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

Shafiee Kamalabad, Mahdi

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 $X_1 \rightarrow Y \leftarrow X_2$, then the interaction $X_1 \rightarrow Y$ can stay constant, while $X_2 \rightarrow Y$ 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 $Z_1, \ldots, Z_N$. 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 $Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_N$ 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 $X_1, \ldots, X_n$ are the covariates.

5.1.1 Bayesian regression with partitioned design matrix

We consider a regression model with response Y and covariates $X_1, \ldots, X_n$. 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 $k \in \{1, \ldots, K\}$ were measured at equidistant time points $t = 1, \ldots, T_k$. 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 $x_{1,k,t}, \ldots, x_{n,k,t}$ 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:

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

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:

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

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 β_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

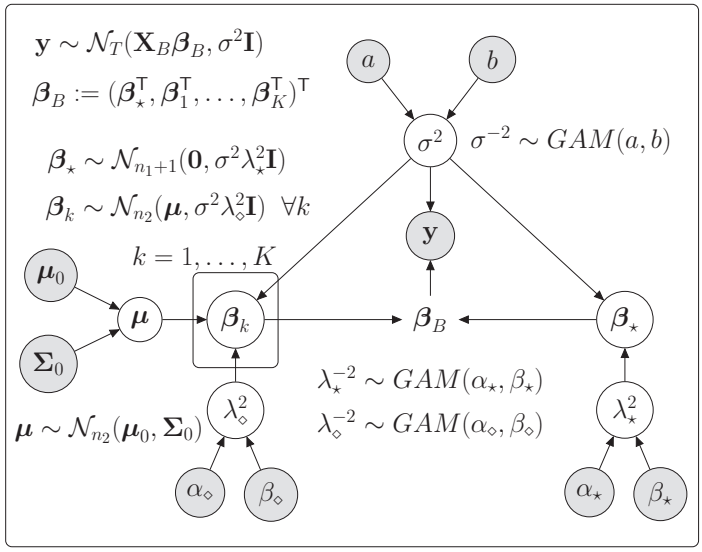

Figure 5.1: Graphical model representation of the regression model with partitioned design matrix. Variables that have to be inferred are in white circles. The data and the fixed hyperparameters are in grey circles. The vector $\boldsymbol{\beta}_B$ deterministically depends on $\boldsymbol{\beta}_\star$ 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 $n = |\mathbf{\Pi}|$. We get: $p(\boldsymbol{\delta} = \mathbf{1}|\mathbf{\Pi}) = \theta_n^{n+1}$ and $p(\boldsymbol{\delta} = \mathbf{0}|\mathbf{\Pi}) = (1-\theta_n)^{n+1}$. From (5.9) we obtain:

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

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

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 $x_{1,k,t}, \ldots, x_{n,k,t}$ is the same for all t. If the data within a component k were measured at time points $t_1, \ldots, t_{T_k}$, 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 $\mathcal{O}^\star = \min\{\mathcal{O}_2, \ldots, \mathcal{O}_{T_k}\}$, where $\mathcal{O}_i := t_i - t_{i-1}$.
- Given the observed data points $\{(t, y_{k,t}) : t = t_1, \ldots, t_{T_k}\}$, use a Gaussian process to predict the whole curve $\{(t, y_{k,t})\}_{t\geq 0}$.
- Extract the response values at the time points: $t_1 + \mathcal{O}^\star, \ldots, t_{T_k} + \mathcal{O}^\star$.
- Build the response vector and design matrix such that the values $x_{1,k,t_i}, \ldots, x_{n,k,t_i}$ are used to explain the predicted response value $\hat{y}_{k,t_i+\mathcal{O}^\star}$ $(i = 1, \ldots, T_k)$. 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

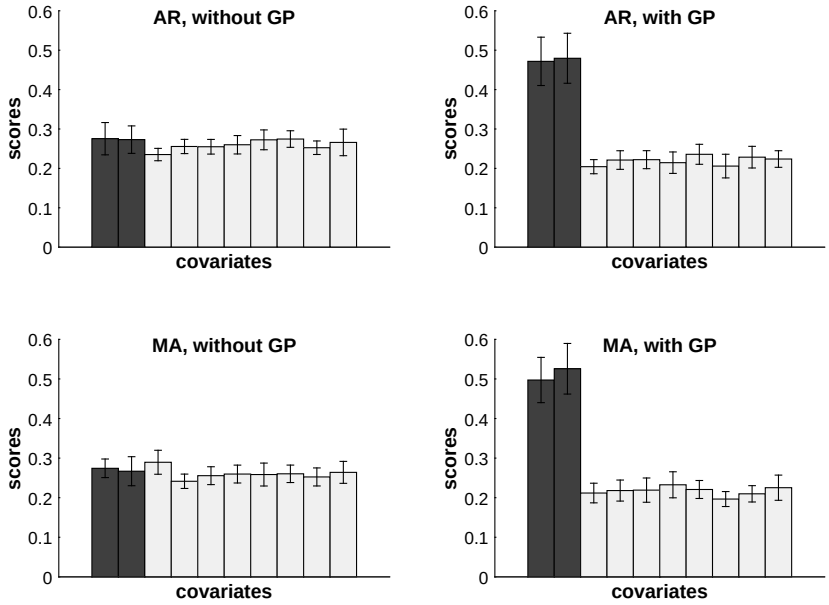

Figure 5.2: Average scores (posterior probabilities). In each histogram, the dark grey bars refer to the scores of the true covariates, and the light grey bars refer to the irrelevant variables. Covariate values were generated via autoregressive [AR] (top) and moving average [MA] processes (bottom). 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

$\mathbf{\Sigma}_0 = \mathbf{I}$. 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 $\mathbf{\Pi}_i^{(r)}$ ($i = 1, \ldots, N$; $r = 1, \ldots, R$) to a network sample $\mathcal{N}^{(1)}, \ldots, \mathcal{N}^{(R)}$, as explained in Section 5.1.6. For each edge $Z_j \to Z_i$ 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 $X_{i,t} \sim N(0, 1)$.

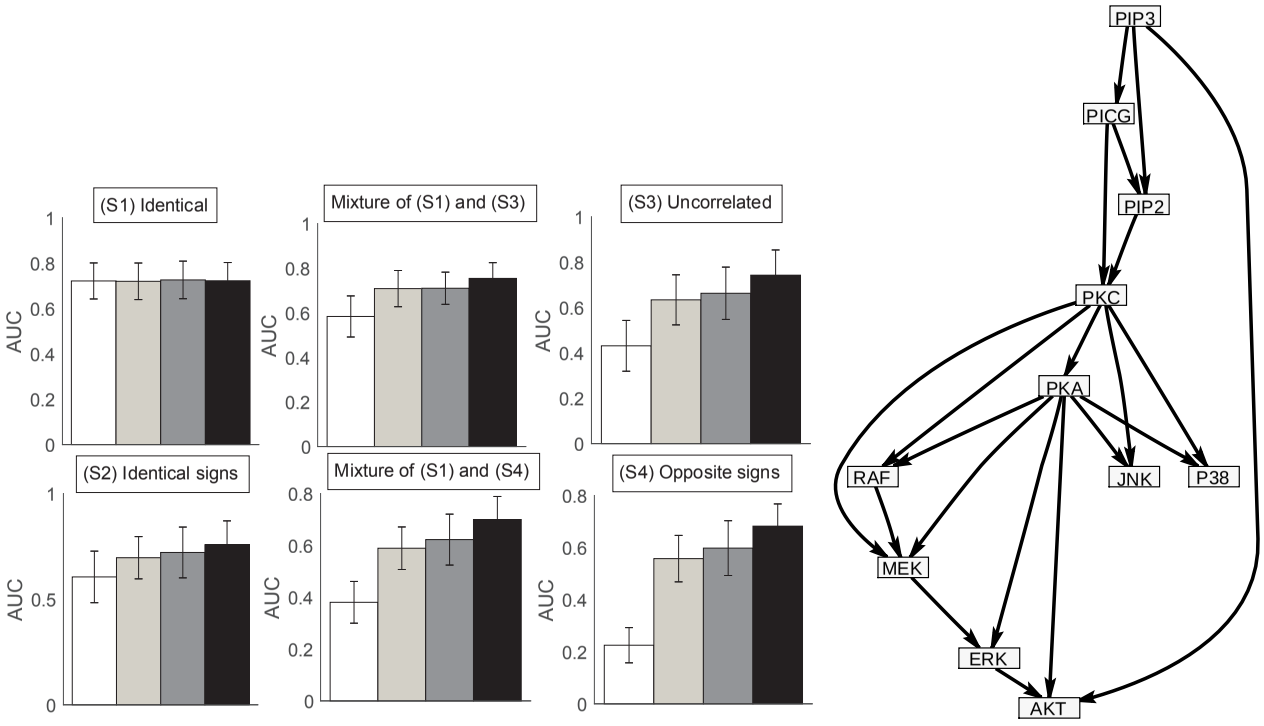

Figure 5.3: Network reconstruction accuracy for RAF pathway data. The histograms show the scenario-specific average precision-recall AUC values. Each AUC is averaged across 25 data sets and the error bars indicate standard deviations. The bars refer to: the homogeneous DBN (white), the NH-DBN model (light-grey), 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.

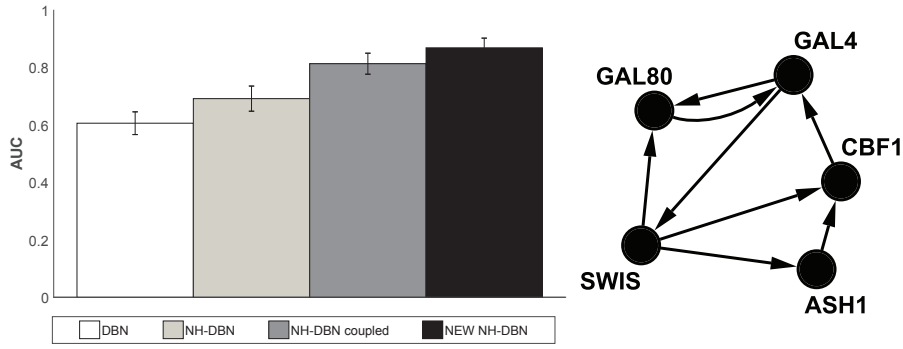

Figure 5.4: Network reconstruction accuracy for yeast gene expression data. The histogram shows the average precision-recall AUC values, averaged across 25 MCMC simulations, with error bars indicating standard deviations. The AUCs are: 0.61 (DBN), 0.69 (NH-DBN), 0.81 (coupled NH-DBN) and 0.87 (new NH-DBN). 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

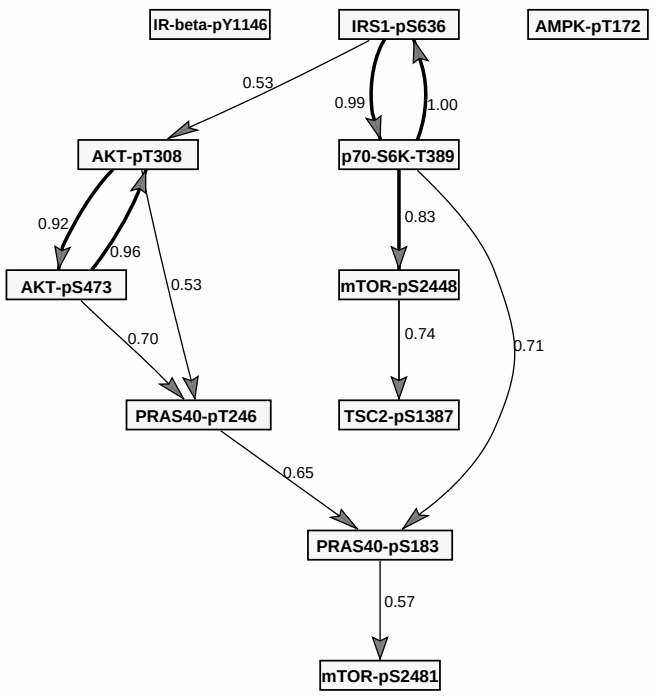

Figure 5.5: Predicted mTORC1 network topology. The 12 interactions whose scores exceeded the threshold $\psi = 0.5$; edges are labelled with their scores. The 5 edges with scores higher than $\psi = 0.8$ are represented in bold. The displayed interactions all had a higher posterior probability for being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

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

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:

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

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 $T := \sum_{k=1}^{K}(T_k - 1)$, 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 $p(\boldsymbol{\mu}|\boldsymbol{\Pi}, \boldsymbol{\delta})$ is a Gaussian, whose dimension is the number of component-specific coefficients, p($\boldsymbol{\Pi}$), is a uniform distribution, truncated to the maximal cardinality $|\boldsymbol{\Pi}| \leq 3$, and $p(\boldsymbol{\delta}|\boldsymbol{\Pi})$ 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 $\boldsymbol{\Pi}$ with $\boldsymbol{\delta}$ and $\boldsymbol{\mu}$. The vector $\boldsymbol{\delta}$ is then always sampled jointly with $\boldsymbol{\mu}$, and $\boldsymbol{\Pi}$ 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:

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

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

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

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

- In the deletion move (D) we randomly select one covariate $X \in \boldsymbol{\Pi}$, and we propose to remove this covariate from $\boldsymbol{\Pi}$. This yields $\boldsymbol{\Pi}_\bullet$. 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 $X \notin \boldsymbol{\Pi}$, and we propose to add this covariate to $\boldsymbol{\Pi}$. This yields $\boldsymbol{\Pi}_\bullet$. 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 $X_\bullet \in \boldsymbol{\Pi}$, and we propose to replace $X_\bullet$ by a randomly selected new covariate $X \notin \boldsymbol{\Pi}$. This yields $\boldsymbol{\Pi}_\bullet$. 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 $[\boldsymbol{\Pi}_\bullet, \boldsymbol{\delta}_\bullet]$, which we complete to a triple by sampling a new $\boldsymbol{\mu}_\bullet$, conditional on $\boldsymbol{\Pi}_\bullet$ 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:

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

where the Hastings-Ratio H is equal to:

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

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

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

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

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

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

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

(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 β_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:

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

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.