

Gene regulatory network inference using sparse probabilistic models

Evelina Gabašová

Supervisor: David Barber

MSc Computational Statistics and Machine Learning
University College London

September 2011

This report is submitted as part requirement for the MSc Degree in CSML at University College London. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

The main task of systems biology is to uncover mechanisms that regulate complex processes that take place in biological cells, especially the mechanisms of gene regulation. This project aims to identify gene regulatory interactions taking place in the early development of neural tube.

Solutions proposed in this work for identification of transcription factors and their target genes are mostly based on Bayesian sparse linear models. Sparse linear regression is used to predict gene expression given a set of transcription factor expressions. Even though transcriptional regulation mechanisms are frequently non-linear, a sparse linear regression model succeeds in identifying some of the transcription factors that are already known to play role in the studied biological processes. The probabilistic sparse model is compared to its frequentist equivalent, the lasso. Both algorithms are evaluated on neural tube development data and Bayesian model produces superior results.

The probabilistic sparse linear model is also extended to take into account non-linear interactions between transcription factors. A simple interaction model is presented and it is integrated into the probabilistic sparse linear model. The resulting model is a novel method to incorporate non-linearity into linear regression.

Variational approximate inference algorithm is derived for the extended model with transcription factor interactions. This model is then applied to a subset of potentially important transcription factors and it succeeds to find a very sparse solution. Again, the results from the extended model succeed to identify transcription factors that seem to have important role.

Both probabilistic models for gene regulation provide a set of straightforwardly testable predictions of potentially interesting transcription factors.

Acknowledgements

I would like to thank David Barber for useful guidance in this project. I would also like to thank Natascha Bushati and James Briscoe from the MRC National Institute for Medical Research for helpful biological insight into the problem of gene regulatory networks. Finally, I also thank all of the above mentioned and Chris Watkins for numerous discussions and inspirational comments.

Contents

1	Introduction	5
2	Sparse linear regression	7
2.1	Lasso	8
2.2	Bayesian sparse linear regression	9
2.2.1	The model	9
2.2.2	Variational approximation	11
3	Sparse linear regression with transcription factor interactions	17
3.1	Transcription factor interactions	17
3.2	The model	21
3.3	Variational approximation	23
4	Data	30
5	Experimental results	33
5.1	Sparse linear regression	33
5.1.1	Dataset filter5+	33
5.1.2	Explants datasets	41
5.1.3	Crossvalidation results	47
5.2	Sparse linear regression with transcription factor interactions	48
6	Summary	57
	Bibliography	59

Chapter 1

Introduction

Activity and development of biological cells is largely governed by expression of genes and their interactions. The main aim of systems biology is to uncover these interactions and relations that govern gene transcription in biological cells. The interconnected networks of interacting genes are so called gene regulatory networks. The most important parts of regulatory networks are transcription factors which are genes that control expression of other genes in the system.

Gene regulatory networks inference mainly aims to identify such gene and transcription factor interactions. The task of reverse-engineering regulatory networks is very difficult especially because of large dimensionality of the microarray data which measures gene expression and typically limited number of samples. This makes the network inference a largely underdetermined problem.

The main aim of this project is to uncover some key interactions that play role in the early development of neural tube. Generally, microarray data used for network inference can be divided into two types - steady state gene expression data or time series data. The steady state expressions correspond to gene expressions measured in specific experimental conditions. Time series data typically contain sequence of measurements that capture temporal dynamics of gene regulation. Data that were used in this project consists partly of very short time series and partly of steady state data. Because of insufficient length of time series, the data was treated as steady state. A detailed description of the data is given in Chapter 4.

Current methods used to discover gene regulatory networks can be grouped into several main types. Widely used group of techniques is based on systems of ordinary differential equations that are constructed to describe changes in gene expression with respect to

external perturbations [3]. By using linear model at the level of differential equations, the models deal with non-linearity on the expression levels. For example [8] uses linear regression on differential equations to identify transcription factors and their targets. These methods generally perform well on smaller scale problems with limited perturbations [3].

Other major group of network inference algorithms uses principles of information theory. For example, algorithm ARACNE [12] uses mutual information to identify co-regulated genes. Third widely used group of methods is based on Bayesian networks [7]. Such algorithms reconstruct the topology of regulatory network from expression data. Generally, this class of algorithms requires a large amount of data to estimate probabilities and evaluate different topologies [3].

Finally, a major group of algorithms deals with non-linearities in gene expression data by discretizing the expression values. Expression is reduced to binary values - gene is expressed or not. Regulatory relations are then modelled by boolean functions [2].

This project concentrates on identification of potentially interesting transcription factors and their targets from gene expression data using probabilistic machine learning methods. The basic model which is used in this project is based on sparse linear model of transcriptional control. A novel model based on linear regression that incorporates also non-linear relations between transcription factors is proposed. Both models are evaluated on the neural tube developmental data.

The report is organized as follows. Chapter 2 introduces the linear model of transcription control. It also provides an overview of two algorithms used for sparse linear regression - lasso and Bayesian model with sparse prior. For the Bayesian model it also describes a variational algorithm for inference in this model.

Chapter 3 introduces an extension of the Bayesian sparse linear model that incorporates non-linear interactions of transcription factors. The chapter also provides a variational approximation algorithm for the model. Chapter 4 gives a brief overview of the experimental data that was used to test the models presented in this report. Finally, Chapter 5 provides an evaluation of the standard and extended sparse linear models on the available experimental data.

Chapter 2

Sparse linear regression

The basic model for gene regulation used in this project is a linear model, similar for example to the model used in [9]. This model simplifies the mechanism of transcriptional control by the assumption of linear regulation. The model can be summarized as

$$x_i^n = \sum_{j=1}^K w_{ij} t_j^n + \epsilon \quad (2.1)$$

where x_i^n is the expression of gene x_i in experiment n , t_j^n is the expression of transcription factor t_j in experiment n , w_{ij} is a linear weight for the regulation of gene x_i by transcription factor t_j and finally ϵ is a random noise. The weight w_{ij} determines the regulatory strength of transcription factor t_j towards gene x_i .

The regulative relation between genes and transcription factors is assumed to be sparse. Genes in biological systems are generally assumed to be regulated only by a small number of transcription factors, which is the main source of sparsity. In terms of the model in Equation 2.1, w_{ij} are assumed to have non-zero values for only a small number of transcription factors for each gene x_i .

Based on the linearity and sparsity assumptions, two models for sparse linear regression were considered. Section 2.1 gives a short summary of the lasso, frequentist model for sparse linear regression. Section 2.2.1 introduces an equivalent Bayesian sparse linear model and finally Section 2.2.2 provides an approximate variational algorithm for inference in the probabilistic model.

2.1 Lasso

Lasso is a standard method to solve sparse linear regression tasks. This section gives a brief overview of the lasso and the least angle regression algorithm that can be used to effectively use the lasso in large linear problems.

Lasso [16] is a general regularization method used to find sparse solutions in linear regression. Lasso fits the standard linear regression model

$$\hat{x}_i^n = w_0 + w_{i1}t_1^n + w_{i2}t_2^n + \cdots + w_{iK}t_K^n \quad (2.2)$$

$$= w_0 + \sum_{j=1}^K w_{ij}t_j^n \quad (2.3)$$

by minimizing the sum of squared errors

$$\sum_{n=1}^N (x_i^n - \hat{x}_i^n)^2 \quad (2.4)$$

subject to a constraint on the total absolute sum of the regression coefficients

$$\sum_{k=1}^K |w_{ik}| \leq t \quad (2.5)$$

where x_i^n is the expression of gene i in experiment n (the target value); t_1^n, \dots, t_K^n are expressions of K transcription factors in experiment n ; variables $w_{i0}, w_{i1}, \dots, w_{iK}$ are regression coefficients corresponding to gene x_i and t is a specified constant.

For sufficiently large values of constant t , the model reduces to standard linear regression. By enforcing the constraint for smaller values of t , weights of non-relevant predictors are effectively driven to zero and the resulting solution is sparse. It is an example of shrinkage method that regularizes the linear regression model by penalizing size of the regression coefficients. Another example of a shrinkage method is for example ridge regression, which does not yield sparse solutions. As a result, the lasso solutions for small constant t select only a small number of transcription factors t_k that best predict the expression of each gene x_i throughout the experimental conditions $n = 1, \dots, N$.

Lasso was previously applied on the gene regulatory data in different settings for example in [14].

Generally, computing lasso solution is a quadratic programming task. However, it is

possible to use a more efficient algorithm, especially for large-scale problems such as the gene expression data. For the purpose of this project, an efficient implementation of the LARS (Least Angle Regression) algorithm [5] for R was used to compute the lasso. This algorithm provides the complete sequence of lasso solutions for increasing values of the constant t , up to a specified number of active predictors.

Application of the lasso on gene expression data is discussed in Chapter 5, where its results are compared to the results obtained from probabilistic sparse linear regression algorithm, which is described in the following section.

2.2 Bayesian sparse linear regression

This section describes a probabilistic model for sparse linear regression and algorithm for inference in such a model. The first part discusses the design of the model and especially the design of sparse priors for weights. The following Section 2.2.2 then shortly introduces the principles of variational approximation and derives variational EM algorithm for the model.

2.2.1 The model

Sparse linear regression in Bayesian settings solves the linear regression model as introduced in Equation 2.1.

The model is essentially the same linear regression model used for fitting lasso solutions in Section 2.1. In this case, priors are put on the variables in the model and sparse solution obtained by assuming that weights w_{ij} belong to a sparse probability distribution. The model then computes the posterior distribution under a sparse prior on the weights.

The sparse prior for weights can be designed in two different ways. The first method uses typically Laplace distribution. For example, in [15] and [?] a sparse Bayesian linear model with Laplace prior on weights is used on ordinary differential equations obtained from gene expression data. This method does not use a truly sparse prior, as the sparse structure is approximated by a continuous Laplace distribution. It is an example of a soft prior.

The other method of putting a sparse prior on linear weights is a so-called ‘spike and slab’ prior. This method models the prior distribution of each weight w_{ij} as a mixture distribution:

$$w_{ij} \sim (1 - \pi)\mathcal{N}(w_{ij}|0, \sigma_w^2) + \pi\delta_{ij} \tag{2.6}$$

where π is the probability that weight w_{ij} is non-zero and δ_{ij} is a discrete ‘spike’. With probability $(1 - \pi)$, the weight w_{ij} is normally distributed with zero mean. This effectively removes the corresponding transcription factor from the set of predictors for gene x_i . On the other hand, with probability π the weight has an arbitrary (non-zero) value. This method produces a truly sparse prior [11].

The main problem with the spike and slab prior is that the computation of the exact posterior is intractable. Because of the combinatorial character of mixture distribution on weights (Equation 2.6), it requires search over exponential number of possible models with respect to the total number of weights [11].

To facilitate approximate inference, the spike and slab distribution is re-parametrised in the following way:

$$\tilde{w}_{ij} = w_{ij}s_{ij} \quad (2.7)$$

such that

$$w_{ij} \sim \mathcal{N}(w_{ij}|0, \sigma_w^2) \quad (2.8)$$

$$s_{ij} \sim \text{Bernoulli}(\pi) \quad (2.9)$$

where \tilde{w}_{ij} is the original weight from Equation 2.6, w_{ij} is a linear regression weight and s_{ij} is a binary variable that controls the sparsity of the prior. The weights w_{ij} have zero-mean Gaussian prior distribution and sparse variables s_{ij} follow Bernoulli distribution. Its parameter π regulates the prior sparsity.

This re-parametrisation separates the sparsity from the weights and allows to treat them separately. Together s_{ij} and w_{ij} form a re-parametrised spike and slab prior on the sparse linear regression weights:

$$p(w_{ij}, s_{ij}) = \mathcal{N}(w_{ij}|0, \sigma_w^2)\pi^{s_{ij}}(1 - \pi)^{(1-s_{ij})} \quad (2.10)$$

The re-formulated Bayesian sparse linear regression model for gene expression data then takes the following form:

$$x_i^n = \sum_{j=1}^K \tilde{w}_{ij}t_j^n = \sum_{j=1}^K s_{ij}w_{ij}t_j^n \quad (2.11)$$

$$p(x_i^n | \mathbf{w}, \mathbf{s}, \mathbf{t}) = \mathcal{N}\left(x_i^n \left| \sum_{j=1}^K s_{ij}w_{ij}t_j^n, \sigma_i^2\right.\right) \quad (2.12)$$

where x_i^n is the expression of gene i in experimental condition n and t_j^n is the expression of transcription factor j in experimental condition n .

Or in vector notation:

$$\mathbf{x}_i = \sum_{j=1}^K \tilde{w}_{ij} \mathbf{t}_j = \sum_{j=1}^K s_{ij} w_{ij} \mathbf{t}_j \quad (2.13)$$

$$p(\mathbf{x}_i | \mathbf{w}, \mathbf{s}, \mathbf{t}) = \mathcal{N} \left(\mathbf{x}_i \left| \sum_{j=1}^K s_{ij} w_{ij} \mathbf{t}_j, \sigma_i^2 I \right. \right)$$

where $\mathbf{x}_i = (x_i^1, \dots, x_i^N)^\top$ are expressions of gene x_i through all experimental conditions $n = 1, \dots, N$, $\mathbf{t}_j = (t_j^1, \dots, t_j^N)^\top$ are expressions of transcription factor t_j in experiments $n = 1, \dots, N$, \tilde{w}_{ij} are sparse linear regression weights which are re-parametrized to $s_{ij} w_{ij}$, where s_{ij} is a binary variable and w_{ij} is Gaussian distributed weight.

Using the priors (2.9,2.8) and the linear regression model (2.13), the complete joint probability for the sparse model is

$$p(\mathbf{x}, \mathbf{s}, \mathbf{w} | \mathbf{t}) = \prod_{i=1}^M \prod_{j=1}^K \{ \pi^{s_{ij}} (1 - \pi)^{(1-s_{ij})} \mathcal{N}(w_{ij} | 0, \sigma_w^2) \} \prod_{i=1}^M \mathcal{N} \left(\mathbf{x}_i \left| \sum_{j=1}^K s_{ij} w_{ij} \mathbf{t}_j, \sigma_i^2 I \right. \right), \quad (2.14)$$

where M is the total number of genes and K is the number of transcription factors. In this model, the gene expressions \mathbf{x} and transcription factor expressions \mathbf{t} are observed variables. The sparse variables s and weights w are hidden variables, and π , σ_w^2 and σ_i^2 are hyperparameters of the prior distribution.

This form of the model now facilitates the use of variational expectation-maximisation (EM) algorithm, which is described in the following section.

2.2.2 Variational approximation

This section gives a brief overview of the variational approximate inference method as it can be applied to the sparse linear regression model (2.13). It discusses a standard mean field approximation technique and a more efficient paired mean field algorithm.

Variational EM algorithm follows the principles of the standard EM algorithm [13] which is widely used for learning in probabilistic models with hidden variables. The EM

algorithm optimises free energy function

$$\begin{aligned}\mathcal{F}(q, \boldsymbol{\theta}) &= \langle \log p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) \rangle_{q(\mathbf{h})} + \mathbf{H}(q) \\ &= \langle \log p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) \rangle_{q(\mathbf{h})} - \langle \log q(\mathbf{h}) \rangle_{q(\mathbf{h})},\end{aligned}\tag{2.15}$$

where q is a distribution over the hidden variables in the model, $\boldsymbol{\theta}$ are the parameters of the model, \mathbf{x} are observed variables and \mathbf{h} are hidden variables of the model. The term $\mathbf{H}(q)$ represents the entropy of q and $\langle \cdot \rangle_q$ denotes the expectation with respect to the distribution q .

The free energy forms a lower bound on the likelihood [13]. By maximizing the free energy, it is possible to find the maximum likelihood setting for the parameters of the model and the posterior distribution over hidden variables in the model.

The EM algorithm proceeds in two alternating steps:

E step

$$q(\mathbf{h})^{(t+1)} \leftarrow \arg \max_{q(\mathbf{h})} \mathcal{F}(q, \boldsymbol{\theta}^{(t)}) = p(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta}^{(t)})\tag{2.16}$$

M step

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \arg \max_{\boldsymbol{\theta}} \mathcal{F}(q^{(t+1)}, \boldsymbol{\theta})\tag{2.17}$$

The E step finds the distribution $q(\mathbf{h})$ that maximises free energy (2.15) while the parameters $\boldsymbol{\theta}$ are held fixed. The M step finds values of parameters $\boldsymbol{\theta}$ that maximise the free energy (2.15) for fixed q . The two steps are iterated until convergence.

For our sparse linear regression model, computation of the posterior $p(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta})$ is intractable. Therefore, an approximate method has to be used. Variational approximation solves the problem by assuming a simplified tractable form of the distribution over hidden variables q . The E step then optimizes free energy $\mathcal{F}(q, \boldsymbol{\theta})$ with respect to a tractable approximation of q . The simplified tractable distribution typically comes from a suitable factorization of q over subgroups of variables in the model.

Standard mean field variational approximation factorizes q over all hidden variables [4]

$$q(\mathbf{h}) = \prod_i q_i(\mathbf{h}_i).\tag{2.18}$$

For our sparse linear regression model (2.13), the mean field approximation fully fac-

torizes over sparse variables \mathbf{s} and weights \mathbf{w}

$$q(\mathbf{s}, \mathbf{w}) = \prod_{i=1}^M \prod_{j=1}^K q(s_{ij})q(w_{ij}) \quad (2.19)$$

$$= \prod_{i=1}^M \prod_{j=1}^K \{\lambda_{ij}^{s_{ij}}(1 - \lambda_{ij})^{(1-s_{ij})}\} \{\mathcal{N}(w_{ij}|\mu_{ij}, \sigma_{ij}^2)\}. \quad (2.20)$$

where $\lambda_{ij} = q(s_{ij} = 1)$ and $(\mu_{ij}, \sigma_{ij}^2)$ are variational parameters for w_{ij} . This factorization was used in the context of sparse models for example in [17].

However, this approximation leads to a very inefficient algorithm. It essentially decouples the variables s_{ij} and w_{ij} and updates them separately. In fact, s_{ij} and w_{ij} are strongly correlated. Using variational approximation, if the sparse variable s_{ij} is set to 0, it blocks adaptation of the weight w_{ij} . A more detailed analysis of some of the problems that this factorization poses is given in [11].

A more efficient factorization, a paired mean field method, is proposed in [11]. This approach does not separate s_{ij} and w_{ij} and treats each pair together. The modified factorization has the following form:

$$q(\mathbf{s}, \mathbf{w}) = \prod_{i=1}^M \prod_{j=1}^K q(s_{ij}, w_{ij}) \quad (2.21)$$

$$= \prod_{i=1}^M \prod_{j=1}^K q(s_{ij})q(w_{ij}|s_{ij}) \quad (2.22)$$

$$= \prod_{i=1}^M \prod_{j=1}^K [q(s_{ij} = 0)q(w_{ij}|s_{ij} = 0) + q(s_{ij} = 1)q(w_{ij}|s_{ij} = 1)] \quad (2.23)$$

This approximation preserves the natural coupling of s_{ij} and w_{ij} .

The conditional distributions $q(w_{ij}|s_{ij} = 0)$ and $q(w_{ij}|s_{ij} = 1)$ turn out to be Gaussian distributed [11]:

$$w_{ij}|s_{ij} = 0 \sim \mathcal{N}(w_{ij}|0, \sigma_w^2) \quad (2.24)$$

$$w_{ij}|s_{ij} = 1 \sim \mathcal{N}(w_{ij}|\mu_{ij}, \sigma_{ij}^2) \quad (2.25)$$

The complete approximation then becomes a mixture of two Gaussian distributions for

each weight w_{ij} :

$$q(\mathbf{s}, \mathbf{w}) = \prod_{i=1}^M \prod_{j=1}^K [q(s_{ij} = 0)q(w_{ij}|s_{ij} = 0) + q(s_{ij} = 1)q(w_{ij}|s_{ij} = 1)] \quad (2.26)$$

$$= \prod_{i=1}^M \prod_{j=1}^K [(1 - \lambda_{ij})^{(1-s_{ij})} \mathcal{N}(w_{ij}|0, \sigma_w^2) + \lambda_{ij}^{s_{ij}} \mathcal{N}(w_{ij}|\mu_{ij}, \sigma_{ij}^2)], \quad (2.27)$$

where $\lambda_{ij} = q(s_{ij} = 1)$. Under this factorization, if the sparse variable s_{ij} is equal to 1, the corresponding weight w_{ij} is normally distributed with a mean μ_{ij} and variance σ_{ij}^2 . On the other hand if s_{ij} is set to 0, the distribution of w_{ij} is equal to its prior distribution with zero mean.

Using the joint distribution for our sparse linear model $p(\mathbf{x}, \mathbf{s}, \mathbf{w}|\mathbf{t})$ defined in Equation 2.14 and the variational distribution $q(\mathbf{s}, \mathbf{w})$ 2.27, the variational approximation algorithm maximizes free energy term

$$\mathcal{F}(q(\mathbf{s}, \mathbf{w}), \boldsymbol{\theta}) = \langle \log p(\mathbf{x}, \mathbf{s}, \mathbf{w}|\mathbf{t}, \boldsymbol{\theta}) \rangle_{q(\mathbf{s}, \mathbf{w})} - \langle \log q(\mathbf{s}, \mathbf{w}) \rangle_{q(\mathbf{s}, \mathbf{w})}, \quad (2.28)$$

where $\boldsymbol{\theta} = (\pi, \sigma_i^2, \sigma_w^2)$ are the hyperparameters of the model. Plugging in the distributions we get the full free energy in Equation 2.29.

E step

The E step of the variational EM algorithm finds the parameters of the variational distribution $q(\mathbf{s}, \mathbf{w})$ that maximize the free energy. By differentiating \mathcal{F} with respect to the parameters of $q(s_{ij}, w_{ij}) = q(s_{ij})q(w_{ij}|s_{ij})$ we get update equations.

Variational update for the distribution of binary variables $q(s_{ij})$ is

$$q(s_{ij} = 1) = \lambda_{ij} = f \left(\log \frac{\pi}{1 - \pi} + \frac{1}{2} \log \frac{\sigma_i^2}{\sigma_w^2} - \frac{1}{2} \log \left(\mathbf{t}_j^\top \mathbf{t}_j + \frac{\sigma_i^2}{\sigma_w^2} \right) + \frac{1}{2\sigma_i^2} \frac{\left(\mathbf{x}_i^\top \mathbf{t}_j - \sum_{k \neq j} \langle s_{ik} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k \right)^2}{\mathbf{t}_j^\top \mathbf{t}_j + \frac{\sigma_i^2}{\sigma_w^2}} \right), \quad (2.30)$$

where f is the logistic function

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (2.31)$$

$$\begin{aligned}
\mathcal{F}(q(\mathbf{s}, \mathbf{w}), \boldsymbol{\theta}) &= -\frac{MN}{2} \log(2\pi) - \frac{N}{2} \sum_{i=1}^M \log(\sigma_i^2) - \frac{1}{2} \sum_{i=1}^M \frac{1}{\sigma_i^2} \mathbf{x}_i^\top \mathbf{x}_i \\
&+ \sum_{i=1}^M \frac{1}{\sigma_i^2} \sum_{j=1}^K \langle s_{ij} w_{ij} \rangle \mathbf{t}_j^\top \mathbf{x}_i - \frac{1}{2} \sum_{i=1}^M \frac{1}{\sigma_i^2} \sum_{j=1}^K \langle s_{ij} (w_{ij})^2 \rangle \mathbf{t}_j^\top \mathbf{t}_j \\
&- \sum_{i=1}^M \frac{1}{\sigma_i^2} \sum_{j=1}^K \sum_{k=j+1}^K \langle s_{ij} w_{ij} \rangle \langle s_{ik} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k \\
&- \frac{1}{2\sigma_w^2} \sum_{i=1}^M \sum_{j=1}^K \langle w_{ij}^2 \rangle \\
&+ \log(\pi) \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle + \log(1 - \pi) \sum_{i=1}^M \sum_{j=1}^K (1 - \langle s_{ij} \rangle) \\
&- \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle \log \langle s_{ij} \rangle - \sum_{i=1}^M \sum_{j=1}^K (1 - \langle s_{ij} \rangle) \log(1 - \langle s_{ij} \rangle) \\
&- \frac{1}{2} \log(\sigma_w^2) \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle + \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle \log(\sigma_{ij}^2)
\end{aligned} \tag{2.29}$$

Figure 2.1: Free energy for the sparse linear model with spike and slab prior.

Variational updates for the conditional distribution $q(w_{ij}|s_{ij})$ are divided into two cases for $s_{ij} = 0$ and $s_{ij} = 1$:

$$q(w_{ij}|s_{ij} = 0) = \mathcal{N}(w_{ij}|0, \sigma_w^2) \quad (2.32)$$

$$q(w_{ij}|s_{ij} = 1) = \mathcal{N}(w_{ij}|\mu_{ij}, \sigma_{ij}^2) \quad (2.33)$$

If $s_{ij} = 0$, $w_{ij}|s_{ij}$ follows its prior distribution. When $s_{ij} = 1$, the weights have Gaussian distribution with parameters

$$\mu_{ij} = \frac{\mathbf{x}_i^\top \mathbf{t}_j - \sum_{k \neq j} \langle s_{ik} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k}{\mathbf{t}_j^\top \mathbf{t}_j + \frac{\sigma_i^2}{\sigma_w^2}} \quad \sigma_{ij}^2 = \frac{\sigma_i^2}{\mathbf{t}_j^\top \mathbf{t}_j + \frac{\sigma_i^2}{\sigma_w^2}}. \quad (2.34)$$

Under the variational distribution $q(s_{ij}, w_{ij})$, the expectations that are used in the updates have the following form:

$$\langle s_{ij} w_{ij} \rangle = \lambda_{ij} \mu_{ij} \quad \langle s_{ij} w_{ij}^2 \rangle = \lambda_{ij} (\mu_{ij}^2 + \sigma_{ij}^2) \quad (2.35)$$

M step

The hyperparameters of the model (2.14) are optimized in the M step:

$$\pi = \frac{1}{MK} \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle \quad (2.36)$$

$$\sigma_w^2 = \frac{1}{\sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle} \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} w_{ij}^2 \rangle \quad (2.37)$$

$$\sigma_i^2 = \frac{1}{N} \left\{ \mathbf{x}_i^\top \mathbf{x}_i - 2 \sum_{j=1}^K \langle s_{ij} w_{ij} \rangle \mathbf{t}_j^\top \mathbf{x}_i + \sum_{j=1}^K \langle s_{ij} w_{ij}^2 \rangle \mathbf{t}_j^\top \mathbf{t}_j + \sum_{j=1}^K \sum_{k \neq j}^K \langle s_{ij} w_{ij} \rangle \langle s_{ik} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k \right\} \quad (2.38)$$

Starting with a random initialization of the hidden variables and unknown parameters, the E and M steps are iterated until convergence.

Application of the described approximate variational EM algorithm on the gene expression data is described in Chapter 5. The method was extensively applied to gene expression data and compared to lasso results.

Chapter 3

Sparse linear regression with transcription factor interactions

One of the main disadvantages of using linear regression especially on gene expression data is that transcription factors frequently interact in a non-linear fashion. Some transcription factors inhibit each other, other transcription factors behave differently in isolation and in combination with different transcription factors [1]. This property of transcription regulatory networks partly limits the use of linear regression models.

This section proposes a novel extension of the Bayesian sparse linear regression model which incorporates simple pairwise non-linear interactions of transcription factors. This approach allows identification of wider range of possible relations in gene regulatory networks.

Sections 3.1 and 3.2 give an overview of the extended linear regression model and how the pairwise interactions are incorporated into the sparse model. Section 3.3 provides variational algorithm for approximate inference in this model.

3.1 Transcription factor interactions

To capture non-linear pairwise interactions of transcription factors, it is possible to expand the set of transcription factor expressions to include also some form of interaction pseudo-expressions. For purpose of the extended model, the following four basic types of pairwise interactions between a pair of transcription factors t_1 and t_2 were considered:

1. **t_1 low, t_2 low** Both transcription factors t_1 and t_2 are underexpressed. Their absence can for example allow expression of other gene.

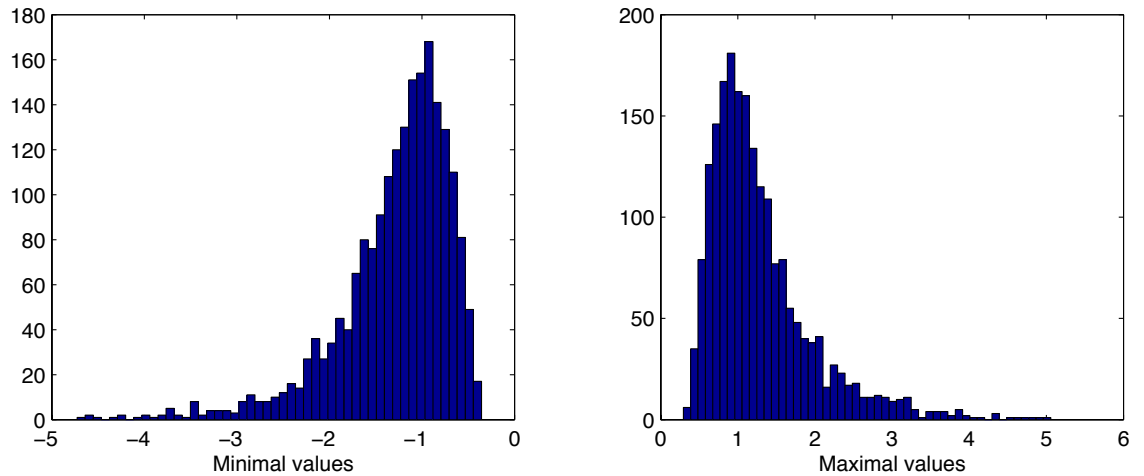


Figure 3.1: Histograms of minimal and maximal values of RMA normalized gene expressions in the `explants` dataset.

2. **t_1 high, t_2 low** Expression of transcription factor t_1 is high whereas t_2 is underexpressed.
3. **t_1 low, t_2 high** Transcription factor t_2 is highly expressed, t_1 has low expression.
4. **t_1 high, t_2 high** Both transcription factors t_1 and t_2 are highly expressed.

The listed types of interactions are considered for each pair of transcription factors that are chosen to be included in the model. To construct the actual interaction pseudo-expressions that would correspond to the four basic interaction types, prototype values were specified for *low* and *high* expressions.

Figure 3.1 shows the empirical distributions of maximal and minimal values of RMA normalized expressions observed in the `explants` dataset. The expressions were also normalized to zero mean for each gene. Based on these distributions, *low* value was set heuristically to -2 and *high* to 2. The expression values of transcription factors were also cut to the interval $[-2,2]$ to avoid extreme values. This essentially eliminates the tails from the distributions in Figure 3.1. The cut-off interval was used because it does not unnecessarily amplify noise in the data. After setting the values, the four interaction types are specified by prototypical expression vectors $(p_1, p_2)^\top$

1. **t_1 low, t_2 low:** $(p_1, p_2)^\top \approx (-2, -2)^\top$
2. **t_1 high, t_2 low:** $(p_1, p_2)^\top \approx (2, -2)^\top$

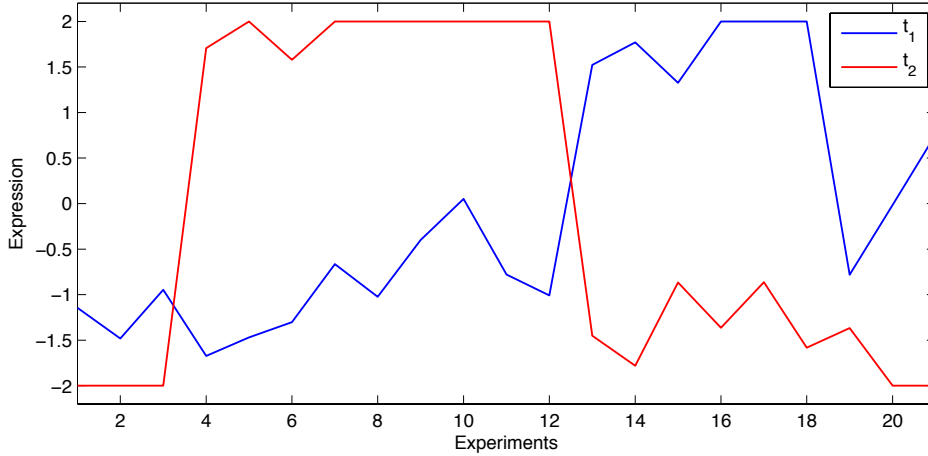


Figure 3.2: Expression of a sample pair of transcription factors.

3. **t_1 low, t_2 high:** $(p_1, p_2)^\top \approx (-2, 2)^\top$

4. **t_1 high, t_2 high:** $(p_1, p_2)^\top \approx (2, 2)^\top$

To include interactions in the model, interaction pseudo-expression vector is computed for each transcription factor pair. Interaction expression is high if the corresponding pair of transcription factor expressions corresponds to its prototypical vector, and low if the transcription factors behave in an opposite way from the prototype.

To produce actual interaction vectors for each pair of transcription factors, the following function was used to compute the pseudo-expression based on the similarity between each prototypical vector and each transcription factor pair \mathbf{t}_j and $\mathbf{t}_k, j \neq k$:

$$f_l(\mathbf{t}_j, \mathbf{t}_k) = \exp \left\{ -\gamma \left((t_j - p_1)^2 + (t_k - p_2)^2 \right) \right\}, \quad (3.1)$$

where $f_l, l = 1, \dots, 4$ represents the four different interaction types and γ is set to 0.1. To bring the values of the interaction vectors closer to real expression values, they are also re-normalized to zero mean and unit variance.

Figure 3.2 shows example expression profiles of a sample pair of transcription factors across 21 experiments, taken from the `explants` dataset. Transcription factor t_1 has mostly low expression in the first half of the experiments (experiments 4 to 12) and mostly high expression in the rest. Transcription factor t_2 behaves in the opposite fashion with high expression in the beginning and low expression in the second half of the experiments.

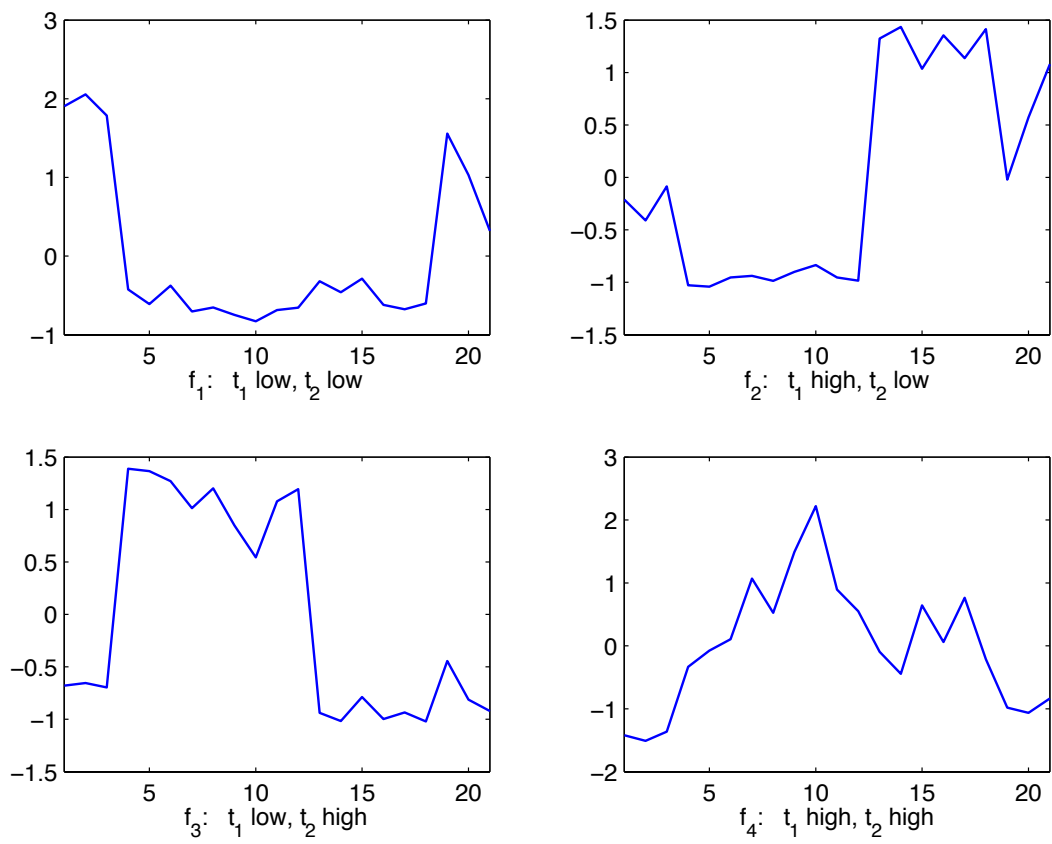


Figure 3.3: Interaction functions for a sample pair of transcription factors.

Figure 3.3 gives the interaction pseudo-expressions for the transcription factor pair shown in Figure 3.2. The figure presents all four interaction types. By comparing both figures, the interaction functions follow the relations which correspond to the four interaction types. For example, f_1 interaction expression is high whenever both t_1 and t_2 are underexpressed.

This model of transcription factor interactions is relatively flexible as it allows any type of non-linear interaction to be incorporated. The presented four basic interaction types are just an example and more sophisticated interaction functions based on biological evidence can be included.

The main limitation is that each interaction effectively significantly increases the size of the model. Number of transcription factor pairs in the model has to be limited, as interactions dramatically increase the possible number of predictors. Given K transcription factors, we get $\frac{K(K-1)}{2}$ possible distinct pairs and each pair is represented by $L = 4$ different interaction expressions. Therefore, the model is suited for smaller scale analyses.

3.2 The model

The most important aim of the design of incorporation of transcription factor interactions into the sparse linear regression model is that it is undesirable to treat the interactions $f_l(\mathbf{t}_j, \mathbf{t}_k)$ in the same way as the individual expressions of transcription factors \mathbf{t}_j and \mathbf{t}_k . The model should use interactions as little as possible, only in necessary cases. As we are looking for a sparse solution, we want as little as possible transcription factors to be involved in the solution.

To incorporate the interactions into the sparse linear model, the original regression model given in Equation 2.13 is modified to the form

$$\mathbf{x}_i = \sum_{j=1}^K \tilde{w}_{ij} \mathbf{t}_j + \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \tilde{w}_{ijk}^l f_l(\mathbf{t}_j, \mathbf{t}_k), \quad (3.2)$$

where $L = 4$ is the number of interaction types and $f_l(\mathbf{t}_j, \mathbf{t}_k)$ represents the corresponding interaction (3.1).

In this model, each interaction $f_l(\mathbf{t}_j, \mathbf{t}_k)$ has its own weight \tilde{w}_{ijk}^l which represents the linear influence of the interaction of type l between transcription factors \mathbf{t}_j and \mathbf{t}_k on gene \mathbf{x}_i . To reflect the fact that we want the interaction terms used only in reasonable cases and that it is desirable to have as few as possible transcription factors involved overall, the

following re-parametrization is applied:

$$\begin{aligned}
\tilde{w}_{ij} &= s_{ij}w_{ij} \\
\tilde{w}_{ijk}^l &= s_{ij}s_{ik}w_{ijk}^l \\
\mathbf{x}_i &= \sum_{j=1}^K s_{ij}w_{ij}\mathbf{t}_j + \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L s_{ij}s_{ik}w_{ijk}^l f_l(\mathbf{t}_j, \mathbf{t}_k),
\end{aligned} \tag{3.3}$$

where s_{ij} are binary variables.

The interaction weight \tilde{w}_{ijk}^l does not have an individual sparse variable s . Instead, it shares the sparse variables s_{ij} and s_{ik} that correspond to both transcription factors \mathbf{t}_j and \mathbf{t}_k . This setting allows the interaction to be effectively used only if both s_{ij} and s_{ik} are set to 1, i.e. only if both corresponding transcription factors are used individually for a given gene \mathbf{x}_i . It prevents the model from using just the non-linear interactions in situations where they are not necessary. The model still allows the interactions to be used in isolation because the weights for transcription factors \mathbf{t}_j and \mathbf{t}_k can be set to 0 even if s_{ij} and s_{ik} are equal to 1. However, this situation may occur only if there is a strong evidence for it.

The re-parametrization given in Equation 3.3 is used to put a sparse spike and slab priors on the weights \tilde{w}_{ij} and \tilde{w}_{ijk}^l :

$$s_{ij} \sim \text{Bernoulli}(\pi) \tag{3.4}$$

$$w_{ij} \sim \mathcal{N}(w_{ij}|0, \sigma_w^2) \tag{3.5}$$

$$w_{ijk}^l \sim \mathcal{N}(w_{ijk}^l|0, \sigma_w^2). \tag{3.6}$$

Parameter π regulates the prior sparsity of the model. Both sets of weights have prior means equal to zero and equal variance σ_w^2 . Then

$$\mathbf{x}_i \sim \mathcal{N}\left(\mathbf{x}_i \left| \sum_{j=1}^K s_{ij}w_{ij}\mathbf{t}_j + \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L s_{ij}s_{ik}w_{ijk}^l f_l(\mathbf{t}_j, \mathbf{t}_k), \sigma_i^2 \mathbf{I} \right.\right), \tag{3.7}$$

where \mathbf{x}_i is the expression of gene i through experiments $n = 1, \dots, N$, \mathbf{t}_j and \mathbf{t}_k are expressions of transcription factors in the same set of experiments and $f_l(\mathbf{t}_j, \mathbf{t}_k)$ represents their interaction of type $l = 1, \dots, L$.

The joint probability for the linear regression model with transcription factor interactions has the following form:

$$\begin{aligned}
p(\mathbf{x}, \mathbf{w}, \mathbf{w}^l, \mathbf{s}|\mathbf{t}) = & \\
& \left\{ \prod_{i=1}^M \prod_{j=1}^K \pi^{s_{ij}} (1 - \pi)^{(1-s_{ij})} \right\} \left\{ \prod_{i=1}^M \prod_{j=1}^K \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{w_{ij}^2}{2\sigma_w^2}\right) \right\} \\
& \left\{ \prod_{i=1}^M \prod_{j=1}^K \prod_{k=j+1}^K \prod_{l=1}^L \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{(w_{ijk}^l)^2}{2\sigma_w^2}\right) \right\} \\
& \left\{ \prod_{i=1}^M \frac{1}{(2\pi\sigma_i^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_i^2} \left(\mathbf{x}_i - \sum_{j=1}^K s_{ij} w_{ij} \mathbf{t}_j - \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L s_{ij} s_{ik} w_{ijk}^l f_l(\mathbf{t}_j, \mathbf{t}_k) \right)^2 \right) \right\}
\end{aligned} \tag{3.8}$$

The following section describes variational inference algorithm developed for this model.

3.3 Variational approximation

This section gives an overview of variational EM algorithm as it can be used to fit the sparse linear regression model with transcription factor interactions described in Equation 3.3. The variational approximation is based on the method of paired mean field method [11] which is discussed in Section 2.2.2. Section 2.2.2 provides also a general description of variational approximate inference and variational EM algorithm.

Variational EM algorithm maximizes the free energy:

$$\mathcal{F}(q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l), \boldsymbol{\theta}) = \langle \log p(\mathbf{x}, \mathbf{s}, \mathbf{w}, \mathbf{w}^l | \mathbf{t}, \boldsymbol{\theta}) \rangle_{q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l)} - \langle \log q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l) \rangle_{q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l)}, \tag{3.9}$$

where q is a distribution over the hidden variables in the model \mathbf{s}, \mathbf{w} and \mathbf{w}^l . Again, exact optimization of \mathcal{F} is intractable and it is necessary to use approximate inference.

To apply variational method on our model with transcription factor interactions, a suitable factorization of the distribution $q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l)$ into simpler tractable distributions is based on the efficient paired mean field method [11]. This method allows the variable groups $(\mathbf{s}_{ij}, \mathbf{w}_{ij})$ and also $(\mathbf{s}_{ij}, \mathbf{s}_{ik}, \mathbf{w}_{ijk}^l)$ to be treated as units and respects strong correlations between them. Again, a more detailed overview of this method [11] is given in Section 2.2.2.

The paired mean field method results in the following factorization of $q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l)$:

$$q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l) = \prod_{i=1}^M \prod_{j=1}^K \left\{ q(s_{ij}) q(w_{ij} | s_{ij}) \prod_{k=j+1}^K \prod_{l=1}^L q(w_{ijk}^l | s_{ij} s_{ik}) \right\} \quad (3.10)$$

$$= \prod_{i=1}^M \prod_{j=1}^K \left\{ \lambda_{ij}^{s_{ij}} (1 - \lambda_{ij})^{1-s_{ij}} q(w_{ij} | s_{ij}) \prod_{k=j+1}^K \prod_{l=1}^L q(w_{ijk}^l | s_{ij} s_{ik}) \right\}, \quad (3.11)$$

where $\lambda_{ij} = q(s_{ij} = 1)$. This factorization respects the natural coupling of weights and corresponding sparse variables, including also the newly introduced interaction weights w_{ijk}^l .

The conditional distributions for weights given values of sparse variables s that arise from the factorization have the following forms:

$$q(w_{ij} | s_{ij}) = \begin{cases} \mathcal{N}(w_{ij} | 0, \sigma_w^2) & \text{if } s_{ij} = 0 \\ \mathcal{N}(w_{ij} | \mu_{ij}, \sigma_{ij}^2) & \text{if } s_{ij} = 1 \end{cases} \quad (3.12)$$

and

$$q(w_{ijk}^l | s_{ij}, s_{ik}) = \begin{cases} \mathcal{N}(w_{ijk}^l | 0, \sigma_w^2) & \text{if } s_{ij} = 0 \text{ or } s_{ik} = 0 \\ \mathcal{N}(w_{ijk}^l | \mu_{ijk}^l, \sigma_{ijk}^2) & \text{if } s_{ij} = 1 \text{ and } s_{ik} = 1 \end{cases} \quad (3.13)$$

The distribution of each weight becomes a mixture distribution of two Gaussians. Under the factorization given in Equation 3.11, if the sparse variable s_{ij} is equal to 1, the corresponding weight w_{ij} is normally distributed with a mean μ_{ij} and variance σ_{ij}^2 . If s_{ij} is set to 0, the distribution of w_{ij} is equal to its prior distribution with zero mean. Similar properties apply for the interaction weights w_{ijk}^l . If both sparse variables s_{ij}, s_{ik} are set to 1, w_{ijk}^l is normally distributed with its specific mean μ_{ijk}^l and variance σ_{ijk}^2 . On the other hand, if either of the sparse variables s_{ij} or s_{ik} is equal to 0, weight w_{ijk}^l retains its prior distribution with zero mean.

Using the factorization (3.11) and the joint probability (3.8), the free energy function (3.9) can be fully instated. Due to the interactions, free energy takes a more complex form:

$$\mathcal{F} = \langle \log p(\mathbf{s}, \mathbf{w}, \mathbf{w}^l, \mathbf{x}|\mathbf{t}) \rangle_{q(\mathbf{s}, \mathbf{s}, \mathbf{s}^l)} - \langle \log q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l) \rangle_{q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l)} \quad (3.14)$$

$$= \log \pi \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle + \log(1 - \pi) \sum_{i=1}^M \sum_{j=1}^K (1 - \langle s_{ij} \rangle) \quad (3.15)$$

$$- \frac{MK}{2} \log 2\pi - \frac{MK}{2} \log \sigma_w^2 - \frac{1}{2\sigma_w^2} \sum_{i=1}^M \sum_{j=1}^K \langle w_{ij}^2 \rangle \quad (3.16)$$

$$- \frac{MK(K-1)L}{4} \log 2\pi \sigma_w^2 - \frac{1}{2\sigma_w^2} \sum_{i=1}^M \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle (w_{ijk}^l)^2 \rangle \quad (3.17)$$

$$- \frac{MN}{2} \log 2\pi - \frac{N}{2} \sum_{i=1}^M \log \sigma_i^2 \quad (3.18)$$

$$- \frac{1}{2} \sum_{i=1}^M \frac{1}{\sigma_i^2} \left(\mathbf{x}_i - \sum_{j=1}^K \langle s_{ij} w_{ij} \rangle \mathbf{t}_j - \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle s_{ij} s_{ik} w_{ijk}^l \rangle f_l(\mathbf{t}_j, \mathbf{t}_k) \right)^\top \quad (3.19)$$

$$\left(\mathbf{x}_i - \sum_{j=1}^K \langle s_{ij} w_{ij} \rangle \mathbf{t}_j - \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle s_{ij} s_{ik} w_{ijk}^l \rangle f_l(\mathbf{t}_j, \mathbf{t}_k) \right) \quad (3.20)$$

$$- \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle \log \langle s_{ij} \rangle - \sum_{i=1}^M \sum_{j=1}^K (1 - \langle s_{ij} \rangle) \log(1 - \langle s_{ij} \rangle) \quad (3.21)$$

$$+ \frac{MK}{2} \log \sigma_w^2 + \frac{MK}{2} \log 2\pi - \frac{1}{2} \log \sigma_w^2 \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle + \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle \log \sigma_{ij}^2 \quad (3.22)$$

$$+ \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle s_{ij} s_{ik} \rangle \log \sigma_{ijkl}^2 \quad (3.23)$$

$$+ \frac{MK(K-1)L}{4} \log 2\pi + \frac{MK(K-1)L}{4} \log \sigma_w^2 - \frac{L}{2} \log \sigma_w^2 \sum_{i=1}^M \sum_{j=1}^K \sum_{k=j+1}^K \langle s_{ij} s_{ik} \rangle. \quad (3.24)$$

By differentiating \mathcal{F} with respect to the unknown parameters, we obtain the E and M steps of the variational EM algorithm, similarly as in Section 2.2.2.

E step

The E step updates parameters of the variational distribution $q(\mathbf{s}, \mathbf{w}, \mathbf{w}^l)$. Updates for distributions of sparse variables $s_{ij}, i = 1, \dots, M, j = 1, \dots, K$ are more complex than in the standard sparse linear regression case (Section 2.2.2) because each variable s_{ij} regulates the sparsity of w_{ij} as well as the sparsity of w_{ijk}^l (together with s_{ik}).

$$q(s_{ij} = 1) = f(u) \quad f(x) = \frac{1}{1 + \exp(-x)} \quad (3.25)$$

and

$$\begin{aligned} u = & \log \frac{\pi}{1 - \pi} + \frac{1}{2} \log \frac{\sigma_{ij}^2}{\sigma_w^2} + \frac{1}{2} \sum_{k \neq j}^K \sum_{l=1}^L \langle s_{ik} \rangle \log \sigma_{ijkl}^2 - \frac{L}{2} \log \sigma_w^2 \sum_{k \neq j}^K \langle s_{ik} \rangle \\ & + \frac{1}{\sigma_i^2} \left\{ \langle w_{ij} \rangle \mathbf{t}_j^\top \mathbf{x}_i - \frac{1}{2} \langle w_{ij}^2 \rangle \mathbf{t}_j^\top \mathbf{t}_j + \sum_{k \neq j}^K \sum_{l=1}^L \langle s_{ik} w_{ijk}^l \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top \mathbf{x}_i \right. \\ & - \sum_{k \neq j}^K \langle s_{ik} w_{ij} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k - \sum_{k \neq j}^K \sum_{l=1}^L \langle s_{ik} w_{ij} w_{ijk}^l \rangle \mathbf{t}_j^\top f_l(\mathbf{t}_j, \mathbf{t}_k) - \sum_{k \neq j}^K \sum_{l=1}^L \langle s_{ik} w_{ik} w_{ijk}^l \rangle \mathbf{t}_k^\top f_l(\mathbf{t}_j, \mathbf{t}_k) \\ & - \sum_{\substack{j' \neq j \\ k \neq j}}^k \sum_{k=j'+1}^K \sum_{l=1}^L \langle w_{ij} s_{ij'} s_{ik} w_{ij'k}^l \rangle \mathbf{t}_j^\top f_l(\mathbf{t}_{j'}, \mathbf{t}_k) - \sum_{j' \neq j}^k \sum_{k \neq j, j'}^K \sum_{l=1}^L \langle s_{ij'} s_{ik} w_{ij'} w_{ij'k}^l \rangle \mathbf{t}_{j'}^\top f_l(\mathbf{t}_j, \mathbf{t}_k) \\ & - \frac{1}{2} \sum_{k \neq j}^K \sum_{l=1}^L \sum_{l'=1}^L \langle s_{ik} w_{ijk}^l w_{ijk}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_j, \mathbf{t}_k) \\ & - \sum_{k \neq j}^K \sum_{\substack{k'=k+1 \\ k' \neq j}}^K \sum_{l=1}^L \sum_{l'=1}^L \langle s_{ik} s_{ik'} \rangle \left(\langle w_{ijk}^l w_{ijk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_j, \mathbf{t}_{k'}) \right. \\ & \quad \left. + \langle w_{ijk}^l w_{ikk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{k'}) + \langle w_{ij'k'}^l w_{ikk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_{k'})^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{k'}) \right) \\ & - \sum_{\substack{j' \neq j \\ k \neq j}}^K \sum_{k=j'+1}^K \sum_{\substack{k'=k+1 \\ k' \neq j}}^K \sum_{l=1}^L \sum_{l'=1}^L \langle s_{ij'} s_{ik} s_{ik'} \rangle \left(\langle w_{ij'j'}^l w_{ikk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_{j'})^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{k'}) \right. \\ & \quad \left. + \langle w_{ijk}^l w_{ij'k'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_{j'}, \mathbf{t}_{k'}) + \langle w_{ij'k'}^l w_{ij'k}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_{k'})^\top f_{l'}(\mathbf{t}_{j'}, \mathbf{t}_k) \right) \Big\} \quad (3.26) \end{aligned}$$

Similarly as in Section 2.2.2, distributions of linear weights is conditioned on the value of s_{ij} according to the factorization given in Equation 3.11.

$$q(w_{ij}|s_{ij} = 0) = \mathcal{N}(w_{ij}|0, \sigma_w^2) \quad (3.27)$$

$$q(w_{ij}|s_{ij} = 1) = \mathcal{N}(w_{ij}|\mu_{ij}, \sigma_{ij}^2) \quad (3.28)$$

For $s_{ij} = 0$, w_{ij} follows its prior distribution. Given $s_{ij} = 1$, the Gaussian distribution of $w_{ij}|s_{ij} = 1$ has the following mean:

$$\mu_{ij} = \frac{1}{\mathbf{t}_j^\top \mathbf{t}_j + \frac{\sigma_i^2}{\sigma_w^2}} \left\{ \mathbf{t}_j^\top \mathbf{x}_i - \sum_{k \neq j} \langle s_{ik} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k - \sum_{k \neq j} \sum_{l=1}^L \langle s_{ik} w_{ijk}^l \rangle \mathbf{t}_j^\top f_l(\mathbf{t}_j, \mathbf{t}_k) \right. \\ \left. - \sum_{k \neq j} \sum_{\substack{j'=k+1 \\ j' \neq j}}^K \sum_{l=1}^L \langle s_{ik} s_{ij'} w_{ikj'}^l \rangle \mathbf{t}_j^\top f_l(\mathbf{t}_k, \mathbf{t}_{j'}) \right\} \quad (3.29)$$

and variance:

$$\sigma_{ij}^2 = \frac{\sigma_i^2}{\mathbf{t}_j^\top \mathbf{t}_j + \frac{\sigma_i^2}{\sigma_w^2}}. \quad (3.30)$$

The variational distribution of w_{ijk}^l is conditioned on the values of s_{ij} and s_{ik} , again following the factorization given in Equation 3.11

$$q(w_{ij}|s_{ij} = 0 \vee s_{ik} = 0) = \mathcal{N}(w_{ijk}^l|0, \sigma_w^2) \quad (3.31)$$

$$q(w_{ijk}^l|s_{ij} = 1, s_{ik} = 1) = \mathcal{N}(w_{ijk}^l|\mu_{ijk}^l, \sigma_{ijk}^2) \quad (3.32)$$

For either of s_{ij} or s_{ik} equal to 0, the corresponding weight w_{ijk}^l follows the prior normal distribution with zero mean. For both s_{ij} and s_{ik} equal to 1, the normal distribution for w_{ijk}^l has mean

$$\begin{aligned}
\mu_{ijkl} = & \frac{1}{f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_l(\mathbf{t}_j, \mathbf{t}_k) + \frac{\sigma_i^2}{\sigma_w^2}} \left\{ f_l(\mathbf{t}_j, \mathbf{t}_k)^\top \mathbf{x}_i - \langle w_{ij} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top \mathbf{t}_j - \langle w_{ik} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top \mathbf{t}_k \right. \\
& - \sum_{\substack{j' \neq j, k \\ j' \neq j, k}}^K \langle s_{ij'} w_{ij'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top \mathbf{t}_{j'} - \sum_{l' \neq l}^L \langle w_{ij'k}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_j, \mathbf{t}_k) \\
& - \sum_{j' \neq j, k}^K \sum_{l'=1}^L \langle s_{ij'} \rangle \left(\langle w_{ij'j'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_j, \mathbf{t}_{j'}) + \langle w_{ikj'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{j'}) \right) \\
& \left. - \sum_{\substack{j' \neq j, k \\ k' \neq j, k}}^K \sum_{k'=j'+1}^K \sum_{l'=1}^L \langle s_{ij'} s_{ik'} w_{ij'k'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_{j'}, \mathbf{t}_{k'}) \right\} \quad (3.33)
\end{aligned}$$

and variance

$$\sigma_{ijkl}^2 = \frac{\sigma_i^2}{f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_l(\mathbf{t}_j, \mathbf{t}_k) + \frac{\sigma_i^2}{\sigma_w^2}}. \quad (3.34)$$

M step

The M step of the variational EM algorithm optimizes values of the hyperparameters of the model $\boldsymbol{\theta} = \{\pi, \sigma_w^2, \sigma_{ij}^2\}$, again by maximizing the value of free energy (3.9) while holding the variational distribution q fixed.

Update equation for the sparsity parameter π is

$$\pi = \frac{1}{MK} \sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle. \quad (3.35)$$

Weight variance is updated as follows

$$\sigma_w^2 = \frac{\sum_{i=1}^M \sum_{j=1}^K \langle w_{ij}^2 \rangle + \sum_{i=1}^M \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle (w_{ijk}^l)^2 \rangle}{\sum_{i=1}^M \sum_{j=1}^K \langle s_{ij} \rangle + \sum_{i=1}^M \sum_{j=1}^K \sum_{k=j+1}^K \langle s_{ij} s_{ik} \rangle}. \quad (3.36)$$

And finally the updates for gene-specific variance have the following form:

$$\begin{aligned}
\sigma_i^2 = \frac{1}{N} & \left\{ \mathbf{x}_i^\top \mathbf{x}_i - 2 \sum_{j=1}^K \langle s_{ij} w_{ij} \rangle \mathbf{t}_j^\top \mathbf{x}_i - 2 \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle s_{ij} s_{ik} w_{ijk}^l \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top \mathbf{x}_i \right. \\
& + \sum_{j=1}^K \langle s_{ij} w_{ij}^2 \rangle \mathbf{t}_j^\top \mathbf{t}_j + \sum_{j=1}^K \sum_{k \neq j}^K \langle s_{ij} w_{ij} \rangle \langle s_{ik} w_{ik} \rangle \mathbf{t}_j^\top \mathbf{t}_k \\
& + 2 \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle s_{ij} s_{ik} w_{ij} w_{ijk}^l \rangle \mathbf{t}_j^\top f_l(\mathbf{t}_j, \mathbf{t}_k) + 2 \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \langle s_{ij} s_{ik} w_{ik} w_{ijk}^l \rangle \mathbf{t}_k^\top f_l(\mathbf{t}_j, \mathbf{t}_k) \\
& + 2 \sum_{j=1}^K \sum_{j'=1}^K \sum_{k=j'+1}^K \sum_{l=1}^L \langle s_{ij} w_{ij} s_{ij'} s_{ik} w_{ij'k}^l \rangle \mathbf{t}_j^\top f_l(\mathbf{t}_{j'}, \mathbf{t}_k) \\
& + \sum_{j=1}^K \sum_{k=j+1}^K \sum_{l=1}^L \sum_{l'=1}^L \langle s_{ij} s_{ik} w_{ijk}^l w_{ijk}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_j, \mathbf{t}_k) \\
& + 2 \sum_{j=1}^K \sum_{k=j+1}^K \sum_{k'=k+1}^K \sum_{l=1}^L \sum_{l'=1}^L \langle s_{ij} s_{ik} s_{ik'} \rangle (\langle w_{ijk}^l w_{ijk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_j, \mathbf{t}_{k'}) + \\
& \quad \langle w_{ijk}^l w_{ikk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{k'}) + \langle w_{ijk'}^{l'} w_{ikk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_{k'})^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{k'})) \\
& + 2 \sum_{j=1}^K \sum_{j'=j+1}^K \sum_{k=j'+1}^K \sum_{k'=k+1}^K \sum_{l=1}^L \sum_{l'=1}^L \langle s_{ij} s_{ij'} s_{ik} s_{ik'} \rangle (\langle w_{ijj'}^l w_{ikk'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_{j'})^\top f_{l'}(\mathbf{t}_k, \mathbf{t}_{k'}) \\
& \quad + \langle w_{ijk}^l w_{ij'k'}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_k)^\top f_{l'}(\mathbf{t}_{j'}, \mathbf{t}_{k'}) + \langle w_{ijk'}^{l'} w_{ij'k}^{l'} \rangle f_l(\mathbf{t}_j, \mathbf{t}_{k'})^\top f_{l'}(\mathbf{t}_{j'}, \mathbf{t}_k)) \left. \right\}. \quad (3.37)
\end{aligned}$$

The updates in the E step and M step are iterated until convergence. Preliminary results obtained from the extended sparse linear model with transcription factor interactions are discussed in Chapter 5.

Chapter 4

Data

One of the aims of this project was to identify some key transcription factors and their target genes that play role in the development of neural tube in vertebrates. The data used for this project was provided by James Briscoe and Natascha Bushati from MRC National Institute for Medical Research. The data come from their research projects on neural tube development and especially Shh (sonic hedgehog) signalling pathway which is responsible for development of different types of neurons.

The microarray gene expression data comes from several *in vivo* and *in vitro* experiments in chick. Table 4.1 gives an overview of the experimental conditions that were present in the available datasets. The conditions are separated into three groups. The first group represents gene expression data measured in an explant culture (*in vitro* isolated tissue), the rest of the groups contain data from *in vivo* experiments (the groups are identified by the person who conducted the experiments). Each experimental condition contains three samples from biological replicates. Therefore there are 21 measurements for the explants data and 15 measurements for each of the *in vivo* datasets. In total the dataset contains 51 experimental samples.

All the gene expression measurements come from Affymetrix microarray chips and were RMA[10] preprocessed. The RMA preprocessing produces log transformed data. The data was also additionally normalized to zero mean.

Table 4.2 shows a list of several versions of the original dataset that were used for the purpose of this project. The filtered datasets were produced by filtering out genes that are not differentially expressed, with different levels of stringency.

Experiment type	Identifier	Description
Explants	0nM_0hr	no Shh,measured immediately after dissection
	0nM_12hrs	no Shh,incubated for 12hrs
	0nM12hrs+4nM12hrs	no Shh for 12hrs,then high Shh concentration for another 12hrs
	1nM_12hrs	intermediate Shh concentration for 12hrs
	4nM_06hrs	high Shh for 6hrs
	4nM_12hrs	high Shh for 12hrs
	4nM_24hrs	high Shh for 24hrs
In vivo, Catarina	GFP_14h	control condition,no manipulation, heterogenous cell population, 14hrs
	GFP_36h	control condition,no manipulation, heterogenous cell population, 36hrs
	GLI1_High_14h	High levels of Gli for 14hrs
	GLI1_High_36h	High levels of Gli for 36hrs
	PatchD_14h	Inhibition of Shh for 14hrs
In vivo, Natascha	SMO_06h	Shh signalling constitutively active (using Smo) for 6hrs
	Smo_12h	Shh signalling constitutively active (using Smo) for 12hrs
	SMO_18h	Shh signalling constitutively active (using Smo) for 18hrs
	Smo_2.2RNAi_18h	Shh signalling constitutively active (using Smo) for 18hrs, but removal of Nkx2.2
	Nkx2.2_RCAS_12h	Forced expression of Nkx2.2 for 12hrs

Table 4.1: List of experimental conditions present in the available gene expression dataset.

Dataset identifier	Number of samples	Number of genes (incl. transcription factors)	Number of identified transcription factors
Complete unfiltered dataset	51	38535	2895
filter5+	51	3410	304
explants_filter2+	21	5341	472
explants_filter3+	21	2012	180

Table 4.2: List of the datasets on patterning neural tube that were used in modelling.

Chapter 5

Experimental results

Evaluation of the presented methods on the dataset described in Chapter 4 is a difficult task. The aim of this project was to design methods and evaluate them on real-world microarray experimental data in order to identify novel relations between transcription factors and genes. Currently, only some key parts of the transcriptional network active during patterning of the vertebrate neural tube are already identified [6].

Table 5.1 lists the key transcription factors are already known to play a role in patterning the neural tube [6] and that are present in the `explants_filter2+` dataset. This list can be used to partially evaluate the results as the key transcription factors should be identified by the tested models.

5.1 Sparse linear regression

This section evaluates the sparse Bayesian linear regression model on the filtered datasets listed in Table 4.2. Each of the results is also compared with a simple model fitted with lasso. For each dataset, transcription factors that regulate the largest number of genes are identified and compared with the list of already identified transcription factors that are important in neural tube development.

5.1.1 Dataset `filter5+`

Dataset `filter5+` contains the complete set of 17 different experimental conditions as listed in Table 4.1. Each condition is represented by three independent samples, i.e. there are 51 independent samples overall. The dataset comprises of 3410 genes out of which 304 are

Name	Affymetrix identifier	Notes
DBX2	Gga.12249.1.S1_at	
FOXJ1	Gga.16555.1.S1_at	
GBX2	Gga.565.1.S1_at	
GLI3	Gga.3297.1.S1_at	
MSX1	GgaAffx.21776.1.S1_s_at	
MSX2	Gga.4977.1.S1_at	
NEUROG1	Gga.407.1.S1_at	
NKX-6.1	Gga.4083.1.S1_at	
NKX2-1	Gga.952.1.S1_at	probe believed to respond to NKX2.2
NKX6-2	Gga.2725.1.S1_at	
OLIG3	Gga.4947.1.S1_at	probe for OLIG2
PAX6	Gga.556.1.S1_at	
PAX7	Gga.555.1.S1_at	
RFX4	GgaAffx.8015.1.S1_at	
SOX10	Gga.4428.1.S2_s_at	
SOX9	Gga.5505.1.S1_a_at	

Table 5.1: List of the transcription factors that are currently known to play role in patterning the neural tube.

identified as transcription factors. Sparse linear regression is used to predict the expression of each gene using only a small subset of regulating transcription factors.

The Bayesian sparse linear regression model with spike and slab prior introduced in Chapter 2 was used to fit the dataset. Figure 5.1 shows sample results for four randomly selected target genes ¹. Each row represents the results for one gene. The first column shows the original original gene expression $\mathbf{x}_i = (x_i^1, \dots, x_i^N)$, $N = 51$ in blue color. The red line represents the mean posterior reconstruction of \mathbf{x}_i using the fitted model. The second column shows the posterior values of sparse variable s_{ij} (see Equation 2.13) for gene \mathbf{x}_i across all transcription factors $\mathbf{t}_j, j = 1, \dots, K, K = 304$. Only a small number of transcription factors is active for each of the sample genes. The third column shows the posterior effective weights $\tilde{w}_{ij} = w_{ij}s_{ij}$, again across all transcription factors $\mathbf{t}_j, j = 1, \dots, K$.

Considering the results for all genes in the filter5+ dataset, the resulting sparsity of the solution was $\pi = 0.014$. Figure 5.2 shows a histogram of expected values of the sparse variable s_{ij} in standard and logarithmic scale. Vast majority of s are equal to zero which reflects the posterior sparsity of the model.

Figure 5.3 shows a histogram the average number of transcription factors that are active for each gene. The values for this histogram were obtained by summing expected values of the variables s_{ij} for each gene \mathbf{x}_i . This does reflect the cumulative influence of transcription factors towards each gene. In average, most genes are regulated by 4-5 transcription factors. This can be compared with Figure 5.4 which shows, how many transcription factors are probably fully active for each gene, i.e. posterior probability that the sparse variable s_{ij} is equal to 1 is larger than 0.5 (for transcription factor \mathbf{t}_j and gene \mathbf{x}_i). The figure shows that most genes are regulated by only two to four transcription factors. Difference between Figures 5.3 and 5.4 is caused by small cumulative influence of other transcription factors.

Figure 5.5 shows a histogram of how many genes are regulated by each single transcription factor. According to the model, most transcription factors regulate only 20 to 40 genes, but there is a smaller group of regulators that influence over 60 and up to 120 transcription factors. The top 10% of transcription factors that influence the largest number of genes are listed in Table 5.2. The list contains 5 of the transcription factors that are known to play role in the patterning of neural tube (Table 5.1). Note that not all of the key transcription factors were present in the filtered dataset.

Overall, the results well correspond to expected biological properties of transcription

¹Code to generate the plots was provided by Michalis K. Titsias.

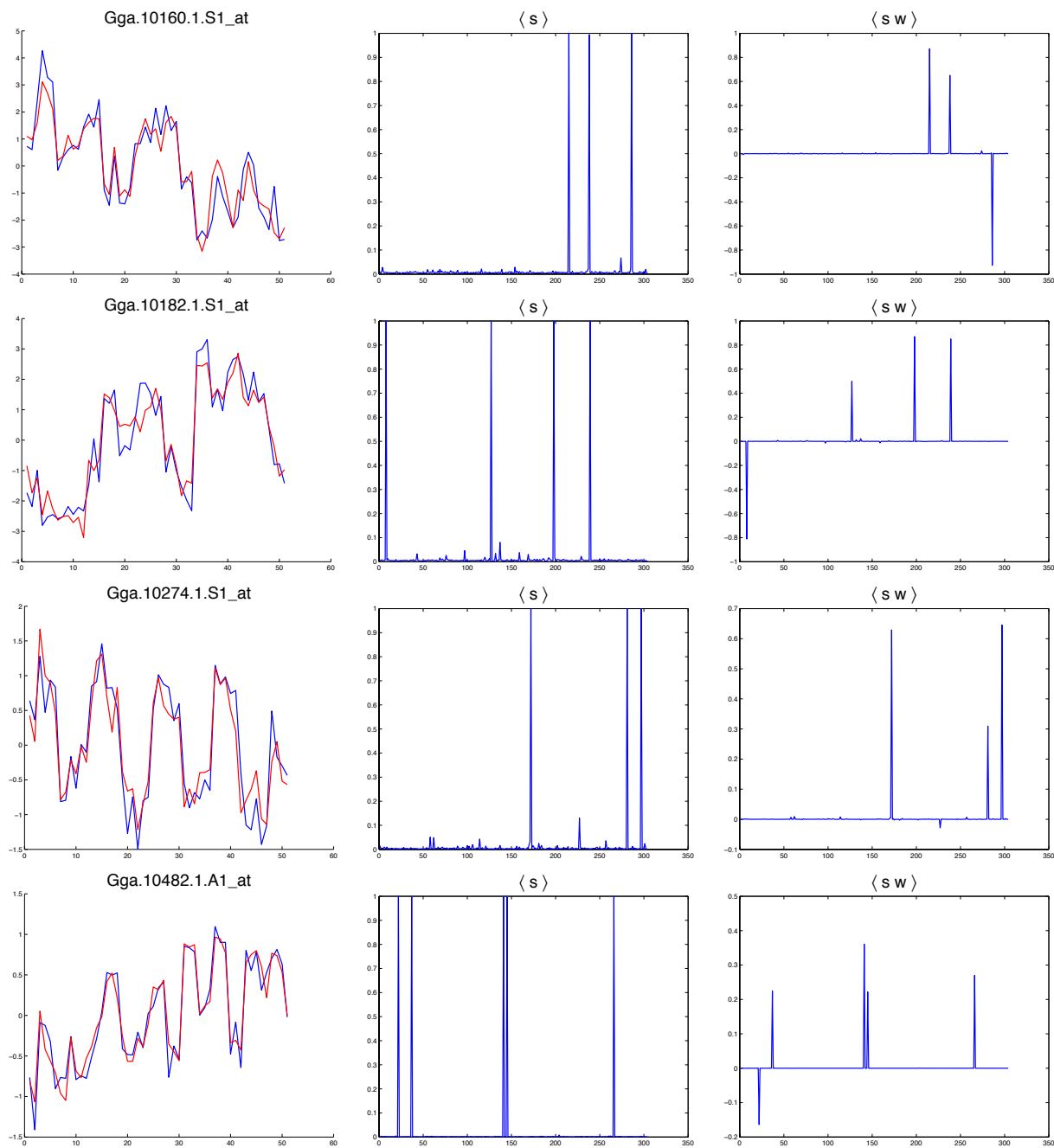


Figure 5.1: Dataset filter5+: Sample results, each row represents a randomly selected gene and the reconstruction of its expression.

factors and regulated genes. Each gene is regulated by a small number of transcription factors. Considering regulators in the model, there are hub transcription factors that regulate a large number of genes and that supposedly play significant role in the biological process.

The results of the sparse linear model were also very stable. Transcription factors identified as active for each gene obtained using five different random initializations for the variational approximation algorithm were on average 99.58% identical.

Additionally, results from the Bayesian model were compared to results produced by lasso on the same dataset. For each gene, the LARS algorithm for lasso (Section 2.1) identified 10 most important transcription factors for prediction of expression of each gene. Average percentage of transcription factors that were selected by lasso if they were also identified as active in Bayesian sparse solution was 76%. The agreement is relatively high.

The expressions of transcription factors were also clustered using Gaussian mixture model and the clusters were used as predictors to avoid ambiguities of correlated gene expressions. The agreement between lasso and Bayesian linear regression increased up to over 90% for clustered transcription factors. However, the clustered results are not presented as they did not appear to be biologically plausible. In the clustered transcription factor groups, the key transcription factors that are assumed to play important role did not correspond well to clusters. However, the result shows that the main differences between lasso and Bayesian solution can be accounted to correlated transcription factor expressions. The main advantage of the Bayesian solution compared to lasso is that it selects automatically the optimal number of transcription factors for each gene.

Overall, Bayesian sparse linear model successfully identified some of the key transcription factors and produced some testable predictions of potentially important transcription factors.

Rank	Affymetrix ID	Name	Number of regulated genes
1	Gga.15090.1.S1_s_at	TSHZ2	124.086
2	Gga.16555.1.S1_at	FOXJ1	121.1593
3	Gga.18136.1.S1_at	ZDHHC21	114.7125
4	Gga.13720.1.S1_at	C21orf66	108.716
5	Gga.2725.1.S1_at	NKX6-2	103.6047
6	Gga.446.1.S1_at	ETV4	103.3071
7	GgaAffx.25626.2.S1_s_at	MED23	97.4397
8	GgaAffx.520.1.S1_at	VPS72	93.1595
9	Gga.4349.2.S1_s_at	SMARCB1	93.0452
10	Gga.2433.1.S1_at	RELB	92.7919
11	Gga.4847.3.S1_a_at	NPM1	88.4459
12	GgaAffx.23312.1.S1_at	GLIS1	81.7817
13	GgaAffx.10000.1.S1_at	NRIP1	78.5015
14	Gga.4949.1.S1_at	SREBF2	76.3873
15	GgaAffx.23684.1.S1_s_at	PLXNA2	75.6867
16	Gga.1491.1.S1_at	RREB1	75.3279
17	Gga.665.1.S1_at	CSRP2	74.224
18	GgaAffx.26573.1.S1_at	ZDHHC7	73.9175
19	GgaAffx.1024.1.S1_at	FOXK2	73.8642
20	Gga.4428.1.S2_s_at	SOX10	73.8054
21	Gga.677.1.S1_at	GNOT1	73.3484
22	Gga.2771.1.S1_at	SUFU	71.8772
23	Gga.4947.1.S1_at	OLIG3	71.7575
24	GgaAffx.3073.1.S1_s_at	RCJMB04_31h7	71.5392
25	Gga.12977.1.S1_s_at	DDB1	71.4038
26	Gga.556.1.S1_at	PAX6	71.2859
27	Gga.1325.1.S1_at	TES	71.0129
28	GgaAffx.349.4.S1_s_at	PER3	70.9938
29	Gga.572.1.S1_at	NR5A2	70.6774
30	Gga.19619.1.S1_at	PHF2	69.7576

Table 5.2: Top 10% of most frequently used transcription factors in filter5+ dataset.

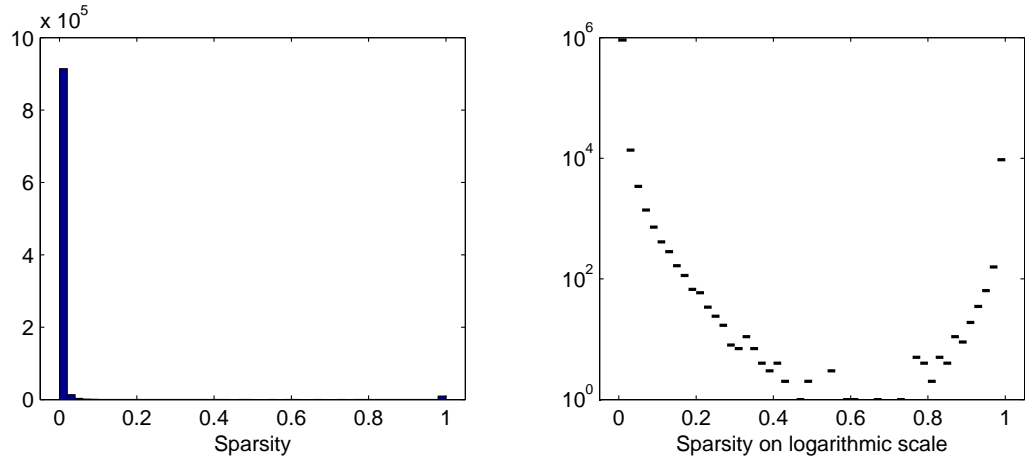


Figure 5.2: Dataset filter5+: sparsity of the solution.

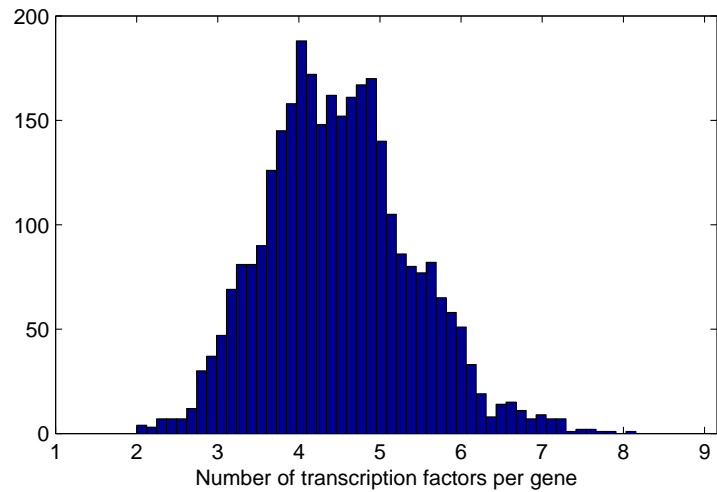


Figure 5.3: Dataset filter5+: histogram of number of transcription factors that are active for each gene.

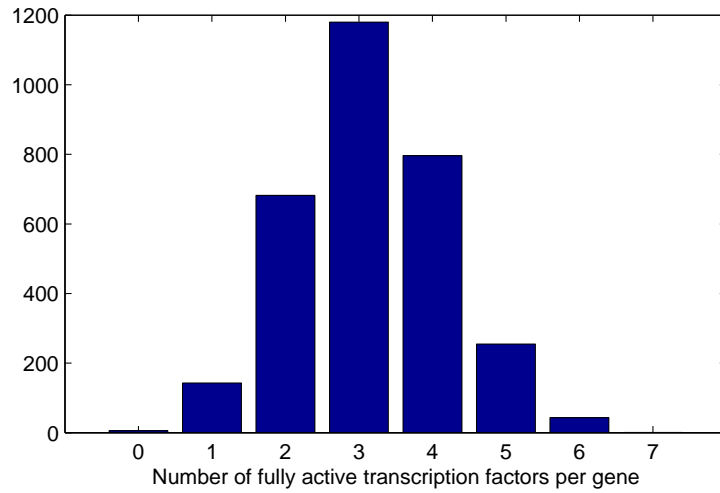


Figure 5.4: Dataset filter5+: number of transcription factors that are fully active for each gene (the probability that their corresponding variable s_{ij} is equal to 1 is larger than 0.5).

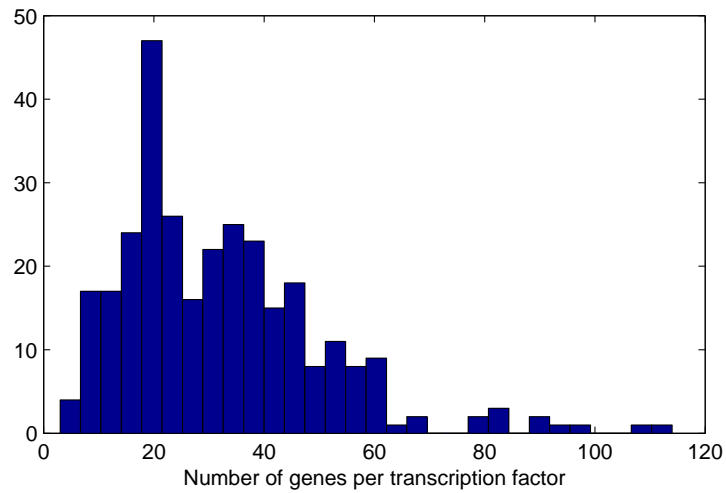


Figure 5.5: Dataset filter5+: histogram of number of genes that are regulated by each transcription factor.

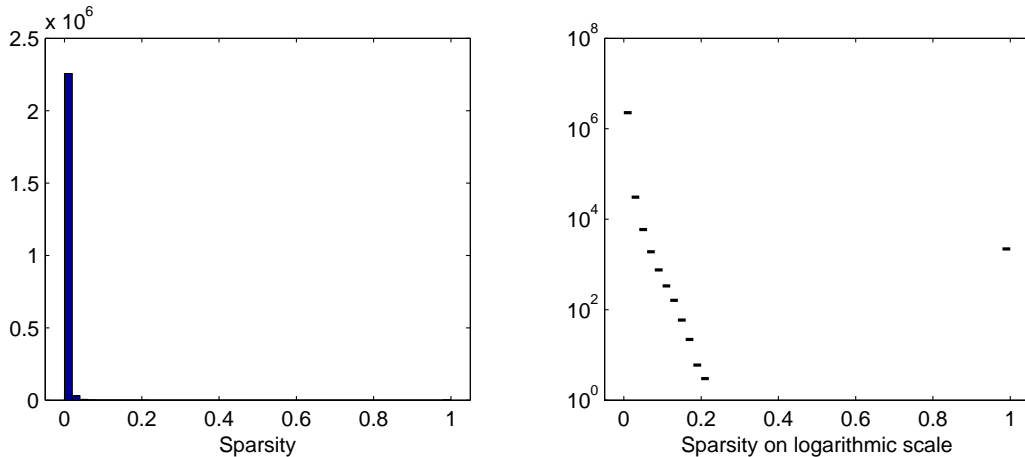


Figure 5.6: Dataset `explants_filter2+`: sparsity of the solution.

5.1.2 Explants datasets

This section describes results obtained from the Bayesian sparse linear model on the explants datasets, `explants_filter2+` with 5341 genes and `explants_filter3+` with 2010 genes. The explants datasets contain only one experiment group that examines the effects of different concentrations of sonic hedgehog (Shh) protein. There are 21 independent samples for each gene in this dataset, coming out of 7 different experimental conditions.

The resulting sparsity when the sparse linear regression model was fitted to the `explants_filter2+` dataset is $\pi = 0.0041$. That corresponds to even sparser solution than was obtained for the full `filter5+` dataset. Figure 5.6 shows the expected values of sparse variable s_{ij} (see Equation 2.13) across all genes and all transcription factors. In the log-scaled plot it is apparent that values of s_{ij} , i.e. probabilities that transcription factor \mathbf{t}_j regulates gene \mathbf{x}_i , are relatively unambiguous as there are no values in the middle of the $[0,1]$ interval.

For the smaller `explants_filter3+` dataset, the results look very similar. Estimated sparsity of the resulting model is $\pi = 0.011$, i.e. 1.1% of transcription factors are estimated to be active overall. Figure 5.7 shows the sparsity of the model in terms of values of the variable s_{ij} . Important transcription factors are again relatively unambiguously identified having the expected values of s_{ij} equal to 1.

Figure 5.8a shows the cumulative number of transcription factors that regulate each gene. The cumulative values were obtained by summing the sparse variables s_{ij} over transcription factors for each gene $\sum_j s_{ij}$. Figure 5.8b shows the number of transcription factors that are fully active for each gene, i.e. for each gene \mathbf{x}_i number of s_{ij} that are larger than 0.5. According to this figure, it is apparent that large number of genes do

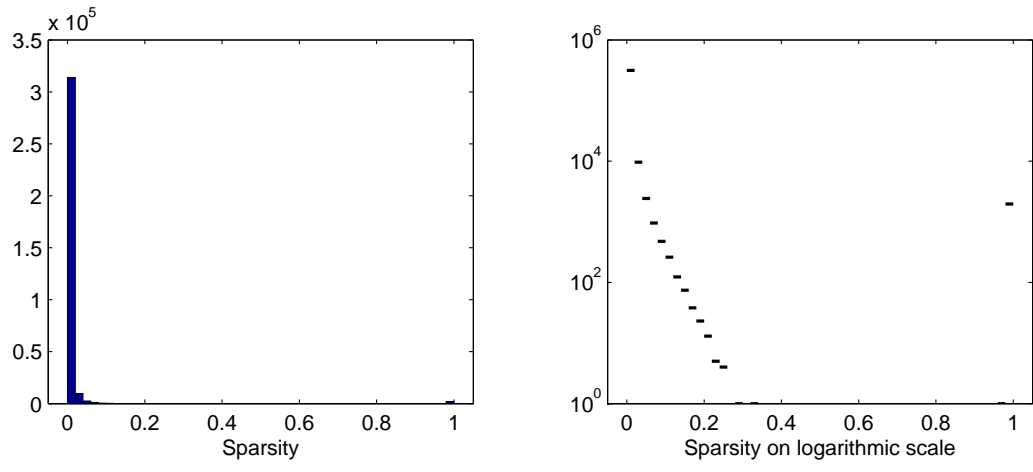


Figure 5.7: Dataset explants_filter3+: sparsity of the solution.

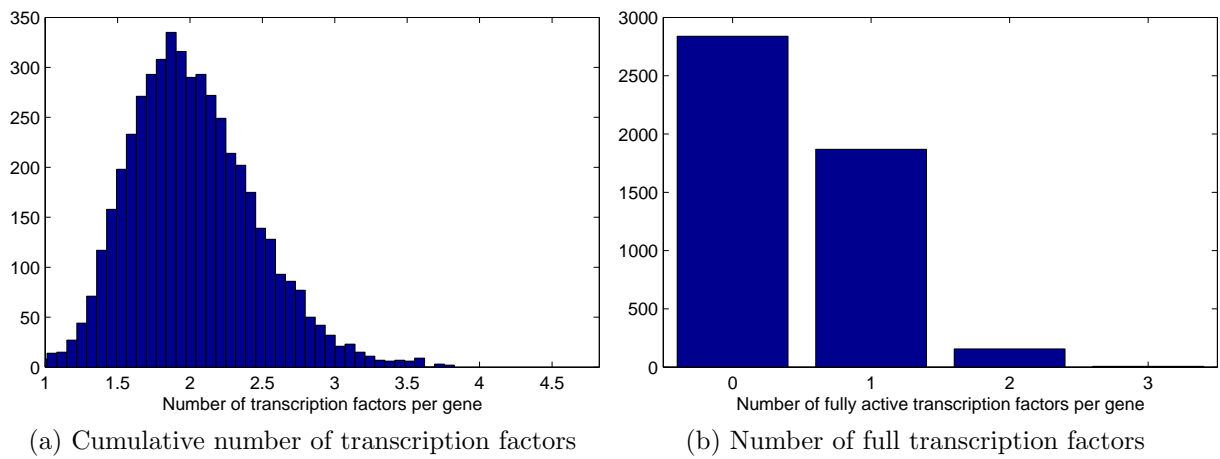


Figure 5.8: Dataset explants_filter2+: histograms of number of transcription factors that are active for each gene.

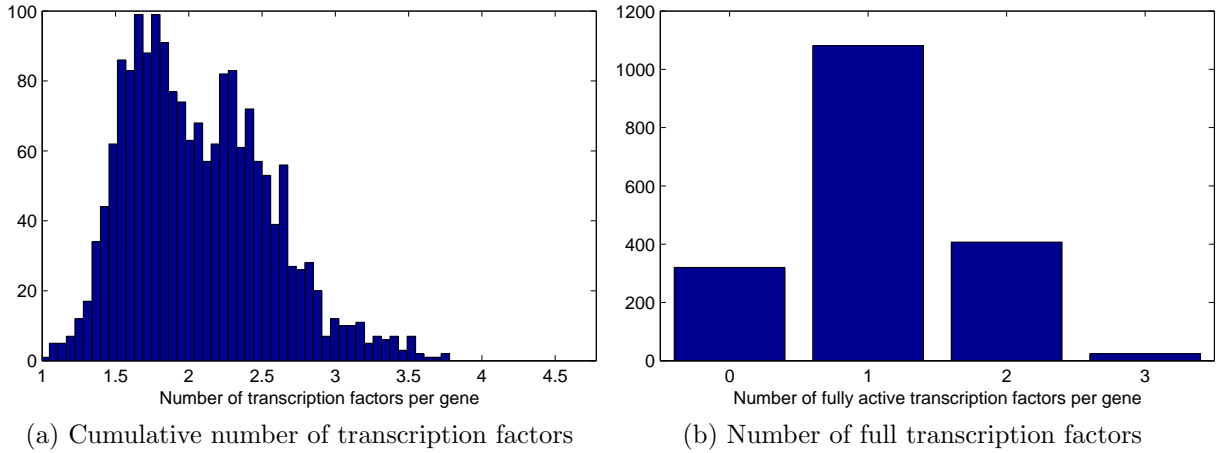


Figure 5.9: Dataset `explants_filter3+`: histograms of number of transcription factors that are active for each gene.

not have any fully active transcription factor. Expressions of these genes are fitted by the model only by using small contributions of larger number of transcription factors that are individually negligible.

Corresponding figures for the smaller `explants_filter3+` dataset that show the number of transcription factors per gene both cumulatively and individually are Figures 5.9a and 5.9b. We can see that although the cumulative effect of transcription factors on each gene is approximately the same, the number of transcription factors that are active for each gene is equal to 1 for most genes. The difference between the results for `explants_filter2+` and `explants_filter3+` datasets might be caused by the fact that `explants_filter3+` is a filtered version of `explants_filter2+` where more genes that are not differentially expressed were removed. Genes that do not change their expression significantly across experimental conditions are typically genes which expression can be approximated just by small contributions from regulators. Also the fit of the probabilistic model may be better for the smaller dataset as it contains less possible predictors.

Figures 5.10a and 5.10b show how many genes are regulated in average by each transcription factor. Again, results for the `explants_filter3+` dataset provide a more reasonable results.

Top 30 transcription factors according to the number of regulated genes are shown in Table 5.3 for the `explants_filter2+` dataset and in Table 5.4 for the smaller `explants_filter3+` dataset. Transcription factors that are known to be important in the development of neural tube (Table 5.1) are highlighted. In case of the larger dataset, 8 out of top 30 transcription

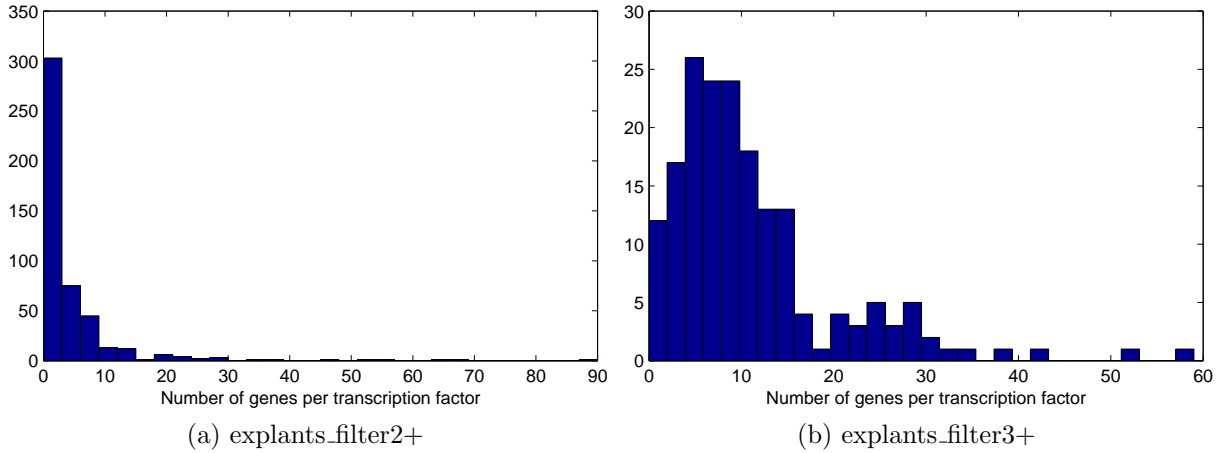


Figure 5.10: Histograms of number of genes that are regulated by each transcription factor in the explants datasets.

factors are known. In the smaller `explants_filter2+` dataset, 6 out of 30 transcription factors are known to play role in the neural tube patterning. The remaining 2 transcription factors that were identified in addition in `explants_filter2+` were filtered out in the smaller dataset.

Results obtained from the Bayesian sparse model were also compared with results computed by lasso. In both cases, lasso was used to identify 5 most important transcription factors for each gene. Agreement with lasso was computed on each of the two datasets as the average fraction of transcription factors that were selected as predictors for each gene in Bayesian sparse linear model, that were also selected by lasso with maximum of 5 predictors.

For the larger `explants_filter2+` dataset, the average agreement was only 30.8%. However, lasso failed to identify any of the known key transcription factors among the top 30 transcription factors as computed by lasso. Among top 50 transcription factors for lasso there were only two of the known regulators, `NKX6-2` and `PAX6`.

Agreement with lasso results on `explants_filter3+` dataset was relatively larger, 37.7%. Again, lasso failed to identify any of the known transcription factors among the 30 most used transcription factors. Only `NKX6-2` and `PAX6` were present among the top 50.

These results indicate that Bayesian sparse linear regression produced more valid results than lasso as it succeeded in identifying significantly larger number of known transcription factors. Overall, the explants datasets are more self-contained and less noisy than the `filter5+` dataset. This is also a factor that led to well identified important transcription factors.

Rank	Affymetrix ID	Name	Number of regulated genes
1	Gga.12249.1.S1_at	DBX2	99.4186
2	Gga.556.1.S1_at	PAX6	76.9185
3	Gga.2649.1.S1_at	CDX4	75.1458
4	Gga.225.1.S1_at	MSGN1	67.1095
5	Gga.952.1.S1_at	NKX2-1	65.1672
6	Gga.4445.1.S1_at	NR2F2	57.2774
7	Gga.2725.1.S1_at	NKX6-2	49.4813
8	Gga.7953.1.S1_at	RPL27A	44.735
9	GgaAffx.12664.1.S1_s.at	CREG1	43.6193
10	Gga.2460.1.S1_at	PROX1	42.4713
11	Gga.6311.1.S1_at	HEY1	42.294
12	Gga.4083.1.S1_at	NKX-6.1	42.2241
13	GgaAffx.11904.1.S1_s.at	KLHL9	39.8811
14	Gga.9087.1.S1_at	CRX	39.0156
15	GgaAffx.23235.1.S1_at	EBF3	35.6238
16	Gga.2070.2.S1_a.at	ID4	34.7699
17	Gga.17627.1.S1_s.at	KRIT1	34.7636
18	Gga.5505.1.S1_a.at	SOX9	34.6164
19	Gga.16555.1.S1_at	FOXJ1	34.5312
20	Gga.4133.1.S1_at	SMAD5	34.2118
21	GgaAffx.4562.1.S1_at	CARHSP1	33.8424
22	GgaAffx.12590.1.S1_at	ZNF509	33.8045
23	Gga.17095.1.S1_at	TSC2	33.1176
24	Gga.8810.1.S1_at	EED	33.0108
25	Gga.15618.1.S1_at	GOLGA4	32.7976
26	Gga.555.1.S1_at	PAX7	32.0389
27	Gga.4905.1.S1_s.at	CNOT7	31.8246
28	Gga.10.1.S1_at	OTX2	31.739
29	GgaAffx.12866.1.S1_s.at	ZDHHC17	31.2239
30	Gga.15881.1.S1_at	ZFP106	31.1283

Table 5.3: Dataset explants_filter2+: top 30 most used transcription factors.

Rank	Affymetrix ID	Name	Number of regulated genes
1	Gga.556.1.S1_at	PAX6	65.1108
2	Gga.225.1.S1_at	MSGN1	58.5771
3	Gga.4445.1.S1_at	NR2F2	47.5954
4	GgaAffx.22363.1.S1_at	ZNF217	43.497
5	Gga.12249.1.S1_at	DBX2	42.1046
6	GgaAffx.12866.1.S1_s.at	ZDHHC17	41.5299
7	Gga.2649.1.S1_at	CDX4	40.4555
8	GgaAffx.7278.1.S1_at	SMARCAL1	39.9749
9	GgaAffx.24812.1.S1_s.at	C21orf66	39.7861
10	GgaAffx.25626.2.S1_s.at	MED23	39.4339
11	Gga.2725.1.S1_at	NKX6-2	38.7963
12	GgaAffx.12664.1.S1_s.at	CREG1	38.7478
13	Gga.6311.1.S1_at	HEY1	37.9462
14	Gga.2070.2.S1_a.at	ID4	37.7121
15	Gga.9087.1.S1_at	CRX	36.0709
16	Gga.18136.1.S1_at	ZDHHC21	35.9812
17	GgaAffx.8015.1.S1_at	RFX4	33.48
18	Gga.952.1.S1_at	NKX2-1	32.7473
19	Gga.2460.1.S1_at	PROX1	31.801
20	Gga.10724.1.S1_at	REL	31.7533
21	Gga.2047.1.S1_a.at	ZNF335	31.4016
22	Gga.9289.1.S1_at	TFAP2B	31.2424
23	Gga.3659.1.S2_at	NFE2L2	31.2172
24	Gga.10.1.S1_at	OTX2	30.6015
25	GgaAffx.23076.1.S1_s.at	STRN3	30.4462
26	GgaAffx.23235.1.S1_at	EBF3	30.3865
27	Gga.555.1.S1_at	PAX7	30.2398
28	Gga.758.1.S1_at	LMX1B	29.4318
29	Gga.4315.1.S2_at	NR1D2	29.3814
30	GgaAffx.349.4.S1_s.at	PER3	28.9812

Table 5.4: Dataset explants_filter3+: top 30 most used transcription factors.

	Test condition	MSE
	None	0.0831
1	GFP_14h	0.22242
2	GFP_36h	0.28314
3	GLI1_High_14h	0.29757
4	GLI1_High_36h	0.18631
5	PatchD_14h	0.27012
6	Nkx2.2_RCAS_12h	0.17728
7	SMO_06h_2.CEL	0.22074
8	Smo_12h	0.33628
9	SMO_18h	0.19435
10	Smo_2.2RNAi_18h	0.28166
11	0nM_0hr	0.46434
12	0nM_12hrs	0.21724
13	0nM12hrs+4nM12hrs	0.21079
14	1nM_12hrs	0.33656
15	4nM_06hrs	0.21267
16	4nM_12hrs	0.22985
17	4nM_24hrs	0.47939
	Average	0.2718

Table 5.5: Crossvalidation results when a given test condition is removed from the training dataset.

5.1.3 Crossvalidation results

To assess the robustness of the model and individual influences of each experimental condition, leave-one-out crossvalidation was used on the filter5+ dataset. Each time, one experimental condition was removed from the training set, including all three independent samples that correspond to the condition. The model was trained using the remaining data. Finally, its performance was tested on predicting gene expressions for the excluded condition.

Table 5.5 shows how the performance of the model deteriorated when each single experimental condition was removed. The performance is measured by mean squared error (MSE) with respect to the test condition:

$$\text{MSE}(c) = \frac{1}{M} \sum_{i=1}^M (x_i^c - \hat{x}_i^c)^2 \quad (5.1)$$

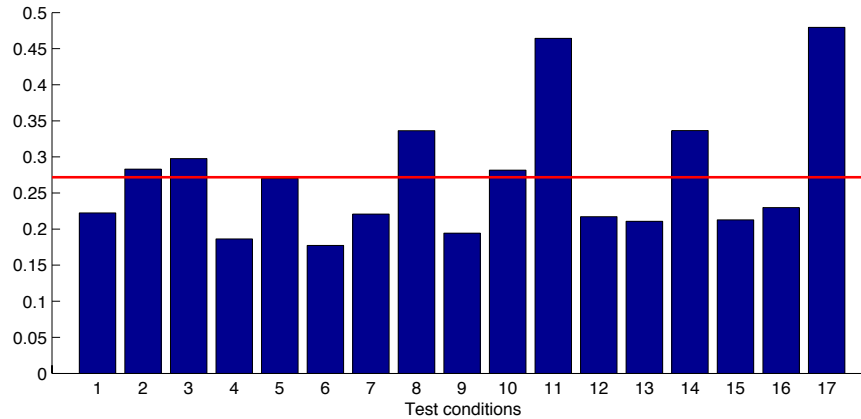


Figure 5.11: Barplot showing the mean squared error for each test condition. The red line indicates the average crossvalidation mean squared error.

where c is the test condition, M is the number of genes, K is the number of transcription factors, x_i^c is the expression of gene x_i in condition c and \hat{x}_i^c is the reconstructed expression of gene x_i . Mean squared errors for each condition are also shown in Figure 5.11.

Overall, the mean squared error was relatively small for most of the conditions. Mean squared error is the largest for conditions number 11 and 17. This may be an indication that these two conditions contain some information that is not present in the other conditions. This result may be also used to assess the relative difference of the processes in individual experimental conditions compared to the rest of the experiments.

5.2 Sparse linear regression with transcription factor interactions

This section presents preliminary results from sparse linear regression model with transcription factor interactions and compares them to the results from simple Bayesian linear regression. The models were tested only on datasets of limited size due to computational limitations.

Table 5.6 shows transcription factors of interest that were selected out of `explants_filter3+` dataset based on their known or anticipated role in the studied processes of Shh-mediated neural tube patterning by Natascha Bushati and James Briscoe. The table contains two groups, group A being presumably more important than group B. Experimental results in this section are presented both using only group A and both group A and B together as

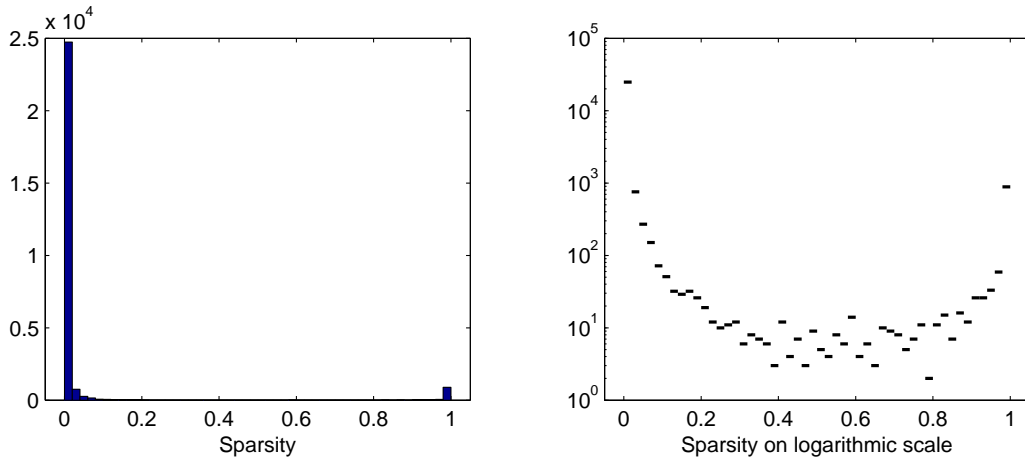


Figure 5.12: Sparsity of the solution of sparse linear regression with transcription factor interactions from group A.

predictors, for genes from the `explants_filter3+` dataset.

Genes that were used as targets for the extended model are genes from the `explants_filter3+` dataset. It contains 21 experimental samples for 1832 genes (with all transcription factors removed).

Considering group A of selected transcription factors, the resulting dataset with transcription factor interactions contains 15 transcription factors and 420 interaction vectors, given four basic interaction types (Section 3.1). After fitting the extended model, the resulting sparsity was estimated as $\pi = 0.0476$. Figure 5.12 shows a histogram with values of the sparse variable s_{ij} as computed by the model. Most values of $s_{ij}, i = 1, \dots, M, j = 1, \dots, K$ are well concentrated around 0 and 1.

Figure 5.14a shows the cumulative number of transcription factors that regulate each gene. According to the model, most genes are regulated only by a single transcription factor from group A, only a small fraction of genes is actually regulated by a larger number of factors. This indicates that transcription factor interactions that were incorporated into the model were used only in a small number of cases.

Figure 5.15 shows how much the transcription factor interactions were actually used by the model. It shows the posterior weights \tilde{w}_{ijk}^l for all four types of interactions $l = 1, \dots, 4$ across all pairs of transcription factors. Usage of the interactions is very sparse, most of the weights were estimated to be 0. This suggests that the proposed model does not use the transcription factor interactions unnecessarily and it does not overfit even with significantly larger number of available predictors.

The results from the model with interactions was also compared with results from the

	Affymetrix ID	Name	Group
1	Gga.952.1.S1_at	NKX2-1	A
2	Gga.3297.1.S1_at	GLI3	A
3	GgaAffx.20087.1.S1_s_at	SALL4	A
4	Gga.4083.1.S1_at	NKX-6.1	A
5	Gga.4947.1.S1_at	OLIG2	A
6	Gga.12249.1.S1_at	DBX2	A
7	GgaAffx.21776.1.S1_s_at	MSX1	A
8	Gga.555.1.S1_at	PAX7	A
9	Gga.2797.1.S1_s_at	PBX3	A
10	Gga.4977.1.S1_at	MSX2	A
11	Gga.556.1.S1_at	PAX6	A
12	GgaAffx.8015.1.S1_at	RFX4	A
13	Gga.6180.1.S2_at	SMAD3	A
14	Gga.2725.1.S1_at	NKX6-2	A
15	GgaAffx.23154.1.S1_at	RFX3	A
16	Gga.9574.1.S1_at	HES5 orthologue	B
17	Gga.11242.1.S1_at	HES5	B
18	Gga.2999.1.S1_at	ETV1	B
19	Gga.2699.1.S1_at	IRX1	B
20	Gga.447.1.S1_s_at	ETV5	B
21	Gga.167.1.S1_at	SALL3	B
22	GgaAffx.13008.1.S1_at	KLF6	B
23	Gga.3316.1.S1_s_at	HHEX	B
24	Gga.6311.1.S1_at	HEY1	B
25	Gga.15090.1.S1_s_at	TSHZ2	B
26	Gga.446.1.S1_at	ETV4	B
27	GgaAffx.23235.1.S1_at	EBF3	B
28	Gga.565.1.S1_at	GBX2	B
29	Gga.3878.1.S1_at	SMAD6	B

Table 5.6: Selected transcription factors for the sparse linear model with interactions.

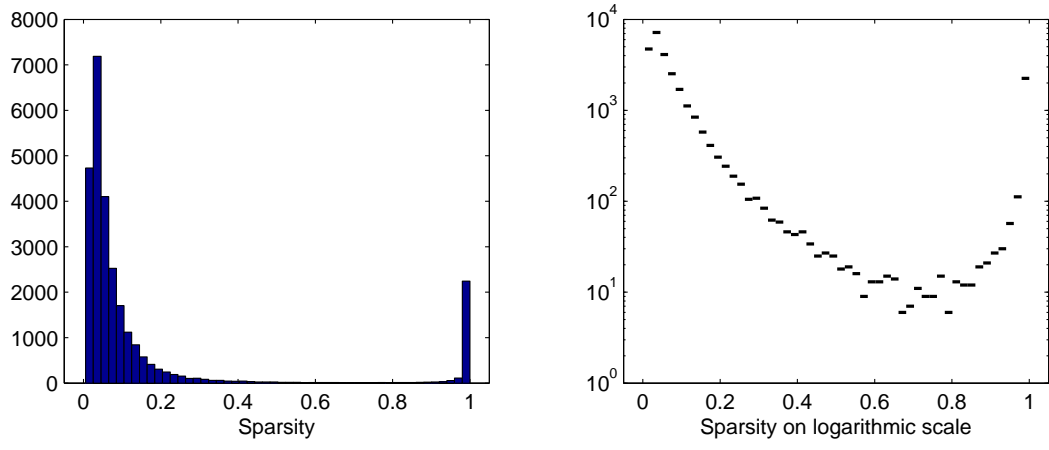


Figure 5.13: Sparsity of the solution of standard sparse linear regression without any interactions with transcription factors from group A.

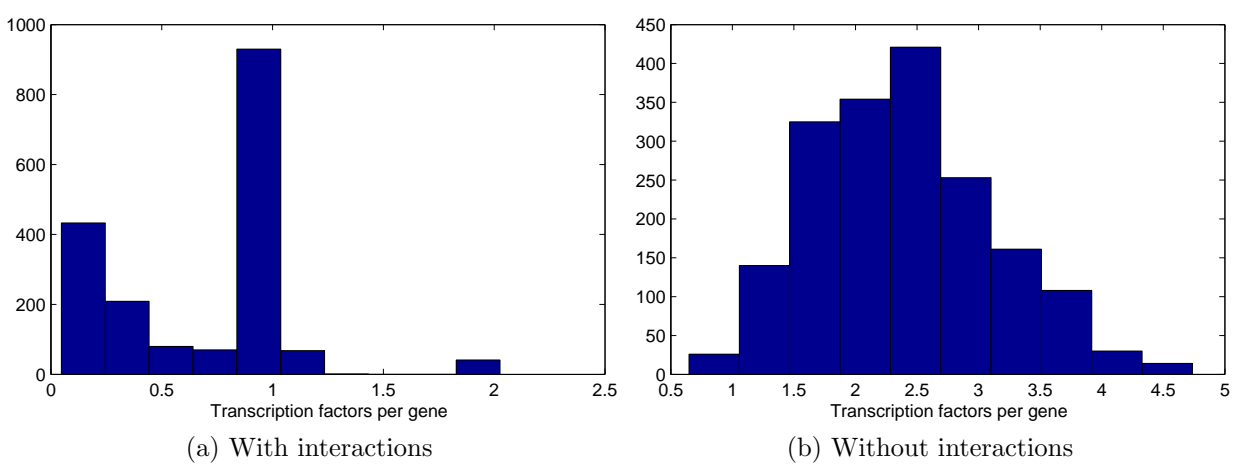


Figure 5.14: Cumulative number of transcription factors from group A that regulate each gene, comparison of the model with interactions and the standard model without interactions.

standard Bayesian linear regression model. The model was fitted using only the 15 transcription factors from group A both without any interactions and with the interaction vectors added as standard predictors. The model with interactions included in the transcription factor set as predictors, i.e. using 15+420 predictors, did not converge to a stable sparse solution. This was caused by low identifiability of interaction vectors and large total number of available predictors.

The model with only the original transcription factors did converge with resulting sparsity $\pi = 0.1593$. This shows that the model with only 15 transcription factors produced less sparse solution than the model with interactions. Figure 5.13 shows the histogram of the sparsity. Compared to Figure 5.12, larger number of sparse variable s_{ij} is larger than zero. This indeed shows that the standard model did not produce a well defined sparse solution.

Figure 5.14b shows the cumulative number of transcription factors per gene in the resulting model. This is again different from the extended model which identified one most important transcription factor for most genes. On the other hand, the standard model provided a better fit the the expression data with mean squared reconstruction error 0.1982. Mean squared error of the expressions reconstructed using extended model is 0.3212.

Finally, Figures 5.16a and 5.16b compare the usage of individual transcription factors from group A computed as $\sum_i \langle s_{ij} \rangle$. The figures show the cumulative number of genes that are regulated by each transcription factor, the numbering of regulators corresponds to Table 5.6. The pattern of usage remained mostly same although in the model with interactions, transcription factors are used less due to larger sparsity.

Given both groups A and B of selected transcription factors, the dataset contains 29 transcription factors and 1642 interaction vectors, again given four basic interaction types (Section 3.1). The model with transcription factor interactions produced a solution with sparsity $\pi = 0.0297$. Figure 5.17 shows the sparse structure of the solution which is very similar to the result of fitting the model with only group A transcription factors with interactions.

We can again compare this to the sparsity of a model fitted with only 29 transcription factors and no interactions. Sparsity of this model is $\pi = 0.0867$ and is shown in Figure 5.18. Similarly to the previous results, the simple model without interactions produces less sparse solution with smaller mean squared reconstruction error $MSE = 0.1353$. Mean square error of the extended model is 0.2547.

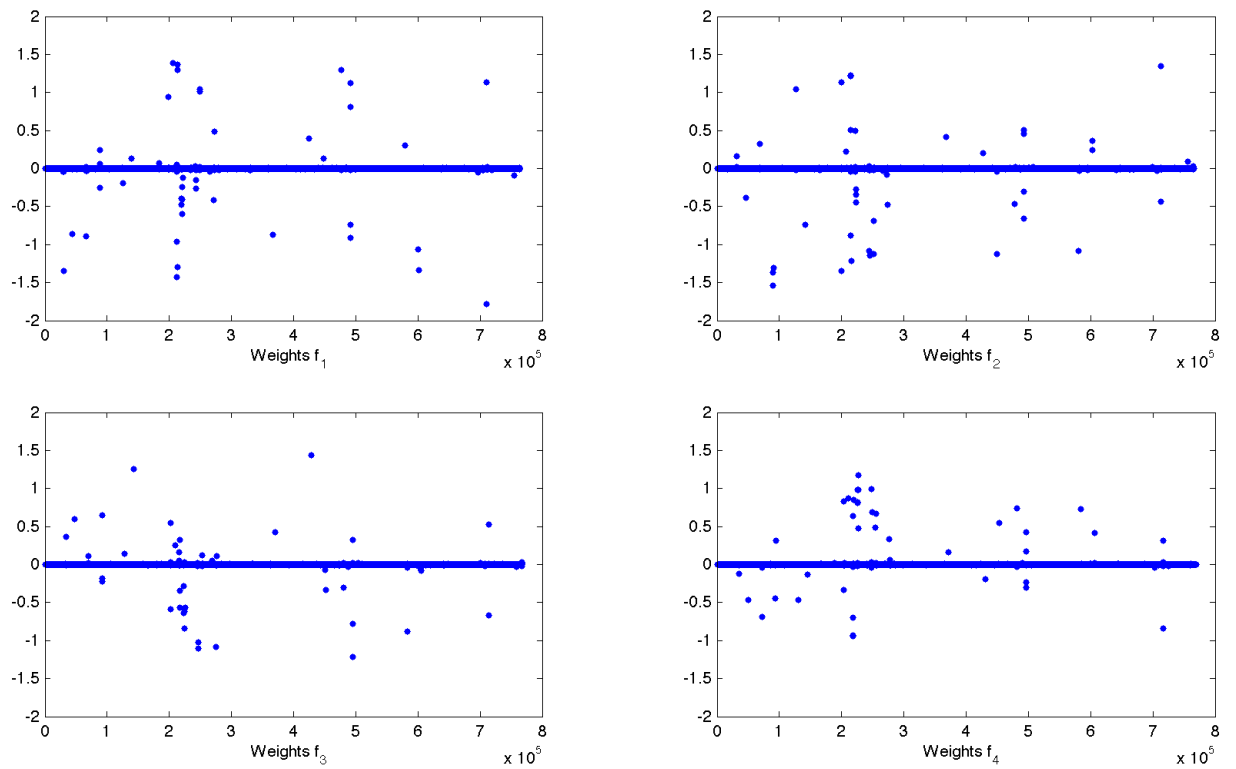


Figure 5.15: Sparsity of the weights of four types of interaction vectors in linear regression with transcription factor interactions from group A.

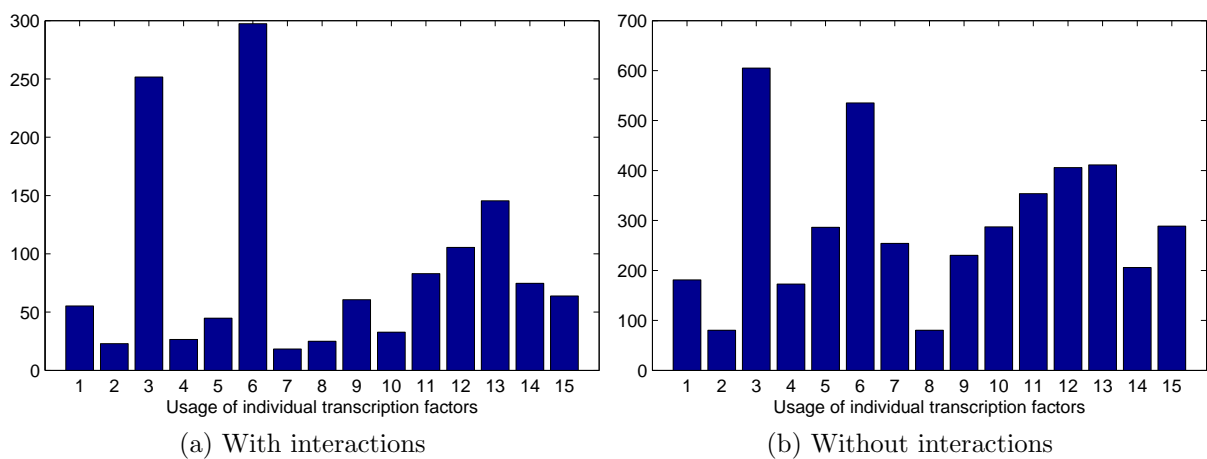


Figure 5.16: Number of genes regulated by each transcription factor from group A.

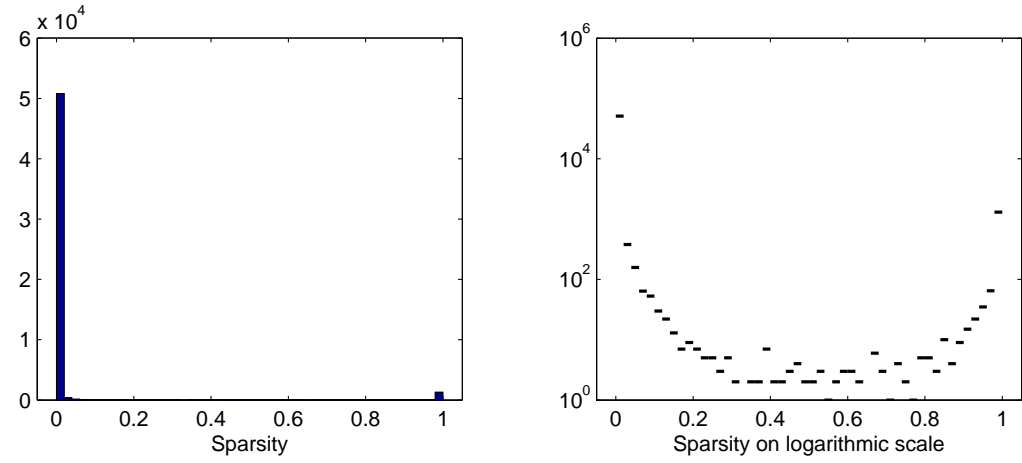


Figure 5.17: Sparsity of the solution of sparse linear regression with transcription factor interactions from groups A and B.

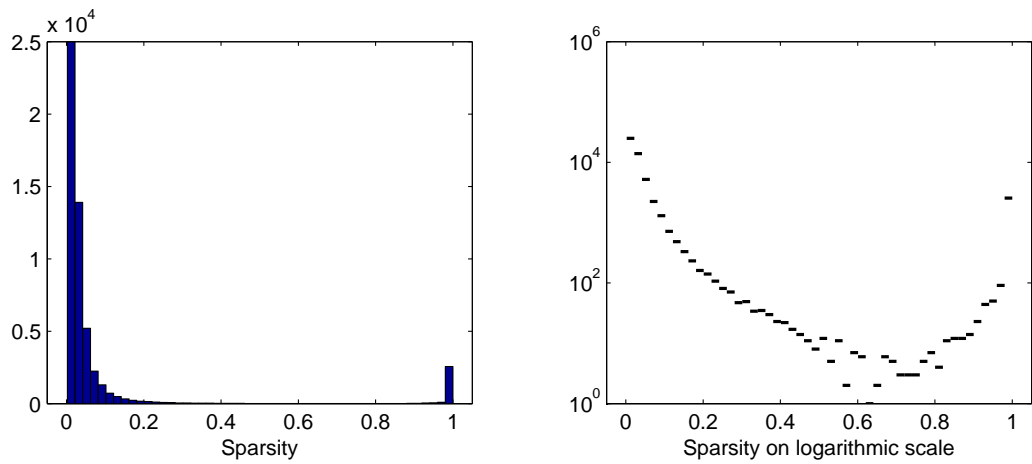


Figure 5.18: Sparsity of the solution of standard sparse linear regression without any interactions with transcription factors from groups A and B.

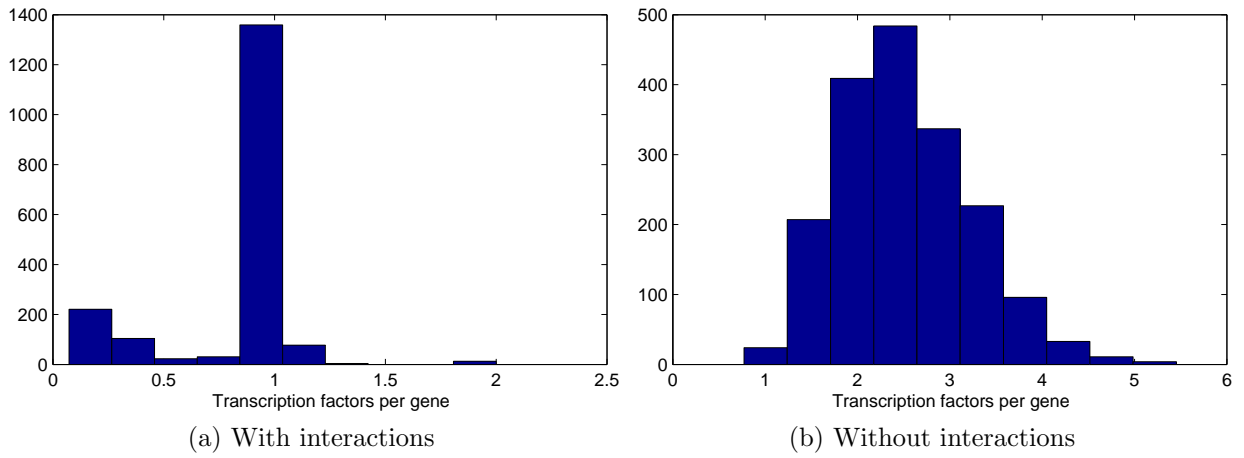


Figure 5.19: Cumulative number of transcription factors from groups A and B that regulate each gene, comparison of the model with interactions and the standard model without interactions.

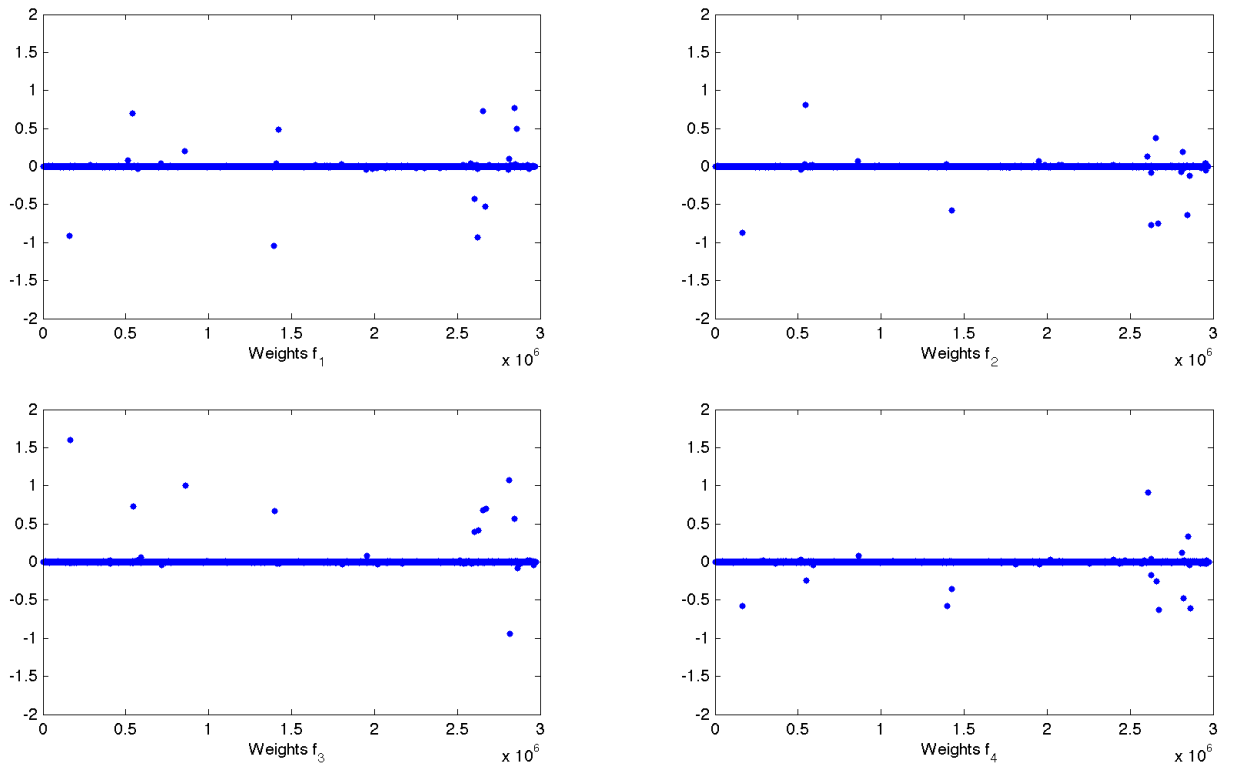


Figure 5.20: Sparsity of the weights of four types of interaction vectors in linear regression with transcription factor interactions from groups A and B.

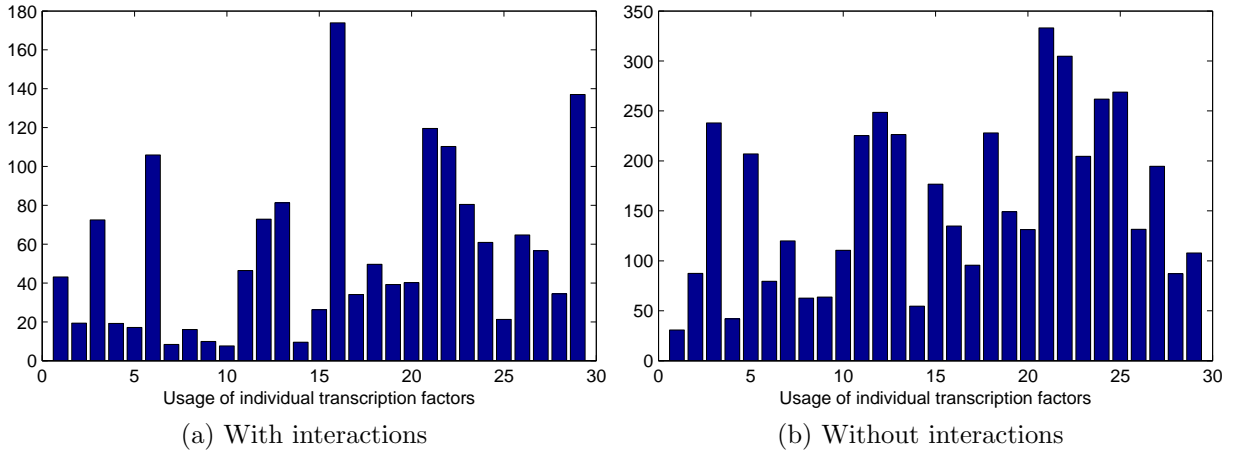


Figure 5.21: Number of genes regulated by each transcription factor from groups A and B.

Figure 5.19 compares, how many transcription factors are active for each gene in both models. Again, the model with interactions identifies only a single transcription factor for most genes. Figures 5.20 show the sparsity of interaction weights \tilde{w}_{ijk}^l . We can see that the model used interactions for only a small number of genes. This behaviour avoids overfitting even though the model uses a very large number of possible predictors.

Finally, Figure 5.21 shows how many genes are regulated by each of the 29 transcription factors from group A and group B. Some of the transcription factors that influence a large number of genes in the interaction model (e.g. transcription factors number 16 and 29) do not seem important in the standard model. The extended model used their interactions to fit the expressions of larger amount of genes.

Overall, the sparse linear model with transcription factor interactions produces solutions that are more sparse than the solutions from standard sparse linear regression. By restricting the use of pairwise interactions in the model, it avoids overfitting or ambiguity in results that would otherwise occur with extension of the predictor space. The proposed model also identifies only a small number of transcription factors for each gene which is well testable from biological point of view.

The extended model is currently applicable to smaller scale problems. It is well suited for a detailed examination of groups of potentially important transcription factors and genes. In such situations, the model produces testable results.

Chapter 6

Summary

In this work we analysed microarray data from early development of neural tube in order to uncover regulatory mechanisms taking place during this biological process. Primary goal was to identify transcription factors that play significant role in the gene regulatory network and also to identify their target genes. For this task, we presented two probabilistic sparse models and evaluated their performance on the experimental data.

The first Bayesian model was sparse linear regression model. Expressions of transcription factors in each experimental condition were used to predict expression of genes in that condition. The goal was to identify as few transcription factors for each gene as possible.

To induce sparsity, spike and slab prior distribution on the linear regression weights was used. An efficient approximate variational algorithm was derived for this model based on the algorithm presented in [11]. This model produced biologically plausible solutions. In the resulting model, each gene is regulated by a small number of transcription factors and on the other hand, some transcription factors simultaneously regulate a very large number of genes. This result well corresponds to the properties of typical gene regulatory network topology.

The linear sparse model also successfully identified a significant number of transcription factors that were already known to play role in the patterning of neural tube. Additionally, it suggests that the rest of the transcription factors that are predicted to influence a large number of genes may play some role in the neural development. These results can be potentially tested in additional microarray experiments.

Although the results that were obtained from the linear sparse model are promising, transcription factors often interact in non-linear fashion. Therefore, the second presented model introduced a novel method that integrates non-linear pairwise interactions of tran-

scription factors into the linear model. The non-linear effects were integrated in a way that prevents the model to overfit although the effective number of predictors in the model increased significantly. Variational approximation algorithm was also proposed for this new model.

Extended sparse model was applied to subset of the experimental data and the preliminary results were compared to results of the original sparse linear model. A small group of up to 49 selected transcription factors was used as predictors in this model, with their pairwise interactions. The extended model converged to very sparse solutions and used transcription factor interactions only in a small number of cases. For each gene it identified only one or two transcription factors that regulate the gene. It provides a detailed analysis of possible regulatory relations that occur in a small group of transcription factors. Biological relevance of the results is yet to be determined by additional microarray experiments. More extensive testing of the presented model is also required.

From the preliminary results presented in this report, the sparse linear regression model succeeded to identify some of the key transcription factors that are known to play important role in the development of neural tube. Additionally, it suggested several transcription factors that seem to be significant as well. These transcription factors can be tested in additional biological experiments. The extended sparse linear model with non-linear interactions provides a new method to incorporate arbitrary interactions between transcription factors into the model. This method creates a more biologically plausible model of transcriptional regulation than the simple linear model. Both methods can identify potentially interesting transcription factors that can be experimentally tested to verify the results.

Bibliography

- [1] B. Alberts. *Molecular Biology of the Cell: Reference edition*. Number v. 1. Garland Science, 2008.
- [2] M. Aldana, S. Coppersmith, and L. Kadanoff. Boolean dynamics with random couplings. *Perspectives and Problems in Nonlinear Science*, pages 23–89, 2003.
- [3] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, (3), February 2007.
- [4] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2011. In press.
- [5] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2002.
- [6] A. P. M. Eric Dessaud and J. Briscoe. Pattern formation in the vertebrate neural tube: a sonic hedgehog morphogen-regulated transcriptional network. *Development*, 135:2489–2503, August 1 2008.
- [7] N. Friedman, M. Linial, and I. Nachman. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.
- [8] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.
- [9] J. M. Hernández-Lobato and T. M. H. Dijkstra. Hub gene selection methods for the reconstruction of transcription networks. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part I*, ECML PKDD’10, pages 506–521, Berlin, Heidelberg, 2010. Springer-Verlag.
- [10] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, Apr. 2003.

- [11] M. Lazaro-Gredilla and M. K. Titsias. Spike and slab variational inference for multi-task and multiple kernel learning. Submitted for publication, 2011.
- [12] A. A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, and A. Califano. ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context. *BMC Bioinformatics*, 7(Suppl 1):S7, 2006.
- [13] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, 1997.
- [14] R. Peeters and R. Westra. On the identification of sparse gene regulatory networks. In *Proc. of the 16th Intern. Symp. on Mathematical Theory of Networks and Systems*.
- [15] F. Steinke, M. Seeger, and K. Tsuda. Experimental design for efficient identification of gene regulatory networks using sparse bayesian models. *BMC Systems Biology*, 1(1):51, 2007.
- [16] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
- [17] R. Yoshida and M. West. Bayesian learning in sparse graphical factor models via variational mean-field annealing. *Journal of Machine Learning Research*, 99:1771–1798, August 2010.