An informative pathway-based prior for combinatorial inference with an application to protein-signalling networks

Steven Hill

Complexity Science Doctoral Training Centre, University of Warwick, UK

September 23, 2008

There is a growing interest in components of biological systems that jointly influence an output or response via combinatorial effects. Boolean functions are a natural way of describing these influences. This paper builds upon an existing noisy Boolean function model and Bayesian inference method which infer the inputs to the underlying Boolean function and its form, using a model prior to promote sparsity. We develop an informative prior to use alongside this existing sparsity prior, enabling us to exploit existing biological knowledge to aid the inference. We use network structure and pathway information to define a prior based on the idea that components that are 'close' in a pathway sense are more likely to be jointly involved in determining an output. We test our approach on synthetic data from two simulation regimes and apply it to proteomic data obtained from a study of signalling in breast cancer, the output of interest being drug response.

1 Introduction

Recent years have seen important advances in experimental methods in molecular biology. In particular, high-throughput technologies have allowed for the generation of high-dimensional genomic and proteomic data [1, 2]. Developments in computational and statistical methods have played a key role in utilising this data to address biological questions, through the use of increasingly realistic models.

The above advancements have accompanied and facilitated a growing interest in complex biological systems in which many heterogeneous components interact. This 'systems' approach moves away from the traditional approach of studying the function of a single protein or gene and moves towards understanding how several proteins or genes interact together [3]. For example, much work is being done on network inference which aims to discover interactions between components [4, 5]. Another area of study is of the combinatorial effects that several variables can jointly have on a certain output or measurement of interest. For example, studies have been done on SNP microarray data which aim to determine the combinatorial effect of variations in DNA data on disease state [6].

If available data is binary or can naturally be transformed into binary, Boolean logic provides a very useful framework for modeling combinatorial influences. The binary output can be written as a Boolean function of k binary arguments. These arguments correspond to the k components that jointly influence the output. It is usually the case that the k inputs of the Boolean function are embedded in a dataset containing many other variables and that the data is subject to noise. Hence we are interested in sparse, noisy Boolean functions. We use a model for noisy Boolean functions developed by Mukherjee *et al.* [7] and also implement their Bayesian inference method to infer the inputs of the Boolean function and its form. The arity of the function is assumed to be unknown but a prior is used to promote sparsity (i.e. keeping the arity small). The size of the space of possible Boolean functions is very large: there are 2^{2^k} different k-ary Boolean functions. Hence the sparsity prior also helps to reduce the size of the model space, making the inference problem computationally and statistically less difficult.

It is worth noting that Boolean functions are generally non-linear (for example, the XOR function). The model in [7] is state-dependent and so can capture complex interactions between components. Marginal statistical methods such as the 'log odds ratio' [8] or 'Naive Bayes classifier' [9] do not have this capability as they consider the influence of each component in isolation. Also, the emphasis in the model used here is on inferring the Boolean function and hence providing insight into the system rather than on prediction. However, the formulation of the problem is essentially the same as that of a classification problem and so the model can also be used for prediction purposes.

The main aim of this paper is to build upon the model in [7] by using an informative prior alongside the existing sparsity prior. An informative prior is one which incorporates existing biological knowledge into inference. This information can be a valuable resource, especially in situations where sample sizes are small. Indeed, it has been shown that informative priors can offer substantial gains over flat priors in these scenarios [10]. Our informative prior incorporates existing pathway information. This is similar in spirit to recent work by Wei and Li [11]. However, our approach and goals differ from theirs.

In this paper we focus on protein signalling networks. These networks contain well-defined pathways of components that perform a specific function. A well-known example is the MAPK pathway [12]. The prior is based on the intuitive idea that biological components that are 'close' in a pathway sense are more likely to be jointly involved in determining

an output or response. Hence, we first define a notion of 'pathway distance' that can be determined from existing knowledge of network structure and pathways. Then for each proposed model (set of inputs to the underlying Boolean function) we use pathway distance to specify a prior probability for the model.

We test this approach with and without the informative pathway distance prior under two different simulation regimes. We find that in both cases, using the informative prior gives clear improvements over using the sparsity prior alone. We then apply the original model, with just the sparsity prior, to data from protein signalling in breast cancer with the output of interest being drug responses. This itself is a novel application (in [7] the same proteomic data was used but in relation to breast cancer subtypes). The enhanced model with the informative prior is then also applied to the data. It is of importance to study this proteomic and drug response data as anti-cancer drugs tend to work effectively in some patients but not others. Insight into the combinatorial effect of activated proteins on drug response could help to understand why this heterogeneity occurs. The model could also potentially serve as a predictor of drug response for new patients.

The rest of the paper is organised as follows. Section 2 outlines the Boolean function probability model and gives details of the Bayesian inference method with sparsity prior as found in [7]. Section 3 then builds upon the model by introducing the new informative prior. In Section 4 we present our experimental results on synthetic data and cancer proteomic data. A brief explanation of the underlying biology is given here also. Section 5 summarises our findings, discusses their implications and highlights directions of further study.

2 Probability Model and Inference

In this section we start by giving basic definitions before outlining the noisy Boolean function probability model as found in [7]. We then describe the inference method used in [7] to learn Boolean functions from noisy data.

2.1 Initial Definitions and Notation

A *Boolean function* of arity k is a function $f : \{0,1\}^k \to \{0,1\}$. There are 2^k possible input states to a k-ary Boolean function. We denote binary input states by $\mathbf{X} = (X_1 \dots X_k)^\mathsf{T}$ and the output by a binary state $Y \in \{0,1\}$. Noise can then be applied to a Boolean function to give a noisy Boolean function.

A *noisy Boolean function* of arity k is a function $\theta : \{0,1\}^k \to [0,1]$. So now the output of the function is a real value in the unit interval. If the inputs are in state $\mathbf{X} = \mathbf{x}$, then we define $\theta_\mathbf{x} = \theta(\mathbf{x})$ as the probability that the output Y is of value 1. In other words we have $Y \mid (\mathbf{X} = \mathbf{x}, \theta_\mathbf{x}) \sim \text{Bernoulli}(\theta_\mathbf{x})$.

Noisy Boolean functions and deterministic Boolean functions can be related by a natural mapping, $f = \Psi(\theta)$, which is given by

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \theta(\mathbf{x}) > 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

In the probability model below we assume that the dataset contains n d-dimensional samples. The true inputs to the underlying Boolean function will be a subset of these d components. We write the complete set of components as $\mathbf{X} = (\mathbf{X}_1 \dots \mathbf{X}_n)$, where $\mathbf{X}_j = (\mathbf{X}_{1j} \dots \mathbf{X}_{dj})^\mathsf{T} \in \{0,1\}^d$. Hence X_{ij} denotes the i^{th} predictor from the j^{th} sample. For a subset of components, $A \subseteq \{1 \dots d\}$, we use $\mathbf{X}_{Aj}$ to denote the j^{th} sample for this subset. Finally, $\mathbf{X}_{A\cdot} = (\mathbf{X}_{A1} \dots \mathbf{X}_{An})$ denotes the full set of samples for components in set A. The full set of binary outputs is given by $\mathbf{Y} = (Y_1 \dots Y_n)$. If a noisy Boolean function is of arity k then we define $\boldsymbol{\theta}$ to be a vector containing all 2^k $\theta_\mathbf{x}$ values required to fully define the function.

2.2 Noisy Boolean Function Probability Model

Using the notation defined above, we describe the probability model for a noisy Boolean function. Suppose $\mathbf{Y} = (Y_1 \dots Y_n)$ is the output from a noisy Boolean function and that the actual inputs to the function are given by a subset $M \subseteq \{1 \dots d\}$. Note that this subset is actually unknown and below we describe a model inference method to determine it. Hence, each unique subset of inputs represents a different model. M is used to denote this subset and the associated model. Recall that from the definition of a noisy Boolean function, we have that $Y_j \mid (\mathbf{X}_{Mj} = \mathbf{x}, \theta_\mathbf{x}) \sim \text{Bernoulli}(\theta_\mathbf{x})$. This can be written explicitly as

$$P(Y_j \mid \mathbf{X}_{Mj} = \mathbf{x}, \theta_\mathbf{x}) = \theta_\mathbf{x}^{Y_j} (1 - \theta_\mathbf{x})^{1 - Y_j} \tag{2}$$

Clearly, the output of the Boolean function should only depend on the components in set M. Hence we assume that, given $\mathbf{X}_{Mj}$ (the j^{th} sample for the actual inputs to the Boolean function), Y_j is conditionally independent of all components not in M. That is,

$$P(Y_j \mid \mathbf{X}_j, M) = P(Y_j \mid \mathbf{X}_{M^\mathsf{c}j}, \mathbf{X}_{Mj}) = P(Y_j \mid \mathbf{X}_{Mj}) \tag{3}$$

where M^c denotes the set of components that are not in M.

We also assume that the outputs $Y_1 \dots Y_n$ are independent and identically distributed given the state of the inputs $\mathbf{X}_{M\cdot}$ and the parameter vector $\boldsymbol{\theta}$. So we have

$$P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}, \boldsymbol{\theta}\right) = P\left(Y_1 \dots Y_n \mid \mathbf{X}_{M1} \dots \mathbf{X}_{Mn}, \boldsymbol{\theta}\right) = \prod_{j=1}^{n} P\left(Y_j \mid \mathbf{X}_{Mj}, \boldsymbol{\theta}\right) \tag{4}$$

For a state $\mathbf{x}$ of the inputs to the Boolean function, let $B_{\mathbf{x}} = \{j \in \{1 \dots n\} : \mathbf{X}_{Mj} = \mathbf{x}\}$. Then let $\mathbf{Y}_{B_{\mathbf{x}}}$ be the collection of outputs Y_j such that $j \in B_{\mathbf{x}}$. Similarly for $\mathbf{X}_{MB_{\mathbf{x}}}$. Then (2) and (4) give

$$P\left(\mathbf{Y}_{B_{\mathbf{x}}} \mid \mathbf{X}_{MB_{\mathbf{x}}}, \theta_{\mathbf{x}}\right) = \prod_{j \in B_{\mathbf{x}}} P\left(Y_j \mid \mathbf{X}_{Mj} = \mathbf{x}, \theta_{\mathbf{x}}\right) = \prod_{j \in B_{\mathbf{x}}} \theta_{\mathbf{x}}^{Y_j} \left(1 - \theta_{\mathbf{x}}\right)^{1 - Y_j} = \theta_{\mathbf{x}}^{s_{\mathbf{x}}} \left(1 - \theta_{\mathbf{x}}\right)^{n_{\mathbf{x}} - s_{\mathbf{x}}} \tag{5}$$

where $n_{\mathbf{x}} = |B_{\mathbf{x}}|$ is the number of samples in which the inputs to the Boolean function are in state $\mathbf{x}$ and $s_{\mathbf{x}} = \sum_{j \in B_{\mathbf{x}}} Y_j$ is the number of these samples that have output $Y = 1$. Since every sample number j is in a single $B_{\mathbf{x}}$, it then follows from (4) that

$$P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}, \boldsymbol{\theta}\right) = \prod_{j=1}^{n} P\left(Y_j \mid \mathbf{X}_{Mj}, \boldsymbol{\theta}\right) = \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \prod_{j \in B_{\mathbf{x}}} P\left(Y_j \mid \mathbf{X}_{Mj} = \mathbf{x}, \theta_{\mathbf{x}}\right) = \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \theta_{\mathbf{x}}^{s_{\mathbf{x}}} \left(1 - \theta_{\mathbf{x}}\right)^{n_{\mathbf{x}} - s_{\mathbf{x}}} \tag{6}$$

So the joint probability of all the Y_j given the states of the inputs to the underlying Boolean function and the parameter vector is a product of Binomial kernels.

2.3 Inference Method

We now describe the inference method employed in [7] to infer the model (i.e. the underlying inputs to the Boolean function). The method uses Bayesian model averaging and Markov chain Monte Carlo.

Using Bayes' rule, the posterior distribution over models is given, up to proportionality, by

$$P\left(M \mid \mathbf{Y}, \mathbf{X}\right) \propto P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}\right) P\left(M\right) \tag{7}$$

since $P\left(M \mid \mathbf{X}\right) = P\left(M\right)$. This is because the state of the components alone tell us nothing about the underlying model.

The first term on the right hand side of (7) is the marginal likelihood which is obtained by integrating over all possible parameters,

$$P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}\right) = \int P\left(\mathbf{Y}, \boldsymbol{\theta} \mid \mathbf{X}_{M\cdot}\right) \mathrm{d}\boldsymbol{\theta} = \int P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right) \mathrm{d}\boldsymbol{\theta} \tag{8}$$

Here we have used Bayes' rule along with the fact that $p\left(\boldsymbol{\theta} \mid \mathbf{X}_{M\cdot}\right) = p\left(\boldsymbol{\theta}\right)$. Integrating over parameters is equivalent to averaging over all possible noisy Boolean functions with inputs M.

We now just need to know the form of the parameter prior $p\left(\boldsymbol{\theta}\right)$. Note that we already know the form of $P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}, \boldsymbol{\theta}\right)$ from (6) above. As in [7] we assume that the parameters $\theta_{\mathbf{x}}$ are independent,

$$p\left(\boldsymbol{\theta}\right) = \prod_{\mathbf{x} \in \{0,1\}^{|M|}} p\left(\theta_{\mathbf{x}}\right) \tag{9}$$

This, along with (6) means that (8) becomes

$$P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}\right) = \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \int \theta_{\mathbf{x}}^{s_{\mathbf{x}}} \left(1 - \theta_{\mathbf{x}}\right)^{n_{\mathbf{x}} - s_{\mathbf{x}}} p\left(\theta_{\mathbf{x}}\right) \mathrm{d}\theta_{\mathbf{x}} \tag{10}$$

The parameter prior $p\left(\theta_{\mathbf{x}}\right)$ is assumed to be Beta distributed with identical hyper-parameters $\alpha = \beta$. Hence the prior is symmetric about $\theta_{\mathbf{x}} = \frac{1}{2}$. This means that given a model M, all possible underlying deterministic Boolean functions with inputs M are a priori equally probable. Also, we assume that $\alpha, \beta < 1$ which results in the probability mass being concentrated around 0 and 1. So the prior prefers noisy Boolean function 'success' parameter values $\theta_{\mathbf{x}}$ that are close to 0 or 1. A full explanation of this is given in [7]. So we now have the marginal likelihood in a closed form,

$$P\left(\mathbf{Y} \mid \mathbf{X}_{M\cdot}\right) = \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \int \theta_{\mathbf{x}}^{s_{\mathbf{x}}} \left(1 - \theta_{\mathbf{x}}\right)^{n_{\mathbf{x}} - s_{\mathbf{x}}} \frac{\Gamma\left(\alpha + \beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)} \theta_{\mathbf{x}}^{\alpha - 1} \left(1 - \theta_{\mathbf{x}}\right)^{\beta - 1} \mathrm{d}\theta_{\mathbf{x}} \tag{11}$$

$$= \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \frac{\Gamma\left(\alpha + \beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)} \int \theta_{\mathbf{x}}^{s_{\mathbf{x}} + \alpha - 1} \left(1 - \theta_{\mathbf{x}}\right)^{n_{\mathbf{x}} - s_{\mathbf{x}} + \beta - 1} \mathrm{d}\theta_{\mathbf{x}} \tag{12}$$

$$= \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \frac{\Gamma\left(\alpha + \beta\right)}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)} \frac{\Gamma\left(s_{\mathbf{x}} + \alpha\right)\Gamma\left(n_{\mathbf{x}} - s_{\mathbf{x}} + \beta\right)}{\Gamma\left(\alpha + \beta + n_{\mathbf{x}}\right)} \tag{13}$$

The last line follows from the fact that the integrand in (12) is a Beta pdf, up to normalisation, with parameters $s_{\mathbf{x}}+\alpha$ and $n_{\mathbf{x}}-s_{\mathbf{x}}+\beta$. Hence it integrates to the inverse of the normalising factor.

Finally, to get the model posterior (up to proportionality, see (7)) we need to define a prior distribution on models. We use the sparsity prior as in [7] and then, in Section 3, we add to it with a novel informative prior. The sparsity prior takes the following form,

$$P(M) \propto \begin{cases} \exp(\lambda_s \min(0, k_0 - |M|)) & \text{if } |M| \leq k_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

In general, the prior prefers models with a smaller number of inputs, but models with size less than or equal to k_0 are equally probable. $k_{\max}$ is an optional strict upper limit on model size and λ_s is the strength of the prior. Methods to determine what the parameters k_0, $k_{\max}$ and λ_s should be are suggested in [7] and [10].

Since the size of the model space is too large to explicity determine the model posterior, we use Markov chain Monte Carlo [13] to sample from it. In particular, a Metropolis-Hastings sampler is used. The reader is referred to [7] for full details of the proposal distribution used.

A particularly useful posterior probability is $P(j \in M \mid \mathbf{X}, \mathbf{Y})$. That is, the probability that a given variable is an input to the underlying Boolean function. An asymptotically valid estimate of this probability can be determined from the MCMC samples, provided the Markov chain has converged. If $M^{(1)} \ldots M^{(T)}$ are the sampled models then we have

$$P(j \in M \mid \mathbf{X}, \mathbf{Y}) \approx \frac{1}{T} \sum_{t=1}^{T} I_{M^{(t)}}(j) \tag{15}$$

where I_A is the standard indicator function on set A.

We will also make use of the posterior distribution over the parameter $\theta_{\mathbf{x}}$. By an application of Bayes' rule we have

$$p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}, M) \propto P(\mathbf{Y} \mid \boldsymbol{\theta}, \mathbf{X}_{M\cdot})\, p(\boldsymbol{\theta} \mid \mathbf{X}_{M\cdot}) \tag{16}$$

This is exactly the integrand in (8) and so from (12) we get

$$p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}, M) \propto \prod_{\mathbf{x} \in \{0,1\}^{|M|}} \theta_{\mathbf{x}}^{s_{\mathbf{x}}+\alpha-1} (1-\theta_{\mathbf{x}})^{n_{\mathbf{x}}-s_{\mathbf{x}}+\beta-1} \tag{17}$$

We find $p(\theta_{\mathbf{x}} \mid \mathbf{Y}, \mathbf{X}, M)$ from this by taking all the $\theta_{\mathbf{x}}$ terms. This gives a Beta distribution, up to a normalising constant. So we have

$$p(\theta_{\mathbf{x}} \mid \mathbf{Y}, \mathbf{X}, M) = \text{Beta}(\theta_{\mathbf{x}} \mid s_{\mathbf{x}}+\alpha, n_{\mathbf{x}}-s_{\mathbf{x}}+\beta) \tag{18}$$

A final posterior probability we will use is a predictive one. Given a new data sample, $\mathbf{X}_{M(n+1)} = \mathbf{x}$, what is the probability of a 'successful' output, $Y_{(n+1)} = 1$. We still use $\mathbf{X}_{M\cdot}$ to denote the first n samples. By integrating over the parameter $\theta_{\mathbf{x}}$ and using Bayes' rule we have

$$\begin{aligned} P\left(Y_{(n+1)} = 1 \mid \mathbf{X}_{M(n+1)} = \mathbf{x}, \mathbf{Y}, \mathbf{X}_{M\cdot}\right) &= \int P\left(Y_{(n+1)} = 1 \mid \theta_{\mathbf{x}}, \mathbf{X}_{M(n+1)} = \mathbf{x}, \mathbf{Y}, \mathbf{X}_{M\cdot}\right) p\left(\theta_{\mathbf{x}} \mid \mathbf{X}_{M(n+1)} = \mathbf{x}, \mathbf{Y}, \mathbf{X}_{M\cdot}\right) \mathrm{d}\theta_{\mathbf{x}} \\ &= \int P\left(Y_{(n+1)} = 1 \mid \theta_{\mathbf{x}}, \mathbf{X}_{M(n+1)} = \mathbf{x}\right) p\left(\theta_{\mathbf{x}} \mid \mathbf{Y}, \mathbf{X}_{M\cdot}\right) \mathrm{d}\theta_{\mathbf{x}} \end{aligned}$$

Now by definition of $\theta_{\mathbf{x}}$, $P\left(Y_{(n+1)} = 1 \mid \theta_{\mathbf{x}}, \mathbf{X}_{M(n+1)} = \mathbf{x}\right) = \theta_{\mathbf{x}}$ and $p(\theta_{\mathbf{x}} \mid \mathbf{Y}, \mathbf{X}_{M\cdot})$ is the Beta distribution from (18). Hence, the predictive probability is simply the expected value of the Beta distribution. So we have

$$P\left(Y_{(n+1)} = 1 \mid \mathbf{X}_{M(n+1)} = \mathbf{x}, \mathbf{Y}, \mathbf{X}_{M\cdot}\right) = \frac{s_{\mathbf{x}}+\alpha}{n_{\mathbf{x}}+\alpha+\beta} \tag{19}$$

3 Informative Prior

In this section, we describe an informative pathway-based distance prior that we use alongside the sparsity prior. In many scenarios there is information available about patterns of interaction between system components. This can be represented by graphs where nodes represent components and edges the interactions between them. Examples include gene regulatory networks and protein interaction networks. Figure 1 is an example of such a network. The components can often also be divided into pathways according to the signal transduction. Pathways carry information, passed from an upstream source in the cell to a downstream target [12]. More information on the biology of protein signalling is in Section 4.2. We aim to exploit this network and pathway information to help with the model inference.

The prior works on the idea that components in the same pathway, which jointly influence an output (drug response for example), are likely to be close to each other (in a network sense). However, there may be components from several different pathways which have a combinatorial influence on the drug response. So we would like to be able to assign a score to any given model M, which is a meaure of how far apart the components of M are from each other. We call this the '*pathway distance*' and denote it by L_M.

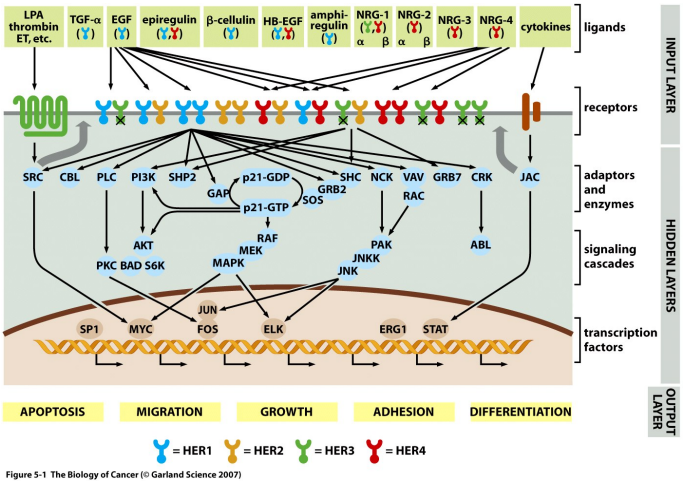

Figure 1: The ErbB signalling network. (Picture taken from [12]).

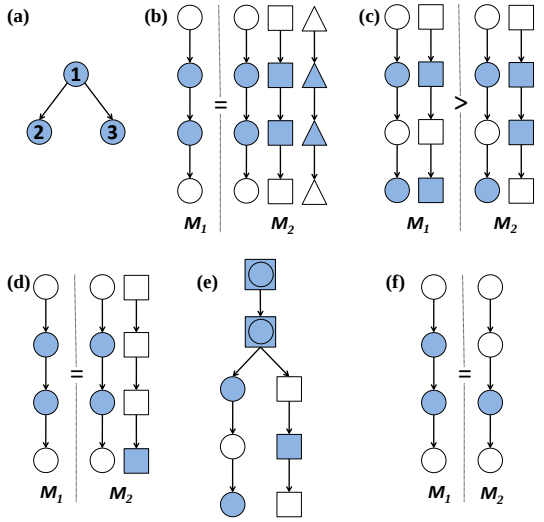

Figure 2: Required properties of the pathway-based distance prior. Shaded components are contained in the model M and different shapes represent different pathways. Where two models (M_1 and M_2) are compared, '=' and'>' signs denote $L_{M_1} = L_{M_2}$ and $L_{M_1} > L_{M_2}$ respectively. (a) Distance is defined in terms of undirected edges in order to capture closeness between components 2 and 3. (b-d&f) Model distance comparisons: see text for details. (e) The upper two components are in both pathways. In calculating L_M, we would like to count the distance between them once only.

Edges in a network can either be directed or undirected. Figure 2(a) shows three components with two directed edges. Components 2 and 3 are clearly close to each other in the network but there is no directed path between them. In order to capture this closeness we define the distance between components i and j, denoted by l_{ij}, as follows:

$$l_{ij} = \text{number of edges in the shortest } \textit{undirected} \text{ path between components } i \text{ and } j$$

If components i and j are not connected by an undirected path, we set $l_{ij} = \infty$.

There are several natural properties that we would like our 'pathway distance' to have. We summarise these below and illustrate them in Figure 2.

1. The distance prior should not act as a sparsity prior. It should be agnostic to the number of components in M and to the number of different pathways they come from.

2. If the components of model M_1 form a certain pattern in a single pathway and the components of model M_2 form the same pattern in each of $p > 1$ pathways, then we would like $L_{M_1} = L_{M_2}$. See Figure 2(b). This is enforcing that the number of pathways has no effect (see Property 1 above).

3. Figure 2(c) depicts another scenario. We again have two models M_1 and M_2. Both have two components in each of two pathways. M_1 has a distance of two between the components in both pathways whilst M_2 has a distance of one in one pathway and a distance of two in another. We would like $L_{M_2} < L_{M_1}$.

We define a *singleton* to be a component in a model M such that there are no other components of M in the same pathway as it.
We now consider three further desiderata relating to singletons.

4. If we add a singleton to a model M, the within pathway distances between pairs of components in M remain unchanged. Hence, we would like L_M to remain unchanged also. See Figure 2(d).

5. A model M could contain components that are in multiple pathways, see Figure 2(e). In calculating L_M, we wish to avoid double-counting of distances.

6. Finally, we note that a model may contain only singletons. There is no concept of within-pathway distance here so it is not obvious what L_M should be in this case. We choose to set $L_M = 0$. However, we do not want our prior to prefer singleton models over the slightly less simple model of two components of M connected by a single edge, see Figure 2(f).

We now discuss the implications of the properties above.
Property 1 implies we cannot simply add pair-wise component distances together to obtain L_M. Also, since components in different pathways are likely to have a big distance between them or not be connected at all, we should only measure the distance between components that are in the same pathway. This avoids automatically penalising models consisting of components from more than one pathway and avoids infinite distances.
Property 3 precludes the use of the maximum (or minimum) of the pair-wise component distances, l_{ij}, to define L_M. However, a mean distance would satisfy the first three properties.
Suppose there are m pathways, numbered $i = 1 \dots m$ and let $P_i = \{j \in \{1 \dots d\} : j \in \text{pathway } i\}$ be the set of components that are in pathway i and let $C_i = M \cap P_i$ be the set of components that are in pathway i and in M. We could either average over the set $\{l_{jk} : j, k \in C_i \text{ for some } i \in \{1 \dots m\}\}$ or we could average over each pathway separately by taking the average over each set $\{l_{jk} : j, k \in C_i\}$ for $i = 1 \dots m$. Then the 'pathway distance' would be found by averaging these m pathway averages.
Properties 4 and 5 preclude this latter option of averaging over the pathways individually. Hence we take the former averaging option leading to the following definition for L_M,

$$L_M = \begin{cases} 0 & \text{if } M \text{ contains only singletons} \\ \left(\frac{2}{\sum_{i=1}^{m} |C_i|(|C_i|-1)} \sum_{i=1}^{m} \sum_{\substack{j,k \in P_i \\ j<k}} l_{jk} \right) - 1 & \text{otherwise} \end{cases} \tag{20}$$

The term in front of the summations is the inverse of the number of l_{jk} we are averaging over. It is simply the inverse of $\sum_{i=1}^{m} \binom{|C_i|}{2}$. We subtract one in the second line in order to satisfy Property 6.

Our distance prior takes the same exponential form as the sparsity prior. So the final model prior becomes

$$P(M) \propto \begin{cases} \exp(\lambda_s \min(0, k_0 - |M|)) \exp(-\lambda_d L_M) & \text{if } |M| \leq k_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

where λ_d is the strength parameter for the informative distance prior.

4 Method and Results

We tested our informative distance prior on several simulated datasets to see if it improves on inferences made with just the sparsity prior. We also applied the inference method, with and without the informative prior, to a proteomic dataset from breast cancer to try to infer combinatorial influences of signalling proteins on drug response. The methods and results are presented below.

4.1 Simulated Data

Synthetic data was produced using two different simulation methods. Both contained 15 possible inputs to the Boolean function (i.e. $d = 15$), forming a network with three pathways as shown in Figure 3. The underlying Boolean function was chosen to be

$$X_3 \wedge \neg X_5 \wedge X_7 \wedge X_{11} \tag{22}$$

for both simulation methods. We call this function f.

This particular model, $M = \{3, 5, 7, 11\}$, was chosen since components X_3,X_5 and X_7 are all close together in the same pathway. Hence we can see if using the distance prior alongside the sparsity prior infers this model more effectively than with a flat prior or with just the sparsity prior.

Put into the context of cancer and drug response, this particular Boolean function means that the drug works only when components X_3,X_7 and X_{11} are 'on' and component X_5 is 'off'. This might arise if, for example, components X_5, X_7 and X_{11} cause cell proliferation but the drug only inhibits X_7 and X_{11}. Hence X_5 has to be 'off', else cell

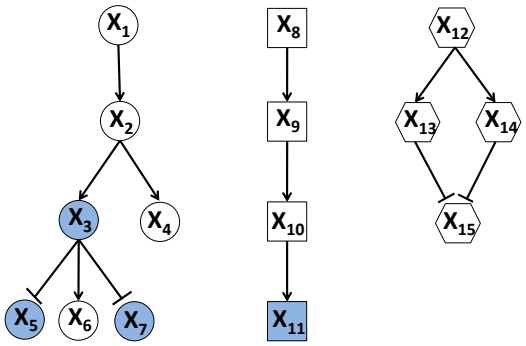

Figure 3: The graph used in the simulations. Shaded components are the actual inputs to the underlying Boolean function and different shapes represent different pathways. Edges with arrowheads represent 'promoting' edges whilst edges with flat heads represent 'inhibiting' edges.

proliferation continues even with the drug applied. Component X_3 could, for example, affect a downstream target not on the network. This target can also cause cell proliferation , but is inhibited when X_3 is turned 'off' by the drug.

Twenty datasets were produced for each simulation method. Ten with 200 samples ($n = 200$) and the other ten with $n = 50$. The data X_{ij} for $i = 1 \dots d$ and $j = 1 \dots n$ is set differently in each simulation method. Both simulations have the following method in common:

The starting components in each pathway are set to one with fixed probabilities (and zero otherwise),

$$P(X_{ij} = 1) = \begin{cases} 0.95 & \text{if } i = 1 \\ 0.7 & \text{if } i = 8 \\ 0.8 & \text{if } i = 12 \end{cases} \tag{23}$$

All other components, except component X_{15}, have a single parent. If the edge that links the component to its parent is a 'promoting' edge (see Figure 3), the component takes on the value of its parent with probability ϑ. If the edge is an 'inhibiting' edge, the component takes on the opposite truth value to its parent with probability ϑ. The exceptions to this are components X_7 and X_{15}.

Component X_7 takes on the value of its parent with probability 0.5. Hence it is essentially random. A motivation for this is if the n samples are from different tissues (for example, different tumours). Then we might expect the majority of the network structure and signalling processes to be identical across the tissues. This lower transition probability represents a signalling process that varies across the cell lines.

The other exception, component X_{15}, has two parents connected by 'inhibiting' edges. If $X_{13,j} = X_{14,j} = 1$, then $X_{15,j} = 0$ with probability ϑ. If $X_{13,j} = 0$ or $X_{14,j} = 0$, then $X_{15,j} = 1$ with probability ϑ.

We now describe the difference between Simulation Method One and Method Two.

Simulation Method One

We take $\vartheta = 0.95$. The purpose of this 'transmission' probability is to represent the noise in a signal being passed from one component to another within a cell.

Simulation Method Two

We take $\vartheta = 0.98$. So here, the 'transmission' of information is less corrupted by noise but we subsequently add in random noise to represent stochasticity in measurements in the laboratory. We do this by changing 1% of the X_{ij} to their opposite truth values. The 1% are chosen at random.

Finally, we describe how the response data Y_j is produced. This is the same for both simulation methods. We make the underlying Boolean function (22), f, noisy by choosing parameters $\theta_{\mathbf{x}}$. If $f(\mathbf{x}) = 1$ we choose $\theta_{\mathbf{x}} = 0.9$ and if $f(\mathbf{x}) = 0$ we choose $\theta_{\mathbf{x}} = 0.1$. This can be written out in full as

$$P(Y_j = 1) = \begin{cases} 0.9 & \text{if } f(\mathbf{x}) = 1 \\ 0.1 & \text{if } f(\mathbf{x}) = 0 \end{cases} \qquad P(Y_j = 0) = \begin{cases} 0.1 & \text{if } f(\mathbf{x}) = 1 \\ 0.9 & \text{if } f(\mathbf{x}) = 0 \end{cases} \tag{24}$$

The noise here represents the stochasticity in measurements and in the output itself; for example due to factors not taken into account in our network and hence, Boolean function.

The same prior hyper-parameters are used for both simulations. We set $k_0 = 4$ and $k_{\max} = 6$. The prior strengths used are $\lambda_s = 3$ and $\lambda_d = 2$. These are set with reference to Jeffreys' scale [14] which relates prior odds ratios to an intuitive 'strength of evidence'. The hyper-parameter of the prior for $\theta_{\mathbf{x}}$ is $\alpha = \beta = 0.9$. We run the MCMC algorithm for 30,000 iterations, discarding the first 3,000 as 'burn-in' (this value was set with reference to convergence diagnostics in [7]).

We now present the results from our simulations. Figure 4(a-b) shows histograms of the posterior distribution over models for one of the ten datasets of Simulation Method One with $n = 200$. The first is from using sparsity prior only

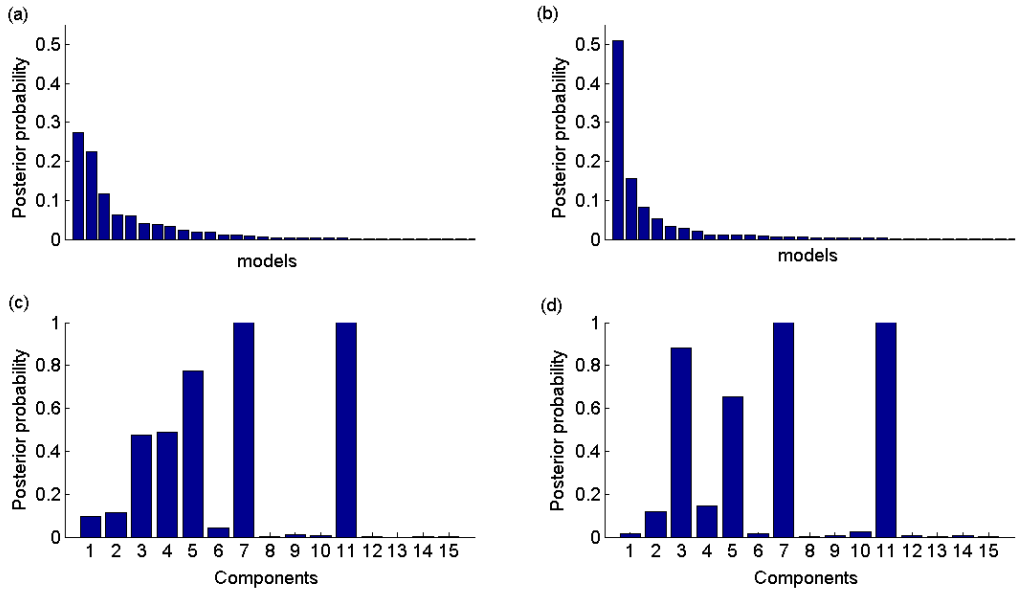

Figure 4: Simulation results from Method One with $n = 200$. Posterior distributions over (a) models (30 highest scoring shown) using sparsity prior only, (b) models (30 highest scoring shown) using sparsity and distance prior, (c) individual components using sparsity prior only, (d) individual components using sparsity and distance prior. Results are from one of the ten datasets.

and the second also uses distance prior. For sparsity prior only, the most probable model is $M = \{4, 5, 7, 11\}$ with a probability of 0.27, whereas adding the distance prior gives the correct most probable model, $M = \{3, 5, 7, 11\}$ with a probability of 0.51. Therefore adding the distance prior has helped to find the correct model in this case and has also made the posterior distribution over models slightly less diffuse, with a clear most probable model. Figure 4(c-d) shows the posterior distribution over individual components (calculated using (15)) for the same dataset. This is a useful way of seeing which components are likely to be inputs to the underlying Boolean function. We can clearly see how the distance prior is helping. Component X_3 is favoured over component X_4 since it is close to components X_5 and X_7 in the pathway.

Figure 5 shows average ROC (Receiver Operating Characteristic) curves for both simulation methods and sample sizes using different priors. The method for plotting ROC curves is as follows. Let $\tau \in [0, 1]$. We threshold, at value τ, posterior probabilities over individual components (see (15)). In other words we find all components X_j such that $P(j \in M \mid \mathbf{X}, \mathbf{Y}) \geq \tau$. Let $x(\tau)$ be the proportion of correct inputs (true positives) selected at threshold τ. Similarly let $y(\tau)$ be the proportion of incorrect inputs (false positives) selected at threshold τ. The proportions are taken with respect to the possible number of correct/incorrect inputs respectively (4 and 11 in our case). The curve is produced by plotting $(x(\tau), y(\tau))$ for $\tau \in [0, 1]$. We also plot ROC curves for the absolute log odds ratio, which is a measure of pairwise association for binary data (see [8]). Here it measures correlations between each individual component and the output. These values are then thresholded in the same way as above to produce the plot. The ROC curves are averages over the ten datasets. Figure 6 shows how the area under the ROC curve (AUC) is affected by choice of prior for each dataset. The AUC is an indicator of the quality of ranking under the posterior probabilities produced by each method. Higher AUC values correspond to lower error rates.

Our results show us that the probability model and Bayesian inference method used have significant benefits over the log odds ratio even when a flat prior is used. This is because the outputs Y_j are a Boolean function of components acting in combination. Our method is able to capture this whereas the log odds ratio just scores components individually.

As expected the AUC values obtained with $n = 50$ are lower than with $n = 200$ due to small sample sizes making the inference much more difficult. Similarly the AUC values suggest that Simulation Method Two is a slightly harder problem than Simulation Method One. However, in both simulation methods and sample sizes we see that using the distance prior alongside the sparsity prior has a positive effect (higher AUC value) over just using the sparsity prior or using a flat prior. This positive effect is the most pronounced in Simulation Method One with $n = 200$. Surprisingly, the AUC values for sparsity prior are only slightly higher on average than those with a flat prior. Indeed, for Simulation Method Two with $n = 50$, the sparsity prior appears to have a negative effect over using a flat prior.

Table 1 shows what proportion of the incorrect components are selected, on average, before all correct components are selected, using the thresholding procedure above. These values come from the ROC curves.

We see that with $n = 200$ the distance prior makes a significant difference, but with $n = 50$ it has a negative effect. However, the AUC values for $n = 50$ were better with the distance prior than with sparsity alone. The ROC curves explain this by showing that the distance prior helps when a high threshold is taken on the posterior over components. This is significant as generally one is interested in high thresholds where only a few inputs are selected.

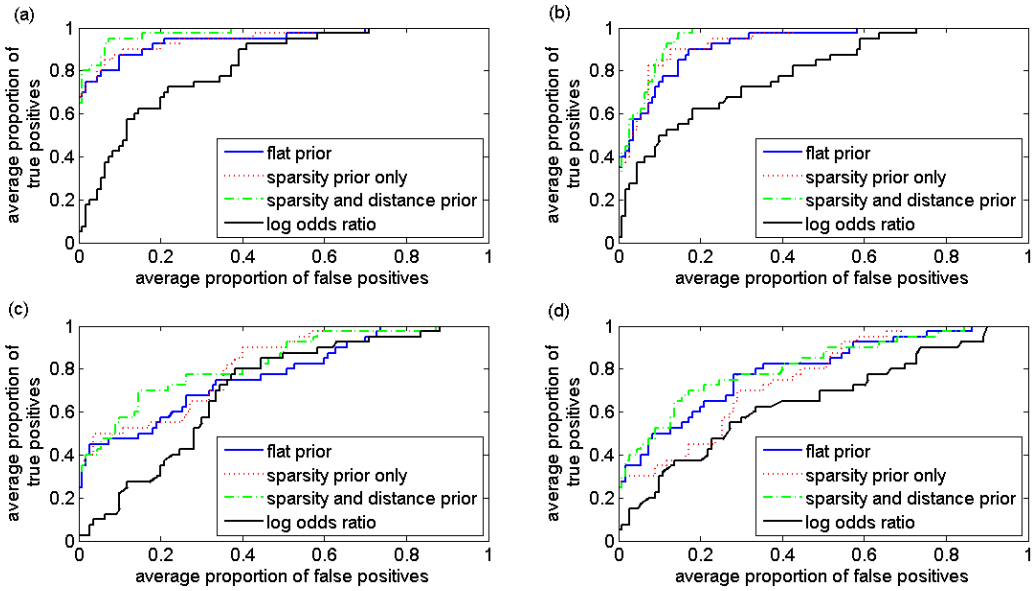

Figure 5: Simulation results. ROC (Receiver Operating Characteristic) curves for different model priors and log odds ratio is also plotted for comparison (see main text for an explanation of these). Each curve is an average over the ten datasets. (a) Simulation Method One with n=200, (b) Simulation Method Two with n=200, (c) Simulation Method One with n=50, (d) Simulation Method Two with n=50.

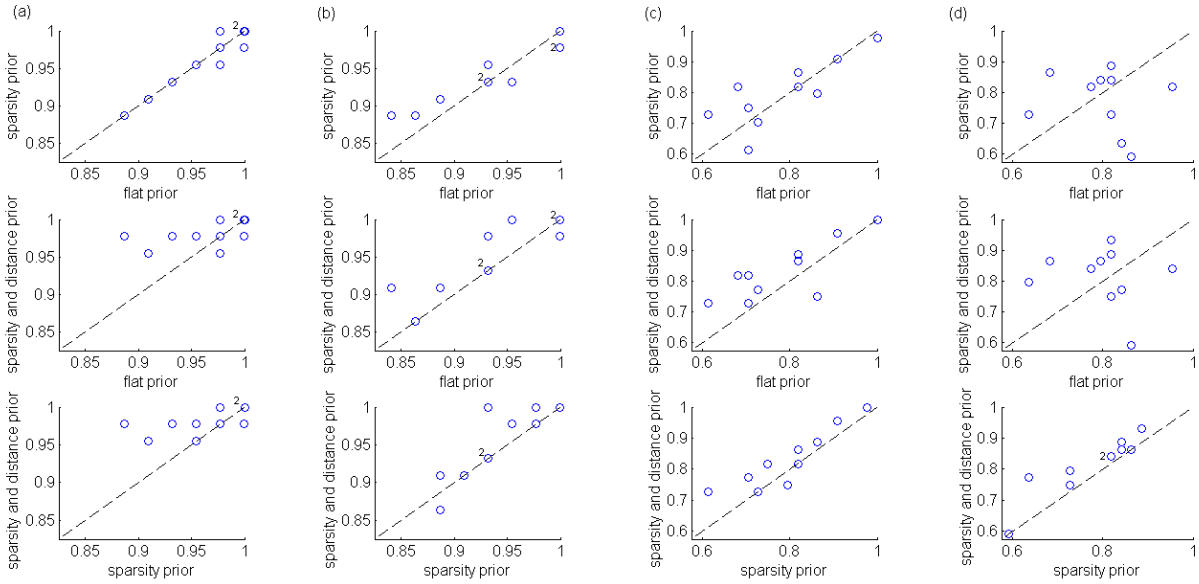

Figure 6: Simulation results. Scatter plots comparing AUC (Area Under the ROC Curve) values for different model priors. Each plot contains ten data points, one for each dataset. A '2' by a data point denotes two data points having the same values (a) Simulation Method One with n=200, (b) Simulation Method Two with n=200, (c) Simulation Method One with n=50, (d) Simulation Method Two with n=50.

Table 1: Simulation results. Proportion of incorrect components selected as inputs to the Boolean function before all correct inputs are selected. These values are averages over ten datasets.

		Simulation Method One	Simulation Method Two
n=200	log odds ratio	0.71	0.73
	flat prior	0.70	0.58
	sparsity prior only	0.61	0.43
	distance and sparsity prior	0.37	0.18
n=50	log odds ratio	0.88	0.90
	flat prior	0.74	0.86
	sparsity prior only	0.66	0.69
	distance and sparsity prior	0.87	0.85

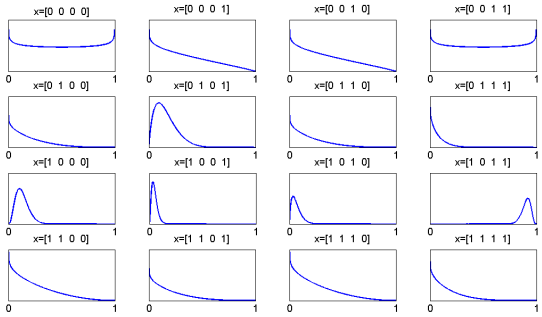

Figure 7: Posteriors over parameters $\theta_{\mathbf{x}}$ for the correct underlying model using Simulation Method One with $n = 200$. These plots are typical of the ten datasets.

As shown in [7], (18) can be used to try to find the form of the Boolean function and (19) can be used to make predictions. We do this for Simulation Method One with $n = 200$. Note that these results are independent of the model prior used.

Figure 7 shows the posterior distributions over parameters $\theta_{\mathbf{x}}$. The Boolean function $X_3 \wedge \neg X_5 \wedge X_7 \wedge X_{11}$ comes through in these plots, but some of the distributions are very diffuse. Correlations within pathways make certain combinations $\mathbf{x}$ of these four inputs very unlikely: these posteriors correctly reflect uncertainty inherent in small sample inference of higher order effects (such as combinatorial influences).

Leave-one-out cross validation (LOOCV) on each of the ten datasets, using predictive probability (19), gives a mean accuracy rate of 89.95% with standard deviation 2.68%. This is an excellent result given that the noisy Boolean function gives the correct truth value with probability 0.9, meaning that we would not expect prediction using these datasets to be more accurate than 90%. We also perform LOOCV on the smaller $n = 50$ sample size (Simulation Method One). The mean accuracy rate is 88.60% with standard deviation 5.50%. This is a very good result again. However we note that we did not perform a true cross-validation here.[1]

4.2 Proteomic Data

We now present results from analysis of proteomic data, obtained from a study of signalling in breast cancer. We start by briefly outlining some of the underlying biology.

Proteins are formed inside a cell through the process of gene transcription and translation. Certain proteins can form signalling pathways. Cells receive signals from their external environment through receptors on the cell surface. This signal can then cause a chain of reactions within the cell through post-translational modifications of the proteins. Phosphorylation is an example of such a modification. This is where a phosphate group gets added onto an amino acid of a protein, allowing it to perform highly specific enzymatic behaviour (the protein is said to be 'activated' when in this state). Generally, only small amounts of phosphorylated proteins are required to produce an enzymatic effect on downstream processes within the cell.

The proteomic data consists of present/absent calls for 32 phosphorylated proteins which relate to the EGFR (Epidermal Growth Factor Receptor) signalling pathway. The network and pathway structure used is shown in Figure 8. This data was obtained using the KinetWorks$^{\text{TM}}$ system (Kinexus Bioinformatics Corporation, Vancouver, Canada) for 24 different breast cancer cell lines. This collection of cell lines has been shown to exhibit the heterogeneity found in primary tumours [15].

[1]We performed LOOCV with a model learnt from a single MCMC run on the whole dataset. A true cross-validation approach would, for each training dataset, perform an MCMC run, learn a model and then use this model to perform the cross-validation. Computational considerations meant we could not carry this out. The difference between our accuracy rates and those of a true cross-validation is likely to be small for $n = 200$, but could be more significant for $n = 50$.

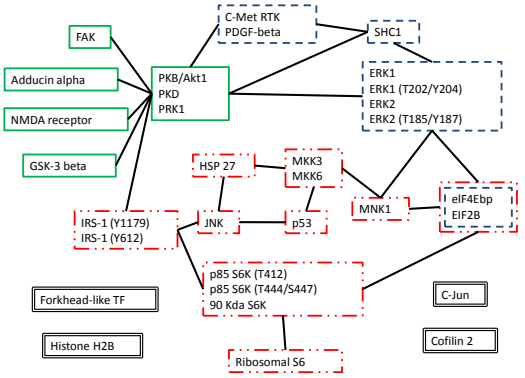

Figure 8: Network and pathway structure for the 32 proteins that constitute the proteomic data. Different colours/box edge patterns represent the different pathways. The components with a double-edged box are all individual pathways.

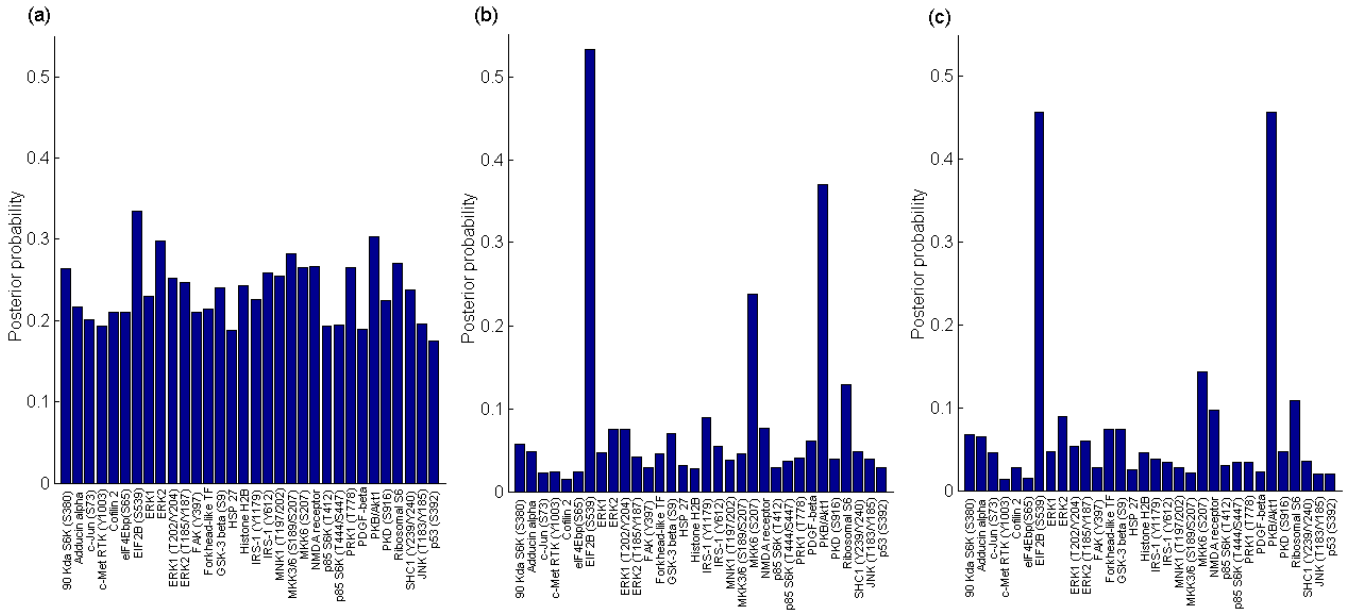

Figure 9: Proteomic data results. Posteriors over individual components when using (a) flat model prior, but a strict limit on model size of $k_{max} = 8$, (b) sparsity prior only , (c) sparsity and distance prior.

We investigate the combinatorial effect these signalling proteins can have on drug response using our Boolean function model and inference method. The drug we study here is 'CGC-11047'; a possible polyamine displacer that could interfere with processes relating to cell growth [16]. 'GI50' data is used to obtain drug responses for the 24 cell lines. 'GI50' is a standard measure of the efficacy of anti-cancer agents. More specifically, it is the concentration that causes 50% growth inhibition compared to a baseline [17]. The data is binarised using the median value as a threshold.

We use the same parameters as in the simulated data with the exception of k_0 and $k_{\max}$ which are now lowered to 2 and 4 respectively on account of the small sample size. Also, we use a strict limit of $k_{\max} = 8$ for the 'flat prior' for computational reasons.

Figure (9) shows the posterior distribution over individual components for different model priors. We can see that the sparsity prior has a significant effect in rendering the inference more tractable. Two proteins stand out in particular. These are 'Eukaryotic Translation Initiation Factor 2B epsilon subunit (S539)' and 'Protein kinase B alpha (Akt1)'. A third protein also stands out slightly: 'MKK6(2) (S207)'. However, this protein has less of a prevalence when the distance prior is also used. Apart from this, the distance prior seems to have little benefit over the sparsity prior. We discuss these results below.

We take the two proteins with the highest posterior probabilities as our model, $M = \{\text{EIF2B}, \text{Akt1}\}$, and try to infer the underlying Boolean relationship for this particular model. Figure (10) shows the posterior distributions over parameters $\theta_{\mathbf{x}}$. Recall that $\theta_{\mathbf{x}} = P(Y_j = 1 \mid \mathbf{X}_{Mj} = \mathbf{x})$. Table 4.2 also shows predictive probabilities (using (19)) for each of the possible combinations of truth values the inputs can take. These both suggest that the drug works according to the Boolean relationship, EIF2B NAND Akt1. These two proteins also have the top two rankings with

Table 2: Proteomic data. Predictive probabilities for each input state using model $M = \{\text{EIF2B}, \text{Akt1}\}$.

EIF2B	Akt1	P(drugworks)
0	0	0.81
0	1	0.84
1	0	0.81
1	1	0.18

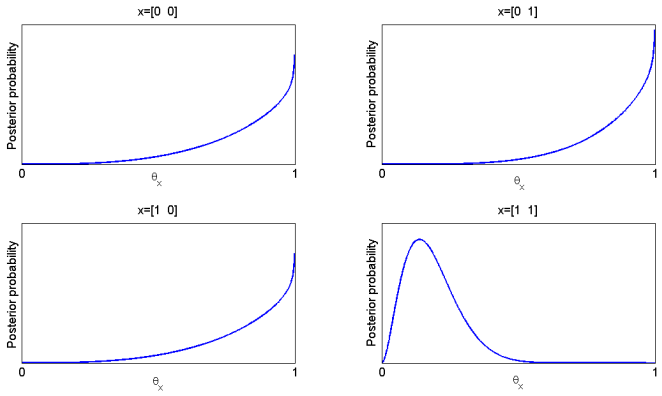

Figure 10: Proteomic data. Posteriors over parameters $\theta_{\mathbf{x}}$ for model $M = \{\text{EIF2B}, \text{Akt1}\}$.

absolute log odds ratio. This is expected due to the simple linear form of the Boolean function, which log odds is fully capable of capturing (unlike the more complicated function in the simulated results).

Finally, we performed Leave-one-out cross validation on the dataset using (19) and model $M = \{\text{EIF2B}, \text{Akt1}\}$. This gave a very good accuracy rate of 91.67%. Leave-one-out cross validation using model $M = \{\text{EIF2B}, \text{Akt1}, \text{MKK6(2)}\}$ gave a lower accuracy rate of 79.17%. This provides a little evidence in favour of using the distance prior because its use reduces the posterior probability for MKK6(2) so that it no longer stands out.

5 Discussion

We have built upon the approach found in Mukherjee *et al.* [7] of using a noisy Boolean function probability model and Bayesian inference to infer combinatorial influences of components on an output of interest. We have constructed an informative pathway-based distance prior to use alongside the existing sparsity prior on models. The aim of this is to incorporate existing biological knowledge into the inference which could be valuable in inferring the correct model and rendering the inference more tractable if the sample size is small.

A benefit of the approach found in [7] is that it is able to model arbitrary Boolean functions and capture complicated combinatorial influences, unlike marginal statistics (for example, log odds ratio) which only consider a single component in isolation. We have seen this very clearly in our results. The MCMC method also allowed us to easily find posterior distribution over individual components, making the selection of a good model easier and giving us more confidence in our inferences.

We saw in our simulation results that our pathway-based distance prior has, on average, a positive effect on the inference of correct inputs to the underlying Boolean function. This was not only the case for a large sample size of 200, but also for a smaller sample size of 50. Notably, with $n = 50$, the sparsity prior alone struggles to make any improvement (on average) over the flat prior, but the addition of the distance prior does lead to some improvement. Small sample sizes should benefit more from the prior than larger ones; since there is less data, the inference relies on the prior more heavily. This is not a noticeable feature of our simulated data or proteomic data results. For example, the sparsity prior appears to have a bigger influence when applied to the data of larger sample size. This could be because our strength parameters λ_s and λ_d are too weak or are not set correctly in relation to each other. Plots of prior odds and Bayes factors could be used to help set these parameters. An alternatively is a more formal 'Empirical Bayes' approach in which the parameters are set automatically through the maximisation of a marginal likelihood.

The inference was effective with both simulation methods. One method emphasized noise in pathway signalling itself whilst the other emphasized noise in measurements. This suggests that the model and method is, to some extent, robust to noise from different sources being in the data.

We also did an analysis of a proteomic dataset from signalling in breast cancer. The sparsity prior was very useful here as it allowed two proteins to be inferred as inputs to the underlying Boolean function. We were also able to infer the Boolean function itself. The posterior distribution over models was very diffuse suggesting that we shouldn't have too much confidence in any one model. However, the posterior distribution over individual components gives us

some confidence that EIF2B and Akt1 are relevant as they appear in sampled models with moderate frequency and significantly more often than other components.

Using the distance prior on the proteomic data yielded few benefits over the sparsity prior. This may be due to the strength parameters mentioned above. Another possibility is that the pathway-based distance prior is, in fact, not appropriate in some settings; closeness of components within a pathway may not be a good indicator of their relevance or there may be a more natural partition of components than pathways. Further work is needed to better understand the role such priors can play in practical problems.

The above discussion suggests some areas for future work. The method of setting strength parameters for the model prior can be improved and an investigation of the sensitivity of our results to prior hyper-parameters could improve confidence in our inferences.

Also, our simulations use two different data generating methods, but it is not clear whether these produce realistic data. Extra work could be done to make the simulation method more realistic and then investigate whether this improves our inference results. For example, the majority of components within pathways had highly correlated values in our simulations. This may or may not be a characteristic of real data. If it is, treating the inference on a pathway level rather than a component level (i.e. inferring which pathways are relevant) could be beneficial.

Our model prior could be modified further. Putting a restriction on the number of pathways the model incorporates could be one possibility or using a notion of clustering instead of distance could prove fruitful.

Finally, our method could be applied to a bigger proteomic dataset. At time of writing, drug response values for all 32 cell lines have become available for analysis.

6 Acknowledgments

The author would like to thank Sach Mukherjee for his dedicated supervision of this work. This work has been funded by the EPSRC through the Warwick Complexity Science Doctoral Training Centre.

References

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

[2] W.P. Blackstock and M.P. Weir, Proteomics: quantitative and physical mapping of cellular proteins, *Trends Biotechnol* **17**, 121 (1999)

[3] H. Kitano, Systems Biology: A brief overview, *Science* **295**, 1662 (2002)

[4] N. Friedman, Inferring cellular networks using probabilistic graphical models, *Science* **303**, 799 (2004)

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

[6] L. Ruczinski, C. Kooperberg and M. LeBlanc, Logic Regression, *J Comput Graph Stat* **12**, 475 (2003)

[7] S. Mukherjee, S. Pelech, R.M Neve, P.T. Spellman, J.W. Gray and T.P. Speed, Sparse combinatorial inference with an application to cancer signalling, (preprint - to appear in *Bioinformatics*)

[8] A.W.F. Edwards, The measure of association in a 22 table, *J R Stat Soc Ser A-G* **126**, 109 (1963)

[9] C.M. Bishop, *Pattern recognition and machine learning* (Springer, New York, 2006)

[10] S. Mukherjee and T.P. Speed, Network inference using informative priors, (preprint - to appear in *P Natl Acad Sci USA*)

[11] Z. Wei and H. Li, Nonparametric pathway-based regression models for analysis of genomic data, *Biostatistics* **8**, 265 (2007)

[12] R.A. Weinberg, *The biology of cancer* (Garland Science, New York, 2007)

[13] C.P. Robert and G. Casella *Monte Carlo statistical methods* (Springer, New York, 2004)

[14] H. Jeffreys, *Theory of probability* (Oxford: Clarendon Press, 1961)

[15] R.M. Neve *et al.*, A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes, *Cancer Cell* **10**, 515 (2006)

[16] National Cancer Institute (U.S. National Institutes of Health), *NCI Drug Dictionary - Polyamine analogue PG11047* [online], available at http://www.cancer.gov/Templates/drugdictionary.aspx?CdrID=467739, [Accessed 15/09/2008]

[17] M.R. Boyd, K.D. Paull, and L.R.Rubinstein, In Cytotoxic Anticancer Drugs: Models and Concepts for Drug Discovery and Development, (Kluwer Academic: Hingham, MA, 1992)