selected $X_j \notin \pi$. Each move yields a new covariate set $\pi^\star$, and we propose to replace the current π by $\pi^\star$. When randomly selecting the move type the acceptance probability for the proposed move is:

$$A(\pi, \pi^\star) \quad = \quad \min\left\{1, \frac{p(\mathbf{y}_1, \ldots, \mathbf{y}_H | \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c, \pi^\star)}{p(\mathbf{y}_1, \ldots, \mathbf{y}_H | \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c, \pi)} \cdot \frac{p(\pi^\star)}{p(\pi)} \cdot HR\right\}$$

$$\text{where } HR = \begin{cases} \frac{|\pi|}{n-|\pi^\star|} & \text{for (D)} \\ \frac{n-|\pi|}{|\pi^\star|} & \text{for (A)} \\ 1 & \text{for (E)} \end{cases}$$

n is the number of potential covariates $X_1, \ldots, X_n$, and $|.|$ denotes the cardinality.

3.1.6 Learning the segmentation

If the segmentation of the data is unknown, we can also infer it from the data. We assume that a changepoint set $\boldsymbol{\tau} := \{\tau_1, \ldots, \tau_{H-1}\}$ with $1 \leq \tau_h < T$ divides the data points $\mathcal{D}_1, \ldots, \mathcal{D}_T$ into disjunct segments $h = 1, \ldots, H$ covering $T_1, \ldots, T_H$ consecutive data points, where $\sum_{h=1}^{H} T_h = T$. Data point $\mathcal{D}_t$ $(1 \leq t \leq T)$ is in segment h if $\tau_{h-1} < t \leq \tau_h$, where $\tau_0 := 1$ and $\tau_H := T$. A priori we assume that the distances between changepoints are geometrically distributed with hyperparameter $p \in (0, 1)$ and that there cannot be more $H = 10$ segments.[2] This implies for the prior density:

$$p(\boldsymbol{\tau}) = \begin{cases} (1-p)^{\tau_H - \tau_{H-1}} \cdot \prod_{h=1}^{H-1} p \cdot (1-p)^{\tau_h - \tau_{h-1} - 1} & \text{if } |\boldsymbol{\tau}| \leq 9 \;\; \text{(i.e. } H \leq 10) \\ 0 & \text{if } |\boldsymbol{\tau}| > 9 \;\; \text{(i.e. } H > 10) \end{cases} \tag{3.23}$$

Let $\mathbf{y}_{\boldsymbol{\tau}} := \{\mathbf{y}_1, \ldots, \mathbf{y}_H\}$ be the segmentation, implied by the changepoint set $\boldsymbol{\tau}$.

---

[2] The assumption that there cannot be more than $H = 10$ segments is made with regard to our applications. Gene expression time series are often rather short; the gene expressions in yeast, described in Subsection 3.2.2, have been measured over $T = 33$ time points only. Restricting the number of segments H avoids segmentations whose individual segments are very short and uninformative.

Again we focus on the improved sequentially coupled model $\mathcal{M}_{1,1}$, and just note that the MCMC algorithm requires only minor adaptions when used for the three competing models, namely the original sequentially coupled model $\mathcal{M}_{0,0}$ (see Subsection 3.1.1) and the two 'in between' models $\mathcal{M}_{1,0}$ and $\mathcal{M}_{0,1}$ (see Subsection 3.1.4).

Using the marginal likelihood from Equation (3.20), the posterior of the improved model takes the form:

$$
\begin{aligned}
p(\pi, \boldsymbol{\tau}, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c | \mathcal{D}_1, \ldots, \mathcal{D}_T) \quad \propto \quad & p(\mathbf{y}_{\boldsymbol{\tau}} | \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c, \pi, \boldsymbol{\tau}) \\
& \cdot p(\pi) \cdot p(\boldsymbol{\tau}) \cdot p(\lambda_u) \cdot \prod_{h=2}^{H} p(\lambda_h | \beta_c) \cdot p(\beta_c)
\end{aligned}
\tag{3.24}
$$

For sampling the changepoint sets we also implement 3 Metropolis Hastings moves. Each move proposes to replace the current changepoint set $\boldsymbol{\tau}$ by a new changepoint set $\boldsymbol{\tau}^\star$, and $\boldsymbol{\tau}^\star$ implies a new data segmentation $\mathbf{y}^\star_{\boldsymbol{\tau}} := \{\mathbf{y}^\star_1, \ldots, \mathbf{y}^\star_{H^\star}\}$. The new segmentation $\mathbf{y}^\star_{\boldsymbol{\tau}}$ contains new segments h that have not been in $\mathbf{y}_{\boldsymbol{\tau}}$, symbolically $\mathbf{y}^\star_h \notin \mathbf{y}_{\boldsymbol{\tau}}$, and for each of those segments h we sample a new segment specific coupling parameter from the prior, $\lambda^\star_h \sim \text{INV-GAM}(\alpha_c, \beta_c)$. For all other segments we do not change the segment-specific coupling parameters. Let $\{\lambda^\star_h\}_{h\geq 2}$ denote the set of coupling parameters associated with the new segmentation $\mathbf{y}^\star_{\boldsymbol{\tau}}$.

In the birth move (B) we propose to set a new changepoint at a randomly selected location. The new changepoint set $\boldsymbol{\tau}^\star$ then contains $H^\star = H+1$ segments. The new changepoint is located in a segment h and divides it into two consecutive sub-segments h and $h+1$. For both we re-sample the segment-specific coupling parameters $\lambda^\star_h, \lambda^\star_{h+1} \sim \text{INV-GAM}(\alpha_c, \beta_c)$. In the death move (D) we randomly select one changepoint $\tau \in \boldsymbol{\tau}$ and delete it. The new changepoint set $\boldsymbol{\tau}^\star$ then contains $H^\star = H - 1$ segments. Removing a changepoint yields that two adjacent segments h and $h+1$ are merged into one single segment h, and we sample $\lambda^\star_h \sim \text{INV-GAM}(\alpha_c, \beta_c)$. In the reallocation move (R) we randomly select one changepoint $\tau \in \boldsymbol{\tau}$ and propose to re-allocate it to a randomly selected position in between the two surrounding changepoints. The re-allocated changepoint yields new bounds for two consecutive segments h (whose upper bound changes) and $h+1$ (whose lower bound changes), and for both segments we re-sample the coupling parameters $\lambda^\star_h, \lambda^\star_{h+1} \sim \text{INV-GAM}(\alpha_c, \beta_c)$.

When randomly selecting the move type, the acceptance probabilities for the move from the changepoints set $\boldsymbol{\tau}$ with segmentation $\mathbf{y}_{\boldsymbol{\tau}} := \{\mathbf{y}_1, \ldots, \mathbf{y}_H\}$ and coupling parameters $\{\lambda_h\}_{h\geq 2}$ to the changepoint set $\boldsymbol{\tau}^\star$ with the new segmentation $\mathbf{y}^\star_{\boldsymbol{\tau}} := \{\mathbf{y}^\star_1, \ldots, \mathbf{y}^\star_{H^\star}\}$ and the new coupling parameters $\{\lambda^\star_h\}_{h\geq 2}$ is:

$$
\begin{aligned}
&A([\boldsymbol{\tau}, \{\lambda_h\}_{h\geq 2}], [\boldsymbol{\tau}^\star, \{\lambda^\star_h\}_{h\geq 2}]) \\
&\qquad = \min\left\{1, \frac{p(\mathbf{y}_{\boldsymbol{\tau}^\star}|\lambda_u, \{\lambda^\star_h\}_{h\geq 2}, \beta_c, \pi, \boldsymbol{\tau}^\star)}{p(\mathbf{y}_{\boldsymbol{\tau}}|\lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c, \pi, \boldsymbol{\tau})} \cdot \frac{p(\boldsymbol{\tau}^\star)}{p(\boldsymbol{\tau})} \cdot HR\right\}
\end{aligned}
$$

$$
\text{where } HR = \begin{cases} \frac{T-1-|\boldsymbol{\tau}^*|}{|\boldsymbol{\tau}|} & \text{for (B)} \\ \frac{|\boldsymbol{\tau}^*|}{T-1-|\boldsymbol{\tau}|} & \text{for (D)} \\ 1 & \text{for (R).} \end{cases}
$$

T is the number of data points, and $|.|$ denotes the cardinality. We note that the prior ratio $\frac{p(\{\lambda^\star_h\}_{h\geq 2})}{p(\{\lambda_h\}_{h\geq 2})}$ has cancelled with the inverse proposal ratio (HR) for re-sampling the coupling parameters for the new segments.

To adapt the MCMC algorithm to the competing models, the marginal likelihood expressions in the acceptance probability have to be replaced. Moreover, for the two models ($\mathcal{M}_{0,0}$ and $\mathcal{M}_{1,0}$) with a shared coupling parameter λ_c, we follow [24] and implement the three changepoint moves such that they do not propose to re-sample λ_c.

3.1.7 Markov Chain Monte Carlo (MCMC) inference

For model inference we use a Markov Chain Monte Carlo (MCMC) algorithm. For the posterior distribution of the improved sequentially coupled model $\mathcal{M}_{1,1}$, described in Subsection 3.1.2, we have:

$$
\begin{aligned}
p(\pi, \boldsymbol{\tau}, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c | \mathcal{D}_1, \ldots, \mathcal{D}_T) \quad \propto \quad & p(\mathbf{y}_{\boldsymbol{\tau}}|\lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c, \pi, \boldsymbol{\tau}) \\
& \cdot p(\pi) \cdot p(\boldsymbol{\tau}) \cdot p(\lambda_u) \cdot \prod_{h=2}^{H} p(\lambda_h|\beta_c) \cdot p(\beta_c)
\end{aligned} \tag{3.25}
$$

We initialize all entities, e.g. $\pi = \{\}$, $\boldsymbol{\tau} = \{\}$, $\lambda_u = 1$, $\lambda_h = 1$ for $h > 1$, and $\beta_c = 1$, before we iterate between Gibbs and Metropolis-Hastings sampling steps:

Gibbs sampling part: We keep the covariate set π and the changepoint set $\boldsymbol{\tau}$ fixed, and we successively re-sample the parameters λ_u, λ_h ($h = 2, \ldots, H$), and β_c from their full conditional distributions. Although the parameters σ^2 and $\boldsymbol{\beta}_h$ ($h = 1, \ldots, H$) do not appear in the posterior above, the full conditionals of λ_u and $\lambda_2, \ldots, \lambda_H$ depend on instantiations of σ^2 and $\boldsymbol{\beta}_h$. The latter parameters thus have to be sampled first, but can then be withdrawn at the end of the Gibbs sampling part. The full conditional distributions have been derived in Subsection 3.1.3. With $\lambda_1 := \lambda_u$ and $\tilde{\boldsymbol{\beta}}_0 = \mathbf{0}$ we have:

(G.1) $\sigma^{-2} \sim GAM$
$\left(\alpha_\sigma + \frac{1}{2} \cdot T, \beta_\sigma + \frac{1}{2}\sum_{h=1}^{H}(\mathbf{y}_h - \mathbf{X}_h\tilde{\boldsymbol{\beta}}_{h-1})^\mathsf{T}\left(\mathbf{I} + \lambda_h\mathbf{X}_h\mathbf{X}_h^\mathsf{T}\right)^{-1}(\mathbf{y}_h - \mathbf{X}_h\tilde{\boldsymbol{\beta}}_{h-1})\right)$

(G.2) $\boldsymbol{\beta}_h \sim \mathcal{N}\left([\lambda_h^{-1}\mathbf{I} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h]^{-1}(\lambda_h^{-1}\tilde{\boldsymbol{\beta}}_{h-1} + \mathbf{X}_h^\mathsf{T}\mathbf{y}_h)\ ,\ \sigma^2[\lambda_h^{-1}\mathbf{I} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h]^{-1}\right)$
$(h = 1, \ldots, H)$

(G.3) $\lambda_u^{-1} \sim GAM\left(\alpha_u + \frac{1\cdot(k+1)}{2}, \beta_u + \frac{1}{2}\sigma^{-2} \cdot \boldsymbol{\beta}_1^\mathsf{T}\boldsymbol{\beta}_1\right)$

(G.4) $\lambda_h^{-1} \sim GAM\left(\alpha_c + \frac{(k+1)}{2}, \beta_c + \frac{1}{2}\sigma^{-2}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})^{\mathsf{T}}(\boldsymbol{\beta}_h - \tilde{\boldsymbol{\beta}}_{h-1})\right) \quad (h = 1, \dots, H)$

(G.5) $\beta_c \sim GAM\left(a + (H-1) \cdot \alpha_c, b + \sum_{h=2}^{H} \lambda_h^{-1}\right)$

We note that each Gibbs step yields parameter updates and that the subsequent full conditional distributions are always based on the newest parameter instantiations (sampled in the previous steps).

Metropolis-Hastings sampling part: We keep λ_u, λ_h $(h = 2, \dots, H)$, and β_c fixed, and we perform one Metropolis-Hastings move on the covariate set π and one Metropolis Hastings step on the changepoint set $\boldsymbol{\tau}$.

(M.1) We propose to change the covariate set $\pi \to \pi^\star$ by adding, deleting or exchanging one covariate. The new covariate set is accepted with probability $A(\pi, \pi^\star)$; see Subsection 3.1.5 for details. If accepted, we replace $\pi \leftarrow \pi^\star$. If rejected, we leave π unchanged.

(M.2) We propose to change the changepoint set $\boldsymbol{\tau} \to \boldsymbol{\tau}^\star$ by adding, deleting or reallocating one changepoint. Along with the changepoint set we propose to update coupling parameters, $\{\lambda_h\}_{h\geq 2} \to \{\lambda_h^\star\}_{h\geq 2}$. The new state is accepted with probability $A([\boldsymbol{\tau}, \{\lambda_h\}_{h\geq 2}], [\boldsymbol{\tau}^\star, \{\lambda_h\}_{h\geq 2}^\star])$; see Subsection 3.1.6 for details. If accepted, we replace $\boldsymbol{\tau} \leftarrow \boldsymbol{\tau}^\star$ and $\{\lambda_h\}_{h\geq 2} \leftarrow \{\lambda_h^\star\}_{h\geq 2}$. If rejected, we leave $\boldsymbol{\tau}$ and $\{\lambda_h\}_{h\geq 2}$ unchanged.

The MCMC algorithm, consisting of seven sampling steps (G.1-5) and (M.1-2) yields a posterior sample:

$$\{\pi^{(r)}, \boldsymbol{\tau}^{(r)}, \lambda_u^{(r)}, \{\lambda_h\}_{h\geq 2}^{(r)}, \beta_c^{(r)}\} \sim p(\pi, \boldsymbol{\tau}, \lambda_u, \{\lambda_h\}_{h\geq 2}, \beta_c | \mathcal{D}_1, \dots, \mathcal{D}_T) \qquad (r = 1, \dots, R)$$

We adapt the MCMC inference algorithm for the three alternative models ($\mathcal{M}_{0,0}$, $\mathcal{M}_{1,0}$, and $\mathcal{M}_{0,1}$) from Subsection 3.1.1 and 3.1.4. To this end, we modify the Gibbs- and Metropolis Hastings- steps as outlined in Subsections 3.1.4 to 3.1.6.

3.1.8 Learning dynamic networks

Dynamic network models are used for learning the regulatory interactions among variables from time series data. The standard assumption is that all regulatory interactions are subject to a time lag $\xi \in \mathbb{N}$. Here we assume that the time lag has the standard value $\xi = 1$. We further assume that the values of n random variables $Y_1, \dots, Y_n$ have been measured at T equidistant time points $t = 1, \dots, T$. Let $\mathbf{D}$ denote the n-by-T data matrix with $\mathbf{D}_{i,t}$ being the observed value of Y_i at time point t. The piecewise linear regression models, described in the previous subsections, can then be applied to each variable Y_i $(i = 1, \dots, n)$ separately.

In the i-th regression model Y_i is the target, and the potential covariates are the $\tilde{n} := n - 1$ remaining variables $Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_n$. Given the time lag ξ, the number of data points, which can be used for the regression model, reduces from T to $\tilde{T} := T - \xi$. For each target Y_i we have the data points $\mathcal{D}_{i,1}, \dots, \mathcal{D}_{i,\tilde{T}}$, and each data point $\mathcal{D}_{i,t}$ $(t = 1, \dots, \tilde{T})$ contains a target value $\mathbf{D}_{i,t+\xi}$ (i.e. the value of Y_i at time point $t + \xi$) and the values of the $\tilde{n}$ potential covariates:

$\mathbf{D}_{1,t}, \dots, \mathbf{D}_{i-1,t}, \mathbf{D}_{i+1,t}, \dots, \mathbf{D}_{N,t}$ (i.e the values of $Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_n$ at the shifted time point t).

The system of all n covariate sets $\{\pi_i\}_{i=1,\dots,n}$, where π_i is the covariate set for Y_i, can be thought of as a network. There is an edge $Y_j \rightarrow Y_i$ in the network if Y_j is a covariate for Y_i, symbolically $Y_j \in \pi_i$. There is no edge from Y_j to Y_i in the network if $Y_j \notin \pi_i$. We represent the resulting network in form of a n-by-n adjacency matrix $\mathcal{N}$ whose elements are binary, $\mathcal{N}_{j,i} \in \{0,1\}$. $\mathcal{N}_{j,i} = 1$ indicates that there is an edge from X_j to X_i (i.e. that $X_j \in \pi_i$).

For each Y_i we can generate a posterior sample, as described in Subsection 3.1.7. For each Y_i we then extract the covariate sets, $\pi_i^{(1)}, \dots, \pi_i^{(R)}$ from the sample, and use the covariate sets to build a sample of adjacency matrices $\mathcal{N}^{(1)}, \dots, \mathcal{N}^{(R)}$ where $\mathcal{N}_{j,i}^{(r)} = 1$ if $X_j \in \pi_i^{(r)}$ ($i, j \in \{1, \dots, n\}; r \in \{1, \dots, R\}$). The mean of the adjacency matrices:

$$\hat{\mathcal{N}} := \frac{1}{R} \sum_{r=1}^{R} \mathcal{N}^{(r)}$$

yields estimates of the marginal posterior probabilities that the individual edges are present. E.g. $\hat{\mathcal{N}}_{j,i} \in [0,1]$ is an estimate for the marginal probability that there is an edge from X_j to X_i (i.e. that $X_j \in \pi_i$). By imposing a threshold $\psi \in [0,1]$ on the edge probabilities, we get a concrete network prediction. The predicted network contains all edges $X_j \rightarrow X_i$ whose probability to be present is equal to or higher than ψ ($\hat{\mathcal{N}}_{j,i} \geq \psi$).

For applications where the true network is known, we can build the n-by-n adjacency matrix of the true network $\mathcal{T}$ with $\mathcal{T}_{j,i} \in \{0,1\}$ and $\mathcal{T}_{j,i} = 1$ if and only if the true network contains the edge $X_j \rightarrow X_i$. For each $\psi \in [0,1]$ we can then compute the recall $\mathcal{R}(\psi)$ and the precision $\mathcal{P}(\psi)$ of the predicted network:

$$\begin{aligned} \mathcal{R}(\psi) &= \frac{|\{X_j \rightarrow X_i | \mathcal{T}_{j,i} = 1, \hat{\mathcal{N}}_{j,i} \geq \psi\}|}{|\{X_j \rightarrow X_i | \mathcal{T}_{j,i} = 1\}|} \\ \mathcal{P}(\psi) &= \frac{|\{X_j \rightarrow X_i | \mathcal{T}_{j,i} = 1, \hat{\mathcal{N}}_{j,i} \geq \psi\}|}{|\{X_j \rightarrow X_i | \hat{\mathcal{N}}_{j,i} \geq \psi\}|} \end{aligned}$$

The curve $\{(\mathcal{R}(\psi), \mathcal{P}(\psi)) | 0 \leq \psi \leq 1\}$ is the precision recall curve ([11]). The area under the precision recall curve (AUC), which can be obtained by numerical integration, is a popular measure for the network reconstruction accuracy. The higher the AUC, the higher the accuracy of the predicted network.

3.1.9 Technical details of our simulation study

Table 3.1 provides an overview to the four models $\mathcal{M}_{i,j}$ ($i, j \in \{0,1\}$) under comparison. We re-use the hyperparameters from the works by [38] and [24], namely

$$\sigma^{-2} \sim GAM(\alpha_\sigma = 0.005, \beta_\sigma = 0.005) \text{ and } \lambda_u^{-1} \sim GAM(\alpha_u = 2, \beta_u = 0.2)$$

Model	coupling parameter(s) for $\boldsymbol{h \geq 2}$	hyperparameter	Graphical model	see Subsection
$\mathcal{M}_{0,0}$	shared: $\lambda_c \sim GAM(\alpha_c, \beta_c)$	β_c fixed	see Figure 3.2	3.1.1
$\mathcal{M}_{1,0}$	shared: $\lambda_c \sim GAM(\alpha_c, \beta_c)$	$\beta_c \sim GAM(a, b)$	see Figure 3.4	3.1.4
$\mathcal{M}_{0,1}$	segment-specific: $\lambda_h \sim GAM(\alpha_c, \beta_c)$	β_c fixed	see Figure 3.5	3.1.4
$\mathcal{M}_{1,1}$	segment-specific: $\lambda_h \sim GAM(\alpha_c, \beta_c)$	$\beta_c \sim GAM(a, b)$	see Figure 3.3	3.1.2

Table 3.1: Overview to the four model instantiations which we cross-compare. $\mathcal{M}_{0,0}$ is the sequentially coupled model from [24]. In this chapter we propose the improved $\mathcal{M}_{1,1}$ model, featuring two modifications. We also include the 'in-between' models ($\mathcal{M}_{0,1}$ and $\mathcal{M}_{1,0}$) with only one modification incorporated. The 1st subscript of $\mathcal{M}$ indicates whether there is a hyperprior on β_c (0=no,1=yes). The 2nd subscript of $\mathcal{M}$ indicates whether the model has segment-specific coupling parameters λ_h (0=no,1=yes).

For the models without hyperprior ($\mathcal{M}_{0,0}$ and $\mathcal{M}_{0,1}$) we further set:

$$\lambda_c^{-1}, \lambda_h^{-1} \sim GAM(\alpha_c = 2, \beta_c = 0.2)$$

while we use for the models with hyperprior ($\mathcal{M}_{1,0}$ and $\mathcal{M}_{1,1}$):

$$\begin{aligned} &\lambda_c^{-1}, \lambda_h^{-1} \sim GAM(\alpha_c = 2, \beta_c) \\ &\text{with } \ \beta_c \sim GAM(a = 0.2, b = 1) \ \text{ so that } \ E[\beta_c] = \frac{a}{b} = 0.2 \end{aligned}$$

For the four models we run the MCMC algorithms with $100,000$ iterations. Withdrawing the first 50% of the samples ('burn-in phase') and thinning out the remaining $50,000$ samples (from the 'sampling phase') by the factor 100, yields $R = 500$ samples from each posterior. To check for convergence, we applied diagnostics based on trace plot and potential scale reduction factors (PSRFs) diagnostics, see, e.g., [20]. All diagnostics indicated perfect convergence for the above setting.

3.2 Data

3.2.1 Synthetic RAF-pathway data

For our cross-method comparison we generate synthetic network data from the RAF pathway, as reported in [50]. The RAF-pathway shows the regulatory interactions among the following $n = 11$ proteins: Y_1: PIP3, Y_2: PLCG, Y_3: PIP2, Y_4: PKC, Y_5: PKA, Y_6: JNK, Y_7: P38, Y_8: RAF, Y_9: MEK, Y_{10}: ERK, and Y_{11}: AKT. There are 20 regulatory interactions (directed edges) in the RAF pathway. We extract the true 11-by-11 adjacency matrix $\mathcal{T}$ where $\mathcal{T}_{j,i} = 1$ if there is an edge from the jth to the ith protein. We get:

$$\mathcal{T} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The covariate set of variable Y_i is then: $\pi_i = \{Y_j : \mathcal{T}_{j,i} = 1\}$. We follow [24] and generate synthetic data sets with $H = 4$ segments having m data points each. For each Y_i we thus require 4 segment-specific regression coefficient vectors $\boldsymbol{\beta}_{i,1}, \ldots, \boldsymbol{\beta}_{i,4}$ each being of length $|\pi_i| + 1$. Given those vectors, we generate 11-by-$(4m + 1)$ data matrices $\mathbf{D}$, where $\mathbf{D}_{i,t}$ is the value of Y_i at t. Let $\mathbf{D}_{.,t}$ denote the tth column of $\mathbf{D}$ (i.e. the values of the variables at t) and let $\mathbf{D}_{\pi_i,t}$ denote the subvector of $\mathbf{D}_{.,t}$ containing only the values of the $|\pi_i|$ covariates of Y_i. We randomly sample the values of $\mathbf{D}_{.,1}$ from independent Gaussian distributions with mean 0 and variance $\sigma^2 = 0.025$, before we successively generate data for the next time points. $\mathbf{D}_{i,t}$ ($i = 1, \ldots, 11; t = 2, \ldots, 4m+1$) is generated as follows:

$$\mathbf{D}_{i,t} = (1, \mathbf{D}^{\mathsf{T}}_{\pi_i,t-1}) \cdot \boldsymbol{\beta}_{i,H(t)} + \epsilon_{i,t}, \tag{3.26}$$

where the $\epsilon_{i,t}$ are iid $\mathcal{N}(0, \sigma^2)$ distributed noise variables, and $H(t)$ is a step function, indicating the segment to which time point t belongs:

$$H(t) = \begin{cases} 1, & 2 \leq t \leq m+1 \\ 2, & m+2 \leq t \leq 2m+1 \\ 3, & 2m+2 \leq t \leq 3m+1 \\ 4, & 3m+2 \leq t \leq 4m+1 \end{cases}.$$

We sample the regression coefficient vectors for the first segment $h = 1$, $\boldsymbol{\beta}_{i,1}$ ($i = 1, \ldots, 11$), from independent standard Gaussian distributions and normalize each vector to Euclidean norm 1: $\boldsymbol{\beta}_{i,1} \leftarrow \frac{\boldsymbol{\beta}_{i,1}}{|\boldsymbol{\beta}_{i,1}|}$. For the segments $h > 1$ we change the vector from the previous segment, $\boldsymbol{\beta}_{i,h-1}$, as follows:

(U) Either we change the regression coefficients drastically by flipping their signs, $\boldsymbol{\beta}_{i,h} = (-1) \cdot \boldsymbol{\beta}_{i,h-1}$. We then say that segment h is uncoupled ('U') from segment $h-1$; i.e. $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ are very dissimilar.

(C) Or we change the regression coefficients moderately. To this end, we first sample the entries of a new vector $\boldsymbol{\beta}_{i,\star}$ from independent standard Gaussians, $\mathcal{N}(0,1)$, and normalize $\boldsymbol{\beta}_{i,\star}$ to Euclidean norm ϵ, $\boldsymbol{\beta}_{i,\star} \leftarrow \epsilon \cdot \frac{\boldsymbol{\beta}_{i,\star}}{|\boldsymbol{\beta}_{i,\star}|}$, where ϵ is a tuning parameter. Then we add the new vector to the vector $\boldsymbol{\beta}_{i,h-1}$ and re-normalize the result to Euclidean norm 1:

$$\boldsymbol{\beta}_{i,h} := \frac{\boldsymbol{\beta}_{i,h-1} + \boldsymbol{\beta}_{i,\star}}{|\boldsymbol{\beta}_{i,h-1} + \boldsymbol{\beta}_{i,\star}|}.$$

We then say that segment h is coupled ('C') to segment $h-1$; i.e. $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ are similar.

We use the symbolic notation: '$C - U - C$' to indicate that segment 2 is coupled to segment 1, segment 3 is uncoupled from segment 2, and segment 4 is coupled to segment 3. In our simulation study we consider all possible scenarios '$S_2 - S_3 - S_4$' with $S_h \in \{C, U\}$ ($h = 2, 3, 4$), where $S_h = U$ ($S_h = C$) indicates that segment h is uncoupled from (coupled to) segment $h-1$; i.e. the regression coefficient vectors $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ are dissimilar (similar).

For coupled segments ('C') the parameter ϵ regulates how similar the vectors $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ are. For our first study we set $\epsilon = 0.25$. In a follow-up study we then investigate the effect of ϵ, and vary this parameter ($\epsilon \in \{0, 0.25, 0.5, 1\}$).

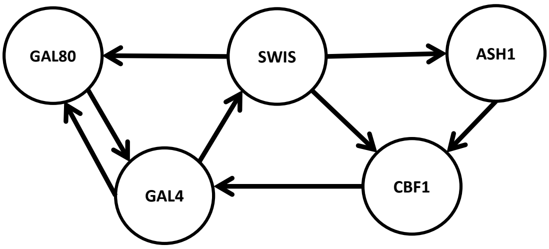


Figure 3.6: The true yeast network, as synthetically designed by [8].

3.2.2 Yeast gene expression data

[8] synthetically generated a small network of $n = 5$ genes in *Saccharomyces cerevisiae* (yeast), depicted in Figure 3.6. The five genes are: $CBF1$, $GAL4$, $SWI5$, $GAL80$, and $ASH1$. The network among those genes was obtained from synthetically designed yeast cells grown with different carbon sources: galactose ("switch on") or glucose ("switch off"). [8] obtained in vivo data with quantitative real-time RT-PCR in intervals of 20 minutes up to 5 hours for the first, and in intervals of 10 minutes up to 3 hours for the second condition. This led to the sample sizes $T_1 = 16$ ("switch on") and $T_2 = 21$ ("switch off"). We follow [24] and pre-process the data, $(\mathbf{D}^{(1)}_{.,1}, \dots, \mathbf{D}^{(1)}_{.,16})$ and $(\mathbf{D}^{(2)}_{.,1}, \dots, \mathbf{D}^{(2)}_{.,21})$, where $\mathbf{D}^{(c)}_{.,t}$ is the t-th observation (vector) of the c-th condition ($c = 1, 2$), as follows: For both conditions we withdraw the initial measurements $\mathbf{D}^{(1)}_{.,1}$ and $\mathbf{D}^{(2)}_{.,1}$, as they were taken while extant glucose (galactose) was washed out and new galactose (glucose) was supplemented. This leaves us with the data vectors: $\mathbf{D}^{(1)}_{.,2}, \dots, \mathbf{D}^{(1)}_{.,16}, \mathbf{D}^{(2)}_{.,2}, \dots, \mathbf{D}^{(2)}_{.,21}$, which we standardize via a log transformation and a subsequent gene-wise mean standardization (to mean 0). We also take into account that $\mathbf{D}^{(2)}_{.,2}$ has no proper relation with $\mathbf{D}^{(1)}_{.,16}$. For each target gene Y_i with covariate set π_i we therefore only use $\tilde{T} = T_1 + T_2 - 4 = 33$ target $\mathbf{D}^{(1)}_{i,3}, \dots, \mathbf{D}^{(1)}_{i,16}, \mathbf{D}^{(2)}_{i,3}, \dots, \mathbf{D}^{(2)}_{i,21}$ and covariate values $\mathbf{D}^{(1)}_{\pi_i,2}, \dots, \mathbf{D}^{(1)}_{\pi_i,15}, \mathbf{D}^{(2)}_{\pi_i,2}, \dots, \mathbf{D}^{(2)}_{\pi_i,20}$.

3.3 Empirical results

3.3.1 Results for synthetic RAF pathway data

In our first empirical evaluation study we cross-compare the network reconstruction accuracies of the four models $\mathcal{M}_{i,j}$ ($i, j \in \{0, 1\}$), listed in Table 3.1, on synthetic RAF pathway data, generated as described in Subsection 3.2.1. In this study we assume the data segmentation into $H = 4$ segments to be known. We can then set the three changepoints at the right locations and we do not perform MCMC moves on the changepoint set $\boldsymbol{\tau}$. That is, we keep $\boldsymbol{\tau}$ fixed

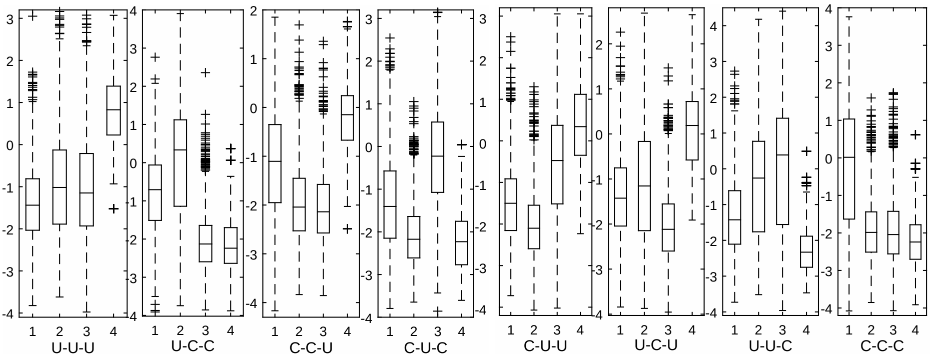


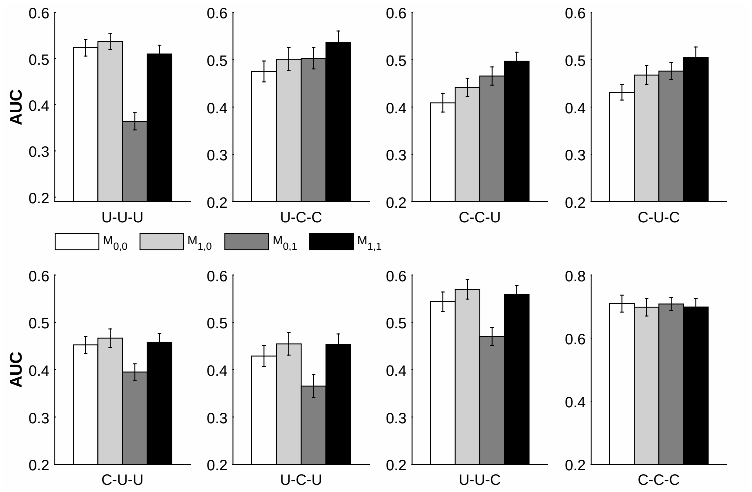
Figure 3.7: Boxplots of the logarithmic segment specific coupling parameters for the RAF pathway data. We generated data with $H = 4$ segments and $m = 10$ data points per segment, and we distinguished eight coupling scenarios of the type: '$S_2 - S_3 - S_4$' with $S_h \in \{U, C\}$. For each scenario there is a panel with four boxplots. The boxplots indicate how the logarithmic sampled coupling parameters $\log(\lambda_1) := \log(\lambda_u)$, $\log(\lambda_2)$, $\log(\lambda_3)$, and $\log(\lambda_4)$ are distributed for the given scenario. For a compact representation we decided to merge all samples taken for $N = 11$ variables from independent MCMC simulations on 25 independent data instantiations. In each panel the h-th boxplot from the left refers to $\log(\lambda_h)$. Our focus is on λ_h with $h > 1$, and it can be seen that the coupling parameters for coupled segments ($S_h = C$) are lower than for uncoupled segments ($S_h = U$). For $m = 5$ data points per scenario we observed the same trends (boxplots not shown).

during the MCMC simulations. The corresponding moves on $\boldsymbol{\tau}$, described in Subsection 3.1.6, are skipped.

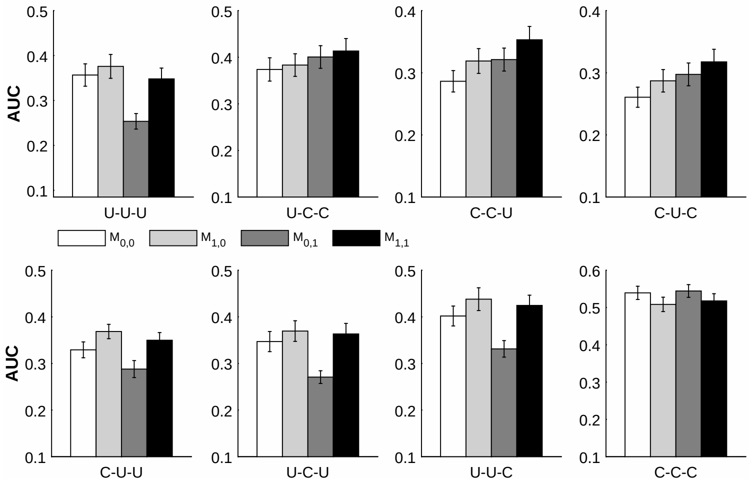
In our first study we generate data for eight different coupling scenarios of the form '$S_2 - S_3 - S_4$' with $S_i \in \{U, C\}$ where $X_h = U$ indicates that segment h is uncoupled from segment $h - 1$ (i.e. $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ are dissimilar), and $S_h = C$ indicates that segment h is coupled to segment $h - 1$ (i.e. $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ are similar). For the technical details we refer to Subsection 3.2.1.

First we perform a sanity check for the proposed $\mathcal{M}_{1,1}$ model: We investigate whether it actually infers different coupling parameters for the segments and whether the segment-specific coupling parameter distributions are consistent with the underlying coupling schemes of the form '$S_2 - S_3 - S_4$' . For uncoupled segments with $S_h = U$ the coupling parameters should on average be greater than for coupled segments with $S_h = C$ ($h = 2, 3, 4$). Figure 3.7 shows boxplots of the inferred segment-specific coupling parameters $\lambda_1 (= \lambda_u)$, λ_2, λ_3 and λ_4. Focusing on λ_h with $h = 2, 3, 4$, it can be seen from the boxplots that the coupling parameters for coupled segments (where $S_h = C$) are consistently lower than for uncoupled segments (where $S_h = U$).

The AUC results, provided in Figure 3.8, show that the proposed generalized model ($\mathcal{M}_{1,1}$) shows, overall, the best performance. It is always among the best models and it never performs substantially worse than any other model. On



(a) Scenarios of the type '$S_2 - S_3 - S_4$' with $H = 4$ and $m = 10$.



(b) Scenarios of the type '$S_2 - S_3 - S_4$' with $H = 4$ and $m = 5$.

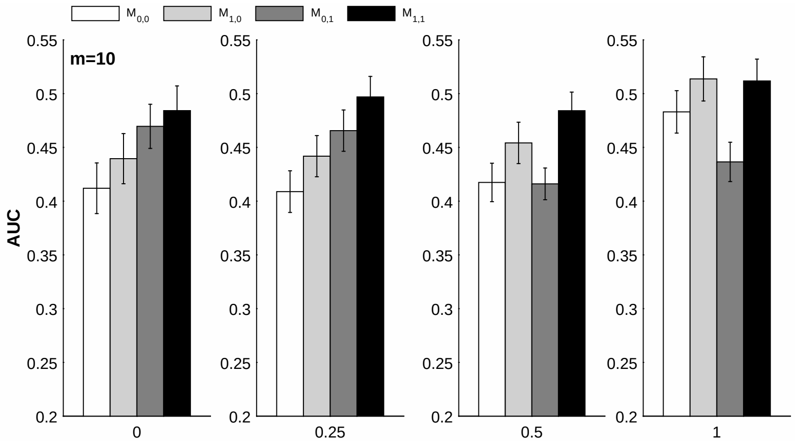
Figure 3.8: Network reconstruction accuracy for the RAF-pathway data. For the RAF pathway we generated synthetic data with $H = 4$ segments and with $m \in \{5, 10\}$ data points per segment. For both m we distinguished 8 coupling scenarios of the type: '$S_2 - S_3 - S_4$' (with $S_h \in \{U, C\}$). For coupled ('C') segments we set the parameter ϵ to 0.25, see Subsection 3.2.1 for details. The histogram bars correspond to the model-specific average precision-recall AUC values, averaged across 25 MCMC simulations. The errorbars correspond to standard deviations.

the other hand, the proposed $\mathcal{M}_{1,1}$ model outperforms its competitors for some settings. Especially for scenarios where 2 out of 3 segments with $h > 1$ are coupled to the previous segment. For the scenarios '$C - C - U$', '$U - C - C$' and '$C - U - C$' the proposed model performs better than the three other models. When comparing the two in-between models ($\mathcal{M}_{1,0}$ and $\mathcal{M}_{0,1}$) with the original $\mathcal{M}_{0,0}$ model from [24] it becomes obvious that imposing a hyperprior on β_c, as implemented in the $\mathcal{M}_{1,0}$ model, almost consistently improves the AUC scores, while making the coupling parameter segment-specific, as done in the $\mathcal{M}_{0,1}$ model, can lead to deteriorations of the AUC scores which is consistent with the results in chapter 2. We draw the conclusion that replacing the coupling parameter λ_c by segment-specific parameters $\lambda_2, \ldots, \lambda_4$ is counter-productive, unless this modification is combined with a hyperprior so that information can be shared among the segment-specific coupling strengths. Just imposing a hyperprior, which then only allows to adjust the prior for the coupling strength parameter λ_c (in light of the data), also improves the network reconstruction accuracy; but the improvement is slightly minor to the improvement that can be achieved by implementing both modifications together, as proposed in this paper.

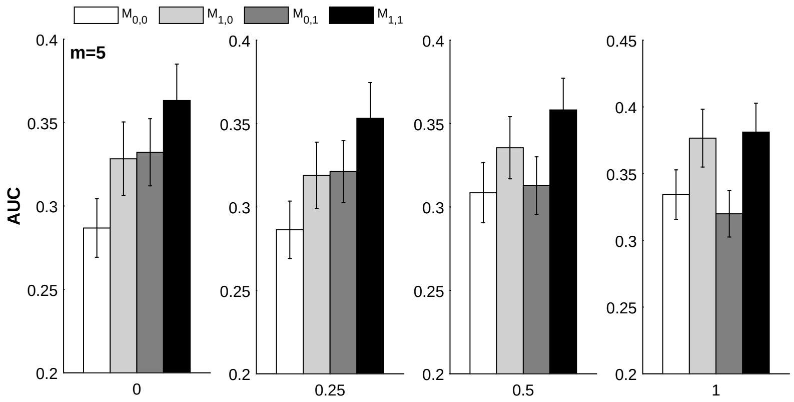
In a follow-up study we then had a closer look at the eight scenarios and varied the tuning parameter $\epsilon \in \{0, 0.25, 0.5, 1\}$ for each of them. With the parameter ϵ the similarity of the regression parameters (i.e. the coupling strength between coupled segments) can be adjusted. The greater ϵ, the weaker the similarity of the regression coefficient $\boldsymbol{\beta}_{i,h}$ and $\boldsymbol{\beta}_{i,h-1}$ of coupled segments; see Subsection 3.2.1 for the mathematical details. As an example, Figure 3.9 shows the results for the scenario: '$C - C - U$', which belongs to the scenarios where the proposed $\mathcal{M}_{1,1}$ model was found to outperform its competitors (see Figure 3.8). We can see from the AUC results in Figure 3.9 that $\epsilon = 0$ and $\epsilon = 0.5$ yield the same trends, as observed earlier for $\epsilon = 0.25$; see Figure 3.8. But for the highest ϵ ($\epsilon = 1$) $\mathcal{M}_{1,0}$ and $\mathcal{M}_{1,1}$ perform equally well and better than the other two models ($\mathcal{M}_{0,0}$ and $\mathcal{M}_{0,1}$). The explanation for this finding is most likely as follows: The similarity of the coupled regression coefficients decreases in ϵ. Hence, for $\epsilon = 1$ even the regression parameters for the two coupled segments get very dissimilar. Thus, $\epsilon = 1$ implies that all four segment-specific regression coefficients $\boldsymbol{\beta}_{i,h}$ ($h = 1, \ldots, 4$) are dissimilar and there is no more need for segment-specific coupling parameters. The reason why for $\epsilon = 1$ the original $\mathcal{M}_{0,0}$ model and the $\mathcal{M}_{0,1}$ model are inferior to the models which possess hyperpriors is probably the following one: The models $\mathcal{M}_{0,0}$ and $\mathcal{M}_{0,1}$ cannot adjust the prior of the coupling parameter (in light of the data). As a consequence they are likely to over-penalize dissimilar regression coefficients (i.e. high coupling parameters) through the prior.

3.3.2 Results for the yeast gene expression data

In this subsection we cross-compare the network reconstruction accuracies of the four models $\mathcal{M}_{i,j}$ ($i, j = 0, 1$), listed in Table 3.1, on the yeast gene expression data,



(a) $'C-C-U'$ with $m=10$, $H=4$ and $\epsilon \in \{0, 0.25, 0.5, 1\}$.



(b) $'C-C-U'$ with $m=5$, $H=4$ and $\epsilon \in \{0, 0.25, 0.5, 1\}$.

Figure 3.9: Network reconstruction accuracy for the RAF-pathway data. Unlike in Figure 3.8, here we focus on the scenario: $'C-C-U'$, and we vary the tuning parameter $\epsilon \in \{0, 0.25, 0.5, 1\}$. Like in Figure 3.8, the model-specific bars and errorbars correspond to the average AUC values and their standard deviations.

described in Subsection 3.2.2. For this application we infer the data segmentation (i.e. the changepoint set $\boldsymbol{\tau}$) along with the network structure from the data.

To vary the segmentations and especially the number of segments (i.e. the number of changepoints in $\boldsymbol{\tau}$), we implement the four models with 8 different hyperparameters $p \in \{0.00625, 0.0125, 0.025, 0.05, 0.1, 0.2, 0.4, 0.8\}$ of the geometric prior on the distance between changepoints; see Equation (3.23) in Subsection 3.1.6 for the mathematical details. We note that the hyperparameters are of the form: $p = 0.1 \cdot 2^i$ with $i = -4, -3, \ldots, 3$.

Figure 3.10 shows histograms of the inferred posterior distributions of the

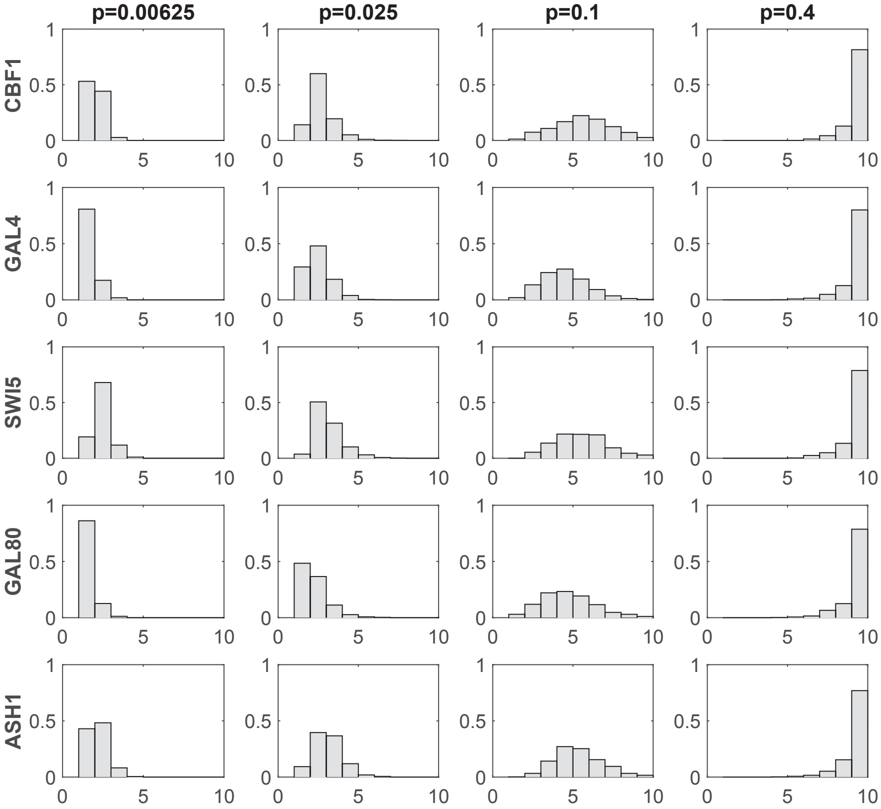


Figure 3.10: Posterior distribution of the numbers of segments H for the yeast data from Subsection 3.2.2. We implemented the models with different hyperparameters p for the geometric distribution on the distance between changepoints. For the proposed $\mathcal{M}_{1,1}$ model the histograms show how the posterior distributions of the numbers of segments H varies with the hyperparameter p. Each row refers to a gene of the yeast network; the four columns refer to the hyperparameters $p \in \{0.00625, 0.025, 0.1, 0.4\}$. For each number of segments $1 \leq H \leq 10$ the bars give the relative frequencies with which the data of the corresponding gene were segmented into H segments in the posterior samples. The relative frequencies are averaged over 25 independent MCMC simulations.

numbers of segments H for the proposed $\mathcal{M}_{1,1}$ model from Subsection 3.1.2. It can be clearly seen that the inferred segmentations strongly depend on the hyperparameter p. For the lowest p the data are rarely divided into more than $H = 2$ segments, while the posterior for $p = 0.4$ peaks at the imposed maximum of $H = 10$ segments. For the other three models we observed almost identical trends (histograms not shown in this chapter).

Figure 3.11 shows how the empirical network reconstruction accuracy, quantified in terms of the average areas under the precision recall curve (AUC) values,

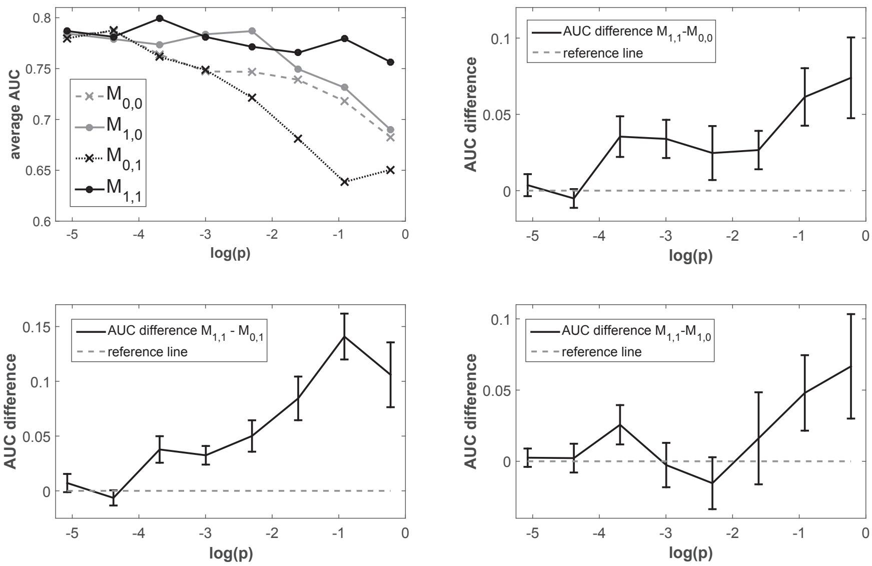


Figure 3.11: Network reconstruction accuracy for the yeast gene expression data from Subsection 3.2.2. For our study we implemented the four models, listed in Table 3.1 with 8 different hyperparameters $p = 0.1 \cdot 2^i$ ($i \in \{-4, -3, \ldots, 3\}$) for the geometric distribution on the distance between changepoints; see Equation (3.23). The upper left panel shows the average precision-recall AUC scores, averaged across 25 MCMC simulations. The other panels show the relative AUC differences in favor of the $\mathcal{M}_{1,1}$ model, with error bars indicating 95% t-test confidence intervals.

varies with the hyperparameter p. From the upper left panel of Figure 3.11, which shows the average AUC scores, the following trends can be observed:

- The $\mathcal{M}_{0,1}$ model (which has segment-specific coupling parameters $\lambda_2, \ldots, \lambda_H$, but does not couple them by a Gamma hyperprior) performs worse than the original sequentially coupled model from [24]. Only for very small hyperparameters $p \leq 0.05$ (i.e. when few changepoints are learned) the two models $\mathcal{M}_{0,0}$ and $\mathcal{M}_{0,1}$ reach approximately the same AUC scores. For higher hyperparameters the introduction of segment-specific coupling parameters has a counter-productive effect, and is, thus, not recommended.

- The $\mathcal{M}_{1,0}$ model, which leaves the coupling parameter λ_c shared among segments and only imposes a hyperprior to adjust the prior on λ_c (in light of the data), performs better than the original model from [24]. For very small hyperparameters $p \leq 0.0125$ (i.e. when very few changepoints are learned) the two models $\mathcal{M}_{0,0}$ and $\mathcal{M}_{1,0}$ reach approximately the same AUC scores. For the 6 higher hyperparameters $p > 0.0125$ the introduction of the hyperprior consistently leads to improved network reconstruction accuracies.

- The $\mathcal{M}_{1,1}$ model, which we propose in this paper, has both modifications implemented: It has segment-specific coupling parameters $\lambda_2, \ldots, \lambda_H$ and couples those parameters by a Gamma hyperprior; see Subsection 3.1.2 for a detailed model description. The combination of both modifications yields that the $\mathcal{M}_{1,1}$ model performs overall best. Its performance even stays stable for rather large hyperparameters p where the AUC scores of the other three models substantially diminish. The AUC difference plots in Figure 3.11 show that most of the improvements are statistically significant in terms of paired t-tests. In particular, the $\mathcal{M}_{1,1}$ model performs better than the original $\mathcal{M}_{0,0}$ model, proposed in [24], for 6 out of 8 hyperparameters p, namely for all $p > 0.0125$.

Based on our network reconstruction accuracy results for the real gene expression data, we conclude that the performance of the proposed generalized model ($\mathcal{M}_{1,1}$) is superior to the performance of the original sequentially coupled model ($\mathcal{M}_{0,0}$). The empirical results obtained with the two 'in between' models ($\mathcal{M}_{1,0}$ and $\mathcal{M}_{0,1}$) suggest further that the main contribution stems from the hyperprior. The capability to adjust the coupling parameter prior in light of the data boosts the performance. The model $\mathcal{M}_{0,1}$, whose segment-specific coupling parameters are not coupled by a hyperprior, led to decreased AUC results. The results, obtained for the yeast gene expression data, are hence in agreement with the earlier results obtained for synthetic network data; see Subsection 3.3.1.

3.4 Discussion and conclusions

In this chapter we have proposed an improved version of NH-DBN model, proposed in [24], in order to refine the generalized coupled model introduced in previous chapter. Unlike the original $\mathcal{M}_{0,0}$ model, our new $\mathcal{M}_{1,1}$ model possesses segment-specific coupling (strength) parameters and a hyperprior on the coupling parameter priors; see Subsection 3.1.2 for the mathematical details. Replacing the shared coupling parameter λ_c by segment-specific coupling parameters $\lambda_1, \ldots, \lambda_H$ increases the model flexibility, while the new hyperprior allows for information-exchange among segments as well as to adjust the coupling parameter prior(s) in light of the data. Our empirical evaluation studies on synthetic RAF pathway data (see Subsection 3.3.1) and on yeast gene expression data (see Subsection 3.3.2) have shown that the new $\mathcal{M}_{1,1}$ model leads to improved network reconstruction accuracies. To gain more insight into the merits of the two individual modifications, we also compared with the performances of the two 'in-between' models ($\mathcal{M}_{1,0}$ and $\mathcal{M}_{0,1}$), which we defined to be subject to only one of the two modifications; see Table 3.1 for an overview to the four models $\mathcal{M}_{i,j}$ ($i, j \in \{0, 1\}$) under comparison. Overall, the proposed $\mathcal{M}_{1,1}$ has reached the highest network reconstruction accuracies among the 4 models. The $\mathcal{M}_{0,1}$ model, which we defined to have segment specific coupling parameters but no hyperprior, performed worse than the original $\mathcal{M}_{0,0}$ model. The $\mathcal{M}_{1,0}$ model, which we defined to have a shared coupling parameter with a hyperprior, performed better than the original $\mathcal{M}_{0,0}$ model and for some scenarios comparable to the proposed $\mathcal{M}_{1,1}$ model. This shows that the major part of the improvement, achieved with the proposed $\mathcal{M}_{1,1}$, stems from imposing a hyperprior onto the coupling parameter prior(s).

To put it in a nutshell, our empirical results show that the model variant $\mathcal{M}_{1,1}$ reaches, overall, the highest network reconstruction accuracies. With regard to future applications, we therefore recommend giving precedence to this model. Moreover, our results for the yeast gene expression data (see Figure 3.11) also suggest that only the $\mathcal{M}_{1,1}$ model is robust with respect to the changepoint process hyperparameter. The network reconstruction accuracies of the other models deteriorate, as the number of inferred changepoints increases. Only the network reconstruction accuracy of the $\mathcal{M}_{1,1}$ model stays high, even if the data are divided into short (uninformative) segments. This is a very important property for applications where the underlying segmentation is unknown and has to be inferred from the data. The number of inferred changepoints (i.e. the data segmentation) strongly depends on the changepoint process hyperparameter (see Figure 3.10). In the absence of any genuine prior knowledge, the changepoint process hyperparameter can easily be misspecified. Non-robust models will then output biased results what might lead to erroneous conclusions.

Our future work will aim to transfer the concept of segment-specific coupling parameters to the globally coupled NH-DBN model from [25]. Unlike the sequential coupling mechanism, which requires a temporal ordering of the segments, the global coupling mechanism treats all segments as interchangeable units. When segmenting a single time series by changepoints, the assumption of interchangeable segments is often not appropriate. But there are other applications in systems biology where data stem from different experiments so that the segments might contain data from different experimental conditions. The segments then do not have any natural order and the sequential coupling scheme should be replaced by the global coupling scheme.

Chapter 4

Partially edge-wise coupled NH-DBNs

In the standard ‘uncoupled’ NH-DBN the segment-specific parameters have to be learned separately for each segment, even if parameters stay identical (or similar). This makes uncoupled model inappropriate for many real world applications. Recently three improved NH-DBNs with coupled network parameters have been proposed, which couple the segment-specific parameters among segments, so as to allow for information exchange between them. When *all* segment-specific parameters stay similar over time, coupled NH-DBNs perform significantly better than the uncoupled NH-DBNs. But the coupled NH-DBNs have the drawback that there is no effective mechanism for uncoupling. When the segment-specific parameters are dissimilar, the parameter coupling can become counter-productive. Partially segment-wise coupled NH-DBN models, introduced in chapter 2, consider both features (coupling and uncoupling) simultaneously. But these models have the pitfall that the (un-)coupling applies for all covariates at each changepoint contemporaneously. In real world applications it is usually not evident how (dis-) similar the segment-specific parameters are. Not rarely there is even a mixture of both parameter types which makes all the models introduced in previous chapters inappropriate for these real world scenarios.

We, therefore, propose a new NH-DBN with partially edge-wise coupled network parameters. The new model operate edge-wise and combines features of the uncoupled and the coupled NH-DBN and infers for each individual edge whether the corresponding parameter should be coupled or stay uncoupled. A beneficial feature of the new model is that it contains the uncoupled and the coupled NH-DBN as limiting cases: When it couples (uncouples) *all* edges, it reduces to the coupled (uncoupled) NH-DBN. Our empirical results show that the new model can significantly improve the network reconstruction accuracy.

The work, presented in this chapter, has been submitted to Journal of Computational and Graphical Statistics (2018). Some parts of this chapter have also been appeared in proceedings of the International Workshop on Statistical Modelling (2018) (see [57]).

4.1 Methods

4.1.1 Bayesian piece-wise linear regression models

Consider a Bayesian piece-wise linear regression model with Y being the response variable and $\pi = \{X_1, \dots, X_k\}$ being a set of k covariates. We assume that the observed data points have a temporal order and can be divided into disjunct segments $h \in \{1, \dots, H\}$, where each h has segment-specific regression coefficients, $\boldsymbol{\beta}_h = (\beta_{h,0}, \dots, \beta_{h,k})^{\mathsf{T}}$. Let $\mathbf{y}_h$ be the vector of the response values and $\mathbf{X}_h$ be the design matrix for segment h, where each $\mathbf{X}_h$ includes a first column of 1's for the intercept. For $h = 1, \dots, H$ we use a Gaussian likelihood:

$$\mathbf{y}_h \sim \mathcal{N}(\mathbf{X}_h \boldsymbol{\beta}_h, \sigma^2 \mathbf{I}) \tag{4.1}$$

where $\mathbf{I}$ denotes the identity matrix, and σ^2 is the noise variance parameter, which is shared among segments. We impose an inverse Gamma prior on σ^2, $\sigma^{-2} \sim GAM(a_\sigma, b_\sigma)$, and we assume that $\boldsymbol{\beta}_1$ has a Gaussian distribution:

$$\boldsymbol{\beta}_1 \sim \mathcal{N}(\mathbf{0}, \sigma^2 \lambda_u \mathbf{I}) \tag{4.2}$$

where $\mathbf{0} := (0, \dots, 0)^{\mathsf{T}}$, λ_u is the *'signal-to-noise ratio parameter for uncoupled regression coefficients'* onto which we also impose an inverse Gamma distribution $\lambda_u^{-1} \sim GAM(a_u, b_u)$.[1] The posterior distribution of $\boldsymbol{\beta}_1$ is:

$$\boldsymbol{\beta}_1 | (\mathbf{y}_1, \sigma^2, \lambda_u) \sim \mathcal{N}(\tilde{\boldsymbol{\beta}}_1, \sigma^2 \mathbf{C}_1) \tag{4.3}$$

where $\mathbf{C}_1 = ([\lambda_u \mathbf{I}]^{-1} + \mathbf{X}_1^{\mathsf{T}} \mathbf{X}_1)^{-1}$ and $\tilde{\boldsymbol{\beta}}_1 = \mathbf{C}_1 \mathbf{X}_1^{\mathsf{T}} \mathbf{y}_1$. Figure 4.1 shows a graphical model for $h = 1$. The conventional *'uncoupled'* (piecewise linear) model uses the same priors for all segments:

$$\boldsymbol{\beta}_h \sim \mathcal{N}(\mathbf{0}, \sigma^2 \lambda_u \mathbf{I}) \quad (h = 1, \dots, H) \tag{4.4}$$

The only information exchange among segments is then w.r.t. σ^2 and λ_u.
The key idea of the *'(fully) sequentially coupled'* model from [24] is to use the posterior expectation $\tilde{\boldsymbol{\beta}}_h$ as prior expectation for $\boldsymbol{\beta}_{h+1}$:

$$\boldsymbol{\beta}_{h+1} \sim \mathcal{N}(\tilde{\boldsymbol{\beta}}_h, \sigma^2 \lambda_c \mathbf{I}) \tag{4.5}$$

where λ_c has been called the *'coupling parameter'* onto which also an inverse Gamma distribution can be imposed $\lambda_c^{-1} \sim GAM(a_c, b_c)$. *'Coupling'* here means

---

[1] Re-employing the parameter σ^2 in Equation (4.2) yields a fully-conjugate prior in both $\boldsymbol{\beta}_1$ and σ^2; this allows both parameter groups to be integrated out in the likelihood, i.e. the marginal likelihood $p(\mathbf{y}_1 | \lambda_u)$ to be computed (see, e.g., Sections 3.3 and 3.4 in [19]).

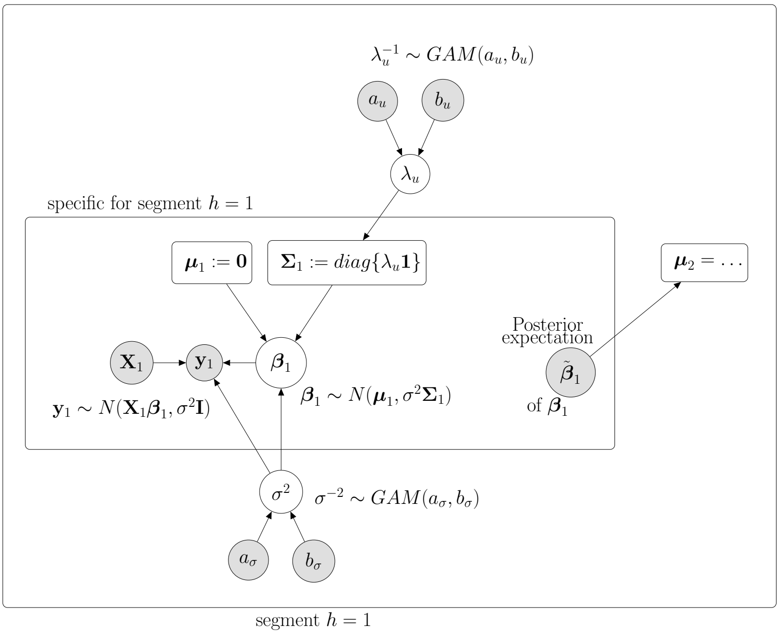

$\lambda_u^{-1} \sim GAM(a_u, b_u)$
a_u
b_u
λ_u
specific for segment $h = 1$
$\boldsymbol{\mu}_1 := \mathbf{0}$
$\boldsymbol{\Sigma}_1 := diag\{\lambda_u \mathbf{1}\}$
$\boldsymbol{\mu}_2 = \ldots$
Posterior expectation
$\tilde{\boldsymbol{\beta}}_1$
of $\boldsymbol{\beta}_1$
$\mathbf{X}_1$
$\mathbf{y}_1$
$\boldsymbol{\beta}_1$
$\mathbf{y}_1 \sim N(\mathbf{X}_1\boldsymbol{\beta}_1, \sigma^2\mathbf{I})$
$\boldsymbol{\beta}_1 \sim N(\boldsymbol{\mu}_1, \sigma^2\boldsymbol{\Sigma}_1)$
σ^2
$\sigma^{-2} \sim GAM(a_\sigma, b_\sigma)$
a_σ
b_σ
segment $h = 1$

Figure 4.1: Graphical Model - Part 1: Graphical presentation of the probabilistic relationships between the random variables for segment $h = 1$. Variables that have to be inferred are represented by white circles. The data and the fixed hyperparameters are represented by grey circles. The two rectangles indicate definitions, which deterministically depend on the parent nodes. All nodes and definitions within the inner plate are specific for segment $h = 1$. The posterior expectation $\tilde{\boldsymbol{\beta}}_1$ is treated like a fixed vector when used as input for segment $h = 2$. Note that $diag\{\lambda_u \mathbf{1}\}$ denotes the diagonal matrix $\lambda_u \mathbf{I}$.

that $\boldsymbol{\beta}_{h+1}$ is coupled to the posterior expectation $\tilde{\boldsymbol{\beta}}_h$ of $\boldsymbol{\beta}_h$. Low values λ_c yield peaked priors in Equation (4.5) and the vectors $\boldsymbol{\beta}_h$ and $\boldsymbol{\beta}_{h+1}$ will tend to be similar (=coupled). One shortcoming of the fully coupled approach is that $\boldsymbol{\beta}_{h+1}$ cannot properly uncouple from the preceding segment. For dissimilar regression coefficients, λ_c has to take large values, so as to make the prior in Equation (4.5) vague. Another bottleneck is that there is only one single coupling parameter λ_c, i.e. *all* coefficients are coupled with the same strength. Generalized coupled model, discussed in previous chapter, refines the latter bottleneck by possessing segment-specific coupling parameter, λ_c^h. But in case the regression coefficients are dissimilar, λ_c^h, has to stay large which makes the prior in Equation (4.5) diffused.

Partially segment-wise coupled model, introduced in chapter 2, infers for each segment $h > 1$ whether it is uncoupled from or coupled to the preceding one. The shortcoming of this model is that the coupling (uncoupling) always applies to all covariates simultaneously. On the other hand, at each changepoint *all* regression coefficients stay either similar or get dissimilar.

In this chapter we propose a new model which infers from the data which regression coefficients stay similar from segment to segment (and should be coupled) and which regression coefficients vary among segments (and should better be re-initialised uninformatively with a prior expectation of 0). We introduce a new vector of indicator variables $\boldsymbol{\delta} = (\delta_0, \dots, \delta_k)$ whose elements are binary variables $\delta_i \in \{0, 1\}$: δ_0 corresponds to the intercept, and δ_i $(i \geq 1)$ refers to the i-th covariate X_i. $\delta_i = 1$ indicates that the segment-specific coefficients $\beta_{1,i}, \dots, \beta_{H,i}$ for X_i are coupled, while $\delta_i = 0$ indicates that they are uncoupled. This definition also holds for intercept. That is, $\delta_0 = 1$ indicates that the corresponding segment-specific intercepts are coupled, whereas $\delta_0 = 0$ indicates that they are uncoupled. We introduce the new prior:

$$\boldsymbol{\beta}_{h+1} \sim \mathcal{N}(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_h, \sigma^2 \cdot diag\{\lambda_c \boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\}) \tag{4.6}$$

where $\odot$ is the Kronecker product ('elementwise multiplication'), $diag\{\mathbf{x}\}$ denotes a diagonal matrix whose diagnoal elements are the elements of the vector $\mathbf{x}$, and $\mathbf{1} := (1, \dots, 1)^{\mathsf{T}}$. As the covariance matrix in Equation (4.6) is a diagonal matrix, each element $\boldsymbol{\beta}_{h+1,i}$ of $\boldsymbol{\beta}_{h+1}$ is independently Gaussian distributed:

$$\boldsymbol{\beta}_{h+1,i} = \begin{cases} \mathcal{N}(0, \sigma^2\lambda_u) & \text{if } \delta_i = 0 \\ \mathcal{N}(\tilde{\boldsymbol{\beta}}_{h,i}, \sigma^2\lambda_c) & \text{if } \delta_i = 1 \end{cases} \qquad (h = 1, \dots, H-1;\ i = 0, \dots, k) \tag{4.7}$$

where $\tilde{\boldsymbol{\beta}}_{h,i}$ is the i-th element of the posterior expectation vector $\tilde{\boldsymbol{\beta}}_h$. The new prior yields a consensus between the uncoupled and the fully coupled approach:

- By setting $\boldsymbol{\delta} = \mathbf{0}$ we obtain $\boldsymbol{\beta}_{h+1} \sim \mathcal{N}(\mathbf{0}, \sigma^2\lambda_u\mathbf{I})$ for all h, as in Equation (4.4). The model is then the **uncoupled** piecewise linear regression model.

- By setting $\boldsymbol{\delta} = \mathbf{1}$, we obtain $\boldsymbol{\beta}_{h+1} \sim \mathcal{N}(\tilde{\boldsymbol{\beta}}_h, \sigma^2\lambda_c\mathbf{I})$ for $h \geq 1$, as in Equation (4.5). The model is then the **(fully) coupled** piecewise linear regression model.

- Our new **partially edge-wise coupled** model infers $\boldsymbol{\delta} = (\delta_0, \dots, \delta_k)$ from the data, to find a trade-off between the uncoupled and the (fully) coupled model.

We assume $\delta_0, \dots, \delta_k$ to be independently Bernoulli distributed with hyperparameter $p = 0.5$.[2]

$$\delta_i \sim BER(p) \quad (i = 0, \dots, k) \tag{4.8}$$

Figure 4.2 shows the relationships within and between segments. The joint distribution is:

---

[2] Alternatively, the parameter p can be assumed to have a Beta hyperprior, $p \sim BETA(a, b)$. For our applications the model extension with $p \sim BETA(1, 1)$ did not lead to any significant improvements.

$$\begin{aligned} p(\{\mathbf{y}_h\},\{\boldsymbol{\beta}_h\},\sigma^2,\lambda_u,\lambda_c,\boldsymbol{\delta}) \quad \propto \quad & p(\lambda_u)\cdot p(\lambda_c)\cdot p(\sigma^2)\cdot p(\boldsymbol{\delta}) \\ & \cdot \prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^2,\boldsymbol{\beta}_h)\cdot P(\boldsymbol{\beta}_1|\lambda_u,\sigma^2) \\ & \cdot \prod_{h=2}^{H} P(\boldsymbol{\beta}_h|\lambda_u,\lambda_c,\sigma^2,\boldsymbol{\delta},\tilde{\boldsymbol{\beta}}_{h-1}) \end{aligned} \tag{4.9}$$

As bivariate function of λ_u and λ_c, $p(\boldsymbol{\beta}_{h+1}|\sigma^2,\lambda_u,\lambda_c,\boldsymbol{\delta},\tilde{\boldsymbol{\beta}}_h)$ in Equation (4.6) has a modular form:

$$\begin{aligned} p(\boldsymbol{\beta}_{h+1}|\lambda_u,\lambda_c,\ldots) &= (2\pi)^{(-k+1)/2}\cdot\det\left(\sigma^2\cdot diag\{\lambda_c\boldsymbol{\delta}+\lambda_u(\mathbf{1}-\boldsymbol{\delta})\}\right)^{-0.5} \\ &\cdot\exp\{-\frac{1}{2}(\boldsymbol{\beta}_{h+1}-\boldsymbol{\delta}\odot\tilde{\boldsymbol{\beta}}_h)^{\mathsf{T}}[\sigma^2 diag\{\lambda_c\boldsymbol{\delta}+\lambda_u(\mathbf{1}-\boldsymbol{\delta})\}]^{-1}(\boldsymbol{\beta}_{h+1}-\boldsymbol{\delta}\odot\tilde{\boldsymbol{\beta}}_h)\} \\ &= (2\pi)^{-(k+1)/2}\cdot\sigma^{-(k+1)}\cdot\lambda_u^{-0.5\sum_{i=0}^{k}(1-\delta_i)}\cdot\lambda_c^{-0.5\sum_{i=0}^{k}\delta_i} \\ &\cdot\exp\{-\frac{1}{2}\sigma^{-2}\lambda_u^{-1}\sum_{i:\delta_i=0}(\beta_{h,i}-0)^2\}\cdot\exp\{-\frac{1}{2}\sigma^{-2}\lambda_c^{-1}\sum_{i:\delta_i=1}(\beta_{h,i}-\tilde{\beta}_{h,i})^2\} \end{aligned}$$

As a function of λ_u^{-1} or λ_c^{-1}, respectively, $p(\boldsymbol{\beta}_{h+1}|\lambda_u,\lambda_c,\ldots)$ is thus proportional to:

$$p(\boldsymbol{\beta}_{h+1}|\lambda_u,\ldots)\propto(\lambda_u^{-1})^{0.5\sum_{i=0}^{k}(1-\delta_i)}\cdot\exp\{-\lambda_u^{-1}\cdot(\frac{1}{2}\sigma^{-2}\cdot\sum_{i:\delta_i=0}\beta_{h,i}^2)\} \tag{4.10}$$

$$p(\boldsymbol{\beta}_{h+1}|\lambda_c,\ldots)\propto(\lambda_c^{-1})^{0.5\sum_{i=0}^{k}\delta_i}\exp\{-\lambda_c^{-1}(\frac{1}{2}\sigma^{-2}\cdot\sum_{i:\delta_i=1}(\beta_{h,i}-\tilde{\beta}_{h,i})^2)\} \tag{4.11}$$

The prior for $\boldsymbol{\beta}_1$ in Equation (4.2) is independent of λ_c. As a function of λ_u^{-1} we get:

$$p(\boldsymbol{\beta}_1|\sigma^2,\lambda_u,\boldsymbol{\delta})\propto(\lambda_u^{-1})^{0.5(k+1)}\cdot\exp\{-\lambda_u^{-1}\cdot(\frac{1}{2}\sigma^{-2}\cdot\sum_{i=0}^{k}\beta_{1,i}^2)\} \tag{4.12}$$

4.1.2 Gibbs sampling of the parameters of the piece-wise regression model

All free parameters of the new model (white circles in Figure 4.2) can be sampled from their full conditional distributions ('Gibbs sampling'). When deriving the full conditionals we make use of the joint distribution in Equation (4.9). For $\boldsymbol{\beta}_h$ we can apply standard rules (see, e.g., Chapters 2-3 of [5]). For the other parameters σ^2, λ_u, λ_c and δ_i ($i=0,\ldots,k$) we derive the full conditionals in this subsection.

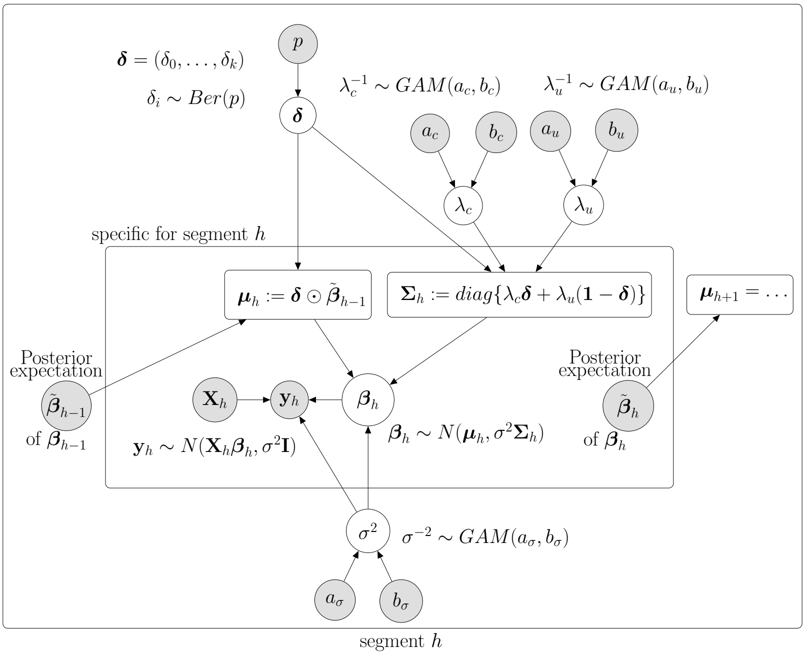

$\boldsymbol{\delta} = (\delta_0, \dots, \delta_k)$
$\delta_i \sim Ber(p)$
p
$\boldsymbol{\delta}$
$\lambda_c^{-1} \sim GAM(a_c, b_c)$
$\lambda_u^{-1} \sim GAM(a_u, b_u)$
a_c
b_c
a_u
b_u
λ_c
λ_u
specific for segment h
$\boldsymbol{\mu}_h := \boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}$
$\boldsymbol{\Sigma}_h := diag\{\lambda_c \boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\}$
$\boldsymbol{\mu}_{h+1} = \dots$
Posterior expectation
$\tilde{\boldsymbol{\beta}}_{h-1}$
of $\boldsymbol{\beta}_{h-1}$
$\mathbf{X}_h$
$\mathbf{y}_h$
$\boldsymbol{\beta}_h$
$\mathbf{y}_h \sim N(\mathbf{X}_h \boldsymbol{\beta}_h, \sigma^2 \mathbf{I})$
$\boldsymbol{\beta}_h \sim N(\boldsymbol{\mu}_h, \sigma^2 \boldsymbol{\Sigma}_h)$
Posterior expectation
$\tilde{\boldsymbol{\beta}}_h$
of $\boldsymbol{\beta}_h$
σ^2
$\sigma^{-2} \sim GAM(a_\sigma, b_\sigma)$
a_σ
b_σ
segment h

Figure 4.2: Graphical model - Part 2: Graphical presentation of the probabilistic relationships within and between segments for $h > 1$; see caption of Figure 4.1 for the terminology.

We now make explicit that the model depends on the covariates $\pi = \{X_1, \dots, X_k\}$, and we assume that the merged response vector $\mathbf{y} := (\mathbf{y}_1^\mathsf{T}, \dots, \mathbf{y}_H^\mathsf{T})^\mathsf{T}$ is segmented into $\mathbf{y}_1, \dots, \mathbf{y}_H$ by a changepoint set $\boldsymbol{\tau} = \{\tau_1, \dots, \tau_{H-1}\}$.

The full conditional distribution of $\boldsymbol{\beta}_1$ has already been provided in Equation (4.3). For $h > 1$ we set $\boldsymbol{\mu}_h := \boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}$ and $\boldsymbol{\Sigma}_h := diag\{\lambda_c \boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\}$ so that the priors take the form: $\boldsymbol{\beta}_h \sim \mathcal{N}(\boldsymbol{\mu}_h, \sigma^2 \cdot \boldsymbol{\Sigma}_h)$. Applying the corresponding rule from Section 3.3 of [5] yields:

$$\boldsymbol{\beta}_h | (\mathbf{y}_h, \sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \sim N(\tilde{\boldsymbol{\beta}}_h, \sigma^2 \mathbf{C}_h) \tag{4.13}$$

where $\mathbf{C}_h = (\boldsymbol{\Sigma}_h^{-1} + \mathbf{X}_h^\mathsf{T} \mathbf{X}_h)^{-1}$ and $\tilde{\boldsymbol{\beta}}_h = \mathbf{C}_h (\boldsymbol{\Sigma}_h^{-1} \boldsymbol{\mu}_h + \mathbf{X}_h^\mathsf{T} \mathbf{y}_h)$.
The noise variance parameter, σ^2, can be re-sampled via a collapsed Gibbs sampling step, where the regression coefficients, $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_H$, have been integrated out. In the Appendix we show that:

$$\sigma^{-2} | (\mathbf{y}_1, \dots, \mathbf{y}_H, \lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \sim GAM\left(a_\sigma + 0.5 \cdot T, b_\sigma + 0.5 \cdot \Delta^2\right) \tag{4.14}$$

where T is the number of data points, i.e. the length of $\mathbf{y} := (\mathbf{y}_1^\mathsf{T}, \dots, \mathbf{y}_H^\mathsf{T})^\mathsf{T}$, and Δ^2 is the sum of the squared Mahalanobis distances:

$$\Delta^2 := \sum_{h=1}^{H} (\mathbf{y}_h - \mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}))^\mathsf{T} (\mathbf{I} + \mathbf{X}_h \boldsymbol{\Sigma}_h \mathbf{X}_h^\mathsf{T})^{-1} (\mathbf{y}_h - \mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1})) \tag{4.15}$$

with $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$, and $\tilde{\boldsymbol{\beta}}_h = (\boldsymbol{\Sigma}_h^{-1} + \mathbf{X}_h^\mathsf{T}\mathbf{X}_h)^{-1}(\boldsymbol{\Sigma}_h^{-1}\boldsymbol{\mu}_h + \mathbf{X}_h^\mathsf{T}\mathbf{y}_h)$ being the posterior expectation of $\boldsymbol{\beta}_h$ ($h \geq 1$), given the prior expectations $\boldsymbol{\mu}_h$ and the prior covariance matrices $\boldsymbol{\Sigma}_h$:

$$\boldsymbol{\mu}_h = \begin{cases} \mathbf{0} & \text{if } h = 1 \\ \boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1} & \text{if } h > 1 \end{cases}, \qquad \boldsymbol{\Sigma}_h = \begin{cases} diag\{\lambda_u \mathbf{1}\} & \text{if } h = 1 \\ diag\{\lambda_c \boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\} & \text{if } h > 1 \end{cases} \tag{4.16}$$

We now derive the full conditional distributions of λ_u^{-1} and λ_c^{-1}. From Equation (4.9) we get:

$$\begin{aligned} p(\lambda_u^{-1}|\ldots) &\propto p(\lambda_u^{-1}) \cdot p(\boldsymbol{\beta}_1|\sigma^2, \lambda_u, \pi, \boldsymbol{\tau}) \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta}, \tilde{\boldsymbol{\beta}}_{h-1}, \pi, \boldsymbol{\tau}) \\ p(\lambda_c^{-1}|\ldots) &\propto p(\lambda_c^{-1}) \cdot \prod_{h=2}^{H} p(\boldsymbol{\beta}_h|\sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta}, \tilde{\boldsymbol{\beta}}_{h-1}, \pi, \boldsymbol{\tau}) \end{aligned}$$

Plugging in the prior densities of λ_u^{-1} and λ_c^{-1} and the results from Equations (4.10-4.12):

$$p(\lambda_u^{-1}|\ldots) \propto (\lambda_u^{-1})^{a_u + \frac{1}{2}k_u - 1} \cdot \exp\{-\lambda_u^{-1}(b_u + \frac{1}{2}\sigma^{-2}D_u^2)\} \tag{4.17}$$

where $D_u^2 := \sum\limits_{i=0}^{k} \beta_{1,i}^2 + \sum\limits_{h=2}^{H} \sum\limits_{i:\delta_i=0} \beta_{h,i}^2$ and $k_u := (k+1) + (H-1) \cdot \sum_{i=0}^{k}(1 - \delta_i)$ is the number of uncoupled regression coefficients.

$$p(\lambda_c^{-1}|\ldots) \propto (\lambda_c^{-1})^{a_c + \frac{k_c}{2} - 1} \exp\{-\lambda_c^{-1}(b_c + \frac{1}{2}\sigma^{-2}D_c^2)\} \tag{4.18}$$

where $D_c^2 := \sum\limits_{h=2}^{H} \sum\limits_{i:\delta_i=1} (\boldsymbol{\beta}_{h,i} - \tilde{\boldsymbol{\beta}}_{h-1,i})^2$ and $k_c := (H-1) \cdot \sum_{i=0}^{k} \delta_i$ is the number of coupled regression coefficients.[3] From the shapes of the full conditionals in Equations (4.17-4.18) it follows:

$$\lambda_u^{-1}|(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_H, \sigma^2, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \sim GAM\left(a_u + \frac{k_u}{2}, b_u + \frac{1}{2}\sigma^{-2}D_u^2\right) \tag{4.19}$$

$$\lambda_c^{-1}|(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_H, \sigma^2, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \sim GAM\left(a_c + \frac{k_c}{2}, b_c + \frac{1}{2}\sigma^{-2}D_c^2\right) \tag{4.20}$$

For the marginal likelihood, with $\boldsymbol{\beta}_h$ ($h = 1, \ldots, H$) and σ^2 integrated out, we apply the rule from Section 2.3.7 of [5]:

---

[3] $k_u + k_c = H \cdot (k+1)$ is the total number of coefficients; for each of H segments there are $k+1$ coefficients.

$$p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) = \frac{\Gamma(\frac{T}{2} + a_\sigma)}{\Gamma(a_\sigma)} \cdot \frac{\pi^{-T/2} \cdot (2b_\sigma)^{a_\sigma}}{\left(\prod_{h=1}^{H} \det(\mathbf{I} + \mathbf{X}_h \boldsymbol{\Sigma}_h \mathbf{X}_h^\mathsf{T})\right)^{1/2}} \cdot (2b_\sigma + \Delta^2)^{-(\frac{T}{2}+a_\sigma)} \tag{4.21}$$

where Δ^2 and $\boldsymbol{\Sigma}_h$ ($h = 1, \ldots, H$) were defined in Equations (4.15-4.16).

Finally, we derive the full conditional distributions of the elements of the vector $\boldsymbol{\delta} = (\delta_0, \ldots, \delta_k)$. As $\delta_0, \ldots, \delta_k$ are i.i.d. $BER(p)$ distributed, we get from Equation (4.9 and 4.21):

$$p(\delta_i|\ldots) \propto p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \cdot p(\boldsymbol{\delta}) = p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \cdot p^{\delta_i} \cdot (1-p)^{1-\delta_i} \tag{4.22}$$

And since δ_i is binary, the full conditional is also a Bernoulli distribution:

$$\delta_i|(\lambda_u, \lambda_c, \{\delta_j : j \neq i\}, \pi, \boldsymbol{\tau}, \mathbf{y}) \sim BER(\theta_i) \tag{4.23}$$

where

$$\theta_i = \frac{p(\delta_i = 1|\ldots)}{p(\delta_i = 1|\ldots) + p(\delta_i = 0|\ldots)} = \frac{p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}^{\delta_i \leftarrow 1}, \pi, \boldsymbol{\tau}) \cdot p}{\sum_{j=0}^{1} p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}^{\delta_i \leftarrow j}, \pi, \boldsymbol{\tau}) \cdot p^j \cdot (1-p)^{1-j}} \tag{4.24}$$

and $\boldsymbol{\delta}^{\delta_i \leftarrow j}$ denotes the vector $\boldsymbol{\delta}$ with δ_i being set to $j \in \{0, 1\}$.

4.1.3 Metropolis-Hastings sampling of the covariate and changepoint set

When the covariates π and the segmentation $\boldsymbol{\tau}$ are unknown, we can infer both from the data. Recall that Y is the response, and let $X_1, \ldots, X_n$ be a set of *potential* covariates. $\mathcal{D}$ denotes a time series of equi-distant data points, indexed $t = 1, \ldots, T$. Each data point $\mathcal{D}_t$ contains a response observation y_t and the observations $x_{t,1}, \ldots, x_{t,n}$ of the covariates. We assume all covariate sets $\pi \subset \{X_1, \ldots, X_n\}$ to be equally likely a priori, and as prior on the number of segments H we take a Poisson distribution with parameter $\lambda = 1$, truncated to $1 \leq H \leq 10$:

$$H \sim POI(\lambda | 1 \leq H \leq 10)$$

Subsequently we identify H segments with $H - 1$ changepoints, $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_{H-1}\}$ on the set $\mathcal{S} := \{2, \ldots, T-1\}$. Data point $\mathcal{D}_t$ is assigned to the h-th segment if and only if $\tau_{h-1} < t \leq \tau_h$, where $\tau_0 := 1$ and $\tau_H := T$. Following [21] and [23] we assume that the changepoints are distributed like the even-numbered order statistics of $L := 2(H-1) + 1$ pairwise different points, being uniformly distributed on $\mathcal{S}$. This yields:

$$p(\boldsymbol{\tau}|H) = \frac{1}{\binom{T-2}{2(H-1)+1}} \cdot \prod_{h=0}^{H-1} (\tau_{h+1} - \tau_h - 1) \tag{4.25}$$

For each combination of π and $\boldsymbol{\tau}$, the model from Subsection 4.1.1 can be applied. The changepoint set $\boldsymbol{\tau}$ yields a segmentation of the data points into H segments with response vectors $\mathbf{y}_1, \ldots, \mathbf{y}_H$. The design matrices $\mathbf{X}_1, \ldots, \mathbf{X}_H$ are built using only the covariates from π. Using the marginal likelihood from Equation (4.21), we obtain for the posterior distribution:

$$p(\pi, \boldsymbol{\tau}, \lambda_u, \lambda_c, \boldsymbol{\delta}|\mathcal{D}) \propto p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \cdot p(\pi) \cdot p(\boldsymbol{\tau}|H) \cdot p(H) \cdot p(\boldsymbol{\delta}) \cdot p(\lambda_u) \cdot p(\lambda_c) \tag{4.26}$$

Given π and $\boldsymbol{\tau}$, the parameters λ_u, λ_c and the elements of $\boldsymbol{\delta}$ can be re-sampled from their full conditional distributions, as described in Subsection 4.1.1.[4] Given λ_u, λ_c, and $\boldsymbol{\delta}$, Metropolis-Hastings steps can be used to sample the covariate set π and the changepoint set $\boldsymbol{\tau}$.

Moves on the covariate set: For sampling π from the posterior we implement 3 moves:

- **Removal (R)**: We randomly select one $X_i \in \pi$ and remove it from π. With the covariate we also delete the corresponding element δ_i of $\boldsymbol{\delta}$.
- **Addition (A)**: We randomly select one $X_i \notin \pi$ and add it to π. With the covariate we also add a new element δ_i to $\boldsymbol{\delta}$. We flip a coin to determine the value of δ_i.
- **Exchange (E)**: We randomly select one $X_i \in \pi$ and replace it by a randomly selected $X_j \notin \pi$. We remove δ_i from $\boldsymbol{\delta}$ and add δ_j to $\boldsymbol{\delta}$. We flip a coin to determine the value of δ_j.

Each move proposes to replace $[\pi, \boldsymbol{\delta}]$ by $[\pi^*, \boldsymbol{\delta}^*]$. When randomly selecting the move type, the acceptance probabilities are:

$$A([\pi, \boldsymbol{\delta}] \to [\pi^*, \boldsymbol{\delta}^*]) = \min\left\{1, \frac{p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}^*, \pi^*, \boldsymbol{\tau})}{p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau})} \cdot \frac{p(\pi^*)}{p(\pi)} \cdot \frac{p(\boldsymbol{\delta}^*)}{p(\boldsymbol{\delta})} \cdot HR\right\} \tag{4.27}$$

where the Hastings Ratio HR depends on the move type:

$$HR_R = \frac{|\pi|}{n - |\pi^*|} \cdot 0.5, \qquad HR_A = \frac{n - |\pi|}{|\pi^*|} \cdot 2, \qquad HR_E = 1 \tag{4.28}$$

where n is the number of potential covariates, $|.|$ denotes the cardinality, and the factors 2 and 0.5 stem from flipping coins for the values of newly introduced indicator variables.[5]

Moves on the changepoint set: For sampling $\boldsymbol{\tau}$ we also implement 3 moves:

- **Birth (B)**: Out of the set of all valid new changepoint locations $\mathcal{B}(\boldsymbol{\tau})$ we randomly sample one element and propose to set a new changepoint at this location. The new changepoint set $\boldsymbol{\tau}^*$ contains $H^* = H + 1$ segments.

---

[4] The parameters σ^2 and $\beta_1, \ldots, \beta_H$ are marginalized out in Equation (4.26). But they have to be sampled, before sampling from the full conditionals of λ_u and λ_c in Equations (4.19-4.20).

[5] For $p = 0.5$ in Equation (4.8) the prior ratio $p(\boldsymbol{\delta}^\star)/p(\boldsymbol{\delta})$ cancels with the factors 2 and 0.5, respectively.

- **Death (D)**: We randomly select one changepoint $\tau \in \boldsymbol{\tau}$ and delete it. The new changepoint set $\boldsymbol{\tau}^*$ contains $H^* = H - 1$ segments.
- **Reallocation (R)**: We randomly select one changepoint $\tau_j \in \boldsymbol{\tau}$ and propose to re-allocate it to a randomly selected position in between the two surrounding changepoints: $\tau_{j-1} + 2, \ldots, \tau_{j+1} - 2$. This yields $\boldsymbol{\tau}^*$, and $H^* = H$.

When randomly selecting the move type, the acceptance probabilities are:

$$A([\boldsymbol{\tau}, H] \to [\boldsymbol{\tau}^*, H^*]) = \min\left\{1, \frac{p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}^*)}{p(\mathbf{y}|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau})} \cdot \frac{p(\boldsymbol{\tau}^*|H^*)}{p(\boldsymbol{\tau}|H)} \cdot \frac{p(H^*)}{p(H)} \cdot HR\right\} \tag{4.29}$$

where the Hastings Ratio HR depends on the move type:

$$HR_B = \frac{|\mathcal{B}(\boldsymbol{\tau}^*)|}{|\boldsymbol{\tau}|}, \qquad HR_D = \frac{|\boldsymbol{\tau}^*|}{|\mathcal{B}(\boldsymbol{\tau})|}, \qquad HR_R = 1 \tag{4.30}$$

where $\mathcal{B}(\boldsymbol{\tau}) := \{\tau | 2 \leq \tau \leq T-1 \ \textbf{and} \ |\tau_j - \tau| \geq 2 \ \text{for } j = 1, \ldots, H-1\}$ is the set of all valid new changepoint locations, given $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_H\}$, and $|.|$ denotes the cardinality.

4.1.4 MCMC sampling algorithm

Given data $\mathcal{D}$ we use Markov Chain Monte Carlo (MCMC) simulations to generate a sample $\{\pi^{(w)}, \boldsymbol{\tau}^{(w)}, \lambda_u^{(w)}, \lambda_c^{(w)}, \boldsymbol{\delta}^{(w)}\}_{w=1,\ldots,W}$ from the posterior $p(\pi, \boldsymbol{\tau}, \lambda_u, \lambda_c, \boldsymbol{\delta}|\mathcal{D})$. In each iteration we re-sample the two parameters λ_u and λ_c and one element of $\boldsymbol{\delta}$ from their full conditional distributions (Gibbs sampling), and we perform two Metropolis-Hastings moves; one on the covariate set π and one on the changepoint set $\boldsymbol{\tau}$. Table 4.1 gives pseudo code for the MCMC algorithm.

4.1.5 Learning dynamic Bayesian networks

Consider a N-by-$(T+1)$ data matrix $\mathbf{D}$ whose rows correspond to the variables $Z_1, \ldots, Z_N$ and whose columns correspond to equi-distant time points $t = 1, \ldots, T+1$. Let $\mathbf{D}_{i,t}$ denote the value of Z_i at t. The variables can then be identified with the nodes of a network, and we can learn how the variables interact with each other. Temporal data are conventionally modelled with dynamic Bayesian networks (DBNs), where all dependencies are subject to a time lag, usually of order $\mathcal{O} = 1$. An edge from node Z_i to node Z_j, $Z_i \to Z_j$, indicates that $\mathbf{D}_{j,t+1}$ (Z_j at $t+1$) depends on $\mathbf{D}_{i,t}$ (Z_i at t). Z_i is then called a parent (node) of Z_j.

Because of the time lag, there is no acyclicity constraint in DBNs. Hence, learning a DBN can be thought of as learning separately for each node Z_j a covariate set π_j ($j = 1, \ldots, N$). In the j-th (piece-wise linear) regression model $Y := Z_j$ is the response, and there are $n := N - 1$ potential covariates:

$\{X_1, \dots, X_n\} := \{Z_1, \dots, Z_{j-1}, Z_{j+1}, \dots, Z_N\}$. Each data point $\mathcal{D}_t$ $(t = 1, \dots, T)$ of the j-th regression model contains a response value $y_j = \mathbf{D}_{j,t+1}$ and the shifted values of the potential covariates $x_{t,1} := \mathbf{D}_{1,t}, \dots, x_{t,n} := \mathbf{D}_{n,t}$.
Having a covariate set π_j for each response $Y := Z_j$, a network can be built by merging the covariate sets: $\mathcal{G} := \{\pi_1, \dots, \pi_N\}$. There is an edge from X_i to X_j if and only if $X_i \in \pi_j$.

4.1.6 Competing regression models (as building blocks for NH-DBNs)

In this subsection we briefly outline alternative regression models, with which NH-DBNs can be built. Like the proposed model, the models can be applied to each variable separately to infer a network among N nodes, see Subsection 4.1.5 for details. To underline that the resulting NH-DBNs are highly competitive, we note that [1] found that the homogeneous DBN and the uncoupled NH-DBN, presented below, performed best among 15 state-of-the-art network reconstruction methods in a cross-method comparison on synthetic network data. The four established ('published') competitors for the newly proposed model are:

- **HOMOGENEOUS DBN**: The conventional homogenous DBN, as discussed in chapter 1, has no changepoints, $H = 1$. This model does neither possess the $\boldsymbol{\delta}$ vector nor the λ_c parameter. The regression coefficient vector $\boldsymbol{\beta}_1$ applies to all data points.

- **UNCOUPLED NH-DBN**: This model corresponds to the model of [38], but unlike in [38] it here does not allow for network changes among segments. Our new model reduces to this model when setting $\boldsymbol{\delta} = \mathbf{0}$ and removing the λ_c parameter. The segment-specific priors are: $\boldsymbol{\beta}_h \sim N(\mathbf{0}, \sigma^2\lambda_u\mathbf{I})$ $(h = 1, \dots, H)$. For $H = 1$ this model reduces to the homogeneous DBN.

- **FULLY (SEQUENTIALLY) COUPLED NH-DBN**: This model from [24] couples all neighbouring regression coefficients with the same strength. Our new model reduces to this model when setting $\boldsymbol{\delta} = \mathbf{1}$. The priors of the regression coefficients are: $\boldsymbol{\beta}_1 \sim N(\mathbf{0}, \sigma^2\lambda_u\mathbf{I})$ and $\boldsymbol{\beta}_h \sim N(\tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2\lambda_c\mathbf{I})$ for $h \geq 2$. For $H = 1$ this model reduces to the homogeneous DBN.

- **GENERALIZED FULLY COUPLED NH-DBN**: This model introduced in chap-ter 3, generalizes the fully coupled NH-DBN. It introduces segment-specific coupling parameters λ_c^h:

  $$\boldsymbol{\beta}_h \sim \begin{cases} N(\mathbf{0}, \lambda_u\sigma^2\mathbf{I}) & \text{if } h = 1 \\ N(\tilde{\boldsymbol{\beta}}_{h-1}, \lambda_c^h\sigma^2\mathbf{I}) & \text{if } h = 2, \dots, H \end{cases}$$

  where $\lambda_c^h \sim GAM(a_c, b_c)$ for $h = 2, \dots, h$. The coupling applies to *all* regression coefficients, but the coupling strengths vary from segment to segment. Under the constraint: $\lambda_c^h = \lambda_c$ for all $h > 1$, this model becomes the fully coupled model.

```
Input: The data $\mathcal{D}$ and the current instantiations of the covariate set $\pi^{(w)}$, the changepoint set $\boldsymbol{\tau}^{(w)}$, the vector $\boldsymbol{\delta}^{(w)}$, and the parameters $\lambda_u^{(w)}$ and $\lambda_c^{(w)}$.
MCMC iteration: $w \to w+1$:
  • Re-sample a new noise variance parameter $\sigma_\diamond^{-2}$ from $\sigma^{-2}|(\mathbf{y}_1,\dots,\mathbf{y}_H,\lambda_u^{(w)},\lambda_c^{(w)},\boldsymbol{\delta}^{(w)},\pi^{(w)},\boldsymbol{\tau}^{(w)})$, see Equation (4.14).
  • For $h = 1,\dots,H$
      – Re-sample the segment-specific regression coefficients vector $\boldsymbol{\beta}_h^\diamond$ from $\boldsymbol{\beta}_h|(\mathbf{y}_h,\sigma_\diamond^2,\lambda_u^{(w)},\lambda_c^{(w)},\boldsymbol{\delta}^{(w)},\pi^{(w)},\boldsymbol{\tau}^{(w)})$, see Equation (4.13).
  • Sample from $\lambda_u^{-1}|(\boldsymbol{\beta}_1^\diamond,\dots,\boldsymbol{\beta}_H^\diamond,\sigma_\diamond^2,\boldsymbol{\delta}^{(w)},\pi^{(w)},\boldsymbol{\tau}^{(w)})$, see Equation (4.19). Invert the sampled value to obtain $\lambda_u^{(w+1)}$
  • Sample from $\lambda_c^{-1}|(\boldsymbol{\beta}_1^\diamond,\dots,\boldsymbol{\beta}_H^\diamond,\sigma_\diamond^2,\boldsymbol{\delta}^{(w)},\pi^{(w)},\boldsymbol{\tau}^{(w)})$, see Equation (4.20). Invert the sampled value to obtain: $\lambda_c^{(w+1)}$
  • Withdraw $\boldsymbol{\beta}_1^\diamond,\dots,\boldsymbol{\beta}_H^\diamond$, and $\sigma_\diamond^2$.
  • Randomly select one of the $k+1$ elements of the vector $\boldsymbol{\delta}^{(w)}$. Replace the selected element $\delta_i^{(w)}$ by a new value $\delta_i^{(w+1)}$ where the latter is sampled from $\delta_i|(\lambda_u^{(w+1)},\lambda_c^{(w+1)},\{\delta_j^{(w)}: j \neq i\},\pi^{(w)},\boldsymbol{\tau}^{(w)},\mathbf{y})$, see Equation (4.23). Replacing the element $\delta_i^{(w)}$ of $\boldsymbol{\delta}^{(w)}$ by $\delta_i^{(w+1)}$ yields the new vector $\boldsymbol{\delta}^{(w+1)}$.
  • Metropolis-Hastings move on the covariate set $\pi^{(w)}$:
      – Randomly select the move type (R, A or E), and propose to move from $[\pi^{(w)},\delta^{(w)}]$ to $[\pi^*,\delta^*]$. Accept the new state $[\pi^*,\delta^*]$ with the acceptance probability given in Equation (4.27) with $\lambda_u = \lambda_u^{(w+1)}$, $\lambda_c = \lambda_c^{(w+1)}$, $\boldsymbol{\delta} = \boldsymbol{\delta}^{(w+1)}$, $\pi = \pi^{(w)}$, $\boldsymbol{\delta} = \boldsymbol{\delta}^{(w)}$.
      – If the move is accepted, set: $\pi^{(w+1)} = \pi^*$ and $\boldsymbol{\delta}^{(w+1)} = \boldsymbol{\delta}^*$.
        Otherwise set: $\pi^{(w+1)} = \pi^{(w)}$ and $\boldsymbol{\delta}^{(w+1)} = \boldsymbol{\delta}^{(w)}$.
  • Metropolis-Hastings move on the changepoint set $\boldsymbol{\tau}^{(w)}$:
      – Randomly select the move type (B, D or R), and propose to move from $[\boldsymbol{\tau}^{(w)},H^{(w)}]$ to $[\boldsymbol{\tau}^*,H^*]$. Accept the new state $[\boldsymbol{\tau}^*,H^*]$ with the acceptance probability given in Equation (4.29) using $\lambda_u = \lambda_u^{(w+1)}$, $\lambda_c = \lambda_c^{(w+1)}$, $\boldsymbol{\delta} = \boldsymbol{\delta}^{(w+1)}$, $\pi = \pi^{(w+1)}$, $\boldsymbol{\tau} = \boldsymbol{\tau}^{(w)}$.
      – If the move is accepted, set: $\boldsymbol{\tau}^{(w+1)} = \boldsymbol{\tau}^*$ and $H^{(w+1)} = H^*$.
        Otherwise set: $\boldsymbol{\tau}^{(w+1)} = \boldsymbol{\tau}^{(w)}$ and $H^{(w+1)} = H^{(w)}$.
Output: The re-sampled instantiations: $\pi^{(w+1)}$, $\boldsymbol{\tau}^{(w+1)}$, $\boldsymbol{\delta}^{(w+1)}$, $\lambda_u^{(w+1)}$, and $\lambda_c^{(w+1)}$.
```

Table 4.1: Pseudo code. The table summarizes one iteration ($w \to w+1$) of the MCMC algorithm.

We now introduce two more competitors which have not been proposed (published) in the literature yet. Those models are similar to the proposed partially edge-wise coupled NH-DBN:

- **SWITCH (UNCOUPLED/COUPLED) NH-DBN**: This model switches between the uncoupled and the fully coupled NH-DBN. The regression coefficient priors are:

$$\boldsymbol{\beta}_h \sim \begin{cases} N(\mathbf{0}, \lambda_u \sigma^2 \mathbf{I}) & \text{if } \delta^\star = 0 \text{ or } h = 1 \\ N(\tilde{\boldsymbol{\beta}}_{h-1}, \lambda_c \sigma^2 \mathbf{I}) & \text{if } \delta^\star = 1 \text{ and } h > 1 \end{cases}$$

  where $\delta^\star \sim BER(0.5)$, indicates the model. For $\delta^\star = 0$ ($\delta^\star = 1$) the model is the uncoupled (fully coupled) NH-DBN. In a network domain for each individual node either the uncoupled or the fully coupled modelling approach is chosen.

- **PARTIALLY SEGMENT-WISE COUPLED NH-DBN**: This model introduced in chapter 2, infers for each segment $h > 1$ whether it is uncoupled from or coupled to the preceding one. The coupling (uncoupling) always applies to all covariates. The priors are:

$$\boldsymbol{\beta}_h \sim \begin{cases} N(\mathbf{0}, \lambda_u \sigma^2 \mathbf{I}) & \text{if } \delta_h^\star = 0 \\ N(\tilde{\boldsymbol{\beta}}_{h-1}, \lambda_c \sigma^2 \mathbf{I}) & \text{if } \delta_h^\star = 1 \end{cases} \quad (h = 1, \ldots, H)$$

  where $\delta_1^\star := 0$, and $\delta_h^\star \sim BER(0.5)$ for $h > 1$. $\delta_h^\star = 1$ indicates that segment h is coupled to segment $h-1$, while $\delta_h^\star = 0$ indicates that it is uncoupled. At each changepoint *all* regression coefficients stay either similar ($\delta_h^\star = 1$) or get dissimilar ($\delta_h^\star = 0$). The vector $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_{k+1})^\mathsf{T}$ is replaced by the vector $\boldsymbol{\delta}^\star = (\delta_2^\star, \ldots, \delta_H^\star)^\mathsf{T}$. The switch NH-DBN model is nested within this model. The switch model is obtained by imposing the constraint that for each domain node *either:* $\delta_h^\star = 0$ for all $h > 1$ *or:* $\delta_h^\star = 1$ for all $h > 1$.

Figure 4.3 shows a graphical overview of the seven changepoint-segmented sequentially coupled models. For each model it can be seen from the figure which other models are nested within it.

The following two NH-DBNs are based on conceptually different approaches:

- **HIDDEN MARKOV MODEL UNCOUPLED NH-DBN (HMM-DBN)**: The HMM-DBN model from [22], uses the priors of the uncoupled NH-DBN, $\boldsymbol{\beta}_h \sim N(\mathbf{0}, \sigma^2 \lambda_u \mathbf{I})$ ($h = 1, \ldots, H$), but unlike the uncoupled NH-DBN, it employs a Hidden Markov model (HMM) to allocate the individual data points to components $h = 1, \ldots, H$. Since the set of data segmentations that can be reached by changepoints is a subset of the segmentation space of a Hidden Markov model, the HMM-DBN model can be thought of as a generalization of the uncoupled NH-DBN, described above.

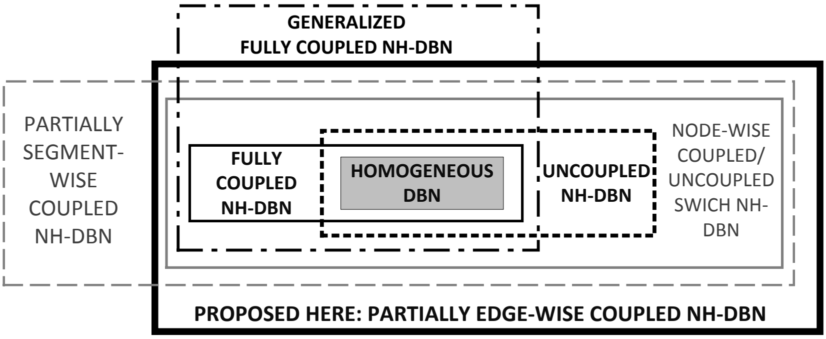

GENERALIZED
FULLY COUPLED NH-DBN
PARTIALLY SEGMENT-WISE COUPLED NH-DBN
FULLY COUPLED NH-DBN
HOMOGENEOUS DBN
UNCOUPLED NH-DBN
NODE-WISE COUPLED/ UNCOUPLED SWICH NH-DBN
PROPOSED HERE: PARTIALLY EDGE-WISE COUPLED NH-DBN

Figure 4.3: Overview of the changepoint-segmented sequentially coupled NH-DBNs. For each model there is a plate covering the plates of the models that are nested within it. The model $\mathcal{M}_1$ is nested in the model $\mathcal{M}_2$ if there are constraints under which $\mathcal{M}_2$ reduces to $\mathcal{M}_1$; see main text for details. The four established (published) competitors of the proposed model have black plates.

- **GLOBALLY COUPLED NH-DBN**: This model was proposed in [25]. It ignores the order of the segments and introduces a new hierarchy:

  $$\boldsymbol{\beta}_h \sim N(\mathbf{m}, \lambda_u \sigma^2 \mathbf{I})$$

  where $\mathbf{m} \sim N(\mathbf{0}, \mathbf{I})$ is a free hyperparameter, which allows for information exchange. This model is conceptually different from the others. It replaces the sequential by a global coupling scheme and does not contain any other model as special case. Even for $H = 1$ it differs from the homogeneous DBN, as it has a hyperprior on the prior expectation $\mathbf{m}$.

In [22] the performances of various allocation models for the uncoupled NH-DBN were compared, and it was found that the HMM-DBN performed best. Marginally, we will therefore also compare the results of the sequentially coupled NH-DBNs from Figure 4.3 with the results of the HMM-DBN model. For fixed changepoints, the HMM-DBN is identical to the uncoupled NH-DBN; and we then compare with the globally coupled NH-DBN instead.

4.1.7 Network reconstruction and convergence diagnostics

Edge scores: Given data $\mathbf{D}$ for a network domain with N variables $Z_1, \dots, Z_N$ we apply the regression model to each variable separately, see Subsection 4.1.5 for details.
For each Z_i the MCMC algorithm in Table 4.1 outputs a sample as follows: $\{\pi_i^{(w)}, \boldsymbol{\tau}_i^{(w)}, \lambda_{u,i}^{(w)}, \lambda_{c,i}^{(w)}, \boldsymbol{\delta}_i^{(w)}\}_{w=1,\dots,W}$ from the i-th posterior distribution. We merge the covariate sets to form a sample of graphs $G^{(w)} = \{\pi_1^{(w)}, \dots, \pi_N^{(w)}\}_{w=1,\dots,W}$, where the w-th graph $G^{(w)}$ has the edge $Z_i \to Z_j$ if $Z_i \in \pi_j^{(w)}$. For each edge

$Z_i \rightarrow Z_j$ we compute the marginal edge posterior probability (score):

$$\hat{e}_{i,j} = \frac{1}{W}\sum_{w=1}^{W} I_{i\rightarrow j}(\mathcal{G}^{(w)}) \text{ where } I_{i\rightarrow j}(\mathcal{G}^{(w)}) = \begin{cases} 1 & \text{if } X_i \in \pi_j^{(w)} \\ 0 & \text{if } X_i \notin \pi_j^{(w)} \end{cases} \quad (i,j \in \{1,\dots n\} : i \neq j)$$

Network reconstruction accuracy: If the true network is known, we evaluate the network reconstruction accuracy in form of precision-recall curves. For each $\psi \in [0,1]$ we extract the $n(\psi)$ edges whose scores $\hat{e}_{i,j}$ exceed ψ, and we count the number of true positives $T(\psi)$ among them. Plotting the *precisions* $P(\psi) := T(\psi)/n(\psi)$ against the *recalls* $R(\psi) := T(\psi)/M$, where M is the number of edges in the true network, gives the precision-recall curve ([11]). We refer to the area under the curve as AUC value.
[11] compared precision recall curves with Receiver Operator Characteristic (ROC) curves and found that precision-recall curves tend to be more adequate. For our applications we computed the areas under both types of curves and observed very similar trends.

Potential Scale Reduction Factors (PSRFs): The MCMC convergence can be monitored in terms of PSRFs; see, e.g. [7]. We perform H independent MCMC simulations and for each simulation h we compute the score $\hat{e}_{i,j}^{(h,s)}$ of edge $X_i \rightarrow X_j$ after $200s$ ($s = 1,\dots,500$) iterations. Assuming a burn-in of $100s$ iterations and thinning out by the factor 100, yields s samples and we compute the "between-chain" and the "within-chain" variances:

$$\mathcal{B}_s(i,j) = \frac{1}{H-1}\sum_{h=1}^{H}(\hat{e}_{i,j}^{(h,s)} - \bar{e}_{i,j}^{(\cdot,s)})^2$$

and

$$\mathcal{W}_s(i,j) = \frac{1}{H(s-1)}\sum_{h=1}^{H}\sum_{w=1}^{s}(I_{i\rightarrow j}(\mathcal{G}_h^{(w)}) - \hat{e}_{i,j}^{(h,s)})^2$$

where $\bar{e}_{i,j}^{[\cdot,s]}$ is the mean of $\hat{e}_{i,j}^{(1,s)},\dots,\hat{e}_{i,j}^{(H,s)}$, and $I_{i\rightarrow j}(\mathcal{G}_h^{(w)})$ is 1 if network w of simulation h has the edge $X_i \rightarrow X_j$, and 0 otherwise. After $200s$ iterations the PSRF of the edge $X_i \rightarrow X_j$ is:

$$PSRF_s(i,j) = \frac{(1-\frac{1}{s})\mathcal{W}_s(i,j) + (1+\frac{1}{H})\mathcal{B}_s(i,j)}{\mathcal{W}_s(i,j)} \tag{4.31}$$

PSRFs near 1 indicate that the MCMC simulations are close to the stationary distribution. We monitor the fraction of edges with $PSRF < 1.01$ against the MCMC iterations $200s$.

4.2 Data

4.2.1 Synthetic RAF-pathway data

We use the synthetic data generation mechanism, described in [24]: For the RAF pathway ([50]) with $N = 11$ nodes and $M = 20$ directed edges, we generate data consisting of $H = 4$ segments with m data points each. For each node Z_i ($i = 1, \ldots, 11$) its parents in π_i are the covariates of a piece-wise linear regression model:

$$z_{i,t+1} = \beta_{i,F(t),0} + \sum_{j:Z_j \in \pi_i} \beta_{i,F(t),j} \cdot z_{j,t} + e_{i,t} \quad (t = 1, \ldots, 4m)$$

where $z_{i,t}$ denotes the value of node Z_i at time point t, the noise values $e_{i,t}$ are sampled from independent $N(0, 0.05^2)$ distributions, and the regression coefficients are subject to temporal changes.[6] In our setting the coefficients change after m data points, so that $F(t) = 1 + \lfloor (t-1)/m \rfloor$. For each node Z_i there are $|\pi_i| + 1$ regression coefficients with $H = 4$ segment-specific values. For each segment h we summarize the coefficients in a vector $\boldsymbol{\beta}_{i,h}$. For $h = 1$ we sample the elements of $\boldsymbol{\beta}_{i,1}$ from a standard $N(0, 1)$ Gaussian distribution and then re-normalize the vector to Euclidean norm one: $\boldsymbol{\beta}_{i,1} \leftarrow \boldsymbol{\beta}_{i,1}/|\boldsymbol{\beta}_{i,1}|$. We distinguish three scenarios:

- **Coupled data:** We keep the regression coefficients fixed among segments, i.e. we set: $\boldsymbol{\beta}_{i,h} = \boldsymbol{\beta}_{i,1}$ ($h = 2, \ldots, 4$). This refers to maximally coupled (identical) network parameters.
- **Uncoupled data:** We set: $\boldsymbol{\beta}_{i,2} = -\boldsymbol{\beta}_{i,1}$, $\boldsymbol{\beta}_{i,3} = \boldsymbol{\beta}_{i,1}$, and $\boldsymbol{\beta}_{i,4} = -\boldsymbol{\beta}_{i,1}$, so that all network parameters switch the signs at changepoints. Neighbouring segments h and $h + 1$ have then very dissimilar network parameters.
- **Partially edge-wise coupled data**: There are $\sum_{i=1}^{11}(|\pi_i| + 1) = M + 11 = 31$ regression coefficients. For each coefficient we flip a coin to decide whether it stays constant or changes its sign from segment to segment. About 50% of the parameters are then coupled, while the others are uncoupled.

For each scenario we generate 25 data sets with $m = 5$ (and $m = 10$) data points per segment, i.e. 150 data sets in total. To each data set we add observational noise using a signal-to-noise ratio of 3. For each node Z_i we compute the standard deviation s_i of its values $z_{i,1}, \ldots, z_{i,4m+1}$, and we then add to each $z_{i,j}$ the realization of a $N(0, \theta^2)$ distribution with variance $\theta^2 = (s_i/3)^2$. Note that [24] rotated the regression coefficient vectors at changepoints. The coupled (uncoupled) scenario corresponds to the rotation angle $\alpha = 0°$ ($\alpha = 180°$) in [24].

4.2.2 Yeast gene expression data

By means of synthetic biology [8] designed a network with $N = 5$ genes and $M = 8$ edges in *S. cerevisiae* (yeast); the true network is shown in Figure 4.9. With quantitative Real-Time Polymerase Chain Reaction (RT-PCR), [8] then measured in vivo gene expression data: first under galactose- and then under glucose-metabolism. For both carbon sources the network structure is identical, but the strengths of the regulatory processes (i.e. the network parameters) change with the carbon source ([8]). For each gene Z_i, 16 measurements were taken in galactose $d_1^i, \dots, d_{16}^i$ and 21 measurements were taken in glucose $d_1^{i,*}, \dots, d_{21}^{i,*}$, with 20 minutes intervals in between measurements. For both parts of the time series the initial measurements d_1^i and $d_1^{i,*}$ were taken while extant glucose (galactose) was washed out and new galactose (glucose) was supplemented. We follow [24] and pre-process the data as follows: We withdraw the initial measurements from the washing period, before we re-merge the two time series parts. After a gene-wise zscore-standardization (to mean 0 and variance 1) we build for each gene Z_i the response vector $\mathbf{y} = (d_3^i, \dots, d_{16}^i, d_3^{i,*}, \dots, d_{21}^{i,*})^{\mathsf{T}}$ and use the other genes Z_j $(j \neq i)$ as covariates. For explaining $\mathbf{y}$ we use the shifted values: $(d_2^j, \dots, d_{15}^j, d_2^{j,*}, \dots, d_{20}^{j,*})^{\mathsf{T}}$.

---

[6] As in [24] the initial values $z_{i,1}$ are sampled from $N(0, 0.05^2)$ distributions.

4.2.3 Arabidopsis gene expression data

The circadian clock in *Arabidopsis thaliana* synchronizes the plant metabolism with the daily 24-h photo period (i.e. with the daily dark:light cycle), which is caused by the rotation of the earth. The circadian clock is capable of anticipating the external cycle and can thus optimize the gene regulatory processes w.r.t. the expected (=entrained) photo period. Thereby the structure of the regulatory network does not change, but the strengths of the gene interactions depend on the entrained photo period. In four experiments (E1-E4) Arabidopsis plants were entrained in different dark:light cycles, before data were collected every 2 or 4 hours under constant light:[7]

- **E1**: Dark:light entrainment: **12h:12h**, then 12 measurements at **4h** intervals.
- **E2**: Dark:light entrainment: **12h:12h**, then 13 measurements at **4h** intervals.
- **E3**: Dark:light entrainment: **10h:10h**, then 13 measurements at **2h** intervals.
- **E4**: Dark:light entrainment: **14h:14h**, then 13 measurements at **2h** intervals.

We concentrate on the $N = 9$ core clock genes: LHY, TOC1, CCA1, ELF4, ELF3, GI, PRR9, PRR5, and PRR3, and we merge the data into one single time series by arranging the individual data successively. For each of the four initial points we do not have values for the potential covariates, so that we cannot use them as response values.

---

[7] RNA was measured using Affymetrix microarrays and an RMA normalisation was applied. For the technical details we refer to [16], [44], and [26].

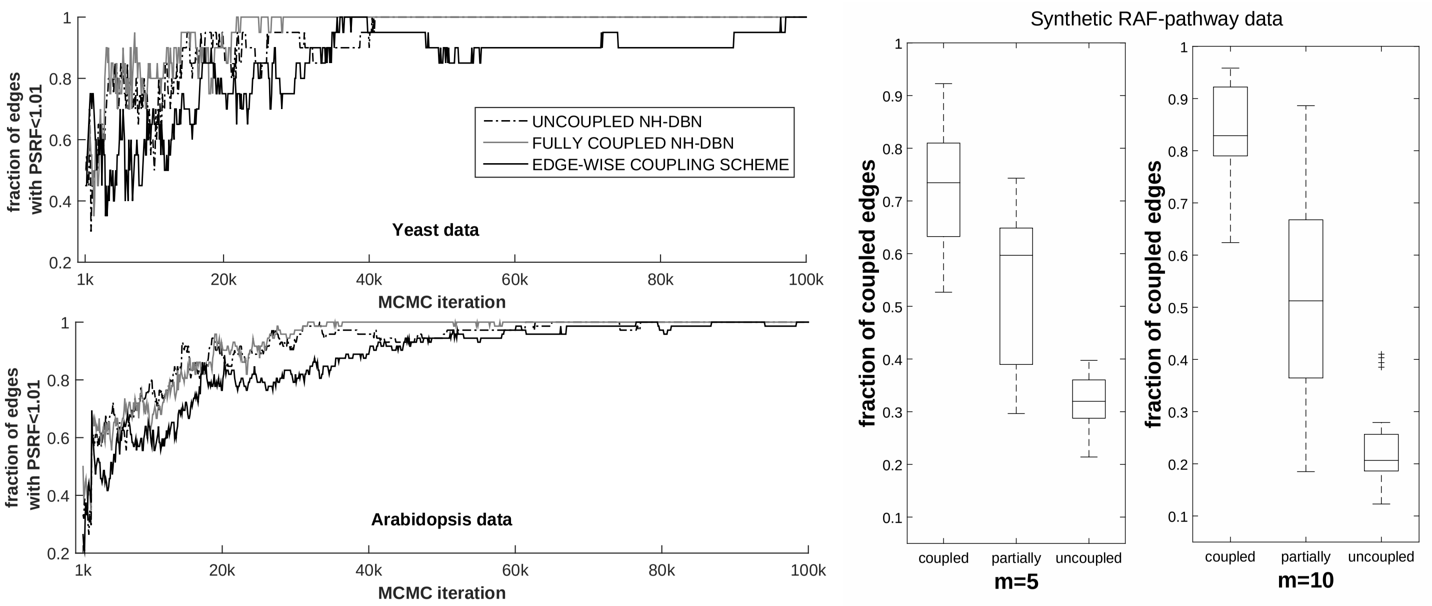


Figure 4.4: Diagnostics. *Left:* Convergence diagnostics based on the potential scale reduction factors (PSRFs). For the yeast and the Arabidopsis data we ran $H = 10$ MCMC simulations and for each edge we computed a PSRF. The plots show the fractions of edges with $PSRF < 1.01$ monitored along the iterations. *Right:* Boxplots of the fractions of coupled edges for the RAF-pathway data scenarios, as inferred with the new partially edge-wise coupled NH-DBN. For each data set we computed the average fraction of coupled edges during the sampling phase.

4.3 Hyperparameter and simulation settings

Figures 4.1-4.2 show a graphical presentation of the proposed model. To be consistent with earlier studies we assume all covariate sets, π, containing up to 3 covariates to be equally likely, $p(\pi) \propto c$, while $p(\pi) = 0$ if $|\pi| > 3$ ('fan-in restriction'). For the inverse Gamma distributed parameters σ^2, λ_u and λ_c we select the shape and rate parameters: $a_\sigma = b_\sigma = 0.005$, $a_c = a_u = 2$ and $b_c = b_u = 0.2$, as in [38] and [24], respectively. Pre-simulations with different hyperparameters confirmed the trends reported in [24], namely robustness w.r.t. the hyperparameters. To ensure a fair comparison we use the same hyperparameters for the competing models. For the real-world applications we infer the segmentations of the time series from the data. For the RAF pathway data we follow [24] and [25] and assume the changepoints to be known so that we keep them fixed. For generating posterior samples we run the MCMC algorithm, outlined in Table 4.1, for 100,000 (100k) iterations. Setting the burn-in phase to 50k and sampling every 100th graph during the sampling phase, yields $W = 500$ samples from the posterior. As described in Subsection 4.1.7, we used potential scale reduction factors (PSRFs) to monitor convergence. For all data sets all PSRF's were below 1.01 after 100k iterations. Figure 4.4 shows the convergence monitors for the yeast and for the Arabidopsis data, and the data-scenario-specific average fractions of coupled edges for the synthetic RAF pathway data, as inferred with the new partially edge-wise coupled NH-DBN.

4.4 Empirical results

4.4.1 Results on synthetic RAF-pathway data

On the synthetic RAF pathway data from Subsection 4.2.1 we compare the performance of the new model with the established ('published') NH-DBNs: the uncoupled, the fully sequentially coupled, and the fully globally coupled NH-DBN, see Subsection 4.1.6 for details.[8] Figure 4.5 shows the results: Neither the uncoupled nor the fully sequentially coupled model yield a significantly better network reconstruction accuracy than the proposed model for any scenario. But there are scenarios where the proposed model significantly outperforms them: The proposed model is significantly superior to the fully sequentially coupled NH-DBN (to the uncoupled NH-DBN) for the partially edge-wise coupled and for the uncoupled data (for the coupled data).

When comparing the proposed model with the globally coupled NH-DBN, we see larger differences for all three scenarios: For the coupled data scenario the globally coupled NH-DBN outperforms the proposed model (AUC differences: -0.04 ($m = 10$) and -0.12 ($m = 5$)), while the proposed model is clearly superior to the globally coupled NH-DBN for the other two scenarios with four

---

[8] As we assume the changepoints to be known, the HMM-DBN here corresponds to the uncoupled NH-DBN.

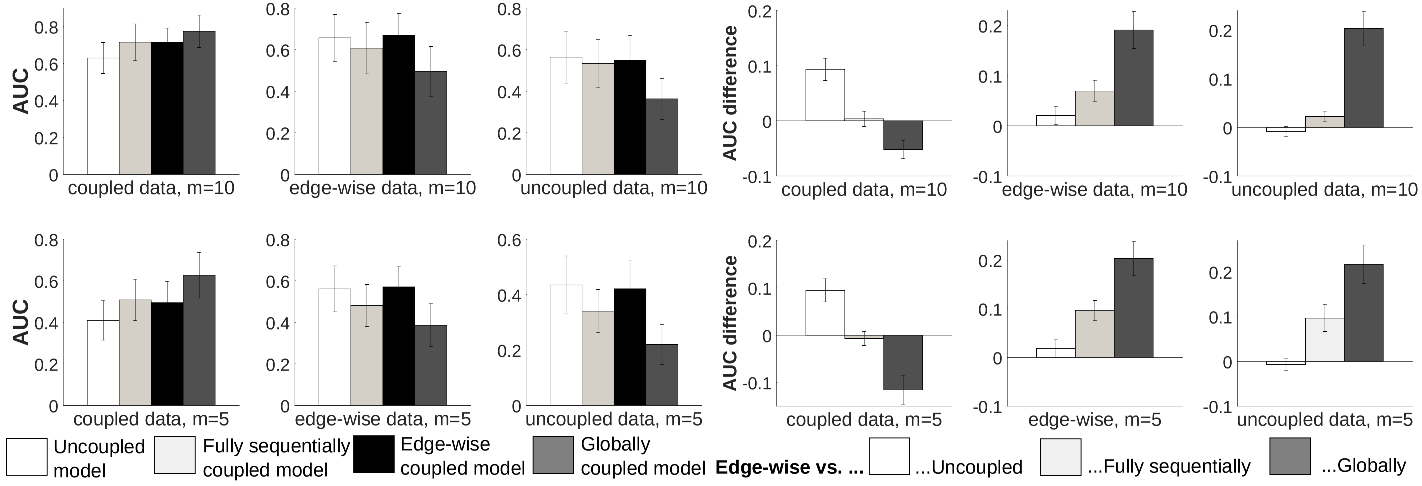


Figure 4.5: RAF-pathway. For the RAF pathway with $N = 11$ nodes and $M = 20$ edges, data with $H = 4$ segments and $m = 5$ and $m = 10$ data points per segment were generated under the scenarios: (i) coupled, (ii) (partially) edge-wise coupled and (iii) uncoupled data, see Section 4.2.1 for details. The left histograms show the average precision-recall AUC values over 25 data instantiations, with errorbars indicating standard deviations. The right histograms show the relative AUC differences in favour of the edge-wise coupled model, with errorbars indicating confidence intervals of paired t-tests.

AUC differences in between 0.19 and 0.22. This shows that the globally coupled model imposes a very strong coupling on the segment-specific regression coefficients. Not surprisingly, this is advantageous for our coupled scenario, where the coefficients stay constant over time, but the strong coupling becomes very counter-productive when all (or about 50% of the) regression coefficients substantially change from segment to segment. Like the fully sequentially coupled model, the globally coupled NH-DBN model has no effective mechanism to uncouple the regression coefficients. This shortcoming is more critical for the globally coupled NH-DBN, as it has only one parameter, λ_u, regulating the variance of the regression coefficient vectors. For uncoupled data λ_u has to take high values what makes all priors diffuse. The fully sequentially coupled model has two separate parameters, λ_u (for $h = 1$) and λ_c (for $h > 1$), and can keep at least the λ_u parameter in a meaningful range, yielding an appropriate prior for the first segment.

4.4.2 Results on yeast gene expression data

The yeast network was designed by means of synthetic biology and is known; see Subsection 4.2.2. We can thus cross-compare the network reconstruction accuracies on real in vivo gene expression data. In this benchmark study we include all seven changepoint segmented sequentially coupled NH-DBNs from Figure 4.3. With each model we run $H = 10$ independent MCMC simulations. Each simulation yields edge scores $\hat{e}_{i,j}$ for all potential edges. We arrange the simulation-specific scores in vectors $\mathbf{v}_{m,h}$, where m indicates the model and h the simulation. In addition we build the true vector $\mathbf{v}^*$ whose entries are 1 if the corresponding edge is present, or 0 otherwise. We then zscore-standardize all vectors,[9] and project them onto the first two principal components. Figure 4.6 shows the resulting PCA plot and a drendogram of the model-specific average score vectors. For the drendogram we clustered the model-specific average score vectors based on their Euclidean distances. The first two principal components (PCs) explain 78% and 10% (together $\approx 90\%$) of the variance, so that the 2-dimensional PCA plot conserves most of the information. Taking into account that the first PC (eigenvalue $\lambda_1 = 1.94$) has much more weight than the second PC (eigenvalue $\lambda_2 = 0.24$), the following trends can be seen from Figure 4.6: (i) The model-specific simulations are always closely grouped together, i.e. independent simulations yield similar edge scores what is a good indicator for convergence. (ii) Nearest to the true network is the proposed model, while the homogenous model has the furthest distance to the true network. The partially segment-wise coupled model is 2nd nearest to the true network. (iii) The fully sequentially coupled model and its generalization (with segment-specific coupling strengths) yield similar edge scores, so that this generalization appears to have a minor effect here. (iv) The points of the gene-wise switch and the partially coupled NH-DBN are near to the uncoupled NH-DBN. We conclude that both NH-DBNs

---

[9] $\mathbf{v} \leftarrow (\mathbf{v} - \bar{v}\mathbf{1})/s$ where $\bar{v}$ and s_v are the mean and the standard deviation of the elements of $\mathbf{v}$.

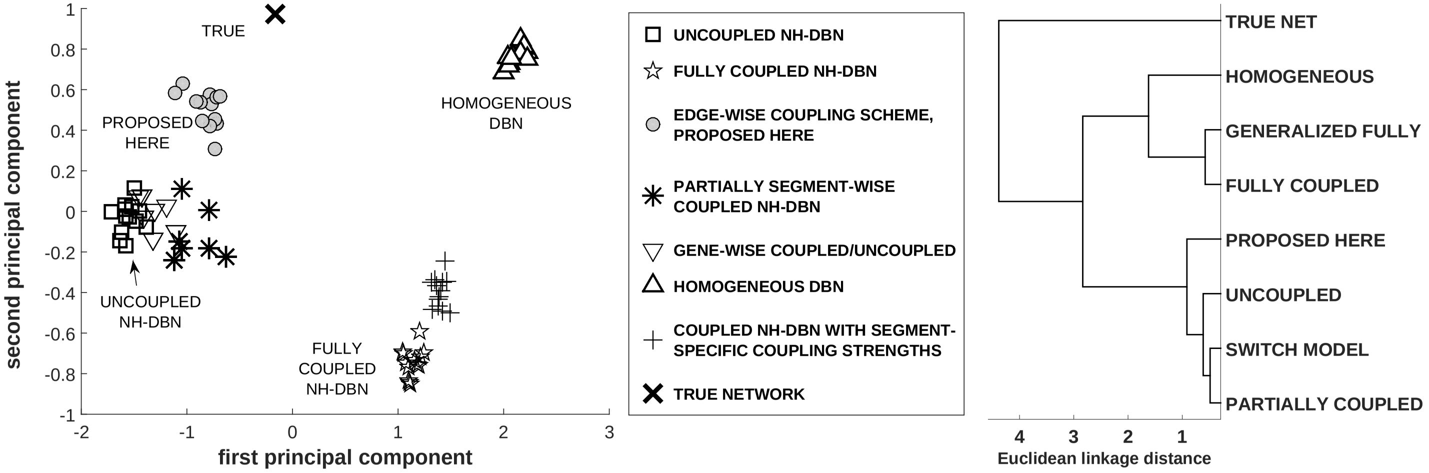


Figure 4.6: PCA and dendrogram plot of the edge scores obtained for the yeast data. Every MCMC simulation outputs edge scores $\hat{e}_{i,j}$ for all edges. We arrange the scores of each simulation vector-wise and standardize all vectors (to mean 0 and variance 1). *Left:* Standard PCA plot to project the set of vectors onto the first two principal components. The two components explain 78%+10% of the variance. *Right:* For each model we then average the score vectors across the simulations and cluster the model-specific average vectors based on their Euclidean distances. The drendogram shows the results.

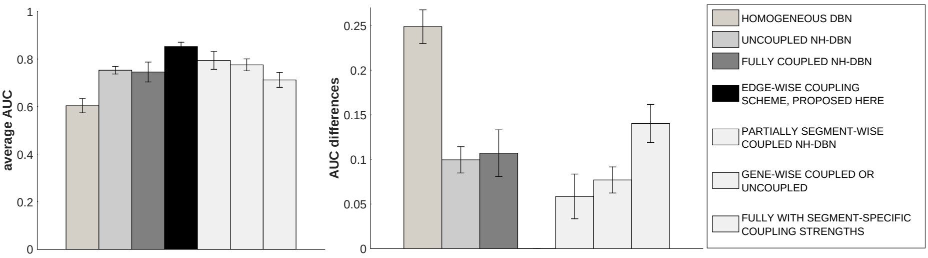


Figure 4.7: Network reconstruction accuracy for yeast. The left histogram shows the precision-recall AUC values, averaged across $H = 10$ MCMC simulations, with error bars indicating standard deviations. The right histogram shows the relative AUC difference in favour of the proposed model, with error bars indicating the confidence intervals of unpaired t-tests. The black bar refers to the proposed model. The bars to the left (right) of the black bar refer to competing models; see Subsection 4.1.6 for an overview.

infer the majority of genes to be uncoupled. From the drendogram it becomes obvious that there are two clusters. In the first cluster the two fully coupled NH-DBNs, which strictly enforce coupling, cluster with the homogeneous DBN. In the second cluster the more flexible NH-DBNs, which feature effective mechanisms to uncouple, cluster with the uncoupled NH-DBN.

Figure 4.7 shows the resulting network reconstruction accuracies in terms of average precision-recall AUC values and AUC differences. The proposed edge-wise coupled model, which has the minimal distance to the true network in the PCA plot, yields the highest average AUC value and performs significantly better than the six competing NH-DBNs. The AUC values and the PCA plot are in good agreement: the AUC values consistently decrease with the distance to the true network in the PCA plot. In addition, we also compare the performance of the proposed model with the conceptually different 'HMM-DBN' model from [22], see Subsection 4.1.6 for a brief review. The HMM-DBN model, which replaces the multiple changepoint process (CPS) of the uncoupled NH-DBN by a Hidden Markov Model (HMM), here yields an average AUC value of 0.8033. The AUC difference in favour of the proposed edge-wise coupled model is 0.049 (t-test p-value: $p < 0.01$).

For the the proposed NH-DBN and its two competitors, the uncoupled and the the fully coupled NH-DBN, we now average the model-specific edge scores across the $H = 10$ simulations to extract for each model a concrete network prediction with $M = 8$ edges (having the highest scores). Figure 4.8 shows the true and the predicted networks. The predictions of the uncoupled and the edge-wise coupled NH-DBN are similar. But the uncoupled NH-DBN assigns its third highest score to the false edge $GAL80 \to ASH1$, while the proposed model assigns its six highest scores to true edges. The fully coupled NH-DBN infers two different false positive edges, and the edge $GAL4 \to CBF1$ gets the third highest score. The similarity of the uncoupled and the edge-wise coupled prediction is consistent with the PCA plot in Figure 4.6, where the points of the uncoupled and the edge-wise coupled models are close together while the points of the coupled NH-DBN are apart. Figure 4.9 shows the resulting model-specific precision-recall curves. The proposed model infers the highest scores for 6 true edges, while the competing NH-DBNs have already a false positive among the three highest-scoring edges, leading to reduced AUC values.

4.4.3 Results on Arabidopsis gene expression data

The absence of a gold standard for the circadian clock network in *A. thaliana* renders an objective evaluation of the network reconstruction accuracy infeasible. We therefore focus on the newly proposed model and use this application to illustrate that the partially edge-wise coupling mechanism allows for more biological insight. Again we run $H = 10$ independent MCMC with the new model and we average the simulation-specific marginal edge posterior probabilities. Onto the scores we impose a threshold ψ such that 20 edges were extracted; the corresponding threshold was around $\psi = 2/3$. Recalling that $\hat{e}_{i,j}$ refers to covariate X_i for response gene Z_j, we then consider the corresponding sampled δ_i indicator variables and estimate the posterior probabilities that the corresponding edge was a mainly 'coupled' (or 'uncoupled') one. If the posterior probability $\hat{p}(\delta_i = 1|\mathbf{D})$ of the state 'coupled' was double as likely as the probability $\hat{p}(\delta_i = 0|\mathbf{D})$ of the state 'uncoupled', we call the edge a 'coupled' edge. Correspondingly we call edges 'uncoupled' if $p(\delta_i = 0|\mathbf{D}) > 2p(\delta_i = 1|\mathbf{D})$, and we call edges 'mix edges' if none of the conditions is satisfied. This way we could classify the 20 edges into 7 coupled, 7 uncoupled and 6 mix edges. The predicted network with different edge symbols for the edge types is shown in Figure 4.10. As many genes of the circadian clock network are co-regulated by the presence (or absence) of light, we would argue that it is a reasonable finding that some of the interactions are stable ('coupled') under constant light condition. In the biological literature we could find evidence for some features of our network. The most important key feature of the circadian clock network is the feedback loop between LHY and $TOC1$. This feedback is already known since [41] to play a central role in circadian regulation (see also more recent works, e.g., [47]). Our new model does not only infer this feedback loop but also indicates that the regulatory effect of LHY on $TOC1$ is stable, while the opposite regulatory effect of $TOC1$ on LHY is time-varying. Focusing our evaluation on those two genes, we further found the following: The regulatory effect of $ELF3$ on $TOC1$, e.g. reported in [43], is also time-varying, while the edge from GI to $TOC1$, also reported in [43], is not. The edges from $ELF3$ to LHY and from LHY to $ELF4$ have been reported in [33]. Our model finds both edges and provides evidence that the regulatory effect of $ELF3$ on LHY changes over time. Finally, for the effects of TOC1 on the $PRR3$ and $PRR9$ (which can be found in the network of [47]) our model switches between both states 'coupled' and 'uncoupled' so that the regulatory effect cannot be specified further.

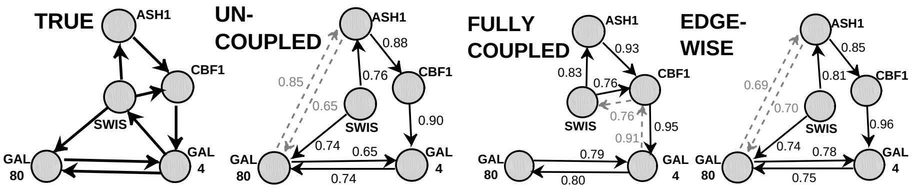


Figure 4.8: True and predicted yeast networks. For the NH-DBNs we averaged the model-specific edge scores across $H = 10$ simulations. As the true network has $M = 8$ edges, we extracted the 8 edges with the highest scores. Grey edges refer to false positive edges. The edge labels give the edge scores.

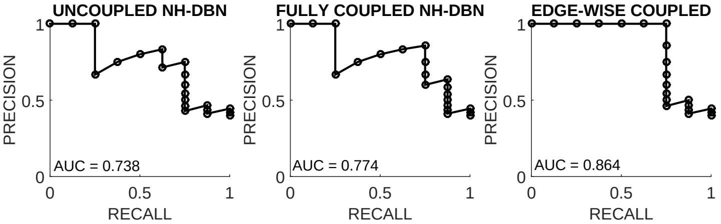


Figure 4.9: Precision-recall curves for yeast network. For the uncoupled, the fully coupled and the (partially) edge-wise coupled model we computed the average edge scores across the $H = 10$ simulations. The figure shows the model-specific precision recall curves with $(0, 1)$ being a pseudo point (the starting point). All three predicted networks in Figure 4.8 give the point $(0.75, 0.75)$ (Recall: 75%, Precision: 75%).

4.5 Discussion and conclusions

We have proposed a new non-homogeneous dynamic Bayesian network (NH-DBN) model with partially edge-wise coupled network parameters. A change-point process is used to divide the temporal data into segments and the network interaction parameters are assumed to change from segment to segment. Unlike in the uncoupled NH-DBN, where *all* interaction parameters have to be learned separately for each segment, and unlike the fully sequentially coupled NH-DBN, which enforces *all* parameters to stay similar among segments, our new model infers for each individual edge (i.e. 'edge-wise') whether the corresponding interaction parameter should be coupled or better stay uncoupled. Loosely speaking, our new model combines features from the uncoupled and the coupled NH-DBN and then follows the paradigm: '*Let the data speak.*'. It comprises the uncoupled and the fully coupled NH-DBN as limiting cases: It effectively becomes the uncoupled (coupled) NH-DBN model when it couples all edges (no edge at all).

In Subsections 4.4.1 and 4.4.2 we have empirically shown on synthetic RAF pathway data and on a benchmark yeast gene expression time series that the new model, overall, reaches a higher network reconstruction accuracy than the competing NH-DBN models. In Subsection 4.4.2 we have used a principal component analysis (PCA) and a cluster analysis to visualize (dis-)similarities

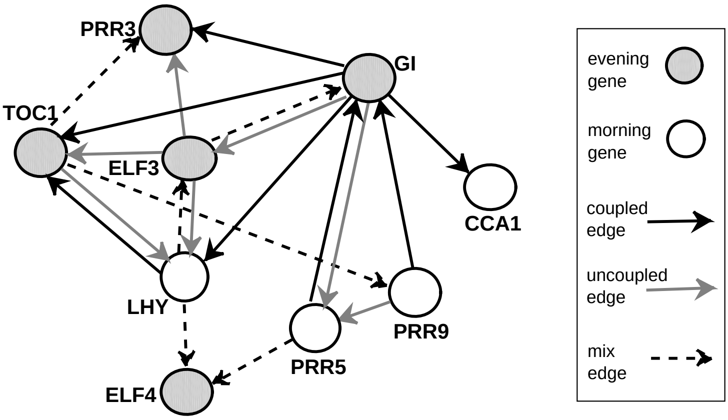


Figure 4.10: Arabidopsis network. The figure shows a prediction which was obtained with the proposed model. Morning (evening) genes are represented as white (grey) nodes. We set the threshold ψ on the edge scores $\hat{e}_{i,j}$ such that 20 edges were extracted. Different edges indicate if the parameters were coupled (black) or uncoupled (grey) or a mixture thereof (black dotted edges), see main text for further details.

between various related NH-DBN models. In Subsection 4.4.3 we have used the new model to infer the circadian clock network in *Arabidopsis thaliana*. Unlike its competitors, our new model here not only outputs a network prediction, but also allows to distinguish between edges whose regulatory effects stay similar across time and edges whose regulatory effects are subject to more substantial temporal changes.

As the proposed 'partial edge-wise parameter coupling' concept is generic, it can also be implemented for many related NH-DBN models. E.g. the (fully) globally coupled NH-DBN model from [25] could be easily implemented in an edge-wise globally coupled version. Also for NH-DBNs with time-varying network structures the new concept can give an improvement. E.g. the NH-DBNs presented in [49], [38] and [14] do not allow for any information-sharing w.r.t. the network interaction parameters. The fully sequential ([24]) and the fully global ([25]) coupling scheme cannot be incorporated into those models, as the parent node sets (i.e. the covariates) vary from segment to segment. Under the condition that parameters associated with non-omnipresent edges (covariates) have to stay 'uncoupled', the edge-wise coupling scheme could be directly transferred to the NH-DBNs with time-varying network structures. The latter adaptation is rather ad-hoc and, thus, most likely suboptimal. There might be more adequate alternatives, whose development and exploration we leave for future research.

4.6 Appendix

Here we derive Equation (4.14) from Subsection 4.1.2. For notational convenience we here do not make explicit that the (marginal) likelihoods depend on the covariate set π and the changepoint set $\boldsymbol{\tau}$.

A standard rule for Gaussian integrals (see, e.g., Section 2.3.2 in [5]) is that:

$$\mathbf{y}|\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}) \text{ where } \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S})$$

implies $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\mu}, \boldsymbol{\Sigma} + \mathbf{X}\mathbf{S}\mathbf{X}^{\mathsf{T}})$ for the marginal distribution: $p(\mathbf{y}) = \int p(\mathbf{y}, \boldsymbol{\beta})\ d\boldsymbol{\beta}$. We use this rule for computing the marginal distributions of $\mathbf{y}_h$, marginalized over $\boldsymbol{\beta}_h$, for our model:

$$\begin{aligned} p(\mathbf{y}_h|\sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta}) &= \int p(\mathbf{y}_h, \boldsymbol{\beta}_h|\sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta})\ d\boldsymbol{\beta}_h \\ &= \int p(\mathbf{y}_h|\boldsymbol{\beta}_h, \sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta}) p(\boldsymbol{\beta}_h|\sigma^2, \lambda_u, \lambda_c, \boldsymbol{\delta})\ d\boldsymbol{\beta}_h \end{aligned}$$

With $\mathbf{y}_h|(\boldsymbol{\beta}_h, \ldots) \sim \mathcal{N}(\mathbf{X}_h\boldsymbol{\beta}_h, \sigma^2\mathbf{I})$ and

$$\boldsymbol{\beta}_h|(\ldots) \sim \begin{cases} \mathcal{N}(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_0, \sigma^2 diag\{\lambda_u \mathbf{1}\}) & \text{if } h = 1 \\ \mathcal{N}(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}, \sigma^2 diag\{\lambda_c\boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\}) & \text{if } h > 1 \end{cases}$$

where $\tilde{\boldsymbol{\beta}}_0 := \mathbf{0}$, so that $\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_0 = \mathbf{0}$, the rule implies for the marginalisation over $\boldsymbol{\beta}_h$:

$$\begin{aligned} &\mathbf{y}_h|(\sigma^{-2}, \lambda_u, \lambda_c, \boldsymbol{\delta}) \\ &\sim \begin{cases} \mathcal{N}(\mathbf{X}_1(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_0), \sigma^2[\mathbf{I} + \mathbf{X}_1 diag\{\lambda_u\mathbf{1}\}\mathbf{X}_1^{\mathsf{T}}]) & \text{if } h = 1 \\ \mathcal{N}(\mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}), \sigma^2[\mathbf{I} + \mathbf{X}_h diag\{\lambda_c\boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\}\mathbf{X}_h^{\mathsf{T}}]) & \text{if } h > 1 \end{cases} \end{aligned}$$

As a function of σ^{-2} we have:

$$\begin{aligned} &p(\mathbf{y}_h|\sigma^{-2}, \ldots) \propto (\sigma^{-2})^{T_h/2} \\ &\quad \cdot \exp\{-\frac{1}{2}\sigma^{-2}(\mathbf{y}_h - \mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}))^{\mathsf{T}}(\mathbf{I} + \mathbf{X}_h\boldsymbol{\Sigma}_h\mathbf{X}_h^{\mathsf{T}})^{-1}(\mathbf{y}_h - \mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}))\} \end{aligned}$$

where T_h is the length of the vector $\mathbf{y}_h$ and (cp. with Equation (4.16)):

$$\boldsymbol{\Sigma}_h := \begin{cases} diag\{\lambda_u\mathbf{1}\} & \text{if } h = 1 \\ diag\{\lambda_c\boldsymbol{\delta} + \lambda_u(\mathbf{1} - \boldsymbol{\delta})\} & \text{if } h > 1 \end{cases}$$

As a function of σ^{-2} it follows for the product of the segment-specific marginal likelihoods:

$$\prod_{h=1}^{H} p(\mathbf{y}_h|\sigma^{-2}, \lambda_u, \lambda_c, \boldsymbol{\delta}) \propto (\sigma^{-2})^{0.5 \cdot T} \exp\{-0.5 \cdot \sigma^{-2} \cdot \Delta^2\}$$

where $\Delta^2 := \sum_{h=1}^{H}(\mathbf{y}_h - \mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}))^{\mathsf{T}}(\mathbf{I} + \mathbf{X}_h\boldsymbol{\Sigma}_h\mathbf{X}_h^{\mathsf{T}})^{-1}(\mathbf{y}_h - \mathbf{X}_h(\boldsymbol{\delta} \odot \tilde{\boldsymbol{\beta}}_{h-1}))$ (cp. Equation (4.15)) and $T = \sum T_h$ is the no. of data points. Using the latter result, we obtain straightforwardly:

$$\begin{aligned} p(\sigma^{-2}|\mathbf{y}_1, \ldots, \mathbf{y}_H, \lambda_u, \lambda_c, \boldsymbol{\delta}) &\propto \left(\prod_{h=1}^{H} p(\mathbf{y}_h|\lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau})\right) \cdot p(\sigma^{-2}) \cdot p(\lambda_u) \cdot p(\lambda_c) \cdot p(\boldsymbol{\delta}) \\ &\propto (\sigma^{-2})^{a_\sigma + 0.5 \cdot T - 1} \exp\{-\sigma^{-2}(b_\sigma + 0.5 \cdot \Delta^2)\} \end{aligned}$$

Equation (4.14):

$$\sigma^{-2}|(\mathbf{y}_1, \ldots, \mathbf{y}_H, \lambda_u, \lambda_c, \boldsymbol{\delta}, \pi, \boldsymbol{\tau}) \sim GAM\left(a_\sigma + 0.5 \cdot T, b_\sigma + 0.5 \cdot \Delta^2\right)$$

follows from the shape of the latter distribution.

Chapter 5

Partially NH-DBNs based on Bayesian regression models with partitioned design matrices

In many real-world applications, e.g. in systems biology, data are often collected under different experimental conditions. That is, instead of one single (long) time series that has to be segmented, there are K (short) time series. The data are then intrinsically divided into K unordered components, and there is no need for inferring the segmentation. In this situation, it is normally not clear a priori whether the network parameters stay constant across components or whether they vary from component to component. If the parameters stay constant, all data can be merged and analysed with one single homogeneous DBN. If the parameters are component-specific, then the data should be analysed by a NH-DBN. The bottleneck of both approaches is that *all* parameters are assumed to be either constant (DBN) or component-specific (NH-DBN). In real-world applications there can be both types of parameters. E.g. if a variable Y is regulated by two other variables, symbolically $X_1 \to Y \leftarrow X_2$, then the interaction $X_1 \to Y$ can stay constant, while $X_2 \to Y$ might be component-specific, e.g. for $K = 2$ and in terms of a regression model:

$$E[Y|X_1 = x_1, X_2 = x_2] = \begin{cases} \alpha x_1 + \beta x_2 & \text{if } k = 1 \\ \alpha x_1 + \gamma x_2 & \text{if } k = 2 \end{cases} \tag{5.1}$$

A DBN ignores that β and γ are different. A NH-DBN has to infer the same parameter α two times separately. This increases the inference uncertainty, and is thus critical when the available data are sparse.

No tailor-made model for the situation in (5.1) has been proposed yet. To fill this gap, we propose a partially non-homogeneous dynamic Bayesian network (partially NH-DBN) model, which infers the best trade-off between a DBN and a NH-DBN. The new partially NH-DBN model operates on the individual interactions (network edges). For each interaction there is a parameter, and the model infers from the data whether the parameter is constant or component-specific. We implement the new model in a hierarchical Bayesian regression framework, since this model class reached the highest network reconstruction accuracy in the cross-method comparison by [1]. But we note that the underlying idea is generic and could also be implemented in other frameworks, e.g. via L1-regularized regression model ('LASSO').

Furthermore, in Section 5.1.5 we propose a Gaussian process (GP) based method to deal with the problem of non-equidistant measurements. The standard assumption for all NH-DBNs is that data are measured at equidistant time points. For applications where this assumption is not fulfilled, we propose to use a GP to predict the values at equidistant data points and to replace the non-equidistant values by predicted equidistant values. We will make use of the GP method when analysing the mTORC1 timecourse data in Section 5.3.4.

The work, presented in this chapter, has been accepted for publication (in press) in Bioinformatics (2018) (see [59]).

5.1 Methods

DBNs and NH-DBNs are used to infer networks showing the regulatory interactions among variables $Z_1, \dots, Z_N$. The interactions are subject to a time lag, so that there is no need for an acyclic network structure. Hence, dynamic network inference can be thought of as inferring the covariate sets for N independent regression models. In the i-th model, Z_i is the response and the remaining $N_\star := N - 1$ variables $Z_1, \dots, Z_{i-1}, Z_{i+1}, \dots, Z_N$ at time point $t-1$ are used as potential covariates for Z_i at time point t. The goal is to infer a covariate set for each Z_i, and the system of covariate sets describes a network; see Section 5.1.6 for details. As the same regression model is applied to each Z_i separately, we describe it using a general notation, where Y is the response and $X_1, \dots, X_n$ are the covariates.

5.1.1 Bayesian regression with partitioned design matrix

We consider a regression model with response Y and covariates $X_1, \dots, X_n$. We assume that data were measured under K experimental conditions, which we refer to as K components. We further assume that the data for each component $k \in \{1, \dots, K\}$ were measured at equidistant time points $t = 1, \dots, T_k$. Let $y_{k,t}$ and $x_{i,k,t}$ denote the values of Y and X_i at the t-th time point of component k. In dynamic networks, the interactions are subject to a time lag $\mathcal{O}$, which is usually set to one time point. That is, the values $x_{1,k,t}, \dots, x_{n,k,t}$ correspond to the response value $y_{k,t+1}$. For each component k we build a component-specific response vector $\mathbf{y}_k$ and the corresponding design matrix $\mathbf{X}_k$, where $\mathbf{X}_k$ includes a first column of 1′s for the intercept:

$$\mathbf{y}_k = (y_{k,2}, \ldots, y_{k,T_k})^\mathsf{T}, \quad \mathbf{X}_k = \begin{pmatrix} \mathbf{1} & \mathbf{x}_{1,k} & \ldots & \mathbf{x}_{n,k} \end{pmatrix}$$

where $\mathbf{x}_{i,k} = (x_{i,k,1}, \ldots, x_{i,k,T_k-1})^\mathsf{T}$

For each k we could assume a separate Gaussian likelihood:

$$\mathbf{y}_k \sim \mathcal{N}_{T_k-1}(\mathbf{X}_k \boldsymbol{\beta}_k, \sigma_k^2 \mathbf{I}) \qquad (k = 1, \ldots, K) \tag{5.2}$$

where $\mathbf{I}$ is the identity matrix, $\boldsymbol{\beta}_k = (\beta_{k,0}, \beta_{k,1}, \ldots, \beta_{k,n})^\mathsf{T}$ is the component-specific vector of regression coefficients, and σ_k^2 is the component-specific noise variance. Imposing independent priors on each pair $\{\boldsymbol{\beta}_k, \sigma_k^2\}$, leads to K independent models. Alternatively, we could merge the data $\mathbf{y} := (\mathbf{y}_1^\mathsf{T}, \ldots \mathbf{y}_K^\mathsf{T})^\mathsf{T}$ and $\mathbf{X} := (\mathbf{X}_1^\mathsf{T}, \ldots, \mathbf{X}_K^\mathsf{T})^\mathsf{T}$ and employ one model for the merged data:

$$\mathbf{y} \sim \mathcal{N}_T(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}) \quad \text{where } T := \sum_{k=1}^{K} (T_k - 1) \tag{5.3}$$

so that $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_n)^\mathsf{T}$ would apply to all components.

When some covariates have a component-specific and other covariates have a constant regression coefficient, both likelihoods (5.2) and (5.3) are suboptimal. For this situation, we propose a new partially non-homogeneous regression model that infers the best trade-off from the data. The key idea is to use a likelihood with a partitioned design matrix.

For now, we assume that we know for each coefficient whether it is component-specific or constant. Let the intercept and the first $n_1 < n$ coefficients stay constant while the remaining $n_2 = n - n_1$ coefficients are component-specific. We then have the regression equation:

$$y_{k,t+1} = \beta_0 + \sum_{i=1}^{n_1} \beta_i \cdot x_{i,k,t} + \sum_{i=n_1+1}^{n} \beta_{k,i} \cdot x_{i,k,t} + \epsilon_{k,t+1}$$

where $\epsilon_{k,t+1} \sim \mathcal{N}(0, \sigma^2)$, and the likelihood takes the form:

$$\mathbf{y} \sim \mathcal{N}_T(\mathbf{X}_B \boldsymbol{\beta}_B, \sigma^2 \mathbf{I}) \tag{5.4}$$

where $\boldsymbol{\beta}_B$ is a vector of $(1 + n_1 + K \cdot n_2)$ regression coefficients, and $\mathbf{X}_B$ is a partitioned matrix with $T = \sum (T_k - 1)$ rows and $(1 + n_1) + (K \cdot n_2)$ columns. E.g. for $K = 2$ the matrix $\mathbf{X}_B$ has the structure:

$$\begin{pmatrix} \mathbf{1} & \mathbf{x}_{1,1} & \ldots & \mathbf{x}_{n_1,1} & \mathbf{x}_{n_1+1,1} & \ldots & \mathbf{x}_{n,1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{1} & \mathbf{x}_{1,2} & \ldots & \mathbf{x}_{n_1,2} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{x}_{n_1+1,2} & \ldots & \mathbf{x}_{n,2} \end{pmatrix},$$

where $\mathbf{x}_{i,k} = (x_{i,k,2}, \ldots, x_{i,k,T_k-1})^\mathsf{T}$, and $\boldsymbol{\beta}_B$ is of the form:

$$((\beta_0, \beta_1, \ldots, \beta_{n_1}), (\beta_{n_1+1,1}, \ldots, \beta_{n,1}), (\beta_{n_1+1,2}, \ldots, \beta_{n,2}))^\mathsf{T}$$

The first subvector of $\boldsymbol{\beta}_B$ is the vector $\boldsymbol{\beta}_\star := (\beta_0, \beta_1, \ldots, \beta_{n_1})^\mathsf{T}$ of the regression coefficients that stay constant, and then there is a subvector $\boldsymbol{\beta}_k := (\beta_{n_1+1,k}, \ldots, \beta_{n,k})^\mathsf{T}$ for each component k with the component-specific regression coefficients. For the noise variance parameter σ^2 we use an inverse Gamma prior, $\sigma^{-2} \sim \text{GAM}(a, b)$, and on $\boldsymbol{\beta}_\star$ we impose a Gaussian prior with zero mean vector:

$$\boldsymbol{\beta}_\star \sim \mathcal{N}_{n_1+1}(\mathbf{0}, \sigma^2 \lambda_\star^2 \mathbf{I}) \tag{5.5}$$

For the component-specific vectors $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$ we adapt the idea from [25], and impose a hyperprior:

$$\boldsymbol{\beta}_k \sim \mathcal{N}_{n_2}(\boldsymbol{\mu}, \sigma^2 \lambda_\diamond^2 \mathbf{I}) \quad (k = 1, \ldots, K) \quad \text{and} \quad \boldsymbol{\mu} \sim \mathcal{N}_{n_2}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{5.6}$$

The hyperprior couples the vectors $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$ hierarchically and encourages them to stay similar across components. Re-using the variance parameter σ^2 in (5.5-5.6) allows the regression coefficient vectors and the noise variance to be integrated out in the likelihood, i.e. the marginal likelihood $p(\mathbf{y}|\lambda_\star^2, \lambda_\diamond^2, \boldsymbol{\mu})$ to be computed analytically (see below). For $\lambda_\star^2$ and $\lambda_\diamond^2$ we also use inverse Gamma priors:

$$\lambda_\star^{-2} \sim \text{GAM}(\alpha_\star, \beta_\star) \text{ and } \lambda_\diamond^{-2} \sim \text{GAM}(\alpha_\diamond, \beta_\diamond)$$

The prior of $\boldsymbol{\beta}_B = (\boldsymbol{\beta}_\star^\mathsf{T}, \boldsymbol{\beta}_1^\mathsf{T}, \ldots, \boldsymbol{\beta}_K^\mathsf{T})^\mathsf{T}$ is a product of Gaussians:

$$p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) = p(\boldsymbol{\beta}_\star|\sigma^2, \lambda_\star^2) \cdot \prod_{k=1}^{K} p(\boldsymbol{\beta}_k|\sigma^2, \lambda_\diamond^2, \boldsymbol{\mu})$$

Given σ^2, $\lambda_\diamond^2$, $\lambda_\star^2$, and $\boldsymbol{\mu}$, the Gaussians are independent, so that:

$$\boldsymbol{\beta}_B|(\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \sim \mathcal{N}_{1+n_1+K\cdot n_2}(\tilde{\boldsymbol{\mu}}, \sigma^2 \tilde{\boldsymbol{\Sigma}})$$

$$\text{with: } \tilde{\boldsymbol{\mu}} = (\mathbf{0}^\mathsf{T}, \boldsymbol{\mu}^\mathsf{T}, \ldots, \boldsymbol{\mu}^\mathsf{T})^\mathsf{T} \text{ and } \tilde{\boldsymbol{\Sigma}} = \begin{pmatrix} \lambda_\star^2 \mathbf{I}_\star & \mathbf{0} \\ \mathbf{0} & \lambda_\diamond^2 \mathbf{I}_\diamond \end{pmatrix}$$

where $\mathbf{I}_\star$ is the $(n_1 + 1)$-dimensional and $\mathbf{I}_\diamond$ the $(K \cdot n_2)$-dimensional identity matrix. We have for the posterior distribution:

$$\begin{aligned} p(\boldsymbol{\beta}_B, \sigma^2, \lambda_\star^2, \lambda_\diamond^2, \boldsymbol{\mu}|\mathbf{y}) \propto p(\mathbf{y}|\sigma^2, \boldsymbol{\beta}_B) \cdot p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \ldots \\ \ldots \cdot p(\boldsymbol{\mu}) \cdot p(\sigma^{-2}) \cdot p(\lambda_\star^{-2}) \cdot p(\lambda_\diamond^{-2}) \end{aligned} \tag{5.7}$$

A graphical model representation of the new regression model is provided in Figure 5.1. The full conditional distributions (FCDs) of $\boldsymbol{\beta}_B$, σ^2, $\lambda_\star^2$, $\lambda_\diamond^2$ and $\boldsymbol{\mu}$ can be computed analytically, so that Gibbs-sampling can be applied to generate a posterior sample. As the derivations are mathematically involved, we relegate them to **part A of the Appendix**.

The marginalization rule from Section 2.3.7 of [5] yields:

$$
\begin{aligned}
p(\mathbf{y}|\lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) &= \frac{\Gamma(\frac{T}{2}+a)}{\Gamma(a)} \cdot \frac{\pi^{-\frac{T}{2}}(2b)^a}{\det(\mathbf{C})^{1/2}} \cdot \ldots \\
\ldots \cdot \big(2b + (\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})^\mathsf{T}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})\big)^{-(\frac{T}{2}+a)} \\
\text{where } T := \sum_{k=1}^{K}(T_k - 1), \text{ and } \mathbf{C} := \mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^\mathsf{T}.
\end{aligned}
\tag{5.8}
$$

5.1.2 Inferring the relevant covariates and their types

In typical applications, there is a set of $N_\star$ variables, and the subset of the relevant covariates has to be inferred from the data. Each covariate can be either constant ($\delta = 1$) or component-specific ($\delta = 0$). Let $\boldsymbol{\Pi} = \{X_1, \ldots, X_n\}$ be a subset of the $N_\star$ variables, and let $\boldsymbol{\delta} = (\delta_0, \delta_1, \ldots, \delta_n)^\mathsf{T}$ be a vector of binary variables, where δ_i indicates whether X_i has a constant ($\delta_i = 1$) or component-specific ($\delta_i = 0$) regression coefficient. The first element, δ_0, refers to the intercept.

The goal is then to infer the covariate set $\boldsymbol{\Pi}$ and the corresponding indicator vector $\boldsymbol{\delta}$ from the data. For any combination of $\boldsymbol{\Pi}$ and $\boldsymbol{\delta}$, the partitioned design matrix $\mathbf{X}_B = \mathbf{X}_{B,\boldsymbol{\Pi},\boldsymbol{\delta}}$ can be built, and the marginal likelihood $p(\mathbf{y}|\lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}, \boldsymbol{\Pi}, \boldsymbol{\delta})$ can be computed with (5.8). We get for the posterior:

$$
\begin{aligned}
p(\boldsymbol{\Pi}, \boldsymbol{\delta}, \lambda_\star^2, \lambda_\diamond^2, \boldsymbol{\mu}|\mathbf{y}) \propto p(\mathbf{y}|\lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}, \boldsymbol{\Pi}, \boldsymbol{\delta}) \cdot p(\boldsymbol{\Pi}) \cdot p(\boldsymbol{\delta}|\boldsymbol{\Pi}) \cdot \ldots \\
\ldots \cdot p(\boldsymbol{\mu}|\boldsymbol{\Pi}, \boldsymbol{\delta}) \cdot p(\lambda_\star^2) \cdot p(\lambda_\diamond^2)
\end{aligned}
$$

where $p(\boldsymbol{\mu}|\boldsymbol{\Pi}, \boldsymbol{\delta})$ is a Gaussian, whose dimension is the number of component-specific coefficients. For the covariate sets, $\boldsymbol{\Pi}$, we follow [25] and assume a uniform distribution, truncated to $|\boldsymbol{\Pi}| \leq 3$. The prior $p(\boldsymbol{\delta}|\boldsymbol{\Pi})$ will be specified in Section 5.1.4.

To generate samples from the posterior, we use a Markov Chain Monte Carlo (MCMC) algorithm, which combines the Gibbs-sampling steps for $\boldsymbol{\beta}_B$, σ^2, $\lambda_\star^2$, $\lambda_\diamond^2$ and $\boldsymbol{\mu}$ with two blocked Metropolis Hastings (MH) moves. In the first MH move the vector $\boldsymbol{\delta}$ is sampled jointly with $\boldsymbol{\mu}$, and in the second MH move $\boldsymbol{\Pi}$ is sampled jointly with $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$. As the implementation of the MCMC algorithm is involved, we relegate the mathematical details to **parts B and C of the Appendix**.

5.1.3 Competing models

A homogeneous model merges all data, while a non-homogeneous model assumes each component k to have specific parameters; see (5.2). The new partially non-homogeneous model infers the best trade-off: Each regression coefficient can be either constant or component-specific.

For a fair comparison, we also allow the non-homogeneous model to switch between a homogeneous and a non-homogeneous state. However, like all models that have been proposed so far, it operates on the covariate sets. All covariates

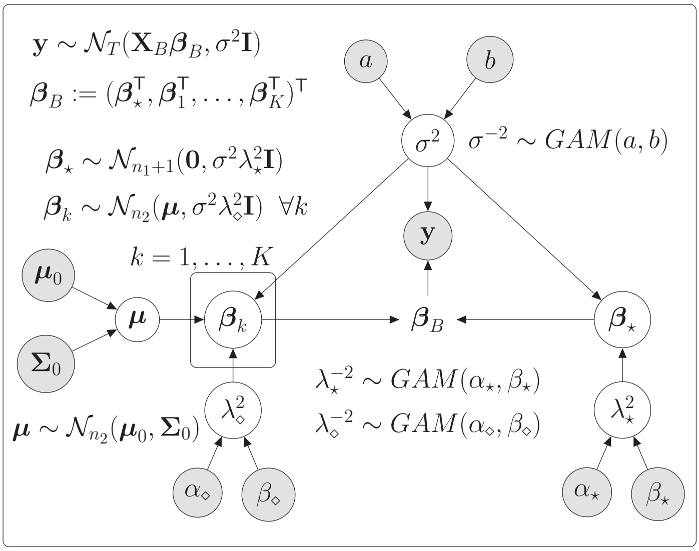

$\mathbf{y} \sim \mathcal{N}_T(\mathbf{X}_B\boldsymbol{\beta}_B, \sigma^2\mathbf{I})$
$\boldsymbol{\beta}_B := (\boldsymbol{\beta}_\star^\mathsf{T}, \boldsymbol{\beta}_1^\mathsf{T}, \ldots, \boldsymbol{\beta}_K^\mathsf{T})^\mathsf{T}$
$\boldsymbol{\beta}_\star \sim \mathcal{N}_{n_1+1}(\mathbf{0}, \sigma^2\lambda_\star^2\mathbf{I})$
$\boldsymbol{\beta}_k \sim \mathcal{N}_{n_2}(\boldsymbol{\mu}, \sigma^2\lambda_\diamond^2\mathbf{I})\ \forall k$
$k = 1, \ldots, K$
a
b
σ^2
$\sigma^{-2} \sim GAM(a, b)$
$\mathbf{y}$
$\boldsymbol{\mu}_0$
$\boldsymbol{\Sigma}_0$
$\boldsymbol{\mu}$
$\boldsymbol{\beta}_k$
$\boldsymbol{\beta}_B$
$\boldsymbol{\beta}_\star$
$\lambda_\star^{-2} \sim GAM(\alpha_\star, \beta_\star)$
$\lambda_\diamond^{-2} \sim GAM(\alpha_\diamond, \beta_\diamond)$
$\lambda_\diamond^2$
$\lambda_\star^2$
$\boldsymbol{\mu} \sim \mathcal{N}_{n_2}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$
$\alpha_\diamond$
$\beta_\diamond$
$\alpha_\star$
$\beta_\star$

Figure 5.1: Graphical model representation of the regression model with partitioned design matrix. Variables that have to be inferred are in white circles. The data and the fixed hyperparameters are in grey circles. The vector $\boldsymbol{\beta}_B$ deterministically depends on $\boldsymbol{\beta}_\star$ and $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$. The vector $\boldsymbol{\beta}_k$ in the plate is condition-specific.

have either component-specific ($S = 0$) or constant ($S = 1$) regression coefficients. In our method comparison, we include:

- **DBN**: A homogeneous model that merges all data, see (5.3).
- **NH-DBN**: The NH-DBN model switches between two states. We have a DBN for $S = 1$, and the likelihood takes the form of (5.2) for $S = 0$.
- **coupled NH-DBN**: This model from [25] is an NH-DBN that globally couples the regression coefficients.

5.1.4 Specifying the covariate type prior

The NH-DBNs can switch between: 'all covariates are constant' ($S = 1$) vs. 'all covariates are component-specific' ($S = 0$). Those states refer to $\boldsymbol{\delta} = \mathbf{1}$ and $\boldsymbol{\delta} = \mathbf{0}$ of the partially NH-DBN. To match the priors, we set:

$$\frac{p(S = 1)}{p(S = 0)} = \frac{p(\boldsymbol{\delta} = \mathbf{1}|\boldsymbol{\Pi})}{p(\boldsymbol{\delta} = \mathbf{0}|\boldsymbol{\Pi})} \tag{5.9}$$

For $\boldsymbol{\Pi} = \{X_1, \ldots, X_n\}$, $\boldsymbol{\delta}$ contains $n + 1$ binary elements, which we assume to be independently Bernoulli distributed. To fulfill (5.9) the Bernoulli parameter must depend on $n = |\boldsymbol{\Pi}|$. We get: $p(\boldsymbol{\delta} = \mathbf{1}|\boldsymbol{\Pi}) = \theta_n^{n+1}$ and $p(\boldsymbol{\delta} = \mathbf{0}|\boldsymbol{\Pi}) = (1 - \theta_n)^{n+1}$. From (5.9) we obtain:

$$r := \frac{p(S=1)}{p(S=0)} = \frac{\theta_n^{n+1}}{(1-\theta_n)^{n+1}} \Leftrightarrow \theta_n = \left(\frac{r}{1+r}\right)^{1/(n+1)}$$

$$\text{and } \; p(\boldsymbol{\delta}|\boldsymbol{\Pi}) = \theta_n^{\sum\limits_{i=0}^{n} \delta_i} \cdot (1-\theta_n)^{\sum\limits_{i=0}^{n} (1-\delta_i)}$$

For mixture models it is often assumed that the number of components $\tilde{K}$ has a Poisson distribution [21]. We truncate it to $\tilde{K} \in \{1, K\}$:

$$p(S=0) = \tfrac{q(K)}{q(1)+q(K)} \text{ and } p(S=1) = \tfrac{q(1)}{q(1)+q(K)}$$

where $q(.)$ is the density of the Poisson distribution with parameter $\theta = 1$.

5.1.5 Gaussian process smoothing for non-equidistant data

The regression models assume that the time lag $\mathcal{O}$ between the response value $y_{k,t+1}$ and the covariate values $x_{1,k,t}, \dots, x_{n,k,t}$ is the same for all t. If the data within a component k were measured at time points $t_1, \dots, t_{T_k}$, with varying distances $\mathcal{O}_i := t_i - t_{i-1}$, the models lead to biased results. For this scenario, we propose to replace the observed non-equidistant response values by predicted equidistant response values. We propose the following Gaussian process (GP) based method:

- Determine the lowest time lag $\mathcal{O}^\star = \min\{\mathcal{O}_2, \dots, \mathcal{O}_{T_k}\}$, where $\mathcal{O}_i := t_i - t_{i-1}$.
- Given the observed data points $\{(t, y_{k,t}) : t = t_1, \dots, t_{T_k}\}$, use a Gaussian process to predict the whole curve $\{(t, y_{k,t})\}_{t \geq 0}$.
- Extract the response values at the time points: $t_1 + \mathcal{O}^\star, \dots, t_{T_k} + \mathcal{O}^\star$.
- Build the response vector and design matrix such that the values $x_{1,k,t_i}, \dots, x_{n,k,t_i}$ are used to explain the predicted response value $\hat{y}_{k,t_i+\mathcal{O}^\star}$ ($i = 1, \dots, T_k$). The new lag is then constant; $\mathcal{O}_t = \mathcal{O}^\star$.

A Gaussian process (GP) is a stochastic process $\{Y_{k,t}\}_{t \geq 0}$, here indexed by time, such that every finite subset of the random variables has a Gaussian distribution. A GP can be used to estimate a non-linear curve $(t, y_{k,t})_{t \geq 0}$ from noisy observations. We here assume the relationship:

$$y_{k,t} = f(t) + \epsilon_t$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ is observational noise, and the non-linear function $f(.)$ is unknown. We estimate $f(.)$ by fitting a GP to the observed data. The GP defines a distribution over the functions $f(.)$, which transforms the input $(t_1, \ldots, t_{T_k})$ into output $(y_{k,t_1}, \ldots, y_{k,t_{T_K}})$, such that

$$(Y_{k,t_1}, \ldots, Y_{k,t_{T_K}})^\mathsf{T} \sim \mathcal{N}_{T_k}(\mathbf{0}, \mathbf{K} + \sigma^2 \mathbf{I}) \tag{5.10}$$

where $\mathbf{I}$ is the identity matrix, and the elements of the T_k-by-T_k covariance matrix $\mathbf{K}$ are defined through a kernel function: $\mathbf{K}_{i,j} = \xi^2 \cdot k(t_i, t_j)$ with signal variance parameter ξ^2. The kernel function $k(.,.)$ is typically chosen such that similar inputs t_i and t_j yield correlated variables Y_{t_i} and Y_{t_j}. A popular and widly used kernel is the squared exponential kernel with: $k(t_i, t_j) = \exp(-\frac{1}{2} \cdot \frac{(t_i - t_j)^2}{l^2})$ where l is the length scale. The predictive expectation $\hat{y}_{k,t}$ for $t \geq 0$ is:

$$\hat{y}_{k,t} = \mathbf{K}_t \cdot (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \cdot \mathbf{y} \tag{5.11}$$

where $\mathbf{K}_t := \xi^2 (k(t, t_1), \ldots, k(t, t_{T_k}))^\mathsf{T}$ and $\mathbf{y} := (y_{t_1}, \ldots, y_{T_K})^\mathsf{T}$.

Before inferring the GP, we standardize $\mathbf{y}$ to mean 0. We impose log-uniform priors on σ^2, ξ^2 and l. We compute the maximum a posteriori (MAP) parameter estimates and plug them into (5.11). This way, we can get predictions for the response values $\hat{y}_{k,t}$ for $t \in \{t_1 + \mathcal{O}^\star, \ldots, t_{T_k} + \mathcal{O}^\star\}$.

5.1.6 Learning topologies of regulatory networks

Assume that the variables $Z_1, \ldots, Z_N$ interact with each other in form of a network and that data were collected under K conditions and that the conditions influence some of the interactions. Let $\mathbf{D}_k$ denote the N-by-T_k data matrix which was measured under condition k. The rows of $\mathbf{D}_k$ correspond to the variables and the columns of $\mathbf{D}_k$ correspond to T_k time points. $\mathbf{D}_{i,k,t}$ denotes the value of Z_i at time point t under condition k.

The goal is to infer the network structure. Interactions for temporal data are usually modelled with a time lag, e.g. of order $\mathcal{O} = 1$. An edge, $Z_j \to Z_i$, indicates that Z_j has an effect on Z_i in the following sense: For all k the value $\mathbf{D}_{i,k,t+1}$ (Z_i at $t + 1$) depends on $\mathbf{D}_{j,k,t}$ (Z_j at t).

There is no acyclicity constraint, and DBN inference can be thought of as inferring N separate regression models and combining the results. In the i-th model $Y := Z_i$ is the response. The remaining $N_\star := N - 1$ variables $Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_N$ are the potential covariates. For each $Y := Z_i$ we infer a covariate set $\mathbf{\Pi}_i$, and the covariate sets $\mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_N$ describe a network $\mathcal{N}$. There is the edge $Z_j \to Z_i$ in the network $\mathcal{N}$ if and only if $Z_j \in \mathbf{\Pi}_i$.

We can thus apply the partially non-homogeneous model to each $Y = Z_i$ separately, to generate posterior samples. We extract the covariate sets, $\mathbf{\Pi}_i^{(1)}, \ldots, \mathbf{\Pi}_i^{(R)}$ ($i = 1, \ldots, N$), and we merge them to a network sample $\mathcal{N}^{(1)}, \ldots, \mathcal{N}^{(R)}$. The r-th network $\mathcal{N}^{(r)}$ possesses the edge $Z_j \to Z_i$ if and only if $Z_j \in \mathbf{\Pi}_i^{(r)}$. For each edge $Z_j \to Z_i$ we can then estimate its marginal posterior

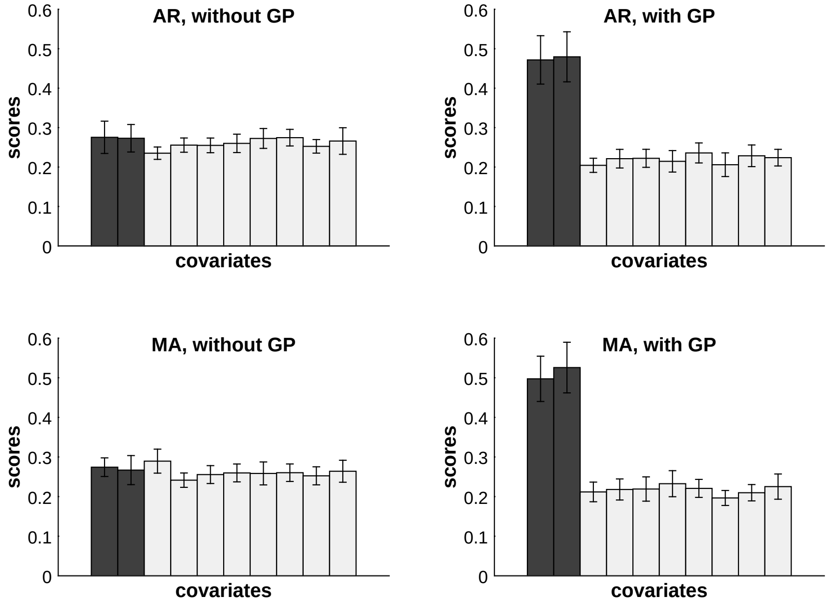


Figure 5.2: Average scores (posterior probabilities). In each histogram, the dark grey bars refer to the scores of the true covariates, and the light grey bars refer to the irrelevant variables. Covariate values were generated via autoregressive [AR] (top) and moving average [MA] processes (bottom). The left histograms show the scores of a standard regression (without GP processing). The left histograms show the scores when the proposed GP method is used. Error bars indicate standard deviations.

probability ('score'):

$$\hat{s}_{j,i} = \frac{1}{R}\sum_{r=1}^{R} I_{j\to i}(\mathcal{N}^{(r)}) \text{ where } I_{j\to i}(\mathcal{N}^{(r)}) = \begin{cases} 1 & \text{if } Z_j \in \mathbf{\Pi}_i^{(r)} \\ 0 & \text{if } Z_j \notin \mathbf{\Pi}_i^{(r)} \end{cases}$$

When the true network is known, we can evaluate the network reconstruction accuracy with precision-recall curves. For each $\psi \in [0,1]$ we extract the $n(\psi)$ edges whose scores $\hat{s}_{j,i}$ exceed ψ, and we count the number of true positives $T(\psi)$ among them. Plotting the *precisions* $P(\psi) := T(\psi)/n(\psi)$ against the *recalls* $R(\psi) := T(\psi)/M$, where M is the number of edges in the true network, gives the precision-recall curve ([11]). We refer to the area under the curve as AUC value. The higher the AUC, the higher the reconstruction accuracy.

5.2 Implementation

For the inverse Gamma distributed parameters (σ^2, $\lambda_\star^2$, $\lambda_\diamond^2$) we use shape and rate parameters from earlier works, e.g. in [38] and [25]: $\sigma^{-2} \sim GAM(0.005, 0.005)$ and $\lambda_\star^{-2}, \lambda_\diamond^{-2} \sim GAM(2, 0.2)$ and for the hyperprior on $\boldsymbol{\mu}$ we use $\boldsymbol{\mu}_0 = \mathbf{0}$ and

$\boldsymbol{\Sigma}_0 = \mathbf{I}$. Other settings led to comparable results what indicates robustness w.r.t. those hyperparameters. To ensure a fair comparison we use the same hyperparameters for the competing models; cf. Section 5.1.3.

For generating posterior samples, we run the MCMC algorithm from Section 5.1.2 for 100,000 (100k) iterations. We set the burn-in phase to 50k and we sample every 100th graph during the sampling phase. This yields $R = 500$ posterior samples for each response $Y = Z_i$. We merge the individual covariate sets $\boldsymbol{\Pi}_i^{(r)}$ $(i = 1, \ldots, N;\ r = 1, \ldots, R)$ to a network sample $\mathcal{N}^{(1)}, \ldots, \mathcal{N}^{(R)}$, as explained in Section 5.1.6. For each edge $Z_j \rightarrow Z_i$ we then compute its edge score $\hat{s}_{j,i}$.

We used potential scale reduction factors (PSRFs) to monitor convergence ([7]). We monitored the fractions of edges which fulfilled $PSRF < 1.01$ against the MCMC iterations. For all data sets all edge-specific PSRF's were below 1.01 after 100k iterations.

The computational costs for $100k$ MCMC iterations are moderate when a computer cluster is available. The computational advantage is that the task to infer a network with N nodes can be subdivided into N independent regression tasks (cf. Section 5.1.6), and the simulations can run in parallel. With our Matlab implementation 100k iterations take 5-10 minutes. We implement the GP method with the squared exponential kernel and used the Matlab package 'GPstuff' [64] to numerically determine the MAP estimates of the parameters via scaled conjugate gradient optimization. We also tested other kernels, such as the Matern 3/2 and 5/2 kernel, and for them we obtained very similar results.

5.3 Data and empirical results

5.3.1 Pre-study on Gaussian Process smoothing

Our first objective is to provide empirical evidence that the proposed GP method from Section 5.1.5 can yield substantial improvements. To this end, we generate values for 10 autoregressive (AR) variables:

$$X_{i,t} = \eta X_{i,t-1} + \epsilon_{i,t} \quad (t = 0, 1, \ldots, 120; i = 1, \ldots, 10) \tag{5.12}$$

where $\epsilon_{i,t} \sim N(0, 0.5^2)$, and X_1 and X_2 are covariates for:

$$Y_{t+1} = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \epsilon_{y,t+1} \tag{5.13}$$

where $\epsilon_{y,t+1} \sim N(0, 0.01^2)$.
In a second scenario we replace (5.12) by moving averages (MA):

$$X_{i,t} = \sum_{j=t-q}^{t} \epsilon_{i,j} \quad (t = 0, 1, \ldots, 120; i = 1, \ldots, 10) \tag{5.14}$$

where $\epsilon_{i,t} \sim N(0, (q+1)^{-1})$, so that $X_{i,t} \sim N(0, 1)$.

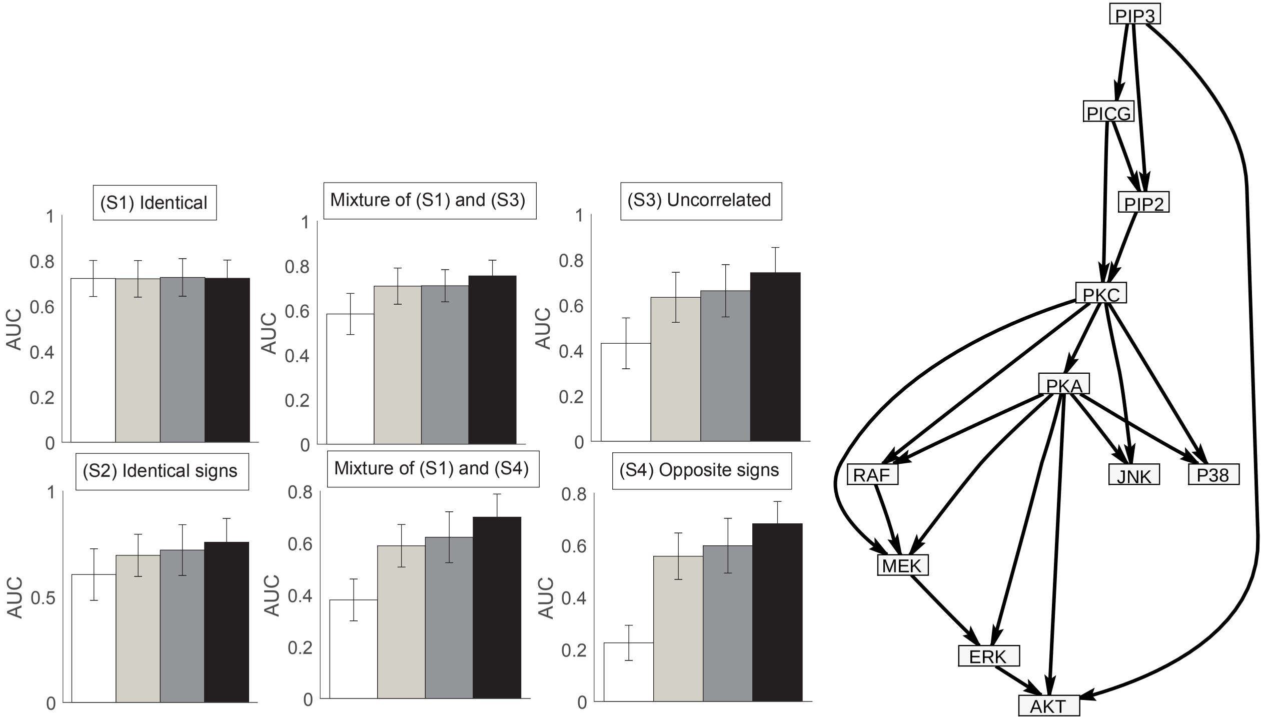


Figure 5.3: Network reconstruction accuracy for RAF pathway data. The histograms show the scenario-specific average precision-recall AUC values. Each AUC is averaged across 25 data sets and the error bars indicate standard deviations. The bars refer to: the homogeneous DBN (white), the NH-DBN model (light-grey), the coupled NH-DBN (dark-grey) and the partially NH-DBN (black). For (S2-5) the AUC differences are in favor of the new model (2-sided paired t-test p-values: $p < 0.05$). **Right**: The RAF pathway.

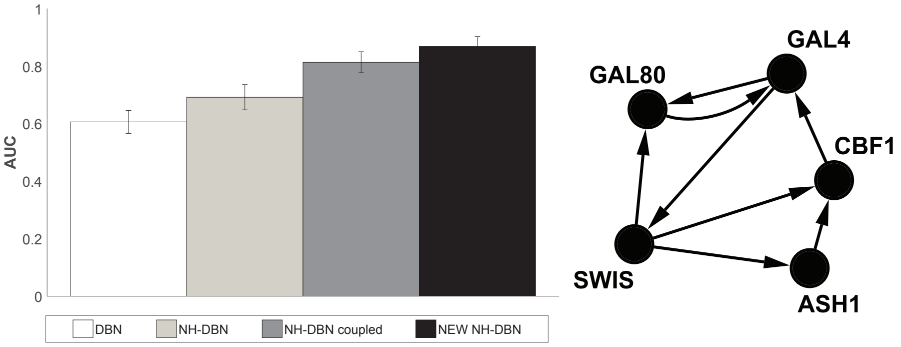


Figure 5.4: Network reconstruction accuracy for yeast gene expression data. The histogram shows the average precision-recall AUC values, averaged across 25 MCMC simulations, with error bars indicating standard deviations. The AUCs are: 0.61 (DBN), 0.69 (NH-DBN), 0.81 (coupled NH-DBN) and 0.87 (new NH-DBN). All three AUC differences are significant in terms of 2-sided t-tests ($p < 10^{-3}$). **Right**: The true yeast network [8].

We generate data for both scenarios (AR and MA) with different parameter settings $(\beta_0, \beta_1, \beta_2)$ in (5.13) and η in (5.12), respective q in (5.14). We thin the data out and keep only the observations at the time points $t \in \{0, 1, 3, 5, 10, 15, 30, 45, 60, 120\}$, as the same time points were measured for the mTORC1 data; see Section 5.3.4. The standard regression approach uses the covariate values at t_i for explaining Y at t_{i+1}, although the time lag steadily increases. The Gaussian Process (GP) method from Section 5.1.5 predicts the response values at $t_i + \mathcal{O}^\star$, and replaces $y_{t_{i+1}}$ (observed Y at t_{i+1}) by $\hat{y}_{t_i+\mathcal{O}^\star}$ (predicted Y at $t_i + \mathcal{O}^\star$), where $\mathcal{O}^\star = 1$.

With both approaches we run MCMC simulations on each data set, and from the MCMC samples we compute for each covariate X_i the score that X_i is a covariate for Y. Our results show that the proposed GP method finds the true covariates X_1 and X_2, while the standard approach cannot clearly distinguish them from the irrelevant variables $X_3, \ldots, X_{10}$. Figure 5.2 shows histograms of the average covariate scores for AR data with $\beta_i = 1$ and $\eta = 1$, and for MA data with $\beta_i = 1$ and $q = 5$.

5.3.2 Pre-study on synthetic RAF-pathway data

The RAF pathway, see [50], consists of $N = 11$ nodes and 20 directed edges; see Figure 5.3. We generate data with $K = 2$ components and $T_k = 10$ data points each. The parent nodes of each node Z_i build its covariate set $\mathbf{\Pi}_i$. We assume a linear model with component-specific regression coefficients:

$$z_{i,k,t+1} = \beta^i_{k,0} + \sum_{j:Z_j \in \mathbf{\Pi}_i} \beta^i_{k,j} \cdot z_{j,1,t} + e^i_{k,t} \quad (k = 1, 2)$$

where $z_{i,k,t}$ denotes the value of node Z_i at time point t in component k, and $\beta^i_{k,j}$ is the regression coefficient for $Z_j \to Z_i$ in component k. The noise values $e^i_{k,t}$ and the initial values $z_{i,k,1}$ are sampled from independent $N(0, 0.05^2)$ distributions. For Z_i there are $2(|\mathbf{\Pi}_i| + 1)$ component-specific regression coefficients. For each Z_i we collect them in two vectors $\boldsymbol{\beta}^i_k$ ($k = 1, 2$), and we sample the elements of $\boldsymbol{\beta}^i_k$ from $N(0, 1)$ Gaussian distributions. We then re-normalize the vectors to Euclidean norm one: $\boldsymbol{\beta}^i_k \leftarrow \boldsymbol{\beta}^i_k / |\boldsymbol{\beta}^i_k|$ ($k = 1, 2$). We distinguish six scenarios:

- **(S1) Identical:** We withdraw $\boldsymbol{\beta}^i_2$ and assume that the same regression coefficients apply to both components. We set: $\boldsymbol{\beta}^i_2 = \boldsymbol{\beta}^i_1$ for all i.
- **(S2) Identical signs (correlated):** We enforce the coefficients to have the same signs, i.e. we replace $\beta^i_{2,j}$ by: $\beta^i_{2,j} := \text{sign}(\beta^i_{1,j}) \cdot |\beta^i_{2,j}|$ for all i and j .
- **(S3) Uncorrelated:** We use the vectors $\boldsymbol{\beta}^i_k$ for component k ($k = 1, 2$). The component-specific coefficients $\beta^i_{1,j}$ and $\beta^i_{2,j}$ are then uncorrelated for all i and all j.
- **(S4) Opposite signs (negatively correlated):** We withdraw the vector $\boldsymbol{\beta}^i_2$ and we set: $\beta^i_{2,j} = (-1) \cdot \beta^i_{1,j}$. The coefficients $\beta^i_{1,j}$ and $\beta^i_{2,j}$ are then negatively correlated.
- **Mixture of (S1) and (S3):** We assume that 50% of the coefficients are identical for both k, while the other 50% are uncorrelated. We randomly select 50% of the coefficients and set: $\beta^i_{2,j} = \beta^i_{1,j}$. The other 50% of the coefficients stay unchanged (uncorrelated).
- **Mixture of (S1) and (S4):** We withdraw $\boldsymbol{\beta}^i_2$ and we assume that 50% of the coefficients are identical for both k, while the other 50% have an opposite sign. We randomly select 50% of the coefficients and set: $\beta^i_{2,j} = \beta^i_{1,j}$. For the other coefficients we set $\beta^i_{2,j} = (-1) \cdot \beta^i_{1,j}$.

For each scenario we generate 25 data sets. We then analyse every data set with each model. Figure 5.3 shows the average AUC values for reconstructing the RAF pathway. Only for scenario (S1), where all coefficients are constant, the models perform equally well. For (S2)-(S6) the homogeneous DBN is substantially worse than the NH-DBNs. The coupled NH-DBN is slightly superior to the (non-coupled) NH-DBN. The proposed partially NH-DBN yields the highest average AUC scores.

5.3.3 Reconstructing the yeast gene network topology

By means of synthetic biology, [8] designed a network with $N = 5$ genes in *S. cerevisiae* (yeast); Figure 5.4 shows the true network. With quantitative Real-Time Polymerase Chain Reaction, [8] then measured in vivo gene expression data: under galactose- ($k = 1$) and glucose-metabolism ($k = 2$). $T_1 = 16$ measurements were taken in galactose and $T_2 = 21$ in glucose. The data have become a

Protein	Full name	Sites
mTOR	mammalian target of rapamycin	pS2481, pS2448
PRAS40	proline-rich AKT/PKB substrate 40 kDa	pT246, pS183
AKT	Protein kinase B	pT308, pS473
IRS1	insulin receptor substrate 1	pS636
IR-beta	insulin receptor beta	pY1146
AMPK	AMP-dependent protein kinase	pT172
TSC2	tuberous sclerosis 2 protein	pS1387
p70-S6K	Ribosomal protein S6 kinase beta-1	pT389

Table 5.1: mTORC1 timecourse data. Overview to the eight proteins and the eleven measured phosporylation sites.

benchmark application, as the network reconstruction accuracies can be cross-compared on real in vivo gene expression data. Figure 5.4 shows the results, and again a clear trend can be seen: The homogeneous DBN yields the lowest AUC value. The non-homogeneous model (NH-DBN) yields higher AUCs and can be further improved by coupling the regression coefficients (coupled NH-DBN). The proposed partially NH-DBN reaches the highest network reconstruction accuracy. The results are thus consistent with the results for the RAF-pathway data in Section 5.3.2.

5.3.4 Reconstructing the topology of the mTOR complex 1 (mTORC1) network

The mammalian target of rapamycin complex 1 (mTORC1) is a serine/threonine kinase which is evolutionary conserved and essential in all eukaryotes [52]. mTORC1 is at the center of a multiply wired, complex signalling network, whose topology is well studied and contains several well-characterised feedback loops [52]. Hence, we used the mTORC1 network as a surrogate based on which we can objectively evaluate the predictive power of our partially NH-DBN model for learning network structures. The signalling network converging on mTORC1 is built by kinases, which inactivate or activate each other by phosphorylation. Thus, a protein can be phosphorylated at one or several sites, and the phosphorylations at these positions determine its activity. Signaling through the mTORC1 network is elicited by external signals like insulin or amino acids. [10] relatively quantified 11 phosphorylation states of 8 key proteins across the mTORC1 signalling network by immunoblotting; for an overview see Table 5.1. Dynamic time course data were obtained under two experimental conditions, namely upon stimulation with amino acids only ($k = 1$), and with amino acids plus insulin ($k = 2$). The phosphorylation states were measured at $T_k = 10$ time points: $t = 0, 1, 3, 5, 10, 15, 30, 45, 60, 120$ minutes, so that the time lag increases from 1 to 60. We therefore apply the Gaussian Process method from Section 5.1.5 to predict

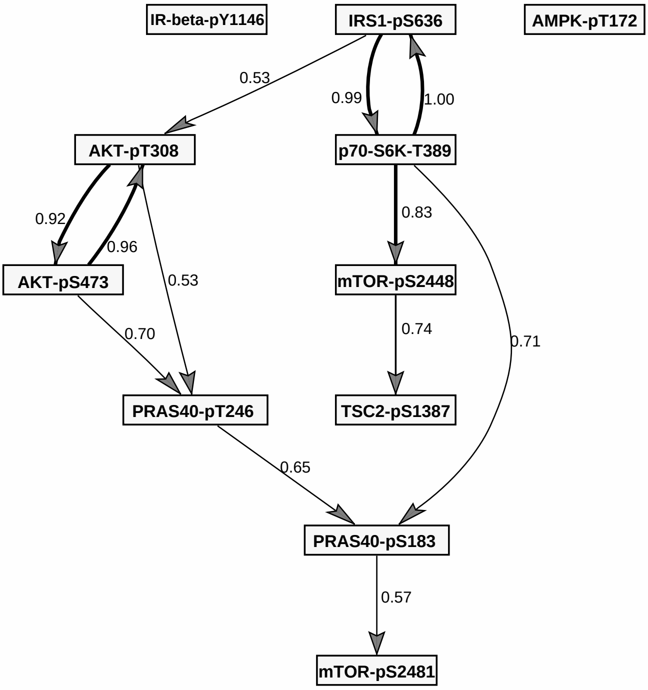


Figure 5.5: Predicted mTORC1 network topology. The 12 interactions whose scores exceeded the threshold $\psi = 0.5$; edges are labelled with their scores. The 5 edges with scores higher than $\psi = 0.8$ are represented in bold. The displayed interactions all had a higher posterior probability for being in the non-homogeneous state ($\delta = 1$).

equidistant response values, before analysing the data with the proposed partially NH-DBN. The 12 edges with scores higher than $\psi = 0.5$ yield the network topology shown in Figure 5.5. A literature review shows that 11 out of the 12 edges have been reported earlier.

We focus first on the five interactions with the highest scores $\psi > 0.8$. Two out of these five interactions are enzyme-substrate relationships: p70-S6K is a kinase which is directly activated by mTORC1 through phosphorylation at threonine 389 [p70-S6K-pT389] [52]. Thus, p70-S6K-pT389 represents a direct readout of mTORC1 activity. p70-S6K phosphorylates IRS1 at serine 636, [IRS1-pS636] [63] and mTOR at serine 2448 [mTOR-pS2448] [13], and both edges are correctly identified by our model [p70-S6K-pT389→IRS1-pS636, p70-S6K-pT389→mTOR-pS2448]. Two other interactions with a high score are between AKT-pT308↔AKT-pS473. The two phosphorylations are predicted by our model to influence each other, and a positive feedback between phosphorylation events on S473 and T308 of AKT has indeed been demonstrated biochemically [42]. Another high score prediction is between IRS1-pS636 and p70-S6K-pT389 [IRS1-pS636→p70-S6K-pT389]. Phosphorylation at S636 inhibits IRS1, thereby leading to inhibition of mTORC1 and its substrate p70-S6K-T389 [63]. Thus, the negative feedback between IRS1-pS636 and p70-S6K-pT389 explains the learned edge between them [IRS1-pS636→p70-S6K-pT389]. In addition, IRS1 inhibition by phosphorylation at S636 results in reduced phosphorylation of AKT at threonine 308, which is in agreement with the learned edge between IRS1-pS636 and AKT-pT308 [IRS1-pS636→AKT-pT308].

We could also find evidence for 6 of the remaining 7 edges with scores in between 0.5 and 0.8. PRAS40 is an endogenous mTORC1 inhibitor [52]. The edge from PRAS40-pT246 to PRAS40-pS183 corresponds to a well-described mechanism of PRAS40 regulation: AKT phosphorylates PRAS40 at T246 [PRAS40-pT246], which allows subsequent phosphorylation of PRAS40-S183 by mTORC1 [45]. This interaction is accurately resembled by our model [PRAS40-pT246→ PRAS40-pS183]. PRAS40's double phosphorylation dissociates PRAS40 from mTORC1, leading to its derepression [45]. This mechanism is resembled by the edge between PRAS40-S183 and mTOR-S2481 [PRAS40-pS183→mTOR-pS2481], the latter being an autophosphorylation site which directly monitors mTOR activity [61]. Furthermore, the model suggests an edge between p70-S6K-pT389 and PRAS40-pS183 [p70-S6K-pT389→PRAS40-pS183]. Both are mTORC1 substrate sites [45, 52] and are therefore often targeted in parallel. The only predicted edge for which there is to the best of our knowledge no literature evidence is between mTOR-pS2448 and TSC2-pS1387 [mTOR-pS2448→TSC2-pS1387]. TSC2 is activated by phosphorylation at S1387 and inhibits mTORC1 [30]. Our model prediction that mTORC1 - when phosphorylated at S2448 by p70-S6K - regulates TSC2 remains to be experimentally tested.

5.4 Discussion and conclusions

We propose a new partially non-homogeneous dynamic Bayesian network (partially NH-DBN) model for learning network structures. When data are measured under different experimental conditions, it is rarely clear whether the data can be merged and analysed within one single model, or whether there is need for a NH-DBN model that allows the network parameters to depend on the condition. The new partially NH-DBN has been designed such that it can infer the best trade-off from the data. It infers for each individual edge whether the corresponding interaction parameter is constant or condition-specific. Our applications to synthetic RAF pathway data as well as to yeast gene-expression data have shown that the partially NH-DBN model improves the network reconstruction accuracy. We have used the partially NH-DBN model to predict the structure of the mTORC1 signalling network. As the measured mTORC1 data are non-equidistant, we have applied a Gaussian process (GP) based method to predict the missing equidistant values. Results on synthetic data (see Section 5.3.1) show that the proposed GP-method (see Section 5.1.5) can lead to substantially improved results.

All but one of the predicted interactions across the mTORC1 network are reflected in experiments reported in the biological literature. [10] built an ODE-based dynamic model which allows to predict signalling responses to perturbations. Like for many ODE-based models, the topology of this model was defined by the authors, based on literature-knowledge. The ODE model simulations could reproduce the measured mTORC1 timecourse data. Interestingly, all the connections predicted by our new partially NH-DBN model form part of the core model by [10]. Hence, we present an alternative unsupervised learning approach, in which the topology of signalling networks is inferred directly from the data. The new model is thus a complementary tool that enhances dynamic model building by predicting the network's topology in a purely data-driven manner.

5.5 Appendix

5.5.1 Part 0 - Summary from this chapter

For the posterior of the proposed partly non-homogeneous dynamic Bayesian network (partly NH-DBN) model we have:

$$\begin{aligned} p(\boldsymbol{\beta}_B, \sigma^2, \lambda_\star^2, \lambda_\diamond^2, \boldsymbol{\mu}|\mathbf{y}) \quad &\propto \quad p(\mathbf{y}|\sigma^2, \boldsymbol{\beta}_B) \cdot p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \cdot p(\boldsymbol{\mu}) \cdot p(\sigma^{-2}) \\ &\cdot p(\lambda_\star^{-2}) \cdot p(\lambda_\diamond^{-2}) \end{aligned} \tag{5.15}$$

and the model likelihood is given by:

$$\mathbf{y} \sim \mathcal{N}(\mathbf{X}_B \boldsymbol{\beta}_B, \sigma^2 \mathbf{I})$$

where $\boldsymbol{\beta}_B$ is a vector of $(1 + n_1 + K \cdot n_2)$ regression coefficients, and $\mathbf{X}_B$ is a partitioned matrix with $\sum(T_k - 1)$ rows and $(1 + n_1) + (K \cdot n_2)$ columns. E.g. for $K = 2$, when the intercept and the first $n_1 < n$ coefficients stay constant while the remaining $n_2 = n - n_1$ coefficients are component-specific, the matrix $\mathbf{X}_B$ has the structure:

$$\mathbf{X}_B = \begin{pmatrix} \mathbf{1} & \mathbf{x}_{1,1} & \dots & \mathbf{x}_{n_1,1} & \mathbf{x}_{n_1+1,1} & \dots & \mathbf{x}_{n,1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{1} & \mathbf{x}_{1,2} & \dots & \mathbf{x}_{n_1,2} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{x}_{n_1+1,2} & \dots & \mathbf{x}_{n,2} \end{pmatrix},$$

In this chapter we have shown that $\boldsymbol{\beta}_B$ has a Gaussian prior:

$$\boldsymbol{\beta}_B|(\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \sigma^2 \tilde{\boldsymbol{\Sigma}}) \quad \text{with: } \tilde{\boldsymbol{\mu}} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\mu} \\ \vdots \\ \boldsymbol{\mu} \end{pmatrix} \quad \text{and } \tilde{\boldsymbol{\Sigma}} = \begin{pmatrix} \lambda_\star^2 \mathbf{I}_\star & \mathbf{0} \\ \mathbf{0} & \lambda_\diamond^2 \mathbf{I}_\diamond \end{pmatrix}$$

where $\mathbf{I}_\star$ is the $(n_1 + 1)$-dimensional identity matrix, and $\mathbf{I}_\diamond$ is the $(K \cdot n_2)$-dimensional identity matrix.

Moreover, we have imposed the following inverse Gamma priors:

$$\sigma^{-2} \sim \text{GAM}(a, b) \quad \text{and} \quad \lambda_\star^{-2} \sim \text{GAM}(\alpha_\star, \beta_\star) \quad \text{and} \quad \lambda_\diamond^{-2} \sim \text{GAM}(\alpha, \beta)$$

When deriving the full conditional distributions we will use the relationship:

$$\mathbf{X}_B \cdot \tilde{\boldsymbol{\mu}} = \mathbf{X}_B^\ddagger \cdot \boldsymbol{\mu} \quad \text{where} = \quad \mathbf{X}_B^\ddagger := \begin{pmatrix} \mathbf{x}_{n_1+1,1} & \dots & \mathbf{x}_{n,1} \\ \mathbf{x}_{n_1+1,2} & \dots & \mathbf{x}_{n,2} \\ \vdots & \dots & \vdots \\ \mathbf{x}_{n_1+1,K} & \dots & \mathbf{x}_{n,K} \end{pmatrix} \tag{5.16}$$

5.5.2 Part A - Deriving the full conditional distributions

A sample from the posterior distribution in Equation (5.15) can be generated by Markov Chain Monte Carlo (MCMC) simulations. In this subsection we derive the full conditional distributions (FCDs) for the model parameters: $\boldsymbol{\beta}_B$, $\lambda_\star^2$, and $\lambda_\diamond^2$. For σ^2 and $\boldsymbol{\mu}$ we implement collapsed Gibbs sampling moves. In collapsed Gibbs sampling steps some of the other variables are integrated out in analytically from the FCDs. Collapsed Gibbs sampling steps are known to be more efficient than standard Gibbs sampling steps. Within a Gibbs MCMC sampling scheme all parameters are iteratively resampled from their FCDs or by a collapsed Gibbs sampling step.

The densities of the FCDs are proportional to the factorized joint density in Equation (5.15). From the shape of the densities we conclude what the full conditional distributions (FCDs) are.

For the **full conditional distribution** of $\boldsymbol{\beta}_B$ we obtain:

$$\begin{aligned}
\text{FCD}(\boldsymbol{\beta}_B) &\propto p(\mathbf{y}|\sigma^2, \boldsymbol{\beta}_B) \cdot p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \\
&\propto \exp\{-\frac{1}{2}(\mathbf{y} - \mathbf{X}_B\boldsymbol{\beta}_B)^\mathsf{T}(\sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{X}_B\boldsymbol{\beta}_B)\} \\
&\quad \cdot \exp\{-\frac{1}{2}(\boldsymbol{\beta}_B - \tilde{\boldsymbol{\mu}})^\mathsf{T}(\sigma^2\tilde{\boldsymbol{\Sigma}})^{-1}(\boldsymbol{\beta}_B - \tilde{\boldsymbol{\mu}})\} \\
&\propto \exp\{-\frac{1}{2} \cdot \boldsymbol{\beta}_B^\mathsf{T}\left(\sigma^{-2}[\tilde{\boldsymbol{\Sigma}}^{-1} + \mathbf{X}_B^\mathsf{T}\mathbf{X}_B]\right)\boldsymbol{\beta}_B \; + \; \boldsymbol{\beta}_B^\mathsf{T}\left(\sigma^{-2}(\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}} + \mathbf{X}_B^\mathsf{T}\mathbf{y})\right)\}
\end{aligned}$$

and from the shape of the latter density we conclude:

$$\boldsymbol{\beta}_B|(\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \sim \mathcal{N}\left([\tilde{\boldsymbol{\Sigma}}^{-1} + \mathbf{X}_B^\mathsf{T}\mathbf{X}_B]^{-1}(\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}} + \mathbf{X}_B^\mathsf{T}\mathbf{y}) \;,\; \sigma^2[\tilde{\boldsymbol{\Sigma}}^{-1} + \mathbf{X}_B^\mathsf{T}\mathbf{X}_B]^{-1}\right) \tag{5.17}$$

For the **full conditional distributions** of $\lambda_\diamond^2$ and $\lambda_\star^2$ we get:

$$\begin{aligned}
\text{FCD}(\lambda_\diamond^2) &\propto p(\lambda_\diamond^{-2}) \cdot p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \\
&\propto p(\lambda_\diamond^{-2}) \cdot \prod_{k=1}^{K} p(\boldsymbol{\beta}_k|\sigma^2, \lambda_\diamond^2) \\
&\propto (\lambda_\diamond^{-2})^{\alpha + \frac{Kn_2}{2} - 1} \cdot \exp\{-\lambda_\diamond^{-2}(\beta + \frac{1}{2}\sigma^{-2}\sum_{k=1}^{K}(\boldsymbol{\beta}_k - \boldsymbol{\mu})^\mathsf{T}(\boldsymbol{\beta}_k - \boldsymbol{\mu}))\} \\
\text{FCD}(\lambda_\star^2) &\propto p(\lambda_\star^{-2}) \cdot p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \\
&\propto p(\lambda_\star^{-2}) \cdot p(\boldsymbol{\beta}_\star|\sigma^2, \lambda_\star^2) \\
&\propto (\lambda_\star^{-2})^{\alpha_\star + \frac{n_1}{2} - 1} \cdot \exp\{-\lambda_\star^{-2}(\beta_\star + \frac{1}{2}\sigma^{-2}\boldsymbol{\beta}_\star^\mathsf{T}\boldsymbol{\beta}_\star)\}
\end{aligned}$$

and from the shapes of the densities it follows for the FCDs:

$$\begin{aligned}
\lambda_\diamond^{-2}|(\sigma^2, \boldsymbol{\beta}_B, \lambda_\star^2, \boldsymbol{\mu}) &\sim GAM\left(\alpha + \frac{Kn_2}{2}, \beta + \frac{1}{2}\sigma^{-2}\sum_{k=1}^{K}(\boldsymbol{\beta}_k - \boldsymbol{\mu})^\mathsf{T}(\boldsymbol{\beta}_k - \boldsymbol{\mu})\right) \\
\lambda_\star^{-2}|(\sigma^2, \boldsymbol{\beta}_B, \lambda_\diamond^2, \boldsymbol{\mu}) &\sim GAM\left(\alpha_\star + \frac{n_1}{2}, \beta_\star + \frac{1}{2}\sigma^{-2}\boldsymbol{\beta}_\star^\mathsf{T}\boldsymbol{\beta}_\star\right)
\end{aligned} \tag{5.18}$$

For the noise variance parameter σ^2 we implement a **collapsed Gibbs sampling step** with $\boldsymbol{\beta}_B$ integrated out. We have:

$$p(\mathbf{y}|\sigma^2, \lambda_\diamond^2, \lambda_\star^2) = \int p(\mathbf{y}, \boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2)d\boldsymbol{\beta}_B = \int p(\mathbf{y}|\boldsymbol{\beta}_B, \sigma^2) \cdot p(\boldsymbol{\beta}_B|\sigma^2, \lambda_\diamond^2, \lambda_\star^2)d\boldsymbol{\beta}_B$$

From a standard rule for Gaussian integrals (see, e.g., Section 2.3.2 in [5]):

$$\mathbf{y}|\boldsymbol{\beta} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma}) \;\text{ with }\; \boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}) \;\text{ implies }\; \mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\mu}, \boldsymbol{\Sigma} + \mathbf{X}\mathbf{S}\mathbf{X}^\mathsf{T})$$

It follows:

$$\mathbf{y}|(\sigma^2, \lambda_\diamond^2, \lambda_\star^2) \sim \mathcal{N}(\mathbf{X}_B\tilde{\boldsymbol{\mu}}, \sigma^2[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]) \tag{5.19}$$

This yields:

$$\begin{aligned} p(\sigma^2|\mathbf{y}, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) &\propto p(\mathbf{y}|\sigma^2, \lambda_\diamond^2, \lambda_\star^2) \cdot p(\sigma^{-2}) \\ &\propto (\sigma^{-2})^{0.5\sum(T_k-1)} \\ &\quad \cdot \exp\{-0.5(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})^{\mathsf{T}}\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})\} \\ &\quad \cdot(\sigma^{-2})^{a-1}\exp\{-b\sigma^{-2}\} \end{aligned}$$

The shape of the density implies the collapsed Gibbs sampling step:

$$\sigma^{-2}|(\mathbf{y}, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \sim GAM$$
$$\left(a + \frac{\sum(T_k-1)}{2}, b + \frac{1}{2}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})^{\mathsf{T}}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})\right)$$

For the FCD of $\boldsymbol{\mu}$ we also use a **collapsed Gibbs sampling step** with $\boldsymbol{\beta}_B$ integrated out (cf. Equation (5.19)) and we use that $\mathbf{X}_B \cdot \tilde{\boldsymbol{\mu}} = \mathbf{X}_B^{\ddagger} \cdot \boldsymbol{\mu}$ (cf. Equation (5.16))

$$\begin{aligned} \text{FCD}(\boldsymbol{\mu}) &\propto p(\mathbf{y}|\sigma^2, \lambda_\diamond^2, \lambda_\star^2) \cdot p(\boldsymbol{\mu}) \\ &\propto \exp\{-0.5\sigma^{-2}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})^{\mathsf{T}}\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})\} \\ &\quad \cdot \exp\{-0.5\boldsymbol{\mu}^{\mathsf{T}}\boldsymbol{\mu}\} \\ &\propto \exp\{-0.5\boldsymbol{\mu}^{\mathsf{T}}\left((\mathbf{X}_B^{\ddagger})^{\mathsf{T}}(\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1})\mathbf{X}_B^{\ddagger} + \mathbf{I}\right)\boldsymbol{\mu} \ldots \\ &\quad \ldots + \boldsymbol{\mu}^{\mathsf{T}}(\mathbf{X}_B^{\ddagger})^{\mathsf{T}}(\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1})\mathbf{y}\} \end{aligned}$$

The latter density is proportional to the density of a Gaussian, so that it follows for the FCD:

$$\boldsymbol{\mu}|(\lambda_\diamond^2, \lambda_\star^2, \sigma^2) \sim \mathcal{N}(\boldsymbol{\mu}^{\ddagger}, \boldsymbol{\Sigma}^{\ddagger}) \tag{5.20}$$

where

$$\boldsymbol{\Sigma}^{\ddagger} = (\mathbf{X}_B^{\ddagger})^{\mathsf{T}}(\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1}\mathbf{X}_B^{\ddagger} + \mathbf{I})^{-1} \tag{5.21}$$

$$\boldsymbol{\mu}^{\ddagger} = \boldsymbol{\Sigma}^{\ddagger} \cdot (\mathbf{X}_B^{\ddagger})^{\mathsf{T}}(\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}]^{-1})\mathbf{y} \tag{5.22}$$

An important model property is that the marginal likelihood, with $\boldsymbol{\beta}_B$ and σ^2 integrated out, can be computed. The marginalization rule from Section 2.3.7 of [5] yields:

$$p(\mathbf{y}|\lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) = \frac{\Gamma(\frac{T}{2} + a)}{\Gamma(a)} \cdot \frac{\pi^{-\frac{T}{2}}(2b)^a}{\det(\mathbf{C})^{1/2}}\left(2b + (\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})^{\mathsf{T}}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})\right)^{-(\frac{T}{2}+a)} \tag{5.23}$$

where $T := \sum\limits_{k=1}^{K}(T_k - 1)$, and $\mathbf{C} := \mathbf{I} + \mathbf{X}_B\tilde{\boldsymbol{\Sigma}}\mathbf{X}_B^{\mathsf{T}}$.

5.5.3 Part B - Blocked Metropolis Hastings moves for inferring the covariate set and the covariate types

We note that the indicator vector $\boldsymbol{\delta}$ depends on the covariate set $\boldsymbol{\Pi}$, as it contains one indicator variable for each covariate in $\boldsymbol{\Pi}$. Moreover, the expression $\mathbf{X}_B$, $\boldsymbol{\mu}$, $\tilde{\boldsymbol{\Sigma}}$, $\tilde{\boldsymbol{\mu}}$ and $\mathbf{C}$ all depend on both $\boldsymbol{\Pi}$ and $\boldsymbol{\delta}$, though we do not make that explicit in our notation.

As described in this chapter, given the covariate set $\boldsymbol{\Pi}$ and the corresponding indicator vector $\boldsymbol{\delta}$, the partitioned design matrix $\mathbf{X}_B = \mathbf{X}_{B,\boldsymbol{\Pi},\boldsymbol{\delta}}$ can be built, and the marginal likelihood $p(\mathbf{y}|\lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}, \boldsymbol{\Pi}, \boldsymbol{\delta})$ can be computed with Equation (5.23). We obtain as posterior distribution for the extended model (and with σ^2 and $\boldsymbol{\beta}_B$ integrated out):

$$\begin{aligned} p(\boldsymbol{\Pi}, \boldsymbol{\delta}, \lambda_\star^2, \lambda_\diamond^2, \boldsymbol{\mu}|\mathbf{y}) &\propto p(\mathbf{y}|\lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}, \boldsymbol{\Pi}, \boldsymbol{\delta}) \cdot p(\boldsymbol{\Pi}) \cdot p(\boldsymbol{\delta}|\boldsymbol{\Pi}) \cdot p(\boldsymbol{\mu}|\boldsymbol{\Pi}, \boldsymbol{\delta}) \cdot p(\lambda_\star^2) \\ &\quad \cdot p(\lambda_\diamond^2) \end{aligned} \tag{5.24}$$

where $p(\boldsymbol{\mu}|\mathbf{\Pi},\boldsymbol{\delta})$ is a Gaussian, whose dimension is the number of component-specific coefficients, p($\mathbf{\Pi}$), is a uniform distribution, truncated to the maximal cardinality $|\mathbf{\Pi}| \leq 3$, and $p(\boldsymbol{\delta}|\mathbf{\Pi})$ has been specified in Section 5.1.4.

We implement two Metropolis Hastings moves, and we use the concept of blocking ([40]). Blocking is a technique by which variables are not sampled separately, but are merged into blocks that are sampled together. We form two blocks, grouping $\boldsymbol{\delta}$ with $\boldsymbol{\mu}$, and grouping $\mathbf{\Pi}$ with $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$. The vector $\boldsymbol{\delta}$ is then always sampled jointly with $\boldsymbol{\mu}$, and $\mathbf{\Pi}$ is always sampled jointly with $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$.

First Metropolis Hastings move:
Each move on $[\boldsymbol{\delta},\boldsymbol{\mu}]$ randomly selects one δ_i of $\boldsymbol{\delta}$ and proposes to switch its value, i.e to replace δ_i by $1-\delta_i$. This yields a new candidate vector $\boldsymbol{\delta}_\bullet$, for which we re-sample $\boldsymbol{\mu}_\bullet$ from its full conditional distribution in Equation (5.20). The acceptance probability for the move is:

$$A([\boldsymbol{\delta},\boldsymbol{\mu}] \to [\boldsymbol{\delta}_\bullet,\boldsymbol{\mu}_\bullet]) = \min\left\{1, \frac{p(\mathbf{y}|\lambda_\diamond^2,\lambda_\star^2,\boldsymbol{\mu}_\bullet,\mathbf{\Pi},\boldsymbol{\delta}_\bullet)}{p(\mathbf{y}|\lambda_\diamond^2,\lambda_\star^2,\boldsymbol{\mu},\mathbf{\Pi},\boldsymbol{\delta})} \cdot \frac{p(\boldsymbol{\delta}_\bullet|\mathbf{\Pi})}{p(\boldsymbol{\delta}|\mathbf{\Pi})} \cdot \frac{p(\boldsymbol{\mu}_\bullet|\mathbf{\Pi},\boldsymbol{\delta}_\bullet)}{p(\boldsymbol{\mu}|\mathbf{\Pi},\boldsymbol{\delta})} \cdot H\right\} \tag{5.25}$$

where the Hastings ratio H is equal to the ratio of full conditional densities:

$$H = \frac{p(\boldsymbol{\mu}|\lambda_\diamond^2,\lambda_\star^2,\sigma^2,\mathbf{\Pi},\boldsymbol{\delta})}{p(\boldsymbol{\mu}_\bullet|\lambda_\diamond^2,\lambda_\star^2,\sigma^2,\mathbf{\Pi},\boldsymbol{\delta}_\bullet)}$$

Second Metropolis Hastings move:
For sampling $[\mathbf{\Pi},\boldsymbol{\delta},\boldsymbol{\mu}]$ we implement 3 moves on the covariate set $\mathbf{\Pi}$, which are accompanied by updates of $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$. Each move proposes to replace $[\mathbf{\Pi},\boldsymbol{\delta},\boldsymbol{\mu}]$ by a new triple $[\mathbf{\Pi}_\bullet,\boldsymbol{\delta}_\bullet,\boldsymbol{\mu}_\bullet]$

- In the deletion move (D) we randomly select one covariate $X \in \mathbf{\Pi}$, and we propose to remove this covariate from $\mathbf{\Pi}$. This yields $\mathbf{\Pi}_\bullet$. Removing X makes the corresponding element δ from $\boldsymbol{\delta}$ redundant, so that we remove it as well to obtain $\boldsymbol{\delta}_\bullet$.
- In the addition move (A) we randomly select one covariate $X \notin \mathbf{\Pi}$, and we propose to add this covariate to $\mathbf{\Pi}$. This yields $\mathbf{\Pi}_\bullet$. We flip a coin to determine the type ($\delta \in \{0,1\}$) of the new covariate. Adding the element δ to $\boldsymbol{\delta}$ yields $\boldsymbol{\delta}_\bullet$.
- In the exchange move (E) we randomly select one covariate $X_\bullet \in \mathbf{\Pi}$, and we propose to replace $X_\bullet$ by a randomly selected new covariate $X \notin \mathbf{\Pi}$. This yields $\mathbf{\Pi}_\bullet$. We then flip a coin to determine the type ($\delta \in \{0,1\}$) of the new covariate. By removing the element $\delta_\bullet$ from $\boldsymbol{\delta}$ and adding the element δ to $\boldsymbol{\delta}$, we obtain $\boldsymbol{\delta}_\bullet$.

Each sub-move (D, A and E) yields a pair $[\mathbf{\Pi}_\bullet,\boldsymbol{\delta}_\bullet]$, which we complete to a triple by sampling a new $\boldsymbol{\mu}_\bullet$, conditional on $\mathbf{\Pi}_\bullet$ and $\boldsymbol{\delta}_\bullet$, from the full conditional distribution in Equation (5.20). When randomly selecting the move type (D, A or E) the acceptance probability is:

$$\begin{aligned} &A([\mathbf{\Pi},\boldsymbol{\delta},\boldsymbol{\mu}],[\mathbf{\Pi}_\bullet,\boldsymbol{\delta}_\bullet,\boldsymbol{\mu}_\bullet]) = \\ &\quad \min\left\{1, \frac{p(\mathbf{y}|\lambda_\diamond^2,\lambda_\star^2,\boldsymbol{\mu}_\bullet,\mathbf{\Pi}_\bullet,\boldsymbol{\delta}_\bullet)}{p(\mathbf{y}|\lambda_\diamond^2,\lambda_\star^2,\boldsymbol{\mu},\mathbf{\Pi},\boldsymbol{\delta})} \cdot \frac{p(\mathbf{\Pi}_\bullet)}{p(\mathbf{\Pi})} \cdot \frac{p(\boldsymbol{\delta}_\bullet|\mathbf{\Pi}_\bullet)}{p(\boldsymbol{\delta}|\mathbf{\Pi})} \cdot \frac{p(\boldsymbol{\mu}_\bullet|\mathbf{\Pi}_\bullet,\boldsymbol{\delta}_\bullet)}{p(\boldsymbol{\mu}|\mathbf{\Pi},\boldsymbol{\delta})} \cdot H\right\} \end{aligned} \tag{5.26}$$

where the Hastings-Ratio H is equal to:

$$H = \frac{p(\boldsymbol{\mu}|\lambda_\diamond^2,\lambda_\star^2,\sigma^2,\mathbf{\Pi},\boldsymbol{\delta})}{p(\boldsymbol{\mu}_\bullet|\lambda_\diamond^2,\lambda_\star^2,\sigma^2,\mathbf{\Pi}_\bullet,\boldsymbol{\delta}_\bullet)} \cdot \mathrm{HR}$$

and the factor HR is move-specific (D, A and E):

$$\mathrm{HR}_D = \frac{|\mathbf{\Pi}|}{N_\star - |\mathbf{\Pi}_\bullet|} \cdot 0.5, \qquad \mathrm{HR}_A = \frac{N_\star - |\mathbf{\Pi}|}{|\mathbf{\Pi}_\bullet|} \cdot 2, \qquad \mathrm{HR}_E = 1 \tag{5.27}$$

where $N_\star$ is the number of potential covariates, $|.|$ denotes the cardinality, and the factors 2 and 0.5 stem from flipping a coin to determine the type ($\delta \in \{0,1\}$) of a new covariate X.

5.5.4 Part C - The Markov Chain Monte Carlo (MCMC) inference algorithm

To generate samples from the posterior distribution in Equation (5.24), we use a Markov Chain Monte Carlo (MCMC) algorithm, which combines the Gibbs-sampling steps from part A with the Metropolis Hastings steps from part B. We initialize all entities, e.g. $\mathbf{\Pi} = \{\}$, $\boldsymbol{\delta} = (\delta_0) = (0)$, $\boldsymbol{\mu} = \mathbf{0}$, $\lambda_\diamond^2 = 1$, $\lambda_\star^2 = 1$, before we iterate among seven sampling steps:

Gibbs part: Given $\mathbf{\Pi}$ and $\boldsymbol{\delta}$, we re-sample the parameters σ^2, $\boldsymbol{\beta}$, $\lambda_\diamond^2$, $\lambda_\star^2$, and $\boldsymbol{\mu}$. Although the parameters σ^2 and $\boldsymbol{\beta}_B$ can be marginalized out, and thus do not appear in the posterior in Equation (5.24), the FCDs of $\lambda_\diamond^2$ and $\lambda_\star^2$, and $\boldsymbol{\mu}$ depend on them. Therefore σ^2 and $\boldsymbol{\beta}_B$ have to be sampled too, but they can be withdrawn after sampling step (5). The Gibbs sampling steps have been derived in part A of this Appendix. Each step updates one parameter, and the subsequent steps are then always conditional on the newest parameter combination:

(1) $\sigma^{-2}|(\mathbf{y}, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \sim$
$$GAM\left(a + \frac{\sum(T_k - 1)}{2}, b + \tfrac{1}{2}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})^\mathsf{T}\sigma^{-2}[\mathbf{I} + \mathbf{X}_B\tilde{\mathbf{\Sigma}}\mathbf{X}_B^\mathsf{T}]^{-1}(\mathbf{y} - \mathbf{X}_B\tilde{\boldsymbol{\mu}})\right)$$

(2) $\boldsymbol{\beta}_B|(\sigma^2, \lambda_\diamond^2, \lambda_\star^2, \boldsymbol{\mu}) \sim \mathcal{N}\left([\tilde{\mathbf{\Sigma}}^{-1} + \mathbf{X}_B^\mathsf{T}\mathbf{X}_B]^{-1}(\tilde{\mathbf{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}} + \mathbf{X}_B^\mathsf{T}\mathbf{y})\ ,\ \sigma^2[\tilde{\mathbf{\Sigma}}^{-1} + \mathbf{X}_B^\mathsf{T}\mathbf{X}_B]^{-1}\right)$

(3) $\lambda_\diamond^{-2}|(\sigma^2, \boldsymbol{\beta}_B, \lambda_\star^2, \boldsymbol{\mu}) \sim GAM\left(\alpha + \frac{Kn_2}{2}, \beta + \frac{1}{2}\sigma^{-2}\sum_{k=1}^K(\boldsymbol{\beta}_k - \boldsymbol{\mu})^\mathsf{T}(\boldsymbol{\beta}_k - \boldsymbol{\mu})\right)$

(4) $\lambda_\star^{-2}|(\sigma^2, \boldsymbol{\beta}_B, \lambda_\diamond^2, \boldsymbol{\mu}) \sim GAM\left(\alpha_\star + \frac{n_1}{2}, \beta_\star + \frac{1}{2}\sigma^{-2}\boldsymbol{\beta}_\star^\mathsf{T}\boldsymbol{\beta}_\star\right)$

(5) $\boldsymbol{\mu}|(\lambda_\diamond^2, \lambda_\star^2, \sigma^2) \sim \mathcal{N}(\boldsymbol{\mu}^\ddagger, \mathbf{\Sigma}^\ddagger)$; see Equations (5.20-5.22).

Metropolis-Hastings part: Withdraw σ^2 and $\boldsymbol{\beta}_B$, and keep $\lambda_\diamond^2$, $\lambda_\star^2$ and $\boldsymbol{\mu}$. Perform the two blocked Metropolis-Hastings moves from part B:

(6) We propose to replace $[\boldsymbol{\delta}, \boldsymbol{\mu}]$ by $[\boldsymbol{\delta}_\bullet, \boldsymbol{\mu}_\bullet]$, and we accept the new pair with the probability given in Equation (5.25). If accepted, we replace $[\boldsymbol{\delta}, \boldsymbol{\mu}]$ by $[\boldsymbol{\delta}_\bullet, \boldsymbol{\mu}_\bullet]$, otherwise we leave $[\boldsymbol{\delta}, \boldsymbol{\mu}]$ unchanged.

(7) We propose to replace $[\mathbf{\Pi}, \boldsymbol{\delta}, \boldsymbol{\mu}]$ by $[\mathbf{\Pi}_\bullet, \boldsymbol{\delta}_\bullet, \boldsymbol{\mu}_\bullet]$, and we accept the new triple with the probability given in Equation (5.26). If accepted, we replace $[\mathbf{\Pi}, \boldsymbol{\delta}, \boldsymbol{\mu}]$ by $[\mathbf{\Pi}_\bullet, \boldsymbol{\delta}_\bullet, \boldsymbol{\mu}_\bullet]$, otherwise we leave $[\mathbf{\Pi}, \boldsymbol{\delta}, \boldsymbol{\mu}]$ unchanged.

The MCMC algorithm generates a posterior sample:

$$\{\mathbf{\Pi}^{(w)}, \boldsymbol{\delta}^{(w)}, \lambda_{\star,(w)}^2, \lambda_{\diamond,(w)}^2, \boldsymbol{\mu}^{(w)}\} \sim p(\mathbf{\Pi}, \boldsymbol{\delta}, \lambda_\star^2, \lambda_\diamond^2, \boldsymbol{\mu}|\mathbf{y}) \qquad (w = 1, \dots, W) \tag{5.28}$$

As described in this chapter, we run the MCMC algorithm for $W = 100,000$ ($W = 100k$) iterations. We set the burn-in phase to $50k$ and we sample every 100th graph during the sampling phase. This yields $R = 500$ posterior samples for each response Y.

Chapter 6

Comparative evaluation of various frequentist and Bayesian non-homogeneous Poisson counting models

The Poisson distribution is one of the most popular statistical standard tools for analysing (homogeneous) count data, i.e. integer-valued samples. For modelling non-homogeneous count data, e.g. time series where the number of counts depends on time and hence systematically differs over time, various extensions of the standard Poisson model have been proposed and applied in the literature. More appropriate non-homogeneous Poisson models can be easily obtained by embedding the standard Poisson model into other statistical frameworks, such as changepoint models (CPS), finite mixture models (MIX), or hidden Markov models (HMM). The three aforementioned modelling approaches have become very popular statistical tools throughout the years for the following three reasons: (i) First, each of the three modelling approaches is of a generic nature so that it can be combined with a huge variety of statistical distributions and models to extend their flexibilities. (ii) Second, the statistical methodology behind those generic models is rather simple, described in lots of textbooks on Statistics and the model inference is feasible. (iii) Third, the three approaches can be easily formulated and implemented in both conceptual frameworks: the standard 'frequentist' framework and the Bayesian framework.

Despite this popularity, the performances of the resulting non-homogeneous models have never been systematically compared with each other in the statistical literature. This chapter tries to fill this gap and presents a comparative evaluation study on non-homogeneous Poisson count data, for which those three well-known statistical models (changepoint models, mixture models and hidden

Markov models) are implemented in both conceptual frameworks: the frequentist framework and the Bayesian framework.
More precisely, for the evaluation study the standard homogeneous Poisson model (HOM) and three non-homogeneous variants thereof, namely a Poisson changepoint model (CPS), a Poisson free mixture model (MIX), and a Poisson hidden Markov model (HMM) are implemented in a frequentist as well as in a Bayesian framework. The goal of the presented study is to systematically cross-compare the performances. Thereby the focus is not only on cross-comparing the generic modelling approach for non-homogeneity (CPS, MIX and HMM), but also on comparing the frequentist model instantiations with the Bayesian model instantiations. The study is performed on various synthetic data sets as well as on real-world taxi pick-up counts, extracted from the recently published New York City Taxi (NYCT) database. In all presented applications it is assumed that the Poisson parameter does *not* depend on any external covariates so that the changes are time-effects only. That is, the non-stationarity is implemented intrinsically by temporal changepoints, at which the Poisson process spontaneously changes its values.

Within this introductory text no literature references have been given, since detailed descriptions of all those generic statistical concepts, mentioned so far, can be found in many standard textbooks on Statistics, and therefore, in principle, will be familiar for most of the readers. However, in Section 6.1, where the models are described and mathematically formulated, explicit literature references will be provided for all models under comparison.

The work, presented in this chapter, has been published in Computational Statistics (2016) (see [28]).

6.1 Methods

6.1.1 Mathematical notations

Let $\mathbf{D}$ denote a n-by-T data matrix, whose columns refer to equidistant time points, $t \in \{1, \ldots, T\}$, and whose rows refer to independent counts, $i \in \{1, \ldots, n\}$, which were observed at the corresponding time points. The element $d_{i,t}$ in the i-th row and t-th column of $\mathbf{D}$ is the i-th count, which was observed at the t-th time point. Let $\mathbf{D}_{.,t} := (d_{1,t}, \ldots, d_{n,t})^\top$ denote the t-th column of $\mathbf{D}$, where "$\top$" denotes vector transposition. $\mathbf{D}_{.,t}$ is then the vector of the n observed counts for time point t.

Assume that the time points $1, \ldots, T$ are linked to K Poisson distributions with parameters $\theta_1, \ldots, \theta_K$. The T time points can then be assigned to K components, which represent the K Poisson distributions. More formally, let the allocation vector $\mathbf{V} = (v_1, \ldots, v_T)^\top$ define an allocation of the time points to components, where component k represents a Poisson distribution with parameter θ_k. $v_t = k$ means that time point t is allocated to the k-th component and that the observations at t stem from a Poisson distribution with parameter θ_k ($t = 1, \ldots, T$

and $k = 1, \dots, K$). Note that the n independent counts within each column are always allocated to the same component, while the T columns (time points) are allocated to different components. Define $\mathbf{D}^{[k]}$ to be the sub-matrix, containing only the columns of $\mathbf{D}$ that are allocated to component k.

The probability density function (pdf) of a Poisson distribution with parameter $\theta > 0$ is:

$$p(x|\theta) = \frac{\theta^x \cdot \exp\{-\theta\}}{x!} \tag{6.1}$$

for $x \in \mathbb{N}_0$. Assuming that all counts, allocated to k, are realisations of independently and identically distributed (iid) Poisson variables with parameter θ_k, the joint pdf is given by:

$$p(\mathbf{D}^{[k]}|\theta_k) = \prod_{t=1}^{T} \mathcal{I}_{\{v_t=k\}}(t) \cdot p(\mathbf{D}_{.,t}|\theta_k) \tag{6.2}$$

where $\mathcal{I}(t)_{\{v_t=k\}}(t)$ indicates whether time point t is allocated to component k, and

$$p(\mathbf{D}_{.,t}|\theta_k) = \prod_{i=1}^{n} p(d_{i,t}|\theta_k) = \frac{(\theta_k)^{\{\sum_{i=1}^{n} d_{i,t}\}} \cdot \exp\{-n \cdot \theta_k\}}{d_{1,t}! \cdot \ldots \cdot d_{n,t}!} \tag{6.3}$$

is the joint pdf of the counts in the t-th column of $\mathbf{D}$. Given the allocation vector $\mathbf{V}$, which allocates the data into K sub-matrices $\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[K]}$, and independent component-specific Poisson distributions with parameters $\theta_1, \dots, \theta_K$, the joint pdf of $\mathbf{D}$ is:

$$p(\mathbf{D}|\mathbf{V}, \boldsymbol{\theta}) = \prod_{k=1}^{K} p(\mathbf{D}^{[k]}|\theta_k) \tag{6.4}$$

where $\boldsymbol{\theta} := (\theta_1, \dots, \theta_K)^\top$, and $p(\mathbf{D}^{[k]}|\theta_k)$ was defined in Eq. (6.2).
Now assume that the allocation vector $\mathbf{V}$ is known and fixed, while the component-specific Poisson parameters are unknown and have to be inferred from the data $\mathbf{D}$.
Following the frequentist paradigm, the parameters can be estimated by the Maximum Likelihood (ML) approach. The ML estimators which maximise the log-likelihood

$$l(\boldsymbol{\theta}|\mathbf{V}, \mathbf{D}) := \log\{p(\mathbf{D}|\mathbf{V}, \boldsymbol{\theta})\} \tag{6.5}$$

are given by $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \dots, \hat{\theta}_K)^\top$, where $\hat{\theta}_k$ is the empirical mean of all counts in $\mathbf{D}^{[k]}$. Assuming that T_k time points are allocated to component k, the matrix $\mathbf{D}^{[k]}$ contains $T_k \cdot n$ counts and

$$\hat{\theta}_k = \frac{1}{n \cdot T_k} \sum_{t=1}^{T} \mathcal{I}_{\{v_t=k\}}(t) \sum_{i=1}^{n} d_{i,t} \tag{6.6}$$

In a Bayesian setting the Poisson parameters in $\boldsymbol{\theta}$ are assumed to be random variables as well, and prior distributions are imposed on them. The standard conjugate prior for a Poisson model with parameter $\theta_k > 0$ is the Gamma distribution:

$$p(\theta_k|a,b) = \frac{b^a}{\Gamma(a)} \cdot (\theta_k)^{a-1} \exp\{-\theta_k \cdot b\} \tag{6.7}$$

where a is the shape and b is the rate parameter. Due to standard conjugacy arguments, for each component k the posterior distribution is a Gamma distribution with parameters $\tilde{a} = a + \xi^{[k]}$ and $\tilde{b} = b + n \cdot T_k$

$$p(\theta_k|\mathbf{D}^{[k]}) = \frac{(b + n \cdot T_k)^{a+\xi^{[k]}}}{\Gamma(a + \xi^{[k]})} \cdot (\theta_k)^{a+\xi^{[k]}-1} \exp\{-\theta_k \cdot (b + n \cdot T_k)\} \tag{6.8}$$

where $\xi^{[k]}$ is the sum of all $n \cdot T_K$ elements of the n-by-T_k (sub-)matrix $\mathbf{D}^{[k]}$. The marginal likelihood can be computed in closed-form:

$$\begin{aligned} p(\mathbf{D}^{[k]}|a,b) &= \int_0^\infty p(\mathbf{D}^{[k]}|\theta_k) p(\theta_k|a,b) d\theta_k \qquad (6.9) \\ &= \frac{b^a}{\Gamma(a)} \cdot \frac{1}{\prod_{t=1}^{T_k} \prod_{i=1}^{n} (d_{i,t}^{[k]})!} \cdot \frac{\Gamma(a + \xi^{[k]})}{(T_k \cdot n + b)^{\xi^{[k]}+a}} \end{aligned}$$

where $d_{i,t}^{[k]}$ is the element in the i-th row and t-th column of $\mathbf{D}^{[k]}$.
Imposing independent Gamma priors on each component $k \in \{1\ldots, K\}$ induced by the allocation vector $\mathbf{V}$, the marginal likelihood for the complete data matrix $\mathbf{D}$ is:

$$p(\mathbf{D}|\mathbf{V}) = \prod_{k=1}^{K} p(\mathbf{D}^{[k]}|a,b) \tag{6.10}$$

where the dependence on the fixed hyperparameters a and b on the left hand side of the last equation was suppressed.

So far it has been assumed that the allocation vector $\mathbf{V}$ is known and fixed, although $\mathbf{V}$ will be unknown for many real-world applications so that $\mathbf{V}$ also has to be inferred from the data $\mathbf{D}$. The next section is therefore on the allocation vector inference.

6.1.2 Allocation vector inference

The standard frequentist Poisson or Bayesian Poisson-Gamma model assumes that the data are homogeneous so that all time points $t = 1, \ldots, T$ always belong to the same component; i.e. $K = 1$ and $\mathbf{V} = (1, \ldots, 1)^\top$. These models are referred to as the homogeneous (HOM) models. The HOM model is not adequate if the number of counts varies over time, and non-homogeneous Poisson models, which infer the underlying allocation, have to be used instead. Prominent approaches to model non-homogeneity include: multiple changepoint processes (CPS), finite mixture models (MIX), and hidden Markov models (HMM). CPS

impose a set of changepoints which divide the time series $1, \ldots, T$ into disjunct segments. Although this is a very natural choice for temporal data, the disadvantage is that the allocation space is restricted, as data points in different segments cannot be allocated to the same component; i.e. a component once left cannot be revisited. E.g. for $T = 6$ the true allocation $\mathbf{V} = (1,1,2,2,2,1)^\top$ cannot be modelled and the best CPS model approximation might be: $\mathbf{V}^{CPS} = (1,1,2,2,2,3)^\top$. The MIX model, on the other hand, is more flexible, as it allows for a free allocation of the time points so that $\mathbf{V}$ is part of the configuration space. But MIX does not take the temporal ordering of the data points into account. It treats the T time points as interchangeable units. This implies in the example above that all allocation vectors, which allocate $T_1 = 3$ time points to component $k = 1$ and $T_2 = 3$ time points to component $k = 2$, are always equally supported a priori; including unlikely allocations, such as: $\mathbf{V}^\star = (1,2,1,2,1,2)^\top$.
A compromise between CPS and MIX is the hidden Markov model (HMM). HMM allows for an unrestricted allocation vector configuration space, but unlike MIX it does not ignore the order of the time points. A homogeneous first-order HMM imposes a (homogeneous) Markovian dependency among the components $v_1, \ldots, v_T$ of $\mathbf{V}$ so that the value of v_t depends on the value of the preceding time point v_{t-1}, and the homogeneous state-transition probabilities can be such that neighbouring points are likely to be allocated to the same component, while components once left can be revisited. The aforementioned Poisson models, can be implemented in a frequentist as well as in a Bayesian framework, yielding 8 non-homogeneous Poisson models in total, see Table 6.1 for an overview.

6.1.3 The frequentist framework

The learning algorithms for the non-homogeneous frequentist models learn the best-fitting model for each number of components K, and the goodness of fit increases in K. Restricting K to be in between 1 and K_{MAX}, for each approach (CPS, MIX and HMM) the best fitting model with K components, symbolically $\mathcal{M}_K$, can be learn from the data $\mathbf{D}$. The Bayesian Information criterium (BIC), proposed by [53], is a well-known model selection criterium and balances between the goodness of fit and model sparsity. According to the BIC, among a set of models $\{\mathcal{M}_1, \ldots, \mathcal{M}_{K_{MAX}}\}$, the one with the lowest BIC value is considered the most appropriate one with the best trade-off (fit vs. sparsity). Given the n-by-T data set matrix $\mathbf{D}$, and models $\mathcal{M}_K$ with K components and $q_{\mathcal{M}_K}$ parameters ($K = 1, \ldots, K_{MAX}$), the BIC of $\mathcal{M}_K$ is defined as

$$BIC(\mathcal{M}_K) = -2 \cdot \log\{p(\mathbf{D}|\mathcal{M}_K)\} + q_{\mathcal{M}_K} \cdot \log(n \cdot T) \qquad (6.11)$$

where $n \cdot T$ is the number of data points in $\mathbf{D}$, and for $K = 1$ each of the three non-homogeneous model becomes the homogeneous model $\mathcal{M}_1$ (see Section 6.1.3).

	Frequentist version (FREQ)	**Bayesian version (BAYES)**
Homogeneous Model (**HOM**) with 1 parameter	**see Section 6.1.3** Well-known standard model from frequentist textbooks with closed-form solution.	**see Section 6.1.4** Well-known standard model from Bayesian textbooks with closed-form solution.
Changepoint Model (**CPS**) with K parameters	**see Section 6.1.3** For each K the best changepoint set can be determined with the Segment Neighbourhood Search Algorithm. BIC is used for model selection.	**see Section 6.1.4** Model averaging via MCMC, based on changepoint birth, death, and reallocation moves.
Finite Mixture Model (**MIX**) with $2K-1$ parameters	**see Section 6.1.3** For each K, the ML estimators of the incomplete model can be inferred with the EM algorithm. BIC is used for model selection.	**see Section 6.1.4** Model averaging via MCMC, based on the moves of the allocation sampler.
Hidden Markov Model (**HMM**) with K^2+2K-1 parameters	**see Section 6.1.3** For each K, the ML estimators of the incomplete model can be inferred with the EM algorithm. BIC is used for model selection.	**see Section 6.1.4** Model averaging via MCMC, based on the moves of the allocation sampler and four additional moves.

Table 6.1: Overview to the eight (non-)homogeneous Poisson models under comparison. Detailed explanations are given in the main text.

The homogeneous frequentist Poisson model (FREQ-HOM)

The homogeneous model $\mathcal{M}_1$ assumes that the counts stay constant over time, i.e. that there is only one single component, $K = 1$, and that the allocation vectors assign all data points to this component, i.e. $\mathbf{V} = \mathbf{1} = (1, \dots, 1)^\top$. Hence, $\mathbf{D}^{[1]} = \mathbf{D}$ and according to Eq. (6.6), the maximum likelihood (ML) estimator of the single ($q_{\mathcal{M}_1} = 1$) Poisson parameter $\theta := \theta_1$ is the empirical mean of all $T \cdot n$ data points in $\mathbf{D}$.

The frequentist changepoint Poisson model (FREQ-CPS)

A changepoint model uses a changepoint set of $K - 1$ changepoints, $C = \{c_1, \dots, c_{K-1}\}$, where $1 \leq c_1 < \dots < c_K < T$, to divide the time points $1, \dots, T$ into K disjunct segments. Time point t is assigned to component k if $c_{k-1} < t \leq c_k$, where $c_0 = 0$ and $c_K = T$ are pseudo changepoints. This means for the t-th element, v_t, of the allocation vector, $\mathbf{V}_C$, implied by C: $v_t = k$ if $c_{k-1} < t \leq c_k$. A changepoint set C with $K - 1$ changepoints implies a segmentation $\mathbf{D}^{[1]}, \dots, \mathbf{D}^{[K]}$ of the data matrix $\mathbf{D}$, and the ML estimators $\hat{\theta}_k$ for the segment-specific Poisson parameters θ_k can be computed with Eq. (6.6). The model fit can be quantified by plugging the ML estimators $\hat{\boldsymbol{\theta}}$ into the log-likelihood in Eq. (6.5):

$$l(\hat{\boldsymbol{\theta}}_C | \mathbf{V}_C, \mathbf{D}) := \log\{p(\mathbf{D} | \mathbf{V}_C, \hat{\boldsymbol{\theta}}_C)\} \tag{6.12}$$

where $\mathbf{V}_C$ is the allocation vector implied by C, and $\hat{\boldsymbol{\theta}}_C$ is the vector of ML estimators. The best fitting set of $K - 1$ changepoints, C^K, i.e. the set maximising Eq. (6.12), can be found recursively by the segment neighbourhood search algorithm. This algorithm, proposed by [4], employs dynamic programming to find the best fitting changepoint set with $K - 1$ changepoints for each K ($2 \leq K \leq K_{MAX}$). The algorithm is outlined in Section 6.8.1 of the Appendix. The best changepoint model $\mathcal{M}_{\hat{K}}$ minimises the BIC in Eq. (6.11), and the output of the algorithm is the corresponding allocation vector $\hat{\mathbf{V}}_{CPS}$ and the segment-specific ML-estimators $\hat{\boldsymbol{\theta}}_{CPS} := \hat{\boldsymbol{\theta}}_{C^{\hat{K}}}$.

The frequentist finitite mixture Poisson model (FREQ-MIX)

In a frequentist finite mixture model with K components the time points $1, \dots, T$ are treated as interchangeable units from a mixture of K independent Poisson distributions with parameters $\theta_1, \dots, \theta_K$ and mixture weights $\pi_1, \dots, \pi_K$, where $\pi_k \geq 0$ for all k, and $\sum_{k=1}^{K} \pi_k = 1$. The columns $\mathbf{D}_{.,t}$ of the data matrix $\mathbf{D}$ are then considered as a sample from this Poisson mixture distribution with pdf:

$$p(\mathbf{D}_{.,t} | \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k \cdot p(\mathbf{D}_{.,t} | \theta_k) \tag{6.13}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)^\top$ is the vector of Poisson parameters, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\top$ is the vector of mixture weights, and $p(\mathbf{D}_{.,t}|\theta_k)$ can be computed with Eq. (6.3). The maximisation of Eq. (6.13) in the parameters $(\boldsymbol{\theta}, \boldsymbol{\pi})$ is analytically not feasible so that the ML estimates have to be determined numerically. For mixture distributions this can be done with the Expectation Maximisation (EM) algorithm ([12]). The mathematical details of the EM algorithm are provided in Section 6.8.1 of the Appendix. The best mixture model $\mathcal{M}_{\hat{K}}$ minimises the BIC in Eq. (6.11), where $q_{\mathcal{K}} = \mathcal{K} + (\mathcal{K} - 1)$ is the number of Poisson and (free) mixture weight parameters.[1] The output of the EM-algorithm is the best number of components $\hat{K}_{MIX}$, the corresponding T-by-$\hat{K}_{MIX}$ allocation probability matrix $\hat{\boldsymbol{\Delta}}_{MIX}$, whose elements $\Delta_{t,k}$ are the probabilities that time point t belongs to component k, and the vector of ML estimators $\hat{\boldsymbol{\theta}}_{MIX}$.

The frequentist Hidden Markov Poisson model (FREQ-HMM)

The key assumption of a hidden Markov model (HMM) with K components ('states') is that the (unobserved) elements $v_1, \ldots, v_T$ of the allocation vector $\mathbf{V}$ follow a (homogeneous) first-order Markovian dependency. That is, $\{v_t\}_{t=1,\ldots,T}$ is considered a homogeneous Markov chain of order $\tau = 1$ with the state space $S = \{1, \ldots, K\}$, the initial distribution $\Pi = (\pi_1, \ldots, \pi_K)$, where $\pi_k \geq 0$ is the probability that v_1 is equal to k, and the K-by-K transition (probability) matrix $\mathbf{A}$, whose elements $a_{i,j} \geq 0$ are the transition probabilities for a transition from state i to state j: $a_{i,j} = P(v_{t+1} = j|v_t = i)$ for all $t \in \{1, \ldots, T-1\}$.[2] Assume that there are K state-dependent Poisson distributions so that each state $k \in \{1, \ldots, K\}$ corresponds to a Poisson distribution with parameter θ_k. The data matrix $\mathbf{D}$ is then interpreted as a sequence of its T columns, $\mathbf{D}_{.,1}, \ldots, \mathbf{D}_{.,T}$, and $v_t = k$ means that column $\mathbf{D}_{.,t}$ is a vector of n realisations of the k-th Poisson distribution with parameter θ_k. Mathematically, this means:

$$p(\mathbf{D}_{.,t}|v_t = k) = p(\mathbf{D}_{.,t}|\theta_k) \tag{6.14}$$

where $p(\mathbf{D}_{.,t}|\theta_k)$ was defined in Eq. (6.3). The Hidden Markov model is now fully specified and has the unknown parameters Π, $\mathbf{A}$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)^\top$. For given parameters $(\Pi, \mathbf{A}, \boldsymbol{\theta})$, the distribution of the unknown ('hidden') state sequence $v_1, \ldots, v_T$ can be inferred recursively with the foward and backward algorithm. And by combining the forward and backward algorithms with the EM algorithm, the best HMM model $\mathcal{M}_{\hat{K}}$, which minimises the BIC in Eq. (6.11), can be numerically determined. The details of the inference procedure are provided in Section 6.8.2 of the Appendix. For a HMM model with K components the total number of parameters is $q_{\mathcal{K}} = \mathcal{K} + (\mathcal{K} - 1) + (\mathcal{K}^2 - \mathcal{K})$, i.e. the sum of the Poisson parameters, the free initial probability parameters and the free transition probability parameters.[3] The output of the EM algorithm, as described in Section 6.8.2 of the Appendix, is the best number of components $\hat{K}_{HMM}$, the corresponding T-by-$\hat{K}_{HMM}$ allocation probability matrix $\hat{\mathbf{\Delta}}_{HMM}$, whose elements $\Delta_{t,k}$ are the probabilities that time point t belongs to component k, and the ML-estimators $\hat{\boldsymbol{\theta}}_{HMM}$.

---

[1] The K mixture weights fulfil: $\sum_{k=1}^{K} \pi_k = 1$.

[2] It holds: $\sum_{k=1}^{K} \pi_k = 1$, and $\sum_{j=1}^{K} a_{i,j} = 1$ for all i.

[3] Note that: $\sum_{k=1}^{K} \pi_k = 1$, and $\sum_{l=1}^{K} a_{k,l} = 1$ for $k = 1, \ldots, K$.

6.1.4 The Bayesian framework

The Bayesian models employ a Poisson-Gamma model, for which the marginal likelihood $p(\mathbf{D}|\mathbf{V})$ can be computed with Eq. (6.10). While the homogeneous model, described in Section 6.1.4, keeps $K = 1$ fixed, the three non-homogeneous models have to infer K and the unknown allocation vector $\mathbf{V}$. In a Bayesian framework this means that prior distributions have to be imposed on $\mathbf{V}$ and K. The three non-homogeneous models, described below, assume that the joint prior distribution can be factorized, $p(\mathbf{V}, K) = p(\mathbf{V}|K) \cdot p(K)$, and impose on K a truncated Poisson distribution with parameter λ and the truncation $1 \leq K \leq K_{MAX}$ so that $p(K) \propto \lambda^K \cdot \exp\{-\lambda\} \cdot (K!)^{-1}$.
Subsequently, the prior on $\mathbf{V}$ is specified conditional on K. The marginal likelihood $p(\mathbf{D}|\mathbf{V})$ and the two prior distributions $p(K)$ and $p(\mathbf{V}|K)$ together fully specify the Bayesian model, and Markov Chain Monte Carlo (MCMC) simulations are used to generate samples $(\mathbf{V}^{(1)}, K^{(1)}), \dots, (\mathbf{V}^{(R)}, K^{(R)})$ from the posterior distribution:

$$p(\mathbf{V}, K|\mathbf{D}) \propto p(\mathbf{D}|\mathbf{V}) \cdot p(\mathbf{V}|K)(K) \tag{6.15}$$

The Bayesian models, described below, differ only by the conditional prior $p(\mathbf{V}|K)$.

The homogeneous Bayesian Poisson-Gamma model (BAYES-HOM)

The homogeneous Bayesian model assumes that the counts do not vary over time, so that $K = 1$ and $\mathbf{V} = (1, \dots, 1)^\top =: \mathbf{1}$ and $\mathbf{D}^{[1]} = \mathbf{D}$. According to Eqns. (6.9-6.10), the marginal likelihood of the BAYES-HOM model is then given by

$$\begin{aligned} p(\mathbf{D}^{[1]}|\mathbf{V} = \mathbf{1}) &= \int_0^\infty p(\mathbf{D}|\theta)p(\theta|a, b)d\theta \\ &= \frac{b^a}{\Gamma(a)} \cdot \frac{1}{\prod_{t=1}^T \prod_{i=1}^n (d_{i,t})!} \cdot \frac{\Gamma(a+\xi)}{(T \cdot n + b)^{\xi+a}} \end{aligned}$$

where $d_{i,t}$ is the element in the i-th row and t-th column of $\mathbf{D}$, and ξ is the sum of all $n \cdot T$ elements of $\mathbf{D}$.

The Bayesian changepoint Poisson-Gamma model (BAYES-CPS)

There are various possibilities to implement a Bayesian changepoint model, and here the classical one from [21] is used. The prior on K is a truncated Poisson distribution, and each K is identified with $K-1$ changepoints $c_1, \dots, c_{K-1}$ on the discrete set $\{1, \ldots, T-1\}$, where $v_t = k$ if $c_{k-1} < t \leq c_k$, and $c_0 := 1$ and $c_K := T$ are pseudo changepoints. Conditional on K, the changepoints are assumed to be distributed like the even-numbered order statistics of $L := 2(K-1)+1$ points uniformly and independently distributed on $\{1, \ldots, T-1\}$. This implies that changepoints cannot be located at neighbouring time points and induces the prior distribution:

$$P(\mathbf{V}|K) = \frac{1}{\begin{pmatrix} T-1 \\ 2(K-1)+1 \end{pmatrix}} \prod_{k=0}^{K-1} (c_{k+1} - c_k - 1) \tag{6.16}$$

The BAYES-CPS model is now fully specified and K and $\mathbf{V}$ can be sampled from the posterior distribution $p(\mathbf{V}, K|\mathbf{D})$, defined in Eq. (6.15), with a Metropolis-Hastings MCMC sampling scheme, based on changepoint birth, death and reallocation moves ([21]).

Given the current state at the r-th MCMC iteration: $(\mathbf{V}^{(r)}, K^{(r)})$, where $\mathbf{V}^{(r)}$ can be identified with the changepoint set: $C^{(r)} = \{c_1, \ldots, c_{K^{(r)}} - 1\}$, one of the three move types is randomly selected (e.g. each with probability $1/3$) and performed. The three move types (i-iii) can be briefly described as follows:

(i) In the changepoint reallocation move one changepoint c_j from the current changepoint set $C^{(r)}$ is randomly selected, and the replacement changepoint is randomly drawn from the set $\{c_{j-1}+2, \ldots, c_{j+1}-2\}$. The new set $C^{\star}$ gives the new candidate allocation vector $\mathbf{V}^{\star}$; the number of components stays unchanged: $K^{\star} = K^{(r)}$.

(ii) The changepoint birth move randomly draws the location of one single new changepoint from the set of all valid new changepoint locations:

$$B^{\dagger} := \Big\{c \in \{1, \ldots, T-1\} : |c - c_j| > 1 \forall j \in \Big\{1, \ldots, K^{(r)} - 1\Big\}\Big\} \tag{6.17}$$

Adding the new changepoint to $C^{(r)}$ yields $K^{\star} = K^{(r)} + 1$, and the new set $C^{\star}$, which yields the new allocation vector $\mathbf{V}^{\star}$.

(iii) The changepoint death move is complementary to the birth move. It randomly selects one of the changepoints from $C^{(r)}$ and proposes to delete it. This gives the new changepoint set $C^{\star}$ which yields the new candidate allocation vector $\mathbf{V}^{\star}$, and $K^{\star} = K^{(i)} - 1$.

For all three moves the Metropolis-Hastings acceptance probability for the new candidate state $(\mathbf{V}^{\star}, K^{\star})$ is given by $A = \min\{1, R\}$, with

$$R = \frac{p(\mathbf{D}|\mathbf{V}^{\star})}{p(\mathbf{D}|\mathbf{V}^{(r)})} \cdot \frac{p(\mathbf{V}^{\star}|K^{\star})p(K^{\star})}{p(\mathbf{V}^{(r)}|K^{(r)})p(K^{(r)})} \cdot Q \tag{6.18}$$

where Q is the Hastings ratio, which can be computed straightforwardly for each of the three move types (see, e.g., [21]). If the move is accepted, set $\mathbf{V}^{(r+1)} = \mathbf{V}^{\star}$ and $K^{(r+1)} = K^{\star}$, or otherwise leave the state unchanged: $\mathbf{V}^{(r+1)} = \mathbf{V}^{(r)}$ and $K^{(r+1)} = K^{(r)}$.

The Bayesian finite mixture Poisson-Gamma model (BAYES-MIX)

Here, the Bayesian finite mixture model instantiation and the Metropolis Hastings MCMC sampling scheme proposed by [46] is employed. The prior on K is a truncated Poisson distribution, and conditional on K, a categorical distribution (with K categories) and probability parameters $\mathbf{p} = (p_1, \ldots, p_K)^\top$ is used as prior for the allocation variables $v_1, \ldots, v_T \in \{1, \ldots, K\}$. That is, $\sum_{k=1}^{K} p_k = 1$ and $p(v_t = k) = p_k$. The probability of the allocation vector $\mathbf{V} = (v_1, \ldots, v_T)^\top$ is then given by:

$$p(\mathbf{V}|\mathbf{p}) = \prod_{k=1}^{K} (p_k)^{n_k} \tag{6.19}$$

where $n_k = |\{t \in \{1, \ldots, T\} : v_t = k\}|$ is the number of time points that are allocated to component k by $\mathbf{V}$. Imposing a conjugate Dirichlet distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)^\top$ on $\mathbf{p}$ and marginalizing over $\mathbf{p}$, yields the closed-form solution:

$$p(\mathbf{V}|K) = \int p(\mathbf{V}|\mathbf{p}) p(\mathbf{p}|\boldsymbol{\alpha}) d\mathbf{p} = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\Gamma(\sum_{k=1}^{K} (n_k + \alpha_k))} \prod_{k=1}^{K} \frac{\Gamma(n_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{6.20}$$

The BAYES-MIX model is now fully specified, and the posterior distribution is invariant to permutations of the components' labels if: $\alpha_k = \alpha$. A Metropolis-Hastings MCMC sampling scheme, proposed by [46] and referred to as the "allocation sampler", can be used to generate a sample from the posterior distribution in Eq. (6.15). The allocation sampler consists of a simple Gibbs move and five more involved Metropolis-Hastings moves. Given the current state at the r-th iteration: $(\mathbf{V}^{(r)}, K^{(r)})$ the Gibbs move keeps the number of components fixed, $K^{(r+1)} = K^{(r)}$, and just re-samples the value of one single allocation variable $v_t^{(r)}$ from its full conditional distribution. This yields a new allocation vector $\mathbf{V}^{(r+1)}$ with a re-sampled t-th component $v_t^{(r+1)}$. As this Gibbs move has two disadvantages, [46] propose to use five additional Metropolis Hastings MCMC moves. (i) As the Gibbs move yields only very small steps in the allocation vector configuration space, [46] propose three additional Metropolis Hastings MCMC moves, referred to as the M1, M2 and M3 move, which also keep $K^{(r)}$ fixed but allow for re-allocations of larger sub-sets of the allocation variables $v_1^{(r)}, \ldots, v_T^{(r)}$. (ii) As neither the Gibbs move nor the M1-M3 moves can change the number of components, [46] also propose a pair of moves, referred to as the Ejection- and Absorption move, which generate a new or delete an existing component, so that $K^{(r+1)} = K^{(r)} + 1$ or $K^{(r+1)} = K^{(r)} - 1$, respectively. The technical details of the moves can be found in [46].

The Bayesian Hidden Markov Poisson-Gamma model (BAYES-HMM)

The focus is on a Bayesian hidden Markov model instantiation, which was recently proposed in [22] in the context of non-homogeneous dynamic Bayesian network models. The prior on K follows a truncated Poisson distribution, and for each K a HMM model with K states is used to model the allocation vector $\mathbf{V}$. To this end, $\mathbf{V}$ is identified with the temporally ordered sequence of its components: $v_1, \ldots, v_T$, and it is assumed that the latter sequence describes a homogeneous first order Markov chain with a uniform initial distribution and a K-by-K transition matrix $\mathbf{A}$.

Let $a_{l,k}$ be the element in the l-th row and k-th column of the transition matrix $\mathbf{A}$. $a_{l,k}$ is then the probability for a transition from component l to component k, and $\sum_{k=1}^{K} a_{l,k} = 1$. For a homogeneous Markov chain this means: $a_{l,k} = P(v_t = k|v_{t-1} = l, \mathbf{A}, K)$ for all t, and hence:

$$\begin{aligned} p(\mathbf{V}|\mathbf{A}, K) = p(v_1, \ldots, v_T|\mathbf{A}, K) &= p(v_1|K) \prod_{t=2}^{T} p(v_t|v_{t-1}, \mathbf{A}, K) \qquad (6.21) \\ &= \frac{1}{K} \prod_{k=1}^{K} \prod_{l=1}^{K} (a_{l,k})^{n_{l,k}} \end{aligned}$$

where $n_{l,k} = |\{t \in \{2, \ldots, T\} : v_t = k \wedge v_{t-1} = l\}|$ is the number of transitions from l to k in the sequence $v_1, \ldots, v_T$.

Each row $\mathbf{A}_{l,.}$ of the transition matrix $\mathbf{A}$ defines the probability vector of a categorical random variable (with K categories), and on each vector $\mathbf{A}_{l,.}$ an independent Dirichlet distribution with parameter vector $\boldsymbol{\alpha}_l = (\alpha_{l,1}, \ldots, \alpha_{l,K})^{\top}$ can be imposed:

$$p(\mathbf{A}_{l,.}|\boldsymbol{\alpha}_l) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_{l,k})}{\Gamma(\sum_{k=1}^{K} \alpha_{l,k})} \prod_{k=1}^{K} (a_{l,k})^{\alpha_{l,k}-1} \qquad (6.22)$$

Marginalizing over the transition matrix $\mathbf{A}$ in Equation (6.21), i.e. marginalizing over the row vectors $\mathbf{A}_{1,.}, \ldots, \mathbf{A}_{K,.}$, where each row vector $\mathbf{A}_{l,.}$ has an independent Dirichlet prior, defined in Eq. (6.22), gives the marginal distribution:

$$p(\mathbf{V}|K) = \int_{\mathbf{A}_{1,.}} \cdots \int_{\mathbf{A}_{K,.}} p(\mathbf{V}|\mathbf{A}, K) \left\{ \prod_{l=1}^{K} p(\mathbf{A}_{l,.}|\boldsymbol{\alpha}_l) \right\} d\mathbf{A}_{1,.} \ldots d\mathbf{A}_{K,.} \qquad (6.23)$$

Inserting Eq. (6.21) into Equation (6.23) yields:

$$P(\mathbf{V}|K) = \frac{1}{K} \prod_{l=1}^{K} \left(\int_{\mathbf{A}_{l,.}} P(\mathbf{A}_{l,.}|\boldsymbol{\alpha}_l) \prod_{k=1}^{K} (a_{l,k})^{n_{l,k}} \, d\mathbf{A}_{l,.} \right) \qquad (6.24)$$

The inner integrals in Eq. (6.23) correspond to Multinomial-Dirichlet distributions, which can be computed in closed form:

$$P(\mathbf{V}|K) = \frac{1}{K} \prod_{l=1}^{K} \frac{\Gamma(\sum_{k=1}^{K} \alpha_{l,k})}{\Gamma(\sum_{k=1}^{K} n_{l,k} + \alpha_{l,k})} \prod_{k=1}^{K} \frac{\Gamma(n_{l,k} + \alpha_{l,k})}{\Gamma(\alpha_{l,k})} \qquad (6.25)$$

The BAYES-HMM model is now fully specified, and with $\alpha_{l,k} = \alpha$ in Eq. (6.22) the marginal distribution $P(\mathbf{V}|K)$ in Equation (6.25) is invariant to permutations of the states' labels.

In principle, the allocation sampler from [46] from Section 6.1.4 can also be used to generate a sample from the posterior distribution in Eq. (6.15). However, the allocation sampler moves have been developed for finite mixture models, where data points are treated as interchangeable units without any order. Hence, the allocation sampler moves are sub-optimal when a Markovian dependency structure among temporal data points is given. In [22] it has been shown that the performance of the allocation sampler can be significantly improved in terms of convergence and mixing by including two new pairs of complementary Metropolis-Hastings moves. These two pairs of moves, referred to as the 'inclusion and exclusion moves' and the 'birth and death moves' in [22], exploit the temporal structure of the data points. A detailed description of these moves can be found in [22].

6.2 Validation

Table 6.1 gives an overview to the models from Section 6.1, and Table 6.2 shows the outputs of those models. The outputs range from a scalar ML estimate (FREQ-HOM) to an MCMC sample of allocation vectors (e.g. BAYES-HMM). For each model the output inferred from $\mathbf{D}$ can be used to estimate the probability of a new validation data set $\tilde{\mathbf{D}}$. Assume that in addition to the n-by-T data matrix $\mathbf{D}$ from Section 6.1.1, another $\tilde{n}$-by-T data matrix $\tilde{\mathbf{D}}$ is given and that the time points $1, \dots, T$ in $\mathbf{D}$ and $\tilde{\mathbf{D}}$ can be mapped onto each other.

Each non-homogeneous **Bayesian model** with K components and allocation vector $\mathbf{V}$ inferred from $\mathbf{D}$ can then be used to subdivide the new data matrix $\tilde{\mathbf{D}}$ into submatrices $\tilde{\mathbf{D}}^{[1]}, \dots, \tilde{\mathbf{D}}^{[K]}$, and the predictive probability for the k-th sub-matrix $\tilde{\mathbf{D}}^{[k]}$ is:

$$\begin{aligned} p(\tilde{\mathbf{D}}^{[k]}|\mathbf{D}^{[k]}) &= \int_0^\infty p(\tilde{\mathbf{D}}^{[k]}|\theta_k)p(\theta_k|\mathbf{D}^{[k]})d\theta_k \qquad (6.26) \\ &= \frac{\tilde{b}^{\tilde{a}}}{\Gamma(\tilde{a})} \cdot \frac{1}{\prod_{t=1}^{T_k}\prod_{i=1}^{\tilde{n}}(\tilde{d}_{i,t}^{[k]})!} \cdot \frac{\Gamma(\tilde{a}+\tilde{\xi}^{[k]})}{(T_k \cdot \tilde{n} + \tilde{b})^{\tilde{\xi}^{[k]}+\tilde{a}}} \end{aligned}$$

where T_k is the number of columns allocated to k, $\tilde{a} = a + \xi^{[k]}$ and $\tilde{b} = b + n \cdot T_k$ are the posterior parameters, defined above Eq. (6.8), $\tilde{\xi}^{[k]}$ is the sum of all $\tilde{n} \cdot T_k$ elements of the $\tilde{n}$-by-T_k sub-matrix $\tilde{\mathbf{D}}^{[k]}$, and $\tilde{d}_{i,t}^{[k]}$ is the element in the i-th row and t-th column of $\tilde{\mathbf{D}}^{[k]}$.

The (logarithmic) predictive probability of $\tilde{\mathbf{D}}$ conditional on K and $\mathbf{V}$ is then given by:

$$\log\{p(\tilde{\mathbf{D}}|\mathbf{D}, \mathbf{V}, K)\} = \sum_{k=1}^{K} \log\{p(\tilde{\mathbf{D}}^{[k]}|\mathbf{D}^{[k]})\} \qquad (6.27)$$

	Frequentist version (FREQ)	**Bayesian version (BAYES)**
Homogeneous Model (**HOM**) with 1 parameter	$K = 1, \mathbf{V} = (1, \ldots, 1)^{\top}$ scalar ML-estimator $\hat{\theta}$ estimated from the n-by-T values in $\mathbf{D}$.	$K = 1, \mathbf{V} = (1, \ldots, 1)^{\top}$ 1-dimensional posterior distribution $p(\theta\|\mathbf{D})$ inferred from the n-by-T values in $\mathbf{D}$.
Changepoint Model (**CPS**) with $\hat{K}$ parameters	$\hat{K}_{CPS}$, one concrete allocation vector instantiation $\hat{\mathbf{V}}_{CPS}$, and a vector of the $\hat{K}_{CPS}$ component-specific ML-estimators $\hat{\boldsymbol{\theta}}_{CPS}$.	A sample $\{K^{(r)}, \mathbf{V}^{(r)}\}_{r=1,\ldots,R}$, and for each r a set of $K^{(r)}$ posterior distributions $p(\theta_{k,r}\|\mathbf{D}^{[k,r]})$ where $\theta_{k,r}$ and $\mathbf{D}^{[k,r]}$ refer to the k-th component of the r-th sample.
Finite Mixture Model (**MIX**) with $2\hat{K} - 1$ parameters	$\hat{K}_{MIX}$, a vector of the $\hat{K}_{MIX}$ component-specific ML estimators $\hat{\boldsymbol{\theta}}_{MIX}$, and a T-by-K matrix $\hat{\boldsymbol{\Delta}}_{MIX}$, whose elements $\hat{\Delta}_{t,k}$ are the estimated allocation probabilities $p(v_t = k\|\mathbf{D})$.	A sample $\{K^{(r)}, \mathbf{V}^{(r)}\}_{r=1,\ldots,R}$, and for each r a set of $K^{(r)}$ posterior distributions $p(\theta_{k,r}\|\mathbf{D}^{[k,r]})$ where $\theta_{k,r}$ and $\mathbf{D}^{[k,r]}$ refer to the k-th component of the r-th sample.
Hidden Markov Model (**HMM**) with $\hat{K}^2 + 2\hat{K} - 1$ parameters	$\hat{K}$, a vector of the $\hat{K}_{HMM}$ component-specific ML estimators $\hat{\boldsymbol{\theta}}_{HMM}$, and a T-by-K matrix $\hat{\boldsymbol{\Delta}}_{HMM}$ whose elements $\hat{\Delta}_{t,k}$ are the estimated allocation probabilities $p(v_t = k\|\mathbf{D})$.	A sample $\{K^{(r)}, \mathbf{V}^{(r)}\}_{r=1,\ldots,R}$, and for each r a set of $K^{(r)}$ posterior distributions $p(\theta_{k,r}\|\mathbf{D}^{[k,r]})$ where $\theta_{k,r}$ and $\mathbf{D}^{[k,r]}$ refer to the k-th component of the r-th sample.

Table 6.2: Overview of the outputs of the eight (non-)homogeneous Poisson models. Detailed explanations are given in the main text.

For the homogeneous Bayesian model with $K = 1$, $\mathbf{D}^{[1]} = \mathbf{D}$, and $\tilde{\mathbf{D}}^{[1]} = \tilde{\mathbf{D}}$, the predictive probability of $\tilde{\mathbf{D}}$ can be computed analytically. For each non-homogeneous Bayesian model $\mathcal{M}$ an MCMC simulation generates a sample $\{\mathbf{V}^{(r)}, K^{(r)}\}_{r=1,\dots,R}$ from the posterior distribution $p(K, \mathbf{V}|\mathbf{D})$ in Eq. (6.15), and the predictive probability of model $\mathcal{M}$ can be approximated by:

$$\log\{p(\tilde{\mathbf{D}}|\mathbf{D}, \mathcal{M})\} \approx \frac{1}{R}\sum_{r=1}^{R} \log\{p(\tilde{\mathbf{D}}|\mathbf{D}, \mathbf{V}^{(r)}, K^{(r)})\} \tag{6.28}$$

For the **frequentist models** it can be proceeded similarly: After data matrix $\mathbf{D}$ has been used to learn a model and its ML-estimates, the probability of the new data matrix $\tilde{\mathbf{D}}$, given the model and the ML estimates learnt from $\mathbf{D}$, is a measure which corresponds to a Bayesian predictive probability. The homogeneous model and the changepoint model both output concrete values for $\hat{K}$ and $\hat{\mathbf{V}}$, and:

$$\log\{p(\tilde{\mathbf{D}}|\hat{K}, \hat{\mathbf{V}}, \hat{\boldsymbol{\theta}})\} = \sum_{k=1}^{\hat{K}} \log\{p(\tilde{\mathbf{D}}^{[k,\hat{\mathbf{V}}]}|\hat{\theta}_k)\} \tag{6.29}$$

where $\hat{K}$, $\hat{\mathbf{V}}$, and $\hat{\boldsymbol{\theta}}$ are those values inferred from the training data $\mathbf{D}$, $\tilde{\mathbf{D}}^{[k,\hat{\mathbf{V}}]}$ is the k-th submatrix of the validation data $\tilde{\mathbf{D}}$ implied by $\hat{\mathbf{V}}$, and $p(\tilde{\mathbf{D}}^{[k,\hat{\mathbf{V}}]}|\hat{\theta}_k)$ can be computed with Eq. (6.2).[4] FREQ-MIX and FREQ-HMM both infer the number of components $\hat{K}$ and the Poisson parameters $\hat{\boldsymbol{\theta}}$ but no concrete allocation vector. They infer a $\hat{K}$-by-T matrix $\hat{\boldsymbol{\Delta}}$, whose elements $\hat{\Delta}_{k,t}$ are the probabilities that time point t is allocated to component k, symbolically $\hat{\Delta}_{k,t} = \hat{p}(v_t = k|\mathbf{D})$. The probability of the new data set $\tilde{\mathbf{D}}$ is then given by:

$$\log\{p(\tilde{\mathbf{D}}|\hat{K}, \hat{\boldsymbol{\Delta}}, \hat{\boldsymbol{\theta}})\} = \sum_{t=1}^{T} \log\{\sum_{k=1}^{\hat{K}} \hat{\Delta}_{k,t} \cdot p(\tilde{D}_{\cdot,t}|\hat{\theta}_k)\} \tag{6.30}$$

6.3 Data

6.3.1 Synthetic data

Synthetic count data matrices are generated as follows: Let $\mathbf{V}^\star = (v_1^\star, \dots, v_T^\star)^\top$ be the true allocation vector, which allocates each time point $t \in \{1, \dots, T\}$ to a component $k \in \{1, \dots, K^\star\}$, where $v_t^\star = k$ means that t is allocated to k. Given $\mathbf{V}^\star$, n-by-T data set matrices $\mathbf{D}^\star$ can be obtained by sampling each matrix element $d_{i,t}^\star$ independently from a Poisson distribution with parameter $\theta_{v_t^\star}$ ($i = 1, \dots, n$ and $t = 1, \dots, T$).

The focus of the study is on different allocation vectors $\mathbf{V}^\star$ with different component-specific Poisson parameters $\theta_1, \dots, \theta_{K^\star}$. Let $\mathbf{P}^\star = (p_1, \dots, p_T)$ denote a row vector whose element p_t is the Poisson parameter for time point t. That is, $p_t = \lambda$ means that $\mathbf{V}^\star$ allocates time point t to a component with Poisson parameter $\theta_{v_t^\star} = \lambda$. The row vector $\mathbf{P}^\star$ will be referred to as the vector of Poisson parameters.

---

[4] For the homogeneous model it holds: $\hat{K} = 1$ and $\tilde{\mathbf{D}}^{[1,\hat{\mathbf{V}}]} = \tilde{\mathbf{D}}$.

Let $\mathbf{s}_m$ denote a row vector of length m, whose elements are all equal to $s \in \mathbb{N}$, $\mathbf{s}_m = (s, \ldots, s)$. The situation, where an allocation vector $\mathbf{V}^\star$ allocates $T = 4 \cdot m$ time points to $K^\star = 4$ equidistant coherent segments of length m, with the four component-specific Poisson parameters $\theta_1 = 1$, $\theta_2 = 5$, $\theta_3 = 3$, and $\theta_4 = 8$, can then be defined compactly:

$$\mathbf{P}^\star = (\underbrace{1, \ldots, 1}_{m-times}, \underbrace{5, \ldots, 5}_{m-times}, \underbrace{3, \ldots, 3}_{m-times}, \underbrace{8, \ldots, 8}_{m-times}) =: (\mathbf{1}_m, \mathbf{5}_m, \mathbf{3}_m, \mathbf{8}_m)$$

For the situation where the allocation vector follows a free mixture model, e.g., by allocating $T = 2 \cdot m$ time points to $K^\star = 2$ components with Poisson parameters $\theta_1 = 1$ and $\theta_2 = 5$, let $\mathbf{P}^\star = \mathbf{MIX}(\mathbf{1}_m, \mathbf{5}_m)$ denote that $\mathbf{P}^\star$ is a row vector whose elements are a random permutation of the elements of the vector $(\mathbf{1}_m, \mathbf{5}_m)$.

With regard to the real-world Taxi data, described in Section 6.3.2, each data matrix $\mathbf{D}$ is built with $T = 96$ columns (time points) and $n \in \{1, 2, 4, 8, 16\}$ rows (independent samples per time point). An overview to the allocation schemes (vectors of Poisson parameters), employed in the comparative evaluation study, is given in Table 4 of the Appendix. For each of the four allocation scenarios (HOM, CPS, MIX, and HMM) two different vectors of Poisson parameters are considered. Data matrices are built with a varying no. of rows $n \in \{1, 2, 4, 8, 16\}$ and $T = 96$ columns. For each of the resulting $4 \cdot 2 \cdot 5 = 40$ combinations, 25 independent data matrix instantiations are generated, i.e. 1000 data matrices in total. Subsequently, for each of those 1000 data matrix instantiations a $\tilde{n}$-by-T validation data matrix with $\tilde{n} = 30$ and $T = 96$ is sampled the same way (using the same vector of Poisson parameters).[5]

6.3.2 The New York City Taxi (NYCT) data from 2013

Through a 'Freedom of Information Law' request from the 'New York City Taxi and Limousine Commission' a dataset, covering information of about 700 million taxi trips in New York City (USA) from the calendar years 2010-2013, was published and stored by the University of Illinois ([15]). In the NYCT database, for each trip various details are provided; e.g. (i) the number of transported passengers, (ii) the pick-up and drop-off dates and daytimes, (iii) the GPS coordinates, where the passenger(s) were picked up and dropped off.[6] In this paper the focus is on the pick-up dates and daytimes of about 170 million taxi rides in the most recent year 2013, so that only a fractional amount of the data is used. Each pick-up is interpreted as a 'taxi call', so that it can be analysed how the number of taxi

---

[5] Note that T and $\tilde{n}$ have been set in accordance with the NYCT data, described in Section 6.3.2.

[6] The NYCT data can be downloaded from: http://dx.doi.org/10.13012/J8PN93H8

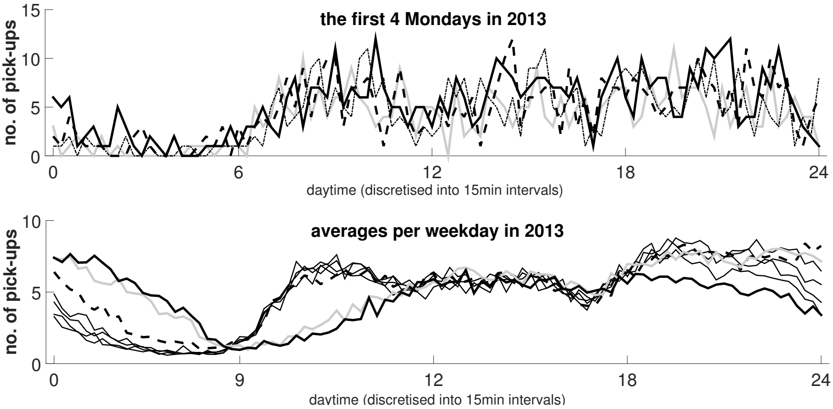


Figure 6.1: New York City Taxi pick-up time series. To shed some light onto the variability of the daily profiles, the upper panel shows the time series of the first 4 Mondays in 2013. The lower panel shows the seven weekday averages in 2013. Three weekdays with slightly deviating profiles have been highlighted: Sunday (bold black), Saturday (grey), and Friday (dotted black).

calls varies over the daytime. The data preparation can be summarised as follows: For each of about 170 million taxi rides from 2013 the pick-up date and daytime are extracted and down-sampled by a factor of 1000 (by randomly selecting 0.1% of the extracted samples), before all entries corresponding to US holidays are withdrawn.[7] Subsequently, there remain 169,596 date-and-time entries, which subdivide onto the 7 weekdays as indicated in Table 6.7 of the Appendix. Discretising the daytimes into $T = 96$ equidistant time intervals[8], each covering 15 minutes of the 24-hour day, and binning the pick-up times of each individual day into the $T = 96$ time intervals, gives a 355-by-96 data matrix $\mathbf{D}$, whose elements $d_{i,t}$ are the number of taxi pick-ups (or taxi calls) on the i-th day in time interval t. Since the seven weekdays might show different patterns, the data set matrix $\mathbf{D}$ is subdivided into seven n_w-by-T sub-matrices $\mathbf{D}_w$ ($w = 1, \ldots, 7$), where w indicates the weekday, and $n_w \in \{46, 50, 51, 52\}$ varies with the weekdays (see Table 6.7 of the Appendix). Figure 6.1 shows the number of Taxi calls for the first four Mondays in 2013 and the weekday averages.

In the study the weekdays are analysed separately, as they are likely to show different patterns. For each weekday $n \in \{1, 2, 4, 8, 16\}$ rows (days) are randomly selected from $\mathbf{D}_w$, before $\tilde{n} = 30$ of the remaining $n_w - n$ rows are randomly selected to build a validation data matrix. Repeating this procedure 5-times independently yields 150 data matrix pairs $\mathbf{D}_{w,n,u}$ and $\widetilde{\mathbf{D}}_{w,\tilde{n},u}$, where $w \in \{1, \ldots, 7\}$

---

[7] The following US holidays in 2013 are excluded: Jan 1 (New Year's Day), Jan 21 (Martin Luther King), Feb 18 (Presidents' Day), May 27 (Memorial Day), Jul 4 (Independence Day), Sep 2 (Labor Day), Oct 14 (Columbus Day), Nov 11 (Veterans Day), Nov 28 (Thanksgiving Day) and Dec 25 (Christmas Day).

[8] The time information is provided in seconds in the format: hh-mm-ss, ranging from 00-00-00 (midnight) to 23-59-59 (last second of the day).

indicates the weekday, $n \in \{1, 2, 4, 8, 16\}$ and $\tilde{n} = 30$ indicate the number of rows of $\mathbf{D}_{w,n,u}$ and $\tilde{\mathbf{D}}_{w,\tilde{n},u}$, and $u \in \{1, \ldots, 5\}$ indicates the replicate. Each $\mathbf{D}_{w,n,u}$ is a n-by-96 matrix and each $\tilde{\mathbf{D}}_{w,n,u}$ is a 30-by-96 matrix.[9]

6.4 Simulation Details

For all models the maximal number of components is set to $K_{MAX} = 10$. In the Gamma priors, see Eq. (6.7), both hyperparameters a and b are set to 1 so as to obtain rather uninformative priors. In terms of equivalent sample sizes this setting corresponds to one ($b = 1$) additional pseudo observation with one single taxi call ($a = 1$) for each component. The hyperparameter of the truncated Poisson prior on the number of components of the non-homogeneous Bayesian models is set to $\lambda = 1$, meaning that a priori only one single component is expected ($K = 1$). Furthermore, all hyperparameters of the Dirichlet priors of the BAYES-MIX and the BAYES-HMM model are set to 1. That is, it was set $\boldsymbol{\alpha} = \mathbf{1}$ above Eq. (6.20) and $\boldsymbol{\alpha}_l = \mathbf{1}$ ($l = 1, \ldots, K$) in Eq. (6.22). In terms of equivalent samples sizes this can be interpreted as one pseudo count per mixture component (BAYES-MIX) or transition (BAYES-HMM), respectively. The two homogeneous models (FREQ-HOM and BAYES-HOM) as well as the frequentist changepoint model (FREQ-CPS) always output deterministic solutions. The EM-algorithm, which is used for inferring the FREQ-MIX and the FREQ-HMM model, can get stuck in local optima. Therefore, the EM algorithm is run 10 times independently for each data set with different randomly sampled initialisations of the Poisson parameters. $\epsilon = 0.001$ is used for the stop-criterion (see Tables 6.4-6.5 in the Appendix). For each K the output with the highest maximal likelihood value was selected, while the other EM algorithm outputs were withdrawn.[10] (The maximal likelihood value was typically reached several times, suggesting that running the EM algorithm 10 times is sufficient for the analysed data.) The non-homogeneous Bayesian models are inferred with MCMC simulations, and a pre-study was performed to determine the required number of MCMC iterations. This pre-study was based on eight data sets with $n = 16$, one from each of the 8 allocation scenarios shown in Table 6.6 of the Appendix. On each of these data sets 5 independent MCMC simulations with different allocation vector initialisations were performed. Trace-plot diagnostics of the quantity: log(Likelihood)+log(Prior), which is proportional to the log posterior probability, as well as scatter plots of the pairwise co-allocation probabilities, $\hat{p}(v_{t_1} = v_{t_2}|\mathbf{D})$ for $t_1, t_2 \in \{1, \ldots, T\}$, indicated that the following MCMC simulation setting is sufficient: The burn-in phase is set to 25,000 MCMC iterations, before $R = 250$ equidistant samples are taken from the subsequent 25,000 MCMC iterations (sampling phase).

---

[9] Note that the same number of validation samples ($\tilde{n} = 30$) is sampled for each n to ensure that the predictive probabilities $p(\tilde{\mathbf{D}}_{w,\tilde{n},u}|\mathbf{D}_{w,n,u})$ are comparable for different n.

[10] Note that the mixture weights and the transition probabilities were always initialised uniformly, i.e. $\pi_k = 1/K$ (FREQ-MIX) and $a_{i,j} = 1/K$ (FREQ-HMM).

6.5 Comparative Evaluation Study

First, the synthetic data from Section 6.3.1 are analysed with the eight models listed in Tables 6.1-6.2. The first finding is that the homogeneous models (FREQ-HOM and BAYES-HOM) yield substantially lower predictive probabilities than the non-homogeneous models for the non-homogeneous data. This is not unexpected, as the homogeneous models can per se not deal with non-homogeneity (e.g. changepoint-segmented data). For clarity of the plots, the results of the homogeneous models are therefore left out whenever their inclusion would have led to substantially different scales.

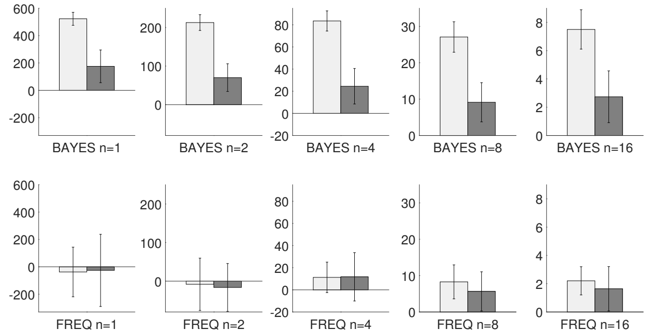
Figures 6.2-6.4 show histograms of the average log predictive probability differences with separate histograms for the Bayesian and the frequentist models. Here, the four models (HOM, CPS, MIX and HMM) are compared independently within the Bayesian and within the frequentist framework without comparing the two paradigms (Bayesian vs. frequentist). In each histogram the models being most consistent with the data (i.e. being most consistent with the data generation process), are used as 'reference' models.[11] In a complementary study the four Bayesian models and the four frequentist models are compared in a pairwise manner. In Figures 6.5-6.6 for each of the four models (HOM, CPS, MIX and HMM) the average log predictive probability differences ('Bayesian results minus frequentist results') are plotted against the average log predictive probability of the Bayesian and the frequentist results. The curves ('differences vs. means') are known as 'Tukey mean-difference' or 'Bland-Altman' plots, see [9] or [6].

(**Global trends, Figures 6.2-6.4**): Before studying the individual results in more detail, two global trends become obvious. Figures 6.2-6.4 show that the predictive probability differences between the non-homogeneous models get consistently lower as the number of samples n per time point t increases. The only exception appears for the mixture data (panels (b) in Figures 6.2-6.3), as the changepoint models (BAYES-CPS and FREQ-CPS) can per se not deal with mixture allocations, even when the sample size n is large. That is, for sufficiently informative data each non-homogeneous model can approximate all kinds of non-homogeneity, unless there is a clear mismatch between the dependency structure in the data and the inference model, as observed for the CPS models on mixture data. The second global finding from Figures 6.5-6.6 is that the pairwise differences between the Bayesian and the frequentist models consistently converge towards zero as the number of samples n increases. That is, asymptotically for all four models the Bayesian variant and the frequentist variant perform equally well for all data scenarios.

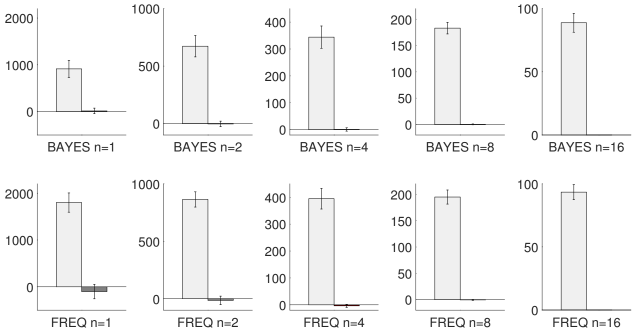
(**Homogeneous data, Figure 6.4**): The differences in the log predictive probabilities are relatively low, except for the frequentist changepoint model (FREQ-CPS). That is, for homogeneous data only FREQ-CPS overfits the data for low sample

---

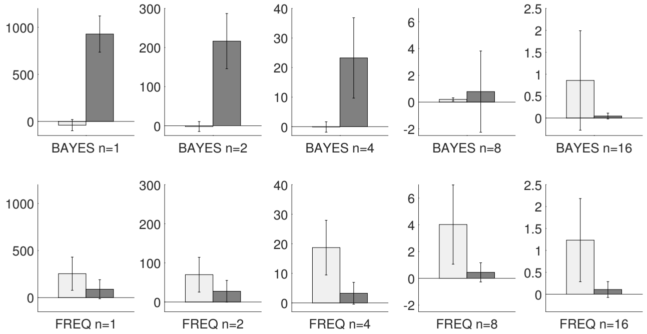
[11] For example the changepoint models (FREQ-CPS and BAYES-CPS) are used as references for the two changepoint-segmented data scenarios: $\mathbf{P}^\star = (\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m)$ and $\mathbf{P}^\star = (\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m, \mathbf{5}_m, \mathbf{6}_m)$.



(a) $\mathbf{P}^{\star} = (\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m)$. Left: **CPS-MIX**, right: **CPS-HMM**.

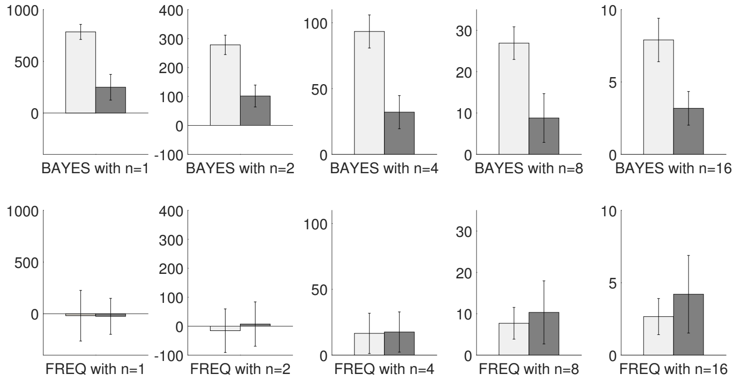


(b) $\mathbf{P}^{\star} = MIX(\mathbf{1}_m, \mathbf{5}_m)$. Left: **MIX-CPS**, right: **MIX-HMM**.

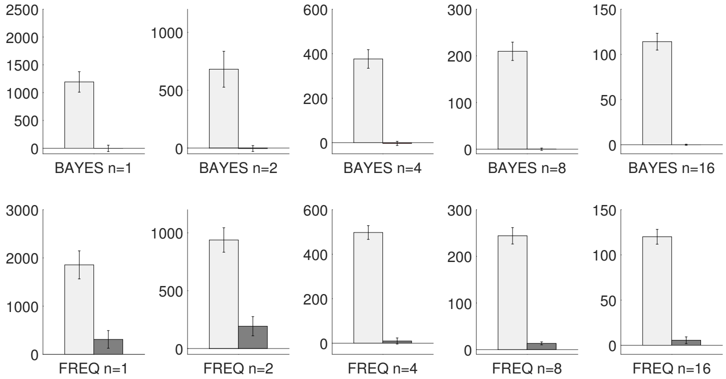


(c) $\mathbf{P}^{\star} = (\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$. Left: **HMM-CPS**, right: **HMM-MIX**.

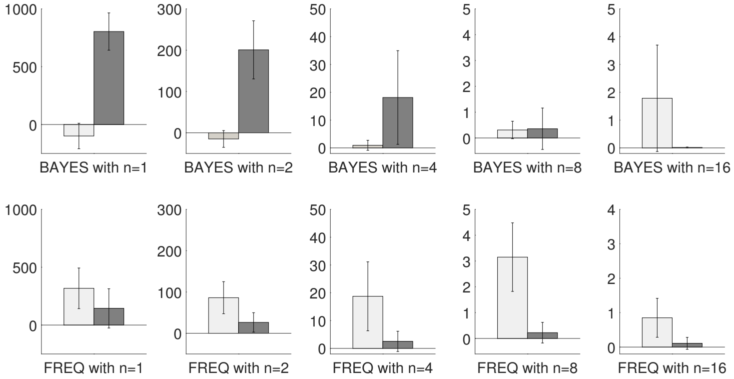
Figure 6.2: Cross-method comparison on synthetic data - Part 1/3. Panels (a-c) show histograms of the average log predictive probability differences for three non-homogeneous allocation scenarios with error bars representing standard deviations. In each panel (a-c) the upper row refers to the Bayesian models while the bottom row refers to the frequentist models. In each panel the differences between the reference (= most consistent with the data) model and the other two non-homogeneous models are shown. The homogeneous models led to substantially lower predictive probabilities and the results are therefore not shown. From left to right the sample size n increases and the scale of the y-axis changes.



(a) $\mathbf{P}^\star = (\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m, \mathbf{5}_m, \mathbf{6}_m)$. Left: **CPS-MIX**, right: **CPS-HMM**.

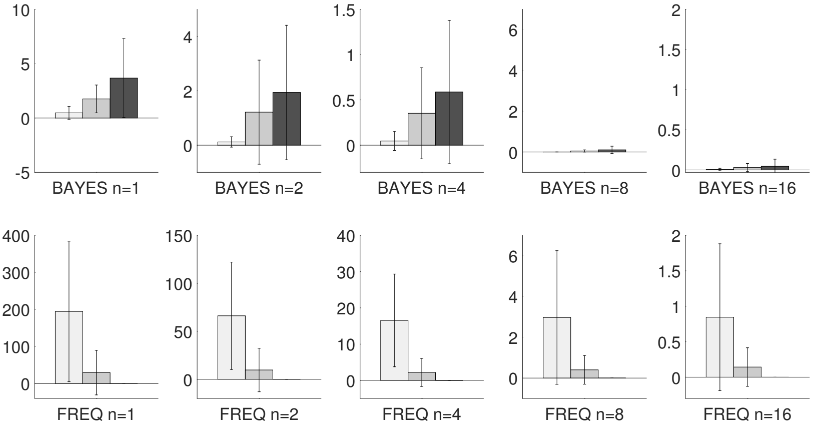


(b) $\mathbf{P}^\star = MIX(\mathbf{1}_m, \mathbf{2}_m, \mathbf{4}_m, \mathbf{8}_m)$. Left: **MIX-CPS**, right: **MIX-HMM**.

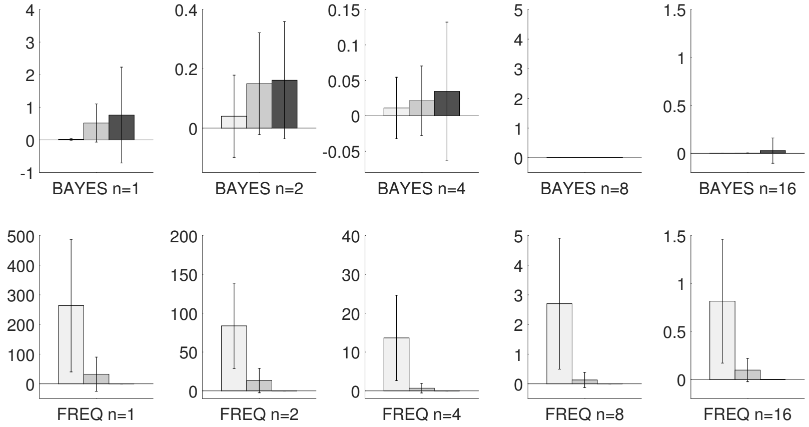


(c) $\mathbf{P}^\star = (\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$. Left: **HMM-CPS**, right: **HMM-MIX**.

Figure 6.3: Cross-method comparison on synthetic data - Part 2/3. See caption of Figure 6.2.



(a) $\mathbf{P}^{\star} = (\mathbf{1}_m)$. Left: **HOM-CPS**, centre: **HOM-MIX**, right: (**HOM-HMM**).



(b) $\mathbf{P}^{\star} = (\mathbf{5}_m)$. Left: **HOM-CPS**, centre: **HOM-MIX**, right: **HOM-HMM**.

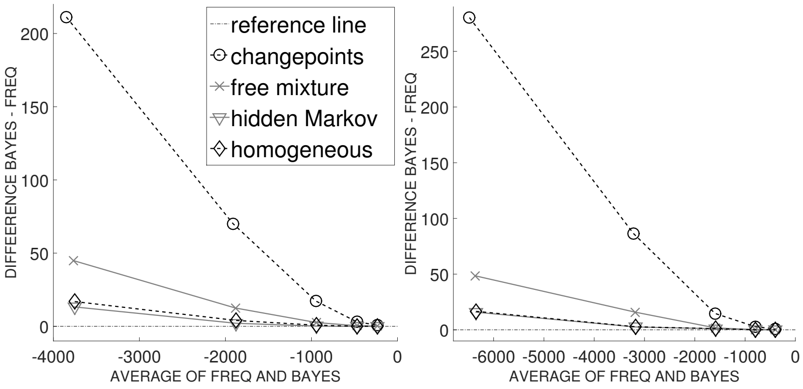
Figure 6.4: Cross-method comparison on synthetic data - Part 3/3. The two panels (a-b) show histograms of the average log predictive probability differences for the two homogeneous data scenarios. In both panels the upper (lower) row refers to the Bayesian (frequentist) models and the sample size n increases from left to right. In each panel the differences between the homogeneous (HOM) model and the three non-homogeneous models (CPS, MIX and HMM) are shown. Note that the FREQ-HMM model results never differed from the FREQ-HOM results and that the scales of the y-axis differ.

sizes n, while the other non-homogenous models are never inferior to the homogeneous reference models. A further analysis (results not shown) reveals that FREQ-CPS yields low predictive probabilities, as it tends to impose too many changepoints. For low sample sizes n, single columns (or coherent sequences of columns) can – by chance – have exceptional large values. Unlike the relatively robust Bayesian variant (BAYES-CPS), the frequentist changepoint model (FREQ-CPS) separates (or 'cuts out') those columns by setting two surrounding changepoints. The Bayesian changepoint variant appears to have a more effective penalty against over-fitting and does not allow for changepoints at neighbouring positions so that single columns cannot be 'cut out'.

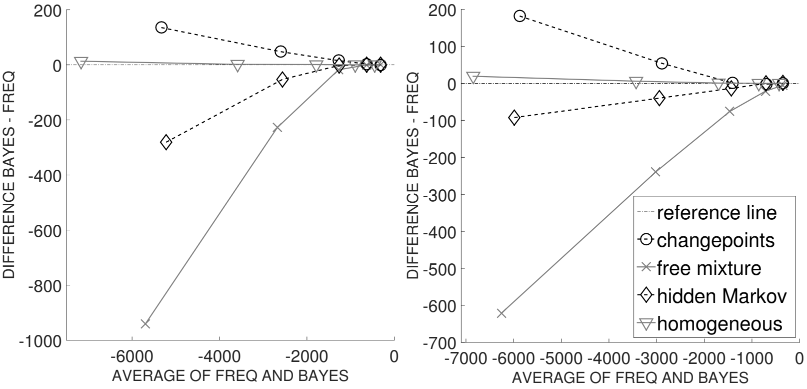
(**Changepoint-segmented data, panels (a) in Figures 6.2-6.3**): For all sample sizes n the Bayesian changepoint model (BAYES-CPS) performs significantly better than the Bayesian mixture model (BAYES-MIX) and the Bayesian hidden Markov model (BAYES-HMM). The differences to the reference model (BAYES-CPS) show that BAYES-MIX performs consistently worse than BAYES-HMM. The reason becomes obvious from Figure 6.8 in Section 6.6: BAYES-HMM approximates the underlying allocation better than BAYES-MIX, as BAYES-HMM – unlike BAYES-MIX – does not ignore the temporal order of the data points. For the frequentist models, the trend on changepoint-segmented data is slightly different: For small $n \leq 2$ there is no difference in the performance of the non-homogeneous models. Only for $n \geq 4$ the changepoint model (FREQ-CPS) performs better than its competitors. Thereby the mixture model (FREQ-MIX) performs better than the hidden Markov model (FREQ-HMM) for $n \geq 4$. Figure 6.9 in Section 6.6 suggests that this can be explained as follows: FREQ-MIX possesses fewer parameters than FREQ-HMM (see Table 6.1) so that its BIC-penalty is lower (see Figure 6.9). Consequently, FREQ-MIX can approximate the underlying segmentation better than FREQ-HMM. For low $n \leq 2$ there is no difference between FREQ-CPS and the other models, as the frequentist changepoint model (FREQ-CPS) tends to overfit the data, as discussed above (see homogeneous data) and demonstrated in Section 6.6 (see Figure 6.10).

(**Free-mixture data, panels (b) in Figures 6.2-6.3**): The Bayesian and the frequentist models show very similar trends. The changepoint models (CPS) are substantially outperformed by the free mixture reference models (MIX), while the hidden Markov models (HMM) are competitive to the mixture models (MIX). Only for small $n \leq 2$ FREQ-HMM appears to be slightly inferior to FREQ-MIX. Figure 6.9 in Section 6.6 suggests that this is due to the higher BIC-penalty of the FREQ-HMM model. However, for the scenario MIX($\mathbf{1}_m, \mathbf{5}_m$) and $n = 1$ the increased BIC-penalty turns out to be advantageous for FREQ-HMM. Unlike FREQ-HMM, FREQ-MIX tends to overfit the data with $n = 1$ by re-allocating outliers (columns with large values) to additional components.

(**Hidden-Markov data, panels (c) in Figures 6.2-6.3**): Among the Bayesian models, the mixture model (BAYES-MIX) is clearly outperformed by the hidden Markov model (BAYES-HMM) for low sample sizes $n \leq 4$. For larger sample sizes $n \geq 8$ the differences decrease. The Bayesian changepoint model (BAYES-CPS) is competitive to BAYES-HMM, as it approximates the underlying dependency



(a) **Homogeneous data**. Left: $\mathbf{P}^{\star} = (\mathbf{1}_m)$, right: $\mathbf{P}^{\star} = (\mathbf{5}_m)$.



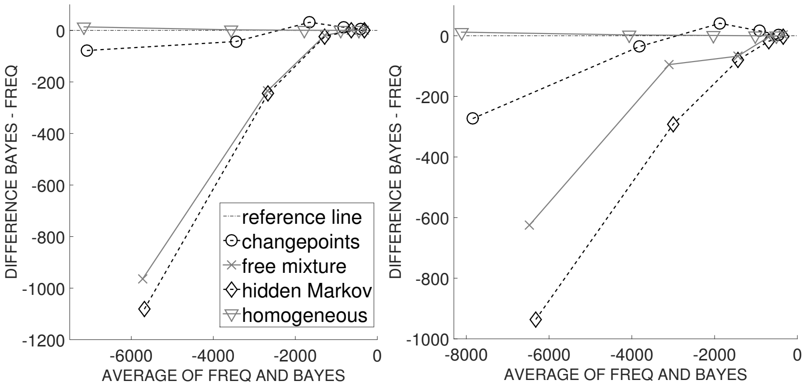
(b) **Changepoint data**. Left: $\mathbf{P}^{\star} = (\mathbf{1}_m, \cdots, \mathbf{4}_m)$, right: $\mathbf{P}^{\star} = (\mathbf{1}_m, \cdots, \mathbf{6}_m)$.

Figure 6.5: Tukey mean-difference plots to compare the performances of the frequentist and the Bayesian models on synthetic data - Part 1/2. The two panels show the average log predictive probability differences for the homogeneous data (a) and for the changepoint segmented data (b). In the four plots for each of the four models (HOM, CPS, MIX and HMM) the log predictive probability differences (Bayesian - frequentist) have been plotted against the average log predictive probabilities (of Bayesian and frequentist). The five symbols on each line correspond to the values obtained for the five sample sizes $n \in \{1, 2, 4, 8, 16\}$.

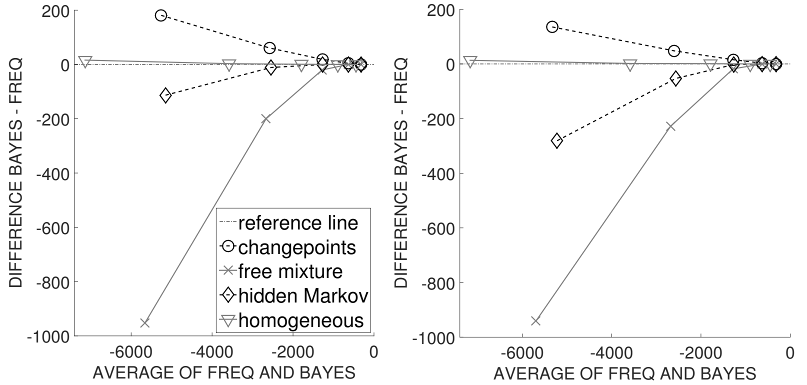
structure by additional changepoints; see Figure 6.8 in Section 6.6.[12] For the frequentist models a complementary trend can be observed: The changepoint model (FREQ-CPS) is consistently inferior to the reference model (FREQ-HMM), while the mixture model (FREQ-MIX) is competitive for all n. Again FREQ-

---

[12] Note that the selected Poisson means ($\theta_1 = 1$ and $\theta = 5$) yield components with very dissimilar values. This makes it easy for the changepoint model to distinguish them and to approximate the non-stationarity by setting an increased number of changepoints, e.g. 3 changepoints for $(\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$.



(a) **Mixture data**. Left: $\mathbf{P}^\star = \mathbf{MIX}(\mathbf{1}_m, \mathbf{5}_m)$, right: $\mathbf{P}^\star = \mathbf{MIX}(\mathbf{1}_m, \mathbf{2}_m, \mathbf{4}_m, \mathbf{8}_m)$.



(b) **Hidden Markov data**. Left: $\mathbf{P}^\star = (\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$, right: $\mathbf{P}^\star = (\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$.

Figure 6.6: Tukey mean-difference plots to compare the performances of the frequentist and the Bayesian models on synthetic data - Part 2/2. The two panels show the average log predictive probability differences plotted against the average log predictive probabilities for the mixture data (a) and for the hidden Markov model data (b); for further details see caption of Figure 6.5.

CPS tends to overfit the data (by cutting out columns with large realisations by surrounding changepoints), see Figure 6.8 in Section 6.6. The disadvantage of FREQ-MIX, to ignore the temporal order of the data points, appears to be compensated by its relatively low BIC-penalty (see Figure 6.9 in Section 6.6).

Bayesian versus frequentist: The Tukey-mean-difference plots of the pairwise predictive probability differences between the four Bayesian and the four frequentist models in Figures 6.5-6.6 show that both paradigms yield nearly identical results for large sample sizes ($n \geq 8$), while significant differences can be observed for small sample sizes n. Most remarkably are the following two trends:

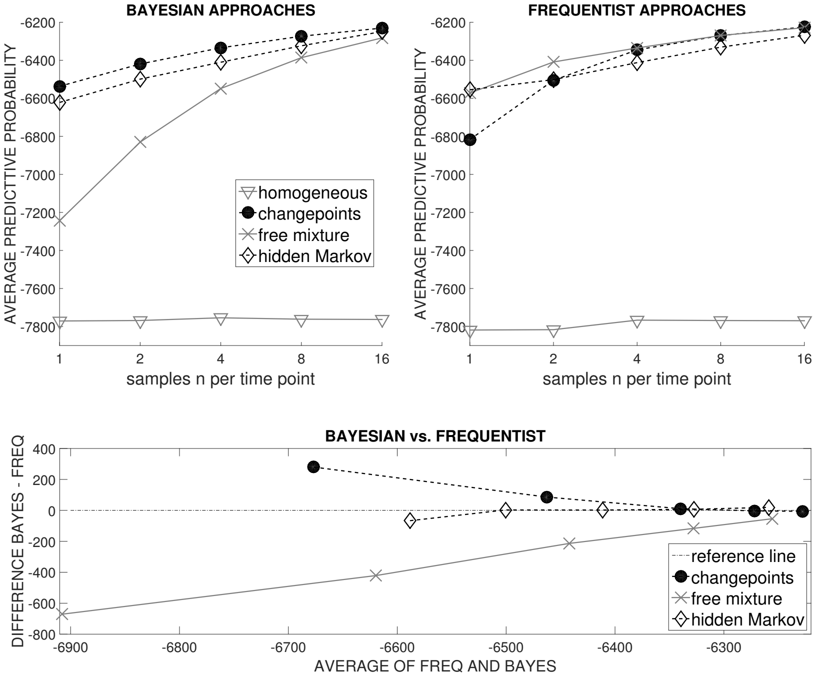


Figure 6.7: Results for the New York City Taxi data. In the upper plots the average log predictive probabilities (averaged across 35 data sets; i.e. 5 randomly sampled data instantiations per weekday) of the Bayesian models (upper left) and the frequentist models (upper right) have been plotted against the number of samples n per time point t. In the lower plot for each of the three non-homogeneous models (CPS, MIX and HMM) the average log predictive probability differences (BAYES-FREQ) have been plotted against the average log predictive probability of FREQ and BAYES. The five symbols on each line correspond to the values obtained for the sample sizes $n \in \{1, 2, 4, 8, 16\}$. In the lower plot the Bayesian (frequentist) model is superior when the curve/symbol is above (below) the reference line.

(i) Except for the mixture data (panel (a) in Figure 6.6), for low sample sizes n the Bayesian changepoint model (BAYES-CPS) is superior to the frequentist changepoint model (FREQ-CPS). (ii) Except for the homogeneous data (panel (a) in Figure 6.5), the frequentist hidden Markov model (FREQ-HMM) and especially the frequentist mixture model (FREQ-MIX) are superior to their Bayesian counterparts (BAYES-HMM and BAYES-MIX). The reason for the superiority of the Bayesian changepoint model (BAYES-CPS) is that the frequentist variant (FREQ-CPS) has a clear tendency towards over-fitting for uninformative data (for low n); see Figures 6.8 and 6.10 in Section 6.6 for more details. Unlike the Bayesian changepoint-model instantiation, FREQ-CPS infers only one single allocation vector (changepoint set) without any model-averaging. The low number of parameters of FREQ-CPS (see Table 6.1) yields a relatively low BIC-penalty.

Single columns of the data matrix, which by chance have larger values than the other columns, can be 'cut out' so that the FREQ-CPS model is very susceptible to over-fitting. On the other hand, the superiority of the frequentist mixture (FREQ-MIX) and the frequentist hidden Markov model (FREQ-HMM) over its Bayesian counterparts can be explained by the Multinomial-Dirichlet prior on the allocation vector. Both Bayesian models (BAYES-MIX and BAYES-HMM) employ Multinomial-Dirichlet priors for the allocation vectors, which can yield very strong prior penalties for non-homogeneous allocation vectors. As shown in Figure 6.9 in Section 6.6, BAYES-MIX is strongly penalized for all forms of non-homogeneity and BAYES-HMM is strongly penalized for mixture allocation vectors. This bottleneck of the Multinomial-Dirichlet prior for allocation vectors has already been analysed and discussed in [27] and renders the Bayesian model variants inappropriate for small samples sizes n, i.e. for uninformative data, where the effect of the likelihood is small compared to the effect of the Multinomial-Dirichlet prior.

The New York City Taxi (NYCT) data: The results are shown in Figure 6.7. The top plots shows the average log predictive probabilities for the Bayesian models (left) and the frequentist models (right) for different sample sizes n. The lower panel provides Tukey mean-difference plots to visualise the pairwise differences between the Bayesian and the frequentist models. The upper plots show that the homogeneous models (FREQ-HOM and BAYES-HOM) perform show the worst performance on the NYCT data. This is not unexpected, as Figure 6.1 shows that the Taxi pick-up data are clearly non-stationary. Among the Bayesian models, the changepoint-model (BAYES-CPS) performs best for all sample sizes n, and asymptotically (i.e. as n increases) the non-homogeneous Bayesian models perform equally well. Among the frequentist models the mixture model (FREQ-MIX) shows the best performance. For $n = 1$ FREQ-MIX and FREQ-HMM perform approximately equally well, while FREQ-CPS performs significantly worse. For $n = 2$ the FREQ-MIX model performs better than both competitors. For larger n ($n \geq 4$) FREQ-MIX and FREQ-CPS perform equally well, while FREQ-HMM performs slightly worse. The Tukey mean-difference plot in the bottom of Figure 6.7 shows that the Bayesian and frequentist models asymptotically perform equally well. For the lower samples sizes n the trends are consistent with the earlier observations for the synthetic data. The Bayesian changepoint model (BAYES-CPS) is superior to its frequentist counterpart (FREQ-CPS), while the opposite trend can be observed for the mixture and the hidden Markov model. The p-values of two-sided one-sample t-tests for the predictive probability differences between the best Bayesian model (BAYES-CPS) and the best frequentist model (FREQ-MIX) are computed to determine whether the performances differ significantly for any n. Given the relatively small t-test sample size of $n_d = 7$ weekdays[13], the five p-values (for $n = 1, 2, 4, 8, 16$) are higher than the standard level $\alpha = 0.05$, indicating that the best Bayesian and the best frequentist model are performing approximately equally well on the NYCT data.[14]

---

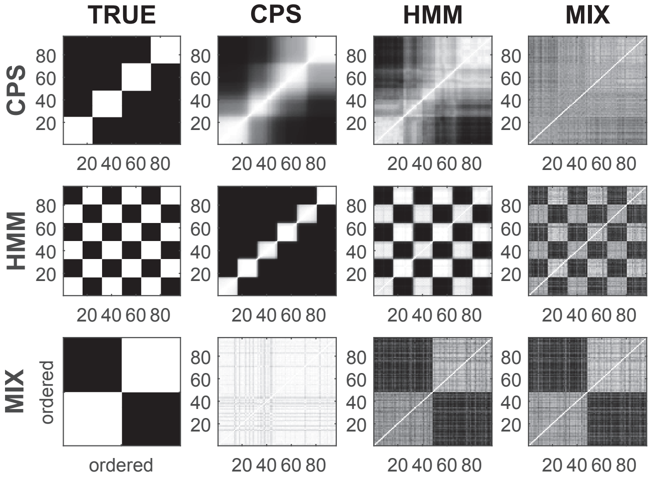
[13] That is one (average) predictive probability difference per weekday; the differences for the 5 data replicates per weekday are averaged, as they are very similar to each other.

6.6 Further model diagnostics

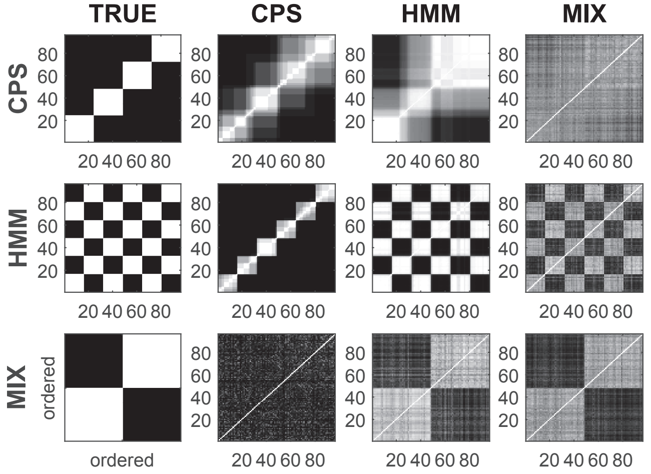
This section provides additional diagnostic plots for the synthetic data, analyzed in Section 6.5. The goal is to shed more light onto the relative merits and shortcomings of the models under comparison and to derive some conclusions of general validity. Since the predictive probabilities differed most significantly for sparse data, the focus of this section is on time series with only $n = 1$ observation per time point. The first analysis investigates to which extent the non-homogeneous models are capable of inferring the true underlying allocation vectors. To this end, for three of the allocation scenarios, namely $(\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m)$, $(\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$, and MIX$(\mathbf{1}_m, \mathbf{5}_m)$, the average probabilities, $p(v_s = v_t|\mathbf{D})$, that two time points s and t are allocated to the same component, are computed. The two panels of Figure 6.8 show heatmap representations of those connectivity probabilities for the Bayesian and for the frequentist model variants. The heatmaps show the following trends: (**1st rows, CPS data**): The MIX models fail to infer the true allocation; the time points are not sufficiently separated and the heatmaps appear unstructured. The HMM models perform better and their heatmaps show that the first time points and the last time points both build connected segments; only the centre changepoint is improperly inferred. The heatmaps of the CPS models are very similar to the true heatmap. Although the centre changepoint is a little bit diffuse, it can be seen that the data consist of 3-4 connected segments. (**2nd rows, HMM data**): The HMM models reconstruct the true allocation almost perfectly, while the CPS models segment the data into too many segments ($K = 6$). This is the reason why changepoint-based models are inappropriate for HMM data. As discussed earlier, CPS models cannot re-visit components once left so that HMM allocations are not in their allocation vector configuration spaces. CPS-models have to approximate HMM allocations by setting additional changepoints; this is the reason for their suboptimal performances on HMM data. The mixture models, in principle, infer the right trends. But it can be seen from the heatmaps that the separations between the components are weaker than those of the HMM models. (**3rd rows, MIX data**): The CPS models fail to infer the segmentation of the mixture data. The HMM and the MIX models perform approximately equally well and correctly divide the time points into $K = 2$ components, though the inferred separations appear to be slightly too weak. (**BAYESIAN vs. FREQ**): The heatmaps of the Bayesian and the frequentist model variants are very similar, except for the heatmaps of the changepoint-models on mixture data (see bottom rows, 2nd columns in Figure 6.8). While CPS-BAYES does not separate the time points, the frequentist counterpart (CPS-FREQ) shows the opposite behaviour: CPS-FREQ separates the time points into (too) many short segments.

---

[14] P-values: $p = 0.30$ ($n = 1$), 0.53 ($n = 2$), $p = 0.96$ ($n = 4$), $p = 0.45$ ($n = 8$), and $p = 0.72$ ($n = 16$).



(a) **Heatmaps of Bayesian model variants.**



(b) **Heatmaps of frequentist model variants.**

Figure 6.8: Heatmap representations of the inferred connectivity structures for the non-homogeneous models. Panel (a) refers to the Bayesian models, panel (b) to the frequentist models. Both panels are arranged as 3-by-4 matrices with rows corresponding to the true allocation vectors: CPS: $(\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m)$ (top), HMM: $(\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$ (centre), and MIX: MIX$(\mathbf{1}_m, \mathbf{5}_m)$ (bottom). The first columns show the true connectivity structures, and columns 2-4 correspond to the three non-homogeneous models: CPS, HMM, and MIX. The heatmaps give the inferred probabilities $p(v_s = v_t|\mathbf{D})$ of two points s and t belonging to the same component. The probabilities are represented by a grey shading, where white corresponds to 1, and black corresponds to 0. The axes refer to the $T = 96$ time points. All connectivity probabilities $p(v_s = v_t)$ are averaged over 25 data instantiations with $n = 1$ observation per time point. The time points of the MIX data in the last rows have been ordered w.r.t. the two mixture components.

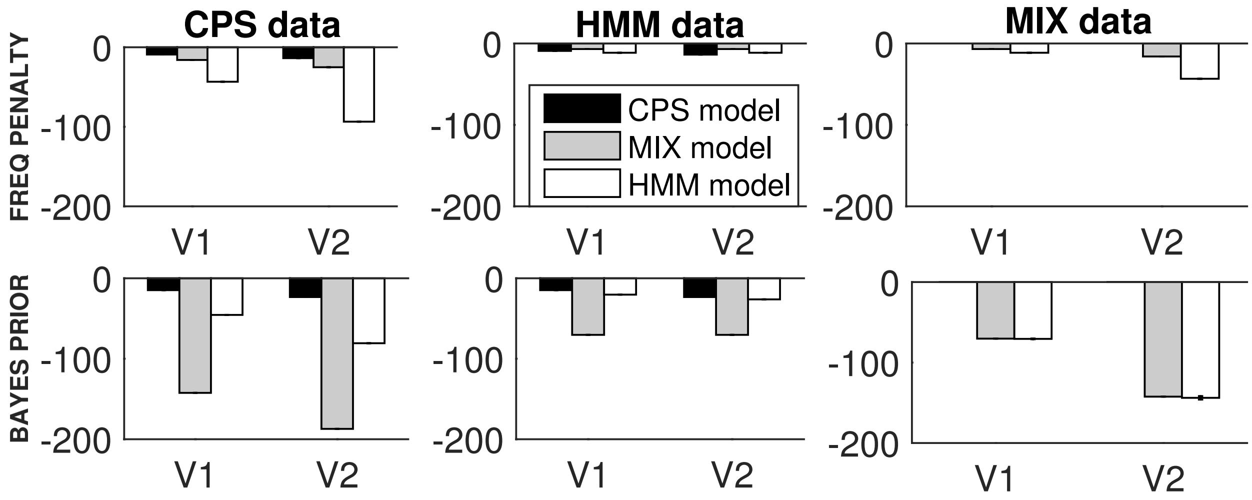


Figure 6.9: Comparison of penalty terms for the non-homogeneous models. The plot is arranged as a 2-by-3 matrix, and the rows refer to the frequentist (top) and the Bayesian (bottom) models. The columns refer to three different allocation scenarios (CPS data, HMM data, and MIX data) and for each scenario two variants (V1 and V2) are distinguished. The segmentation schemes correspond to those used in the comparative evaluation study in Section 6, see Table 2 in the Appendix for an overview. The bars give the penalties (BIC or prior probability) of the three models (CPS, MIX and HMM) for the true underlying allocation. As the CPS models cannot infer the true allocation of mixture data, the bars are not shown. For the CPS models it is assumed that they approximate HMM data by additional changepoints, e.g. (HMM, V1): $(\mathbf{1}_m, \mathbf{5}_m, \mathbf{1}_m, \mathbf{5}_m)$ is approximated by setting 3 changepoints.

Overall, the findings are in agreement with the results from Section 6.5: The heatmaps in Figure 6.8 confirm that the HMM models can properly infer HMM data and mixture data while their performances on changepoint-segmented data are suboptimal (the changepoints are not properly inferred). The CPS models and the MIX models completely fail for certain allocation scenarios: The CPS model cannot deal with mixture data and can only approximate the segmentation of HMM data by setting too many changepoints. The mixture models, which do not exploit the temporal order of the data points, are inappropriate for changepoint-segmented data. Another finding is the difference between the two changepoint models: CPS-BAYES and CPS-FREQ. For mixture data CPS-BAYES infers 'undercomplex' allocations with too few changepoints (mostly $K = 1$), while CPS-FREQ infers 'overcomplex' allocations with too many changepoints (even the maximum of $K = 10$ changepoints is reached). The latter finding suggests that the frequentist changepoint model has a tendency towards overfitting the data by setting too many changepoints.

A comparison of the penalty terms for the true allocations is given in Figure 6.9. The top row of Figure 6.9 shows the BIC-penalties of the frequentist models, the bottom row shows the Bayesian (log) prior probability penalties. (**CPS data and HMM data**): The penalties of the frequentist and the Bayesian models are comparable except for the MIX model. For both types of data the penalties of the Bayesian mixture model are substantially higher than the penalties of its frequentist counterpart. (**MIX data**): The CPS model cannot infer the true mixture allocations. The Bayesian HMM model and the Bayesian MIX model are penalized significantly stronger than their frequentist counterparts. (**MIX-BAYES vs. HMM-BAYES**): The bottom row of Figure 6.9 shows that the penalties of HMM-BAYES and MIX-BAYES are nearly identical for mixture data, while MIX-BAYES has substantially higher penalties for CPS data and MIX data. That is, the Bayesian mixture model (MIX-BAYES) is 'over-penalized' for non-mixture data. The last diagnostic compares the performances of the Bayesian and the frequentist models with respect to over-fitting issues. To this end, certain 'features' of the 'best' inferred models (FREQ-models minimising the BIC score; BAYES-models with the highest posterior score) are compared with the corresponding 'features' of the true models, i.e. models which are based on the true allocation vectors. The 'features' are: (i) the scores, (ii) the (marginal) likelihood values, (iii) the prior penalty terms, and (iv) the predictive probabilities for new data. Figure 6.10 gives scatter plots in which the features of the best inferred models are plotted against the features of the true models. The scatter plots can be interpreted as follows: Symbols are above (below) the diagonal when the feature of the best inferred model is higher (lower) than the feature of the true model. (**Figure 6.10(a), changepoint-divided data**): The best inferred models yield higher scores and higher likelihoods than the true models. That is, both models variants (BAYES and FREQ) fit the data better than the true models. But the scatter plot of the penalty terms show that the Bayesian CPS model consistently infers 'under-penalized' models (i.e. models with too few changepoints) while the frequentist CPS model also infers 'over-penalized' models (i.e. models with too many changepoints). The scatter plot of the predictive probabilities shows the implication. The inferred models are inferior to the true models (all symbols are below the diagonal), but the predictive probabilities of the 'under-complex' CPS-BAYES model are better than those of the 'over-complex' CPS-FREQ model. Thereby the most 'over-complex' CPS-FREQ models yield the lowest predictive probabilities. This clearly shows that the frequentist changepoint model is more susceptible to over-fitting than its Bayesian counterpart. (**Figure 6.10(b), mixture data**): The upper panels show that MIX-FREQ again consistently overfits the data, while the MIX-BAYES model sometimes yields lower likelihoods than the true model (see diamond symbols in the upper right panel). The scatter plot of the penalty term shows that MIX-BAYES infers 'undercomplex' models (with too few mixture components) for scenario (V2, mixture with $K = 4$ components) and sometimes for scenario (V1, mixture with $K = 2$ components). This suggests that MIX-BAYES overpenalizes the complexity of the allocation vectors, so that 'undercomplex' models are inferred. This explains why MIX-FREQ is superior to the over-penalized (and thus 'undercomplex') MIX-BAYES model (see scatter plot of the predictive probabilities). Figure 6.11 in the Appendix shows the same diagnostics for time series with $n = 16$ data points per time point. The results show that those issues of under- and over-penalisation diminish/disappear as the data get more informative.

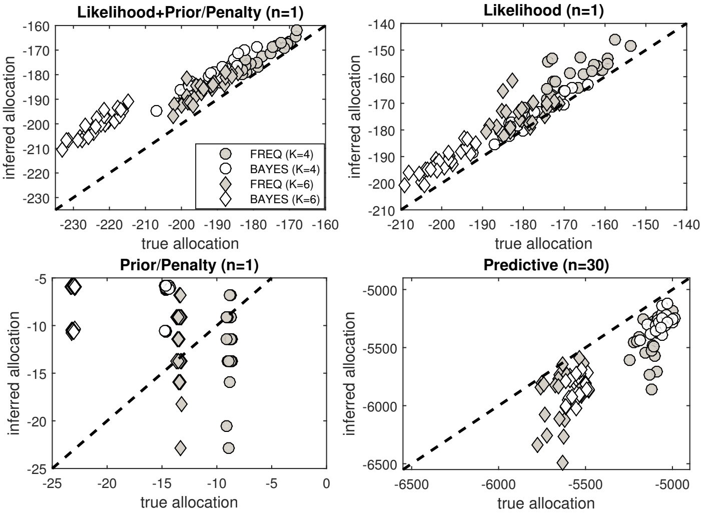
Summary: The additional diagnostics, shown in Figures 6.8-6.10, confirm four of the empirical findings from Section 6.5.

(1) In Section 6.5 the CPS-models showed suboptimal performances for HMM and MIX data. Figure 6.8 shows that CPS models are inferior to HMM models for both types of data because the true allocations are not part of their allocation vector configuration spaces. Only for changepoint segmented data, where the HMM models do not properly infer the true allocation, the CPS models are superior to the HMM models.

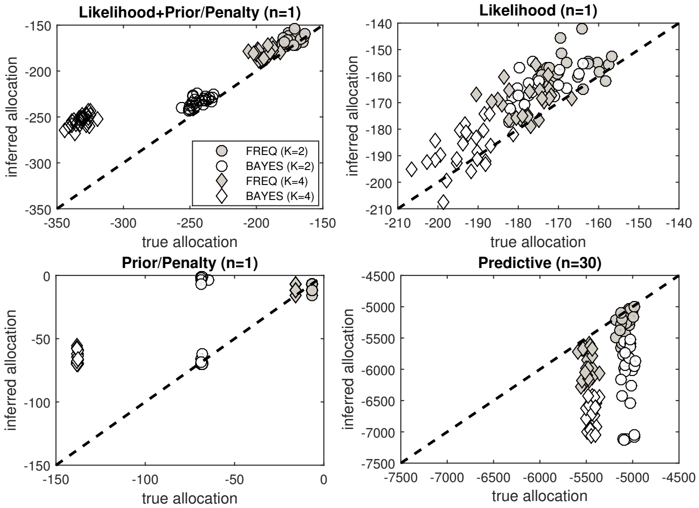
(2) In Section 6.5 the MIX-models showed suboptimal performances for CPS and HMM data. Figure 6.8 shows that the MIX models cannot properly infer CPS and HMM allocations, while the HMM models show moderate performances for all types of data.

(3) In Section 6.5 the Bayesian mixture model (MIX-BAYES) was found to be inferior to its frequentist counterpart (MIX-FREQ). As seen from Figures 6.9-6.10, the Bayesian mixture variant is over-penalised. This renders the frequentist HMM model preferable to the Bayesian mixture model.

(4) In Section 6.5 it was also found that the Bayesian CPS model is superior to its frequentist counterpart (CPS-FREQ). As seen from Figure 6.10(a), the frequentist changepoint model tends to overfit the data. This renders the Bayesian changepoint model preferable to the frequentist CPS model.



(a) **Changepoint models on changepoint data with n=1.**



(b) **Mixture models on mixture data with n=1.**

Figure 6.10: Comparing the best scoring models with the 'true' models. Panel (a) shows diagnostics for the changepoint models (CPS-FREQ and CPS-BAYES) on changepoint-segmented data: $(\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m)$ (circles) and $(\mathbf{1}_m, \mathbf{2}_m, \mathbf{3}_m, \mathbf{4}_m, \mathbf{5}_m, \mathbf{6}_m)$ (diamonds). Panel (b) shows diagnostics for the mixture models (MIX-FREQ and MIX-BAYES) on mixture data: MIX$(\mathbf{1}_m, \mathbf{5}_m)$ (circles) and MIX$(\mathbf{1}_m, \mathbf{2}_m, \mathbf{4}_m, \mathbf{8}_m)$ (diamonds). In both panels there are 4 scatter plots, in which features of the 'best' models (BAYES: highest posterior, FREQ: best BIC) are plotted against the corresponding features of the 'true' models, using the true allocations. Symbols that refer to Bayesian models are white-colored, the frequentist symbols are grey-colored. **Upper left**: scores vs. scores (BAYES: log(likelihood+prior), FREQ: BIC value); **upper right**: log-likelihood vs. log-likelihood; **lower left** penalty vs. penalty (BAYES: log(prior), FREQ: BIC-penalty); and **lower right**: predictive probability vs. predictive probability. Compare to Figure 6.11 of Appendix.

6.7 Discussion and conclusions

In this chapter the results of a comparative evaluation study on eight (non-) homogeneous models for (Poisson) count data were presented. The study was performed on various synthetic data sets and on taxi pick-up counts, extracted from the recently published New York City Taxi (NYCT) database, described in Section 6.3. For the study the standard homogeneous Poisson model (HOM) and three non-homogeneous Poisson models, namely a changepoint model (CPS), a free mixture model (MIX) and a hidden Markov model (HMM), were implemented following the frequentist paradigm (FREQ) and the Bayesian paradigm (BAYES); see Tables 6.1-6.2 in Section 6.1 for an overview. The empirical findings from Section 6.5-6.6 suggest the following conclusions:
Asymptotically, i.e. for sufficiently informative data (here: quantified in terms of the sample size n per time point t), there is no difference between the paradigms. The Bayesian and the frequentist models perform equally well. For less informative data (here: for small n) there are significant differences, as described in more detail below. While the homogeneous model variants (FREQ-HOM and BAYES-HOM) cannot deal with non-homogeneity, the non-homogeneous models, except for the frequentist changepoint model (FREQ-CPS), do not overfit homogeneous data. Thus, it can be recommended applying non-homogeneous approaches, even if the data might be homogeneous. Moreover, if the data is informative enough, in both frameworks (Bayesian and frequentist) all three non-homogeneous models can approximate all kinds of non-homogeneity, unless there is a clear mismatch between the model and the underlying data. E.g. in Section 6.5-6.6 it was found that the changepoint models (FREQ-CPS and BAYES-CPS) perform badly for mixture data. The hidden Markov models (FREQ-HMM and BAYES-HMM) appear to be superior to the mixture models (FREQ-MIX and BAYES-MIX), since they are competitive on free-mixture data, and superior on hidden Markov and changepoint-segmented data.[15] In a pairwise comparison of the four Bayesian and the four frequentist models it was found for less informative data (here: small n) that the Bayesian changepoint model (BAYES-CPS) is superior to its frequentist counterpart (FREQ-CPS), while the opposite trend could be observed for the mixture model and the hidden Markov model. The superiority of the Bayesian changepoint model (BAYES-CPS) over the frequentist variant (FREQ-CPS) is due to the fact the the frequentist model variant is very susceptible to over-fitting (see Figure 6.10(a) in Section 6.6). The inferiority of the Bayesian free mixture model (BAYES-MIX) and the Bayesian hidden Markov (BAYES-HMM) model to their frequentist counterparts is caused by the allocation vector priors. Both Bayesian models employ the Multinomial-Dirichlet prior, which is known to impose a very strong penalty on non-homogeneous allocations (see Figure 6.9 in Section 6.6), rendering the Bayesian variants inappropriate for less informative data sets (here: for small samples sizes n) than the two frequentist variants

---

[15] Though still worse than the changepoint models (FREQ-CPS and BAYES-CPS), which can be seen as reference models for changepoint-segmented data.

FREQ-MIX and FREQ-HMM (see, e.g., Figure 6.10(b)).

For the real-world New York City Taxi (NYCT) data very similar trends could be observed. For sufficiently informative data (here: for large n) all non-homogeneous models led to approximately identical results, and for uninformative data (here: for small n) it was found that the Bayesian changepoint model (BAYES-CPS) performs better than its frequentist counterpart, while the frequentist mixture model (FREQ-MIX) performs better than its Bayesian counterpart (see Figure 6.7). It was also found that the performances of the best Bayesian model (BAYES-CPS) and the best frequentist model (FREQ-MIX) do not differ significantly for any n. Finally, it should be noted that potential 'overdispersion' problems (i.e. potential violations of the Poisson model assumption) were not taken into account within the presented study. Unlike for the synthetic data, where all data points were actually sampled from Poisson distributions so that over-dispersion problems could not arise, overdispersion could have been present for the real-world NYCT data application. Therefore, the (undispersed) models, considered here, might have been suboptimal for the NYCT data and better results could perhaps have been obtained by taking the potential over-dispersion properly into account; e.g. by replacing the Poisson distribution by the more flexible negative binomial distribution or by applying more advanced Poisson model approaches.

6.8 Appendix

6.8.1 Details on the segment neighbourhood search algorithm for the changepoint Poisson model

Let $\mathbf{D}^{[s:t]}$ define the data subset containing the data points for the successive time points from s to t. In a first step for all pairs of time points $s, t \in \{1, \dots, T\}$ with $s \leq t$ a cost function $\Psi(\mathbf{D}^{[s:t]})_1$ has to be pre-computed, where the subscript '1' indicates that the segment from s to t builds one component. (That is, there is no changepoint in between s and t, and only the time points from s to t belong to this component.) As the goal is to maximise Eq. (12), while the segment neighbourhood search algorithm is usually formulated such that it minimises a cost function, the negative of the contribution of the data segment $\mathbf{D}^{[s:t]}$ to the log-likelihood can be used as cost function:

$$\Psi(\mathbf{D}^{[s:t]})_1 := (-1) \cdot \log\{p(\mathbf{D}^{[s:t]}|\hat{\theta}_{[s:t]})\}$$

where $\hat{\theta}_{[s:t]}$ is the ML estimator for the Poisson parameter of the segment $\mathbf{D}^{[s:t]}$, which can be computed with Eq. (6), while $p(\mathbf{D}^{[s:t]}|\hat{\theta}_{[s:t]})$ can be computed using Eq. (2). To determine the best changepoint sets $C^K := \{c_1^K, \dots, c_{K-1}^K\}$ subdividing the data into K subsets for each $K \in \{2, \dots, K_{MAX}\}$, the recursive SNS algorithm proceeds as outlined in the pseudo code, provided in Table 6.3.

```
• Pre-Computation: For all $s, t \in \{1, \dots, T\}$ with $s \leq t$ pre-compute the cost function
  $\Psi(\mathbf{D}^{[s:t]})_1 := (-1) \cdot \log\{p(\mathbf{D}^{[s:t]}|\hat{\theta}_{[s:t]})\}$ for the segment $s, \dots, t$.
• Start the SNS algorithm: For $K = 2, \dots, K_{MAX}$:
    – For $s = 2, \dots, T$
      compute $\Psi(\mathbf{D}^{[1:s]})_K = \min_{\nu \in \{1,\dots,s-1\}}\{\Psi(\mathbf{D}^{[1:\nu]})_{K-1} + \Psi(\mathbf{D}^{[(\nu+1):s]})_1\}$.
    – Set: $c_1^K := argmin_{\nu \in \{1,\dots,T-1\}}\{\Psi(\mathbf{D}^{[1:\nu]})_{K-1} + \Psi(\mathbf{D}^{[(\nu+1):T]})_1\}$.
    – If $K \geq 3$ continue as follows:
      For $i = 2, \dots, K-1$:
      Set $c_i^K := argmin_{\nu \in \{K-2,\dots,c_{i-1}^K-1\}}\{\Psi(\mathbf{D}^{[1:\nu]})_{K-i-1} + \Psi(\mathbf{D}^{[(\nu+1):c_{i-1}^K]})_1\}$.
  Output: For $K = 2, \dots, K_{MAX}$ output the best changepoint set with $K-1$ changepoints
  $C^K := \{c_1^K, \dots, c_{K-1}^K)\}$ that minimises Eq. (12) of this chapter. For this changepoint
  set the model fit, quantified by Eq. (12) of this chapter, is $l(\hat{\boldsymbol{\theta}}_{C^K}|\mathbf{V}_{C^K}, \mathbf{D}) = (-1) \cdot \Psi(\mathbf{D}^{[1:T]})_K$.
```

Table 6.3: Pseudo-code for the segment neighbourhood search algorithm. The algorithm infers the best changepoint segmentations with $K = 1, \dots, (K_{MAX} - 1)$ changepoints and was proposed by [4]; see the chapter for further details.

```
For each $K = 2, \dots, K_{MAX}$ perform the EM algorithm for a Poisson finite mixture model with $K$ components:
  • Initialise the Poisson parameter vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)^\mathsf{T}$ and the mixture weights vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\mathsf{T}$ with $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\pi}^{(1)}$.
  • Inner loop: For $i = 1, 2, 3, \dots$
    - E-step: Compute $\mathbb{E}[l(\boldsymbol{\theta}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \boldsymbol{\theta}^{(i)}, \boldsymbol{\pi}^{(i)}]$, defined in Eq. (6.32).
    - M-step: Maximsise $\mathbb{E}[l(\boldsymbol{\theta}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \boldsymbol{\theta}^{(i)}, \boldsymbol{\pi}^{(i)}]$ in $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. This gives the updated parameter vectors $\boldsymbol{\pi}^{(i+1)}$ and $\boldsymbol{\theta}^{(i+1)}$, whose elements were defined in Eqns. (6.34-6.35).
    - Check the stop criterion:
      If $|\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^{(i)}|_2 := \sqrt{\sum_{k=1}^{K} (\theta_k^{(i+1)} - \theta_k^{(i)})^2} < \epsilon$, where $\epsilon$ is a small pre-defined value, stop the iterations and output the parameter vectors $\boldsymbol{\pi}^K := \boldsymbol{\pi}^{(i+1)}$ and $\boldsymbol{\theta}^K := \boldsymbol{\theta}^{(i+1)}$ as well as $\hat{\Delta}_{k,t}^K := \hat{\Delta}_{k,t}^{(i+1)}$, where the expected values of the allocation variables can be computed with Eq. (6.33).
      Else perform the next iteration, i.e. increment $i$.
```

Table 6.4: Pseudo-code for the EM algorithm for the Poisson finite mixture model. An algorithm to infer the ML-estimators of the Poisson mixture models with $K = 2, \dots, K_{MAX}$ mixture components, see the chapter for further details.

6.8.2 Details on the EM algorithm for the finite mixture Poisson model

First assume that the allocation vector $\mathbf{V} = (v_1, \dots, v_T)^\mathsf{T}$ was known, and re-write the log-likelihood given in Eq. (5) of this chapter as follows:

$$l(\boldsymbol{\theta}|\mathbf{V}, \mathbf{D}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \Delta_{k,t} \cdot \log\{p(\mathbf{D}_{.,t}|\theta_k)\} \tag{6.31}$$

where $\Delta_{k,t}$ is a indicator variable which indicates whether time point t is allocated to component k or not, i.e. $\Delta_k(t) = 1$ if $v_t = k$ and $\Delta_k(t) = 0$ if $v_t \neq k$. The EM algorithm was introduced by [12] to maximise (log-)likelihoods for incomplete data, i.e. data with missing values. Here, the allocation vector is unknown ('missing'), i.e. the values of the indicator variables $\Delta_{k,t}$ are missing, and the goal is to maximise Eq. (6.31) in $\boldsymbol{\theta}$. The EM algorithm proceeds as follows: After initialisation of the parameters, e.g. by $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\pi}^{(1)}$, the Expectation (E) and the Maximisation (M) steps are performed iteratively, until convergence is reached. The E-step calculates the expectation of Eq. (6.31) given the current instantiation of the parameters $\boldsymbol{\theta}^{(i)}$ and $\boldsymbol{\pi}^{(i)}$ and conditional on the observed data $\mathbf{D}$, before the M-step maximises the expectation of Eq. (6.31) w.r.t. the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$ to obtain new parameter instantiations $\boldsymbol{\theta}^{(i+1)}$ and $\boldsymbol{\pi}^{(i+1)}$.

Since the E-step computes the expectation of Eq. (6.31) conditional on the data $\mathbf{D}$ and the current parameter vectors $\boldsymbol{\theta}^{(i)}$ and $\boldsymbol{\pi}^{(i)}$, symbolically

$$\mathbb{E}[l(\boldsymbol{\theta}_{\mathbf{V}}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \boldsymbol{\theta}^{(i)}, \boldsymbol{\pi}^{(i)}],$$

the expectation is only taken w.r.t. the indicator variables $\Delta_{k,t}$. This yields:

$$\mathbb{E}[l(\boldsymbol{\theta}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \boldsymbol{\theta}^{(i)}, \boldsymbol{\pi}^{(i)}] = \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbb{E}[\Delta_{k,t}|\mathbf{D}, \boldsymbol{\theta}^{(i)}, \boldsymbol{\pi}^{(i)}] \cdot \log\{p(\mathbf{D}_{.,t}|\theta_k)\} \tag{6.32}$$

and it can be easily shown that

$$\mathbb{E}[\Delta_{k,t}|\mathbf{D},\boldsymbol{\theta}_{\mathbf{V}}^{(i)}] = \frac{\pi_k^{(i)} \cdot p(\mathbf{D}_{.,t}|\theta_k^{(i)})}{\sum_{l=1}^{K} \pi_l^{(i)} \cdot p(\mathbf{D}_{.,t}|\theta_l^{(i)})} =: \hat{\Delta}_{k,t}^{(i+1)} \tag{6.33}$$

where $p(\mathbf{D}_{.,t}|\theta_k^{(i)})$ can be computed with Eq. (3) of this chapter.

Subsequently, the M-step maximises Eq. (6.32) in the parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$. This yields parameter updates: $\boldsymbol{\theta}^{(i+1)} = (\theta_1^{(i+1)}, \dots, \theta_K^{(i+1)})^\mathsf{T}$ and $\boldsymbol{\pi}^{(i+1)} = (\pi_1^{(i+1)}, \dots, \pi_K^{(i+1)})^\mathsf{T}$, where

$$\pi_k^{(i+1)} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\Delta_k(t)|\mathbf{D},\boldsymbol{\theta}_{\mathbf{V}}^{(i)}] = \frac{1}{T}\sum_{t=1}^{T} \hat{\Delta}_{k,t}^{(i+1)} \tag{6.34}$$

is the expected fraction of time points being allocated to component k, and

$$\theta_k^{(i+1)} = \frac{\sum_{t=1}^{T}\{\sum_{j=1}^{n} d_{.,t}\} \cdot p(\mathbf{D}_{.,t}|\theta_k^{(i)})}{\sum_{t=1}^{T} \cdot p(\mathbf{D}_{.,t}|\theta_k^{(i)})} \tag{6.35}$$

where $\sum_{j=1}^{n} d_{.,t}$ is the sum of the elements in the t-th row of the data matrix $\mathbf{D}$. (Note that the mixture model allocates the T columns of $\mathbf{D}$, symbolically $\mathbf{D}_{.,1}, \dots, \mathbf{D}_{.,T}$, to K mixture components and that each column $\mathbf{D}_{.,t}$ contains realisations $d_{1,t}, \dots, d_{n,t}$ of n iid Poisson variables.)
Iterating the E- and the M-step until convergence is reached, yields the ML estimators. As the algoritm can be get stuck in local optima, it should be run several times with different initialisations $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\pi}^{(1)}$. Pseudo code for computing the ML-estimators for each $K = 2, \dots, K_{MAX}$ is provided in Table 6.4.

```
For each $K = 2, \dots, K_{MAX}$ perform the EM algorithm for a Poisson hidden Markov model
with $K$ states:
  • Initialise the Poisson parameter vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_K)^\mathsf{T}$, the transition matrix $\mathbf{A}$, and
    the initial distribution $\Pi^{(1)} = (\pi_1, \dots, \pi_K)^\mathsf{T}$ with $\boldsymbol{\theta}^{(1)}$, $\mathbf{A}^{(1)}$, and $\Pi^{(1)}$.
  • Inner loop: For $i = 1, 2, 3, \dots$
    - E-step: Compute $\mathbb{E}[l(\Pi, \mathbf{A}, \boldsymbol{\theta}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \Pi^{(i)}, \mathbf{A}^{(i)}, \boldsymbol{\theta}^{(i)}]$, defined in Eq. (6.40).
    - M-step: Maximise $\mathbb{E}[l(\Pi, \mathbf{A}, \boldsymbol{\theta}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \Pi^{(i)}, \mathbf{A}^{(i)}, \boldsymbol{\theta}^{(i)}]$ in $\boldsymbol{\theta}$, $\mathbf{A}$ and $\Pi$. This gives the
      updated parameter vectors $\boldsymbol{\theta}^{(i+1)}$, $\mathbf{A}^{(i+1)}$ and $\Pi^{(i+1)}$, whose elements were defined in
      Eqns. (6.42-6.43).
    - Check the stop criterion:
      If $|\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^{(i)}|_2 := \sqrt{\sum_{k=1}^{K}(\theta_k^{(i+1)} - \theta_k^{(i)})^2} < \epsilon$, where $\epsilon$ is a small pre-defined
      value, stop the iterations and output the vectors $\Pi^K := \Pi^{(i+1)}$, $\mathbf{A}^K = \mathbf{A}^{(i+1)}$, and
      $\boldsymbol{\theta}^K := \boldsymbol{\theta}^{(i+1)}$ as well as $\hat{\Delta}_{k,t}^K := \hat{\Delta}_{k,t}^{(i+1)}$, which can be computed with Eq. (6.41).
      Else perform the next iteration, i.e. increment $i$.
```

Table 6.5: Pseudo-code for the EM algorithm for the Poisson Hidden Markov model. An algorithm to infer the ML-estimators of the Poisson hidden Markov models with $K = 2, \dots, K_{MAX}$ states (components), see the chapter for further details.

6.8.3 Details on the Hidden Markov model inference

Given the parameters Π, $\mathbf{A}$ and $\boldsymbol{\theta}$ define for $t = 1, \dots, T$ the forward probability $\alpha_t(k)$ to be the joint pdf of the data sequence $\mathbf{D}_{.,1}, \dots, \mathbf{D}_{.,t}$ and state k at time point t ($v_t = k$):

$$\alpha_t(k) := P(\mathbf{D}_{.,1}, \dots, \mathbf{D}_{.,t}, v_t = k|\Pi, \mathbf{A}, \boldsymbol{\theta})$$

For $t = 1$ it holds: $\alpha_1(k) = P(\mathbf{D}_{.,1}, v_1 = k|\Pi, \mathbf{A}, \boldsymbol{\theta}) = \pi_k \cdot p(\mathbf{D}_{.,1}|\theta_k)$, and the remaining forward probabilities can be computed recursively with the forward algorithm.
For $t = 2, \dots, T$:

$$\alpha_t(k) = \left(\sum_{l=1}^{K} \alpha_{t-1}(l) \cdot a_{l,k}\right) \cdot p(\mathbf{D}_{.,t}|\theta_k) \tag{6.36}$$

Similarly, the backward probabilities can be defined as:

$$\beta_t(k) := P(\mathbf{D}_{.,t+1}, \dots, \mathbf{D}_{.,T}|v_t = k, \Pi, \mathbf{A}, \boldsymbol{\theta})$$

and the backward algorithm computes the backward probabilities recursively.
For $t = T-1, \dots, 1$:

$$\beta_t(k) = \sum_{l=1}^{K} a_{l,k} \cdot p(\mathbf{D}_{.,t+1}|\theta_l) \cdot \beta_{t+1}(l) \tag{6.37}$$

where $\beta_T(k) = 1$ for all k serves as initialisation. The so called 'forward-backward formula', which holds true for all $t \in \{1, \dots, T\}$, can then be used to compute the marginal (marginalized over all possible state sequences $v_1, \dots, v_T$) probability of the data $\mathbf{D}$:

$$P(\mathbf{D}|\Pi, \mathbf{A}, \boldsymbol{\theta}) = P(\mathbf{D}_1, \dots, \mathbf{D}_T|\Pi, \mathbf{A}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \alpha_t(k) \cdot \beta_t(k) \tag{6.38}$$

Unfortunately, the maximisation of Eq. (6.38) in the parameters $(\Pi, \mathbf{A}, \boldsymbol{\theta})$ is analytically not feasible so that the ML estimates have to be determined numerically. Like for the frequentist finite mixture model, the EM algorithm can be used.
To this end, assume that the allocation vector $\mathbf{V} = (v_1, \dots, v_T)^\mathsf{T}$ was known, and re-write the HMM log-likelihood:

$$l(\Pi, \mathbf{A}, \boldsymbol{\theta}|\mathbf{V}, \mathbf{D}) = \sum_{t=1}^{T} \sum_{k=1}^{K} \Delta_{k,t} \cdot \log\{p(\mathbf{D}_{.,t}|\theta_k)\} \tag{6.39}$$

where $\Delta_{k,t}$ is a indicator variable with $\Delta_k(t) = 1$ if $v_t = k$ and $\Delta_k(t) = 0$ if $v_t \neq k$. As the values of the indicator variables $\Delta_{k,t}$ are missing, the EM algorithm is used for maximisation. After initialisation, say $(\Pi^{(1)}, \mathbf{A}^{(1)}, \boldsymbol{\theta}^{(1)})$, the M-step and the E-step are performed iteratively till convergence is reached. In the i-th iteration the E-step calculates the expectation of Eq. (6.39) conditional on the data $\mathbf{D}$ and the current parameters $(\Pi^{(1)}, \mathbf{A}^{(1)}, \boldsymbol{\theta}^{(1)})$, so that the expectation is only taken w.r.t. the $\Delta_{k,t}$'s:

$$\mathbb{E}[l(\Pi, \mathbf{A}, \boldsymbol{\theta}|\mathbf{V}, \mathbf{D})|\mathbf{D}, \Pi^{(i)}, \mathbf{A}^{(i)}, \boldsymbol{\theta}^{(i)}] = \sum_{t=1}^{T} \sum_{k=1}^{K} \mathbb{E}[\Delta_{k,t}|\mathbf{D}, \Pi^{(i)}, \mathbf{A}^{(i)}, \boldsymbol{\theta}^{(i)}] \cdot \log\{p(\mathbf{D}_{.,t}|\theta_k)\} \tag{6.40}$$

and it can be easily shown that

$$\mathbb{E}[\Delta_{k,t}|\mathbf{D}, \Pi^{(i)}, \mathbf{A}^{(i)}, \boldsymbol{\theta}^{(i)}] = \frac{\alpha_t^{(i)}(k) \cdot \beta_t^{(i)}(k)}{P(\mathbf{D}|\Pi^{(i)}, \mathbf{A}^{(i)}, \boldsymbol{\theta}^{(i)})} =: \hat{\Delta}_{k,t}^{(i+1)} \tag{6.41}$$

where $\alpha_t^{(i)}(k)$ and $\beta_t^{(i)}(k)$ can be computed with the forward and the backward algorithm, using the current parameter instantiations $\Pi^{(i)}$, $\mathbf{A}^{(i)}$, and $\boldsymbol{\theta}^{(i)}$ in Eqns. (6.36) and (6.37), and the denominator can be computed by plugging $\alpha_t^{(i)}(k)$ and $\beta_t^{(i)}(k)$ into Eq. (6.38).
The M-step then maximises Eq. (6.40) in the parameters $(\Pi, \mathbf{A}, \boldsymbol{\theta})$ to obtain updated parameters $(\Pi^{(i+1)}, \mathbf{A}^{(i+1)},$
$\boldsymbol{\theta}^{(i+1)})$, whose elements are given by:

$$\pi_k^{(i+1)} = \frac{\alpha_1^{(i)}(k) \cdot \beta_1^{(i)}(k)}{\sum_{l=1}^{K} \alpha_T^{(i)}(l)} \tag{6.42}$$

$$\begin{aligned} a_{l,k}^{(i+1)} &= \frac{\sum_{t=1}^{T-1} \alpha_t^{(i)}(l) \cdot a_{l,k} \cdot p(\mathbf{D}_{\cdot,t+1}|\theta_k^{(i)}) \cdot \beta_{t+1}^{(i)}(k)}{\sum_{t=1}^{T-1} \alpha_t^{(i)}(l)\beta_t^{(i)}(l)} \\ \theta_k^{(i+1)} &= \frac{\sum_{t=1}^{T} \alpha_t^{(i)}(k) \cdot \beta_t^{(i)}(k) \cdot \{\sum_{i=1}^{n} d_{i,t}\}}{\sum_{t=1}^{T} \alpha_t^{(i)}(k) \cdot \beta_t^{(i)}(k)} \end{aligned} \tag{6.43}$$

Iterating the E- and the M-step until convergence is reached, yields the ML estimators of the hidden Markov model. As before, to avoid getting stuck in local optima, the algorithm should be run several times with different initialisations. Pseudo code for computing the ML-estimators for each $K = 2, \ldots, K_{MAX}$ is provided in Table 6.5.

The application of the forward and backward algorithms can lead to serious underflow problems (i.e. to too low probabilities). These issues can be solved easily by re-scaling. For the empirical data analyses in this chapter the scaling procedure from [39] was implemented to protect against numerical underflows.

6.8.4 Additional tables and figures

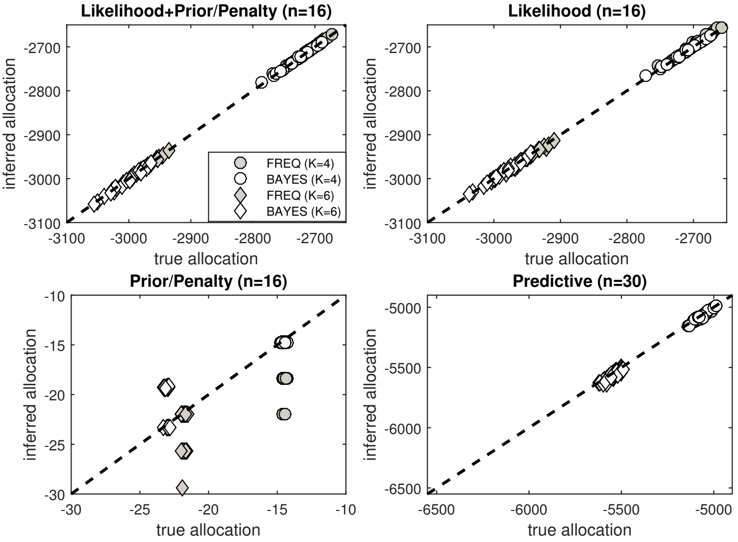
This section of the supplementary material provides two additional tables (Tables 4-5) for Section 4 of this chapter.

Allocation scenario	**Variant no. 1: $\mathbf{P}_1^\star$**	**Variant no. 2: $\mathbf{P}_2^\star$**
Homogeneous time points	$(\mathbf{1}_{96})$	$(\mathbf{5}_{96})$
Change-point segmented	$(\mathbf{1}_{24}, \mathbf{2}_{24}, \mathbf{3}_{24}, \mathbf{4}_{24})$	$(\mathbf{1}_{16}, \mathbf{2}_{16}, \mathbf{3}_{16}, \mathbf{4}_{16}, \mathbf{5}_{16}, \mathbf{6}_{16})$
Finite mixture model	MIX$(\mathbf{1}_{48}, \mathbf{5}_{48})$	MIX$(\mathbf{1}_{24}, \mathbf{2}_{24}, \mathbf{4}_{24}, \mathbf{8}_{24})$
Hidden Markov model	$(\mathbf{1}_{24}, \mathbf{5}_{24}, \mathbf{1}_{24}, \mathbf{5}_{24})$	$(\mathbf{1}_{16}, \mathbf{5}_{16}, \mathbf{1}_{16}, \mathbf{5}_{16}, \mathbf{1}_{16}, \mathbf{5}_{16})$

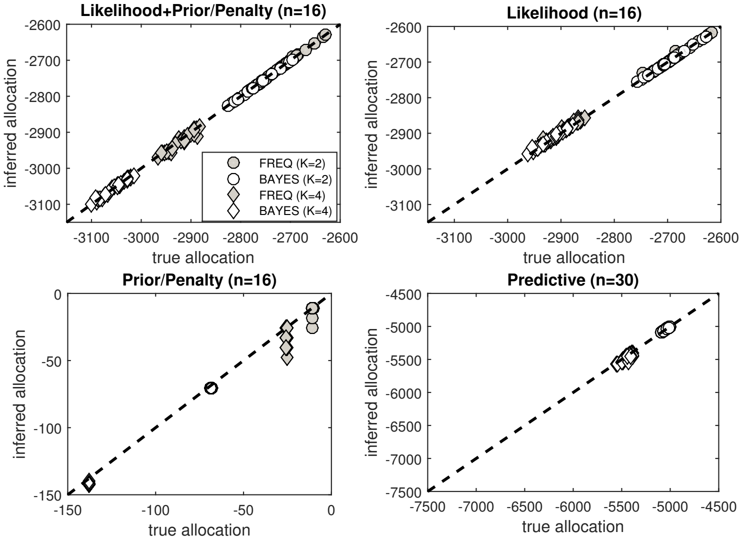
Table 6.6: Overview to the eight allocation schemes, employed in the comparative evaluation study. For each of the four allocation scenarios (HOM, CPS, MIX, and HMM) two different vectors of Poisson parameters ($\mathbf{P}_1^\star$ and $\mathbf{P}_2^\star$) are considered; see Section 4.1 for further details.

Weekday	**Sunday**	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**
no. of these weekdays in 2013 without holidays	$n_1 = 52$	$n_2 = 46$	$n_3 = 52$	$n_4 = 51$	$n_5 = 50$	$n_6 = 52$	$n_7 = 52$
extracted no. of entries from that weekday	23147	20127	24215	24551	25025	26291	26240

Table 6.7: Summary of the extracted New York City Taxi (NYCT) data. The table gives the number of non-holiday weekdays in 2013, and the total number of the extracted taxi ride pick-up date-and-time entries per weekday. See Section 4.2 of this chapter for further details.



(a) **Changepoint models (FREQ and BAYES) on changepoint data with n=16.**



(b) **Mixture models (FREQ and BAYES) on mixture data with n=16.**

Figure 6.11: Comparing the best inferred models with the true models for time series with n=16 data points per time point. This figure corresponds to Figure 6.10 of this chapter. Unlike in Figure 6.10 of this chapter, the performances of the best inferred models and the true models are here compared for more informative data, i.e. for time series with $n = 16$ data points per time point. See caption of Figure 6.10 in this chapter for further details.

Summary

One statistical challenge in many fields is to infer network topologies of interacting units from time series data. One class of statistical models, which has been widely applied to deal with this task, is dynamic Bayesian network models (DBNs). The underlying assumption of the conventional DBNs is that the underlying process is a homogeneous Markov process, so that DBNs do not allow the network parameters to change in time. Therefore, DBNs cannot deal with non-homogeneous and non-stationary regulatory processes, which arise in many important real world applications.

Recently, non-homogeneous dynamic Bayesian network models (NH-DBNs) have been introduced and become an important statistical tool to relax this restrictive assumption. NH-DBNs have been implemented with various allocation models to divide the temporal data into disjoint data subsets. Those models infer the data segmentation, the joint network structure and the segment- or component-specific interaction parameters from the data.

In this thesis we have focused on improving changepoint (CPS) divided NH-DBNs which have become the most widely applied NH-DBNs to model complex systems. These models infer changepoints which divide the data into disjoint segments and the segment-specific network parameters are learned for each segment separately. In many real world applications these NH-DBNs divide a time series into even shorter segments. Learning the network parameters for each segment separately ('uncoupled' NH-DBN models), leads to over flexibility and inflated inference uncertainties. Moreover, these models do not incorporate the reasonable prior assumption that neighboring segments are often more likely to have similar network interaction parameters than distant segments. To address these bottlenecks, Bayesian models with coupling mechanisms between the segment-specific parameters have been proposed.

Bayesian models with parameter coupling can lead to significantly improved network reconstruction accuracies when the segment-specific parameters are similar. However, recently we have found that coupling can become counterproductive when the segment-specific parameters are dissimilar. The reason for that is that neither the sequential nor the global coupling scheme has an effective mechanism for uncoupling. For many real-world applications this is a constraint. We have addressed these bottlenecks in this thesis by introducing four novel

NH-DBNs.

Another scenario corresponded to many real-world applications, happens when time series data are often collected under different experimental conditions. That is, instead of one single time series, which can be divided into segments with natural temporal order, there are K (short) time series with no natural order. They are exchangeable units and there is no need for inferring the segmentation. In this situation it is often unclear a priori whether the network parameters are actually component-specific or whether they are constant across components. Whereas in real-world applications there can be both types of parameters simultaneously. We, therefore, have addressed this problem by introducing novel partially NH-DBNs based on Bayesian regression models with partition design matrix.

In **chapter 1** we have given an introduction and an overview to existing network models and we have outlined this thesis.

In **chapter 2**, we have proposed two new models based on piecewise Bayesian regression models: The partially segment-wise coupled NH-DBN model and the generalized fully sequentially coupled model. Our empirical results have shown that the partially coupled model leads to improved network reconstruction accuracies. For the generalized coupled model we have not seen consistent improvements over the fully coupled NH-DBN model.

In **chapter 3**, we have therefore refined the generalized fully sequentially coupled model. For the refined model with a hyperprior onto the second hyperparameter of the coupling parameter prior we have seen improved network reconstruction accuracies.

In **chapter 4**, we have presented a novel NH-DBN model with partially edge-wise coupled segment-specific network parameters. This model operates on the individual edges. Instead of enforcing *all* edges to be coupled, our model operates edge-wise and infers for each individual edge from the data whether the associated parameters should be coupled or stay uncoupled across all segments. We have empirically shown on yeast gene expression time series that the new model reaches a highest network reconstruction accuracy. For Arabidopsis thaliana gene expression data, we have shown that our new model not only outputs a network prediction, but also allows to distinguish between edges whose regulatory effects stay similar across time and edges whose regulatory effects are subject to more substantial temporal changes.

In **chapter 5**, we have introduced a partially NH-DBN model, which is effectively a Bayesian regression model with partitioned design matrix. The new model aims to infer the best trade-off between a homogeneous model and a non-homogeneous model. For each network interaction there is a parameter, and the new model infers from the data whether this parameter is constant or whether it varies among segments. We, moreover, have proposed to employ a Gaussian process based approach to deal with non-equidistant measurements. Our applications to yeast data have shown that the new model improves the network reconstruction accuracy. We have used the new model to reconstruct the topologies of the mTORC1 data. The inferred network topologies showed features that are consistent with the biological literature.

Chapter 6 has been on a comparative evaluation study on popular non-homogeneous Poisson models for count data. For this study the standard homogeneous Poisson model (HOM) and three non-homogeneous variants, namely a Poisson changepoint model (CPS), a Poisson free mixture model (MIX), and a Poisson hidden Markov model (HMM) have been implemented in both frequentist and Bayesian framework.The first major objective has been cross-comparing the performances of the four aforementioned models independently for both modelling frameworks (Bayesian and frequentist). Subsequently, a pairwise comparison between the four Bayesian and the four frequentist models has been performed to elucidate to which extent the results of the two paradigms (‘Bayesian versus frequentist’) differ.

Samenvatting

Eén van de statistische uitdagingen in veel onderzoeksgebieden is het uit tijdseriedata afleiden van de topologie van netwerken van op elkaar inwerkende eenheden. Een klasse van statistische modellen die veel toegepast wordt om met deze opgave om te gaan, is de klasse van dynamische Bayesiaanse netwerkmodellen (DBN's). De onderliggende aanname van conventionele DBN's is dat het onderliggende proces een homogeen Markov proces is, zodat voor DBN's de netwerkparameters niet mogen veranderen in de tijd. Daardoor kunnen DBN's niet omgaan met niet-homogene en niet-stationaire regelgevende processen, die voorkomen in veel belangrijke toepassingen in de werkelijkheid.

Onlangs zijn niet-homogene dynamische Bayesiaanse netwerkmodellen (NH-DBN's) geïntroduceerd, welke een belangrijk statistisch hulpmiddel zijn geworden om deze beperkende veronderstelling te omzeilen. NH-DBN's zijn geïmplementeerd met verschillende toewijzingsmodellen om de tijdseriedata te verdelen in afzonderlijke dataverzamelingen. Deze modellen leiden de datasegmentatie, de gezamenlijke netwerkstructuur en de segment- of componentspecifieke interactieparameters uit de data af.

In dit proefschrift hebben we ons gefocust op het verbeteren van changepoint (CPS) verdeelde NH-DBN's; dit zijn de NH-DBN's die het meest toegepast worden om complexe systemen te modelleren. Deze modellen leiden changepoints af die de data verdelen in afzonderlijke segmenten en de segmentspecifieke netwerkparameters worden voor elk segment apart geleerd. In veel toepassingen in de werkelijkheid verdelen deze NH-DBN's een tijdserie in nog kortere segmenten. Het leren van de netwerkparameters voor elk segment afzonderlijk ('ontkoppelde' NH-DBN-modellen) leidt tot te grote flexibiliteit en opgedreven inferentieonzekerheden. Bovendien, deze modellen bevatten niet de redelijke aanname vooraf dat naburige segmenten vaak meer kans hebben op vergelijkbare netwerkinteractieparameters dan segmenten ver van elkaar. Om deze knelpunten aan te pakken zijn Bayesiaanse modellen met koppelingsmechanismen tussen de segmentspecifieke parameters voorgesteld.

Bayesiaanse modellen met parameterkoppeling kunnen leiden tot significant verbeterde nauwkeurigheden van netwerkreconstructies wanneer de segmentspecifieke parameters vergelijkbaar zijn. Onlangs hebben we echter ontdekt dat koppeling contraproductief kan worden wanneer de segmentspecifieke parameters verschillend zijn. De reden daarvoor is dat zowel het sequentiële als het globale koppelingsschema geen effectief mechanisme voor ontkoppeling heeft. Dit is voor veel echte toepassingen een beperking. We hebben deze knelpunten in dit proefschrift aangepakt door vier nieuwe NH-DBN's te introduceren.

Een ander scenario, dat overeenkomt met veel werkelijke toepassingen, komt voor wanneer tijdseriedata vaak wordt verzameld onder verschillende experimentele omstandigheden. Hierbij zijn er, in plaats van één enkele tijdserie die verdeeld kan worden in segmenten met een normale tijdsorde, K (korte) tijdseries zonder normale orde. Dit zijn inwisselbare eenheden en er is geen noodzaak om de segmentatie af te leiden. In deze situatie is het a priori vaak onduidelijk of de netwerkparameters werkelijk componentspecifiek zijn of dat zij constant zijn voor alle componenten. In echte toepassingen daarentegen, kunnen beide soorten parameters tegelijkertijd voorkomen. We hebben daarom dit probleem aangepakt door nieuwe gedeeltelijke NH-DBN's te introduceren, gebaseerd op Bayesiaanse regressiemodellen met partitieontwerpmatrix.

In **hoofdstuk 1** hebben we een introductie en een overzicht van de bestaande netwerkmodellen gegeven. Ook hebben we in hoofdlijnen aangegeven waar dit proefschrift over gaat.

In **hoofdstuk 2** hebben we twee nieuwe modellen geboden, gebaseerd op stuksgewijs Bayesiaanse regressiemodellen, namelijk het gedeeltelijk segmentgewijs gekoppeld NH-DBN-model en het gegeneraliseerde, volledig sequentieel gekoppelde model. Onze empirische resultaten laten zien dat het gedeeltelijk gekoppelde model leidt tot een verbeterde nauwkeurigheid van netwerkreconstructies. Voor het gegeneraliseerde gekoppelde model hebben we geen consequente verbeteringen gezien ten opzichte van het volledig gekoppelde NH-DBN-model.

In **hoofdstuk 3** hebben we daarom het gegeneraliseerde, volledig sequentieel gekoppelde model verfijnd. Voor het verfijnde model met een hyperprior distributie op de tweede hyperparameter van de a priori-distributie van de koppelingsparameter zien we een verbetering in de nauwkeurigheid van netwerkreconstructies.

In **hoofdstuk 4** hebben we een nieuw NH-DBN-model gepresenteerd met gedeeltelijk zijdegewijs gekoppelde en segment-specifieke netwerkparameters. Dit model werkt op de individuele zijden. In plaats van *alle* randen te forceren gekoppeld te zijn, werkt ons model zijdegewijs en leidt het voor elke afzonderlijke zijde uit de data af of de bijbehorende parameters gekoppeld moeten worden of juist in alle segmenten ontkoppeld moeten blijven. Dit nieuwe model heeft ook het ongekoppelde en het gekoppelde NH-DBN als limietgevallen. We hebben empirisch aangetoond op gist-genexpressie tijdseries dat het nieuwe model een hoogste nauwkeurigheid van netwerkreconstructie behaald. Voor Arabidopsis thaliana-genexpressiedata laten we zien dat ons nieuwe model niet alleen een netwerkvoorspelling geeft, maar ook onderscheid kan maken tussen zijden waarvan de regulerende effecten gelijk blijven in de tijd en zijden waarvan de regulerende effecten meer substantiële veranderingen in de tijd ondergaan.

In **hoofdstuk 5** hebben we een gedeeltelijk NH-DBN-model geïntroduceerd, dat in feite een Bayesiaans regressiemodel is met partitieontwerpmatrix. Het nieuwe model beoogt de beste afweging te maken tussen een homogeen model en een niet-homogeen model. Voor elke netwerkinteractie is er een parameter en het nieuwe model leidt uit de data af of deze parameter constant is of tussen segmenten varieert. Daarnaast stellen we voor om een Gaussisch proces gebaseerde benadering te gebruiken om niet-equidistante metingen te kunnen verwerken.

Door dit nieuwe model toe te passen op gistdata hebben we aangetoond dat het de nauwkeurigheid van netwerkreconstructies verbetert. We hebben het nieuwe model gebruikt om de topologieën van de mTORC1-data te reconstrueren. De afgeleide topologieën van netwerken vertoonden kenmerken die overeenkomen met de biologie-literatuur.

Hoofdstuk 6 ging over een vergelijkend evaluatieonderzoek naar populaire niet-homogene Poisson-modellen voor count data. Voor dit onderzoek zijn het standaard homogene Poisson model (HOM) en drie niet-homogene varianten, namelijk een Poisson changepoint-model (CPS), een Poisson free-mixture-model (MIX) en een Poisson hidden-Markov-model (HMM), geïmplementeerd in zowel een frequentistisch als een Bayesiaans kader. Het eerste hoofddoel is het onafhankelijk vergelijken van de prestaties van de vier bovengenoemde modellen voor beide modelleringskaders (Bayesiaans en frequentistisch). Daarna voeren we een paarsgewijze vergelijking uit tussen de vier Bayesiaanse en de vier frequentistische modellen om op te helderen in hoeverre de resultaten van de twee paradigma's ('Bayesiaans versus frequentistisch') verschillen.

Acknowledgments

First and above all, I would like to express my deepest thanks to my God for helping me through all the difficulties and always strongly supporting me. Thank you God for your great love and care, and for your many blessings on us.

I would like to express my sincere gratitude to my day-to-day supervisor, Marco Grzegorczyk, for his strong scientific support, encouragement, and patience throughout my PhD studies. His guidance and supervision have been essential for their completion. I would also like to thank my supervisor, Ernst Wit for his support, and valuable comments during my PhD studies.

I would like to express my appreciation to Dirk Husmeier, Casper Albers, and Joris Mulder for assessing the present thesis and sharing their time and knowledge with me.

I am grateful to Dirk van Kekem for translating the summary of my thesis into Dutch and thanks to Hassan and Georg for being my paranymphs. I am also grateful to the supervisors of my master's thesis, Nader Nematollahi and Ahmad Parsian, for all I learned from them during my master's, which proved to be helpful in my PhD. Furthermore, I want to acknowledge Mohsen Ghanbary for encouraging me to study at the University of Groningen, and my special thanks go to Fardin for welcoming us when we arrived in the Netherlands for the first time.

I am in debt to all my friends and colleagues in the Netherlands, especially in Groningen, for creating a pleasant environment and providing stimulating discussions. Most notably I would like to thank Ahmad, Ahmadreza, Alef, Ali, Ben, Bohuan, Dirk, Fentaw, Francisco, Fred, Gabriel, Georg, Hamed, Hassan, Jin, Jorge, Jris, Luca, Maartje, Mahdi, Marcus, Mathijs, Matthijs, Maurits, Mehdi, Mohammad, Nazia, Naomi, Nikolay, Reka, Reza, Rianne, Pariya, Sam, Spyros, Venus, Wim and Yongjiao.

Words cannot express how grateful I am to my family: my father and my mother, my brother and sisters for all of the sacrifices that they have made on my behalf. Your prayers for me are what have sustained me thus far. I also wish to thank my in-laws, especially my mother-in-law, for their love and encouragement.

Most importantly, it would have been impossible to finish my PhD studies and write this thesis without the unconditional love, support, care, and encourage-

ment from my beloved wife Mozhgan and my two dearly loved and wonderful children: Alicenna and Delina, who provide me with unending inspiration. I would like to dedicate this thesis to you from the bottom of my heart. My deepest appreciation is for you forever.

Mahdi Shafiee Kamalabad
Groningen, January, 2019.

Bibliography

[1] Andrej Aderhold, Dirk Husmeier, and Marco Grzegorczyk. Statistical inference of regulatory networks for circadian regulation. *Statistical Applications in Genetics and Molecular Biology*, 13(3):227–273, 2014.

[2] Andrej Aderhold, Dirk Husmeier, and V. Anne Smith. Reconstructing ecological networks with hierarchical bayesian regression and mondrian processes. In Carlos M. Carvalho and Pradeep Ravikumar, editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 75–84, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL: `http://proceedings.mlr.press/v31/aderhold13a.html`.

[3] A. Ahmed and E.P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106:11878–11883, 2009.

[4] I.E. Auger and C.E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51:39–54, 1989.

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

[6] J.M. Bland and D.G. Altman. Comparing methods of measurement: why plotting difference against standard method is misleading. *Lancet*, 346:1085–1087, 1995.

[7] S.P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphial Statistics*, 7:434–455, 1998.

[8] I. Cantone, L. Marucci, F. Iorio, M.A. Ricci, V. Belcastro, M. Bansal, S. Santini, M. di Bernardo, D. di Bernardo, and M.P. Cosma. A yeast synthetic network for in vivo assessment of reverse-engineering and modeling approaches. *Cell*, 137:172–181, 2009.

[9] W.S. Cleveland. *The Elements of Graphing Data*. Hobart Press, 2nd edition, 1994.

[10] Piero Dalle Pezze, Stefanie Ruf, Annika G. Sonntag, Miriam Langelaar-Makkinje, Philip Hall, Alexander M. Heberle, Patricia Razquin Navas, Karen van Eunen, Regine C. Tölle, Jennifer J. Schwarz, Heike Wiese, Bettina Warscheid, Jana Deitersen, Björn Stork, Erik Fäßler, Sascha Schäuble, Udo Hahn, Peter Horvatovich, Daryl P. Shanley, and Kathrin Thedieck. A systems study reveals concurrent activation of AMPK and mTOR by amino acids. *Nature Communications*, 7:1–19, 2016.

[11] J. Davis and M. Goadrich. The relationship between precision-recall and ROC curves. In *ICML '06: Proceedings of the 23rd international conference on Machine Learning*, pages 233–240, New York, NY, USA, 2006. ACM. `doi:http://doi.acm.org/10.1145/1143844.1143874`.

[12] A. P. Dempster, N. M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39(1):1–38, 1977.

[13] C.C. Dibble and L.C. Cantley. Regulation of mTORC1 by PIP3K signaling. *Trends Cell Biology*, 25:545–555, 2015.

[14] F. Dondelinger, S. Lèbre, and D. Husmeier. Non-homogeneous dynamic Bayesian networks with Bayesian regularization for inferring gene regulatory networks with gradually time-varying structure. *Machine Learning*, 90:191–230, 2012.

[15] B. Donovan and D. Work. Using coarse GPS data to quantify city-scale transportation system resilience to extreme events. In *Proceedings of the Transportation Research Board 94th Annual Meeting*, Washington, 2015. to appear.

[16] K.D. Edwards, P.E. Anderson, A. Hall, N.S. Salathia, J.C.W. Locke, J.R. Lynn, M. Straume, J.Q. Smith, and A.J. Millar. Flowering locus C mediates natural variation in the high-temperature response of the Arabidopsis circadian clock. *The Plant Cell*, 18:639–650, 2006.

[17] N. Friedman and D. Koller. Being Bayesian about network structure. *Machine Learning*, 50:95–126, 2003.

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

[19] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, London, 2nd edition, 2004.

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

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

[22] M. Grzegorczyk. A non-homogeneous dynamic Bayesian network with a hidden Markov model dependency structure among the temporal data points. *Machine Learning*, 102(2):155–207, 2016.

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

[24] M. Grzegorczyk and D. Husmeier. 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), 2012. Article 7.

[25] M. Grzegorczyk and D. Husmeier. Regularization of non-homogeneous dynamic Bayesian networks with global information-coupling based on hierarchical Bayesian models. *Machine Learning*, 91:105–154, 2013.

[26] M. Grzegorczyk, D. Husmeier, K. Edwards, P. Ghazal, and A. Millar. Modelling non-stationary gene regulatory processes with a non-homogeneous Bayesian network and the allocation sampler. *Bioinformatics*, 24(18):2071–2078, 2008.

[27] M. Grzegorczyk, D. Husmeier, and R. Rahnenführer. Modelling non-stationary gene regulatory processes. *Advances in Bioinformatics*, 2010. vol. 2010, Article ID 749848.

[28] Marco Grzegorczyk and Mahdi Shafiee Kamalabad. Comparative evaluation of various frequentist and bayesian non-homogeneous poisson counting models. *Computational Statistics*, 32(1):1–33, 2017.

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

[30] S.K. Hindupur, A. González, and M.N. Hall. The opposing actions of target of rapamycin and AMP-activated protein kinase in cell growth control. *Cold Spring Harbor Perspectives in Biology*, 7, 2015. a019141.

[31] Dirk Husmeier. Introduction to learning Bayesian networks from data. In Dirk Husmeier, Richard Dybowski, and Stephen Roberts, editors, *Probabilistic Models in Bioinformatics and Medical Informatics*, London, 2003. Springer.

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

[33] E.A. Kikis, R. Khanna, and P.H. Quail. ELF4 is a phytochrome-regulated component of a negative-feedback loop involving the central oscillator components CCA1 and LHY. *Plant J.*, 44(2):300–313, 2005.

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

[35] M. Kolar, L. Song, A. Ahmed, and E. Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, 4:94–123, 2010.

[36] M. Kolar, L. Song, and E. Xing. Sparsistent learning of varying-coefficient models with structural changes. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1006–1014. 2009.

[37] S. Lèbre. *Stochastic process analysis for Genomics and Dynamic Bayesian Networks inference.* PhD thesis, Université d'Evry-Val-d'Essonne, France, 2007.

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

[39] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. *The Bell System Technical Journal*, 62:1035–1074, 1983.

[40] F. Liang, C. Liu, and R.J. Carroll. *Advanced Markov chain Monte Carlo methods: Learning from past samples*. Wiley Series in Computational Statistics. John Wiley and Sons, Cornwall, UK, 2010.

[41] James C W Locke, László Kozma-Bognár, Peter D Gould, Balázs Fehér, Eva Kevei, Ferenc Nagy, Matthew S Turner, Anthony Hall, and Andrew J Millar. Experimental validation of a predicted feedback loop in the multi-oscillator clock of *Arabidopsis thaliana*. *Molecular Systems Biology*, 2(1), 2006.

[42] B.D. Manning and A. Toker. AKT/PKB Signaling: Navigating the Network. *Cell*, 169:381–405, 2017.

[43] K. Miwa, M. Serikawa, S. Suzuki, T. Kondo, and T. Oyama. Conserved expression profiles of circadian clock-related genes in two lemna species showing long-day and short-day photoperiodic flowering responses. *Plant and Cell Physiology*, 47(5):601–612, 2006.

[44] T. C. Mockler, T. P. Michael, H. D. Priest, R. Shen, C. M. Sullivan, S. A. Givan, C. McEntee, S. A. Kay, and J. Chory. The diurnal project: Diurnal and circadian expression profiling, model-based pattern matching and promoter analysis. *Cold Spring Harbor Symposia on Quantitative Biology*, 72:353–363, 2007.

[45] E.B.M. Nascimento, M. Snel, B. Guigas, G.C.M. van der Zon, J. Kriek, Maassen J.A., I.M. Jazet, M. Diamant, and D.M. Ouwens. Phosphorylation of PRAS40 on Thr246 by PBK/AKT facilitates efficient phosphorylation of Ser183 by mTORC1. *Cellular Signalling*, 22:961–967, 2010.

[46] A. Nobile and A.T. Fearnside. Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing*, 17(2):147–162, 2007.

[47] Alexandra Pokhilko, Paloma Mas, Andrew J Millar, et al. Modelling the widespread effects of TOC1 signalling on the plant circadian clock and its outputs. *BMC Systems Biology*, 7(1):1–12, 2013.

[48] J.W. Robinson and A.J. Hartemink. Non-stationary dynamic Bayesian networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 1369–1376. Morgan Kaufmann Publishers, 2009.

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

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

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

[52] R.A. Saxton and D.M. Sabatini. mTOR Signaling in Growth, Metabolism, and Disease. *Cell*, 168:960–976, 2017.

[53] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[54] M. Shafiee Kamalabad and M. Grzegorczyk. A non-homogeneous dynamic Bayesian network model with partially sequentially coupled network parameters. In *Proceedings of the 31st International Workshop on Statistical Modelling*, volume 1, pages 139–144, 2016. URL: `https://www.lebesgue.fr/content/sem2016-iwsm2016`.

[55] M. Shafiee Kamalabad and M. Grzegorczyk. A sequentially coupled non-homogeneous dynamic Bayesian network model with segment-specific coupling strengths. In *Proceedings of the 32nd International Workshop on Statistical Modelling*, volume 1, pages 173–178, 2017. URL: `https://iwsm2017.webhosting.rug.nl/`.

[56] M. Shafiee Kamalabad and M. Grzegorczyk. A new partially coupled piecewise linear regression model for statistical network structure inference. In *Proceedings of the 15th International Conference on Computational Intelligence methods for Bioinformatics and Biostatistics*, page 30, 2018. URL: `https://eventos.fct.unl.pt/cibb2018/`.

[57] M. Shafiee Kamalabad and M. Grzegorczyk. Non-homogeneous dynamic Bayesian networks with edge-wise coupled parameters. In *Proceedings of the 33rd International Workshop on Statistical Modelling*, volume 1, pages 270–275, 2018. URL: `https://people.maths.bris.ac.uk/~sw15190/IWSM2018/`.

[58] Mahdi Shafiee Kamalabad and Marco Grzegorczyk. Improving non-homogeneous dynamic Bayesian networks with sequentially coupled parameters. *Statistica Neerlandica*, 72(3):281–305, 2018.

[59] Mahdi Shafiee Kamalabad, Alexander Martin Heberle, Kathrin Thedieck, and Marco Grzegorczyk. Partially non-homogeneous dynamic bayesian networks based on Bayesian regression models with partitioned design matrices. *Bioinformatics*, page bty917, 2018. URL: `http://dx.doi.org/10.1093/bioinformatics/bty917`, `doi:10.1093/bioinformatics/bty917`.

[60] V Anne Smith, Jing Yu, Tom V Smulders, Alexander J Hartemink, and Erich D Jarvis. Computational inference of neural information flow networks. *PLoS computational biology*, 2(11):e161, 2006.

[61] G.A. Soliman, H.A. Acosta-Jaquez, E.A. Dunlop, B. Ekim, N.E. Maj, A.R. Tee, and D.C. Fingar. mTOR Ser-2481 autophosphorylatyion monitors mTORC-specific catalytic activity and clarifies rapamycin mechanism of action. *Journal of Biological Chemistry*, 285:7866–7879, 2010.

[62] Thomas Thorne and Michael P. H. Stumpf. Inference of temporally varying Bayesian networks. *Bioinformatics*, 28(24):3298–3305, 2012. `doi:10.1093/bioinformatics/bts614`.

[63] A. Tzatsos and K.V. Kandor. Nutrients suppress phosphatidylinositol 3-kinase/AKT signaling via raptor-dependent mTOR-mediated insulin receptor substrate 1 phosphorylation. *Molecular Cell Biology*, 26:63–76, 2006.

[64] Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. GPstuff: Bayesian modeling with Gaussian processes. *The Journal of Machine Learning Research*, 14(1):1175–1179, 2013.

Curriculum Vitae

Mahdi Shafiee Kamalabad was born on 21 September, 1982 in Tehran, Iran. In 2004, he obtained his bachelor's degree in Statistics with the honor of top-ranked student from the University of Tabriz. He continued his studies with a master's in Statistics at Allame Tabatabaie University (ATU). Mahdi wrote his master's thesis under the supervision of Nader Nematollahi and Ahmad Parsian, and graduated as a top-ranked student from ATU in 2006. Afterwards, he received the Elite Certificate from the Ministry of Energy in Iran and was employed as a statistician and data scientist in the department of Strategic Management, Quality and Productivity in the Tehran Regional Electric Company (TREC) until 2014. During the course of his employment there he carried out statistical analyses and taught courses in Statistics at various universities. From 2015 to 2019, he was a PhD student at the University of Groningen where he wrote the present thesis under the supervision of Marco Grzegorczyk and Ernst Wit. His main research interests are Complex data Analysis, Network Inference, (Bayesian) Statistical Inference, Computational Statistics.