# Application of Bayesian networks to two classification problems in bioinformatics

Christopher Needham*[1], James Bradford*[2], Andrew Bulpitt[1] and David Westhead*[2]

[1] School of Computing, University of Leeds
[2] School of Biochemistry and Molecular Biology, University of Leeds

## 1 Introduction

The application of machine learning techniques to bioinformatics problems has become increasingly popular in recent years. Of particular interest are *probabilistic graphical models* since they provide a concise representation for inferring models from data. Current applications include the learning of gene regulatory networks (Friedman, 2004) and protein function prediction.

Bayesian networks are the subset of probabilistic graphical models that can be expressed as directed acyclic graphs (Jordan, 1996; Jensen, 2001). They use a combination of domain knowledge and data to provide a framework that can be used to model the relationships between sets of variables in a probabilistic manner. They are a particularly powerful tool in bioinformatics since they can handle incomplete data sets allowing the building of a model from a training set with (different) missing data, or facilitating the prediction of a variable's state/value based on a restricted set of evidence. For example, an important variable may be unknown during testing, yet marginal probabilities can still be calculated. Causal relationships between variables can also be learnt giving us information on the contribution of each variable to the prediction, and any independencies between variables can be exploited to reduce the complexity of the model.

In this work, we use Bayesian networks for two classification tasks in bioinformatics. Our results show an improvement over previous machine learning methods applied to these problems. This improvement may be due to the Bayes nets' ability to capture the interactions between multiple variables. Preliminary results show that the framework of probabilistic graphical models provides a good basis for modelling bioinformatics data.

## 2 Predicting functional effects of single nucleotide polymorphisms

Proteins consist of amino acids which are in turn encoded in an organism's DNA. A single DNA base mutation, often called a single nucleotide polymorphism (SNP), can cause the exchange of one amino acid for another. Such a change can effect protein function, although the mutation may be neutral (no effect). In this work, the goal is to predict the consequences of an SNP to protein function, i.e. "effect" or "no effect", using lac repressor and lysozyme data.

In previous machine learning approaches to the problem Chasman and Adams (2001) proposed a probabilistic method, and Krishnan and Westhead (2003) evaluated decision trees and support vector machines. Verzilli *et al.* (2005) applied a hierarchical Bayesian multivariate adaptive regression spline (hierarchical BMARS) model for binary classification of the functional consequences of SNPs. Within the BMARS model, samples from the posterior distribution can be used to highlight the factors which are most frequent and indicative of being

important predictors.

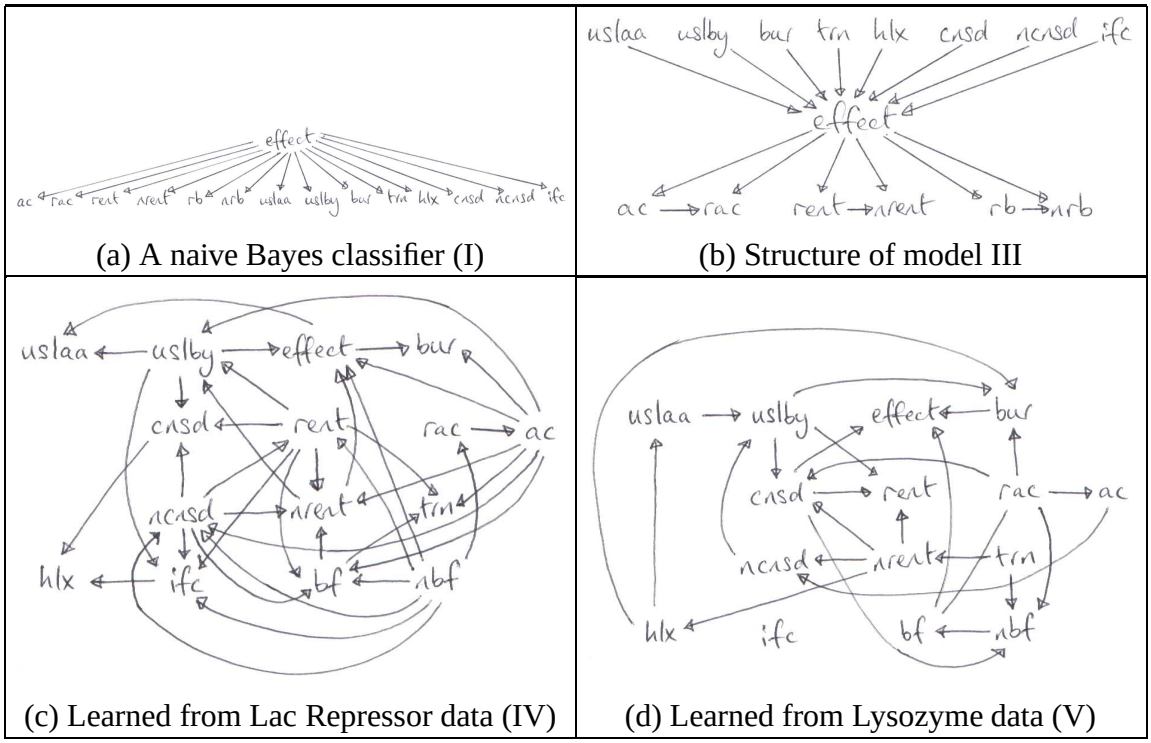
In this work, we learn Bayesian networks to predict the consequences of an SNP to protein function, and evaluate the methods using data from Verzilli*et al.* (2005) of two proteins: the Lac Repressor and Lysoyzme. In addition to the 'class' attribute (whether or not the mutation effects protein function), we use a set of fourteen variables (Table 1) of which six are continuous (such as accessibility and B-factor of the native amino acid mutated in the experiment), and eight are discrete binary (such as whether or not the mutant amino acid is charged at the buried site, or whether or not the native amino acid is at a conserved position in the phylogenetic profile). There are many challenges buried in these data. The continuous data is non-Gaussian, making it unsuitable for modelling as a continuous Gaussian node in a Bayes net. There are also no obvious boundaries at which to split the data into discrete categories. Our solution (D-fit) has been to fit a number of Gaussians to the data using an Expectation-Maximisation based algorithm (which automatically chooses the number of classes). This has proved useful in forming discrete classes from the continuous data giving better performance than simply splitting the data into three classes of equal range (D-eq).

| attribute | attribute description |
|---|---|
| ac | solvent accessible area of native AA |
| nrent | phylogenetic entropy of structural neighbourhood of native AA |
| bf | B-factor of native AA |
| nbf | B-factor of structural neighbourhood of native AA |
| uslaa | mutant AA is not in phylogenetic profile |
| uslby | mutant AA is not in the smallest AA class that includes the phylogenetic profile |
| bur | mutant AA is charged AA at buried site |
| trn | mutant AA occurs at glycine or proline in a turn |
| hlx | mutant AA occurs in helical region and involves glycine or proline |
| cnsd | native AA is at conserved position in phylogenetic profile |
| ncnsd | native AA is near conserved position in phylogenetic profile |
| ifc | native AA is near subunit interface |
| effect | effect of mutation on functionality |

Table 1: SNP Attributes

Experiments were performed using the Bayes Net Toolbox for MATLAB (BNT) (Murphy, 2001). Numerous structures were evaluated and results of heterogeneous cross validation are shown in Table 2. As expected, error rates observed in homogeneous tests were a little lower (results not shown). Structures used are illustrated in Figure 1. The structures (IV) and (V) were learned from the Lac Repressor and Lysozyme data respectively using a structure learning package (SLP) (Leray and Francois, 2004). Using discretised data (D-fit introduced above), a directed acyclic graph structure was learned using a greedy search algorithm.

These results show that Bayesian networks can reduce the error rates in predicting functional effects of SNPs over decision trees, SVMs and other probabilistic approaches, which perform the task with over 30% error rates. The use of structure learning algorithms can improve the performance further, with good results shown where structure is learned on data from one protein and parameters of the model for this structure are learned from another.




(a) A naive Bayes classifier (I)

(b) Structure of model III

(c) Learned from Lac Repressor data (IV)

(d) Learned from Lysozyme data (V)

Figure 1: Network Structures. Key to node labels is shown in Table 1.

| Model | Trained on Lac repressor Tested on Lysozyme | | | Trained on Lysozyme Tested on Lac repressor | | |
|---|---|---|---|---|---|---|
| | C+D | D-eq | D-fit | C+D | D-eq | D-fit |
| (I) naive Bayes | 27.6 | 29.5 | 23.5 | 22.7 | 21.4 | 20.1 |
| (II) inverted naive | - | 23.3 | 21.0 | - | 24.9 | 23.5 |
| (III) adapted | 26.0 | 23.3 | 21.6 | 22.4 | 21.8 | 20.9 |
| (IV) learned (Lac) | - | 22.1 | 17.6 | - | 23.4 | 21.5 |
| (V) learned (Lyso) | - | 17.7 | 17.8 | - | 23.8 | 23.2 |

Table 2: Classification error rates for five network structures. Evaluated using continuous and discrete nodes (C+D), discrete nodes formed by splitting into categories of equal range (D-eq), and data discretised fitting a number of Gaussians to the data (D-fit).

# 3 Protein-protein interactions

Structural genomics projects are beginning to produce protein structures with unknown function so accurate, automated predictors of protein function are required if all these structures are to be annotated in reasonable time. Identifying the interface between two interacting proteins provides important clues as to the function of a protein. Consequently we have developed an interface prediction method that involves combining a Bayesian Network approach with protein surface patch analysis.

First we train the Bayes net to distinguish between patches that form part of an interface (interacting patches) and those that are found outside the interface (non-interacting patches) using a training set of 180 interacting and 180 non-interacting patches. Our set of patch proper-

ties consists of 15 continuous variables (the means and standard deviations of hydrophobicity, residue interface propensity, electrostatic potential, solvent accessible surface area, sequence conservation, shape index and curvedness, and patch desolvation energy) and three discrete variables (interaction type (transient or obligate), patch secondary structure (helix, sheet, other, mixed), and the class variable (whether a patch is interacting or not)). In the prediction step, enough patches on the test protein are generated to ensure complete surface coverage and then the probability of each of these patches being an interacting patch is calculated by the Bayes net. The three patches with the highest probabilities are seen as most likely to be part of the interface. When the interface is known during cross validation, a prediction is deemed a success if any of the three patches contains at least 50% interface and actual interface coverage is over 20%.

In leave-one-out cross validation, we are able to predict the location of the interface on 78% of the 180 proteins in our training set using a naive Bayes net. This compares favourably with our earlier, similar method that uses a support vector machine to achieve 76% success (Bradford and Westhead, 2005). Prediction success was increased still further to 81% when expert knowledge was added to the network structure in the form of a number of connections between variables, such as hydrophobicity and residue interface propensity.

# References


Bradford J.R. and Westhead D.R. (2005). Improved prediction of protein-protein binding sites using a support vector machines approach. *Bioinformatics*, **21(8)**, 1487-1494.

Chasman D. and Adams R.M. (2001). Predicting the functional consequences of non-synonymous single nucleotide polymorphisms: Structure-based assessment of amino acid variation. *J. Mol. Biol.*, **307**, 683-706.

Friedman N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, **303**, 799-805.

Jensen F.V. (2001). *Bayesian Networks and Decision Graphs*. Springer.

Jordan M.I. (1996). *Learning in Graphical Models*. Kluwer Academic.

Krishnan V.G. and Westhead D.R. (2003). A comparative study of machine learning methods to predict the effects of single nucleotide polymorphisms on protein function. *Bioinformatics*, **19(17)**, 2199-2209.

Leray P. and Francois O. (2004). BNT structure learning package: Documentation and experiments. Technical report, Laboratoire PSI, Université et INSA de Rouen.

Murphy K.P. (2001). The Bayes Net toolbox for Matlab. *Computing Science and Statistics*, 331-350.

Verzilli C.J., Whittaker J.C., Stallard N. and Chasman D. (2005). A hierarchical Bayesian model for predicting the functional consequences of amino-acid polymorphisms. *Applied Statistics*, **54**, 191-206.