BIOINFORMATICS

Vol. 18 no. 6 2002
Pages 788–801

A Bayesian network model for protein fold and remote homologue recognition

A. Raval[1], Z. Ghahramani[2] and D. L. Wild[3,]*

[1]Keck Graduate Institute of Applied Life Sciences, 535 Watson Drive, Claremont, CA 91711, USA, [2]Gatsby Computational Neuroscience Unit, University College London, Queen's Square, London WC1N 3AR, UK and [3]Keck Graduate Institute of Applied Life Sciences, 535 Watson Drive, Claremont, CA 91711, USA

Received on December 1, 2000; revised on May 15, 2001; December 12, 2001; accepted on December 18, 2001

ABSTRACT

Motivation: The Bayesian network approach is a framework which combines graphical representation and probability theory, which includes, as a special case, hidden Markov models. Hidden Markov models trained on amino acid sequence or secondary structure data alone have been shown to have potential for addressing the problem of protein fold and superfamily classification.

Results: This paper describes a novel implementation of a Bayesian network which simultaneously learns amino acid sequence, secondary structure and residue accessibility for proteins of known three-dimensional structure. An awareness of the errors inherent in predicted secondary structure may be incorporated into the model by means of a confusion matrix. Training and validation data have been derived for a number of protein superfamilies from the Structural Classification of Proteins (SCOP) database. Cross validation results using posterior probability classification demonstrate that the Bayesian network performs better in classifying proteins of known structural superfamily than a hidden Markov model trained on amino acid sequences alone.

Contact: alpan_raval@kgi.edu; zoubin@gatsby.ucl.ac.uk; david_wild@kgi.edu

INTRODUCTION

The functional and structural annotation of the proteins coded by a newly sequenced genome is usually initially performed by pairwise sequence similarity searches against protein sequence databases, with the subsequent transfer of annotation, although a number of authors have drawn attention to the limitations of this approach (Richards *et al.*, 1995; Smith and Zhang, 1997; Brenner *et al.*, 1998). Since the sequences of remotely homologous proteins may have diverged beyond the point at which their similarity may be detected by pairwise sequence comparisons, typically some 30% or more of open reading frames (ORFs) may remain functionally uncharacterized after this type of analysis. Recent benchmarking experiments (Brenner *et al.*, 1998; Park *et al.*, 1998; Mueller *et al.*, 1999) using the Structural Classification of Proteins (SCOP) classification of protein structural domains (Murzin *et al.*, 1995) have shown that techniques based on multiple sequence profiles, such as PsiBlast (Altschul *et al.*, 1997) and hidden Markov models (HMMs) (Krogh *et al.*, 1994) provide more sensitive methods for detecting remote homologies than database search methods which rely only on pairwise sequence similarity. These benchmark studies also demonstrated that all sequence-based methods miss many important remote homologies between proteins with less than 20% sequence similarity. In addition, since three-dimensional (3D) structure is more highly conserved than primary sequence in evolution, and homologous proteins nearly always have similar 3D structures, fold recognition techniques, which attempt to utilize the additional information encoded by the 3D structure to identify the fold which an protein of unknown structure is most likely to adopt, also provide a sensitive method of detecting remote homologies (Fischer and Eisenberg, 1997; Rychlewski *et al.*, 1998, 1999; Jones, 1999). As structural genomics projects get underway, there will be a dramatic increase in the number of experimentally determined protein structures available, which will increase the coverage of fold recognition template libraries and the utility of these techniques for remote homologue detection (Burley *et al.*, 1999).

There have been two main approaches to fold assignment to date: threading using empirically derived pairwise pseudo-potentials (Sippl, 1990; Sippl and Weitckus, 1992; Godzik *et al.*, 1992; Jones *et al.*, 1992; Bryant and Lawrence, 1993) and profile alignment methods (Bowie *et al.*, 1991; Rost, 1995; Fischer and Eisenberg, 1996; Russell *et al.*, 1997). These methods are summarized below.

*To whom correspondence should be addressed.

© Oxford University Press 2002

In the first approach an amino acid sequence is 'threaded' onto the backbone coordinates of each protein in a library of protein folds, and the compatibility of the sequence with the structure is evaluated by a scoring function based on empirical pairwise interaction potentials that are derived using the statistics of known protein structures and the equilibrium Boltzmann distribution (Sippl, 1990). Potentials corresponding to short-, medium-, and long-range interactions have been utilized. In this approach, the score of the sequence-to-structure alignment is interpreted as a pseudo 'free energy' of the sequence in the conformation imposed by the structural template, and the assumption made is that the most probable sequence-to-structure alignment is the one with the lowest 'free energy'. However, as Bienkowska *et al.* (2000) have pointed out, if one assumes that the structural templates used for fold recognition represent an ensemble of similar structures or expected variants around a particular fold topology, then the most probable (or lowest 'free energy') sequence-to-structure alignment represents only one of the possible variants in the ensemble.

The second approach to fold recognition involves the use of a profile, or position-specific scoring matrix and gap penalties, derived from a multiple alignment of related sequences (Gribskov *et al.*, 1987). The similarity of any other sequence to the profile can be obtained through the use of dynamic programming to obtain an optimal (or most probable) sequence-to-profile alignment. The '3D profile' approach to fold recognition extends this concept by incorporating structural information into the profiles (Bowie *et al.*, 1991). Substitution matrices have been derived for different secondary structural environments, and may also include additional information such as the degree of solvent accessibility of an amino acid residue. The matching of both the primary sequence and the predicted secondary structure of an unknown sequence with the sequences and observed secondary structures of a fold library is then performed using a dynamic-programming algorithm and these structural profiles. However, the optimal sequence-to-structure alignment produced by dynamic programming techniques is rarely the correct one (Rost *et al.*, 1997; Levitt, 1997; Russell *et al.*, 1996). Bienkowska *et al.* (2000) have pointed out that the set of *suboptimal* alignments can be seen as a set of *optimal* alignments under expected statistical fluctuations in the scoring function, and can be interpreted as optimal alignments to structural variants of the same fold. White *et al.* (1994) and Lathrop *et al.* (1998a,b), using an approach based on Bayesian theory have proposed that summing the probabilities of all possible sequence-to-structure-model alignments should give a more rigorous approach to fold recognition than relying on the optimal or most probable sequence–structure alignment. The HMM approach to this problem uses the *forward algorithm* to calculate the likelihood (the probability of observing the sequence given the model; $P(\text{sequence} \mid \text{model})$). The posterior normalization of the sequence-model probabilities is given by Bayes' rule, which calculates the probability of observing a particular model given the sequence; $P(\text{model} \mid \text{sequence})$. Using an approach based on these ideas, Bienkowska *et al.* (2000) have shown that fold recognition accuracy may be increased by 40% compared to an approach based on optimal sequence–structure alignment probability.

Hidden Markov models have been successfully applied to protein fold recognition by using two other strategies. Homologous sequences of the unknown probe are first used to build a model which is then searched against sequences of a library of folds, or a model is built directly from a library of fold sequences and the probe sequence is scored against each model in turn. HMMs trained on either amino acid sequence or secondary structure data have been shown to have potential for addressing the problem of protein fold and superfamily recognition, where the goal is to classify a new protein sequence as belonging to a particular fold or superfamily (Karplus *et al.*, 1997; Di Francesco *et al.*, 1997). In the benchmarking experiments of Park *et al.* (1998), sequence-based HMMs were shown to be three times more effective than pairwise methods at detecting remote protein homologies. Further improvement in the sensitivity of remote homologue classification has been obtained by Jaakkola *et al.* (1999) who have used sequence-based HMMs to develop a discriminative model, based on a Fisher kernel function, estimated using both positive and negative training examples. The Fisher kernel method is a special case of the more general method of constructing a support vector machine (Cristianini and Shawe-Taylor, 2000) for the purpose of classification. The aim of a support vector machine is to find a hyperplane that separates training examples in one class (the positive or like class) from training examples in another (negative, or unlike class). This is done by mapping the training data into a higher-dimensional space and defining an appropriate separating hyperplane there. The map used by Jaakkola *et al.* (1999) was the Fisher score of a sample training sequence, derived from the parameters of a HMM trained on positive sequences examples.

We have previously proposed a method to combine features of the 3D profile alignment fold recognition method with a HMM formalism by developing Bayesian network models which incorporate both primary sequence and structural information (Wild and Ghahramani, 1998). The Bayesian network approach is a framework which combines graphical representation and probability (Pearl, 1988) and may be thought of as a generalization of HMMs (Smyth *et al.*, 1997; Ghahramani, 2001). Our model simultaneously learns amino acid sequence, secondary

structure and residue solvent accessibility for proteins of known 3D structure. A consideration of the errors inherent in the predicted secondary structure and residue accessibility for a query sequence may be incorporated into the model by means of a confusion matrix. We have used posterior probabilities (obtained from the likelihoods calculated by summing over all possible sequence-model alignments) for each of our structural models to identify the model which best classifies a particular sequence.

Several groups have recently presented methods which combine amino acid and secondary structure information in a HMM framework, with promising results. Thorne *et al.* (1996) have described the use of a HMM with three fully connected states and evolutionary trees for secondary structure prediction. Yu *et al.* (1998) describe a new class of probabilistic models (discrete state-space models) in which amino acid probability distributions associated with particular secondary structural states are replaced by the distributions of conserved sequence patterns. Hargbo and Elofsson (1999) describe modifications to the HMMER hidden Markov model package (Eddy, 1998) which combine the use of predicted secondary structure and pre-existing multiple sequence alignments (Sander and Schneider, 1991) and their application to the problem of protein fold (not superfamily) recognition. The work described here significantly extends this area of research by examining various Bayesian network structures for combining amino acid and secondary structure sources of information and the use of Fisher kernel discriminants derived from these Bayesian networks to evaluate classification performance. In the next section we provide tutorial material on the Bayesian network approach.

BAYESIAN NETWORKS

Bayesian networks are a graphical tool for representing conditional independencies between a set of random variables in a probabilistic model (Pearl, 1988; Cowell *et al.*, 1999; Jensen, 1996). A random variable A is *conditionally independent* from B given C if $P(A, B \mid C) = P(A \mid C)P(B \mid C)$, or equivalently $P(A \mid B, C) = P(A \mid C)$, where the notation $P(Y \mid X)$ denotes the probability of Y given X. For example, in a medical diagnosis model, the probability of observing a symptom (A) may be conditionally independent of the results of a test for the disease (B), if we are given (i.e. we know) whether the patient suffers from the disease (C). Using these conditional independecies, the joint probability of all the variables in the model can be factored into a product of conditional probabilities. For example,

$$P(A, B, C) = P(C)P(A \mid C)P(B \mid C).$$

In a Bayesian network, each variable in the model is represented by a node in a graph and the conditional independence relationships by directed arrows in the graph. We draw arrows from the variables on the right-hand side of the conditionals in the factorization to the variables on the left-hand side, calling the right-hand side variables the *parents* of the left-hand side variables (called the *children*). The Bayesian network for the above factorization is shown in Figure 1.

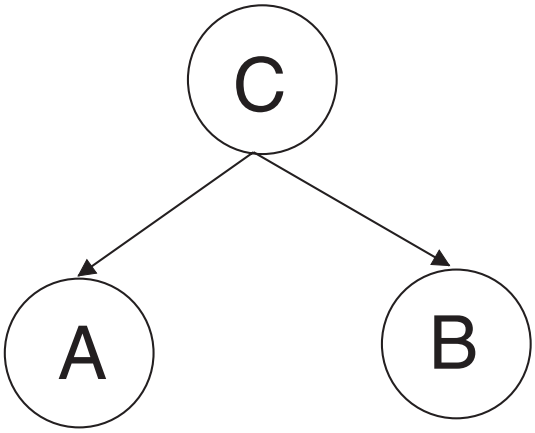

Fig. 1. A Bayesian network consistent with the conditional independence relations in $P(A, B, C) = P(C)P(B \mid C)P(A \mid C)$.

We now turn to HMMs, which are a special case of Bayesian networks. A HMM is a tool for representing probability distributions over sequences of observations. Let us denote the observation at location (or time) t in the sequence by the variable P_t. This can be, for example, the amino acid residue observed at location t in a protein sequence. The HMM gets its name from two defining properties. First, it assumes that the observation at location t was generated by some process whose state S_t is *hidden* from the observer. The state of the HMM is assumed to be a discrete variable: i.e. S_t can take on K values which we can denote by the integers $\{1, \ldots, K\}$. Second, it assumes that the state of this hidden process satisfies the *Markov property*: that is, given the value of S_{t-1}, the current state S_t and all future states are independent of all the states prior to $t - 1$. In other words, the state at some time or location encapsulates all we need to know about the history of the process in order to predict the future of the process. The outputs also satisfy a Markov property with respect to the states: given S_t, P_t is independent of the states and observations at all other time indices. A Bayesian network specifying the conditional independence relationships for a HMM is shown in Figure 2. For HMMs, the algorithm that infers the hidden states given the observations efficiently by exploiting conditional independence relations is called the *forward–backward* algorithm, which is a special case of both dynamic programming and the general propagation

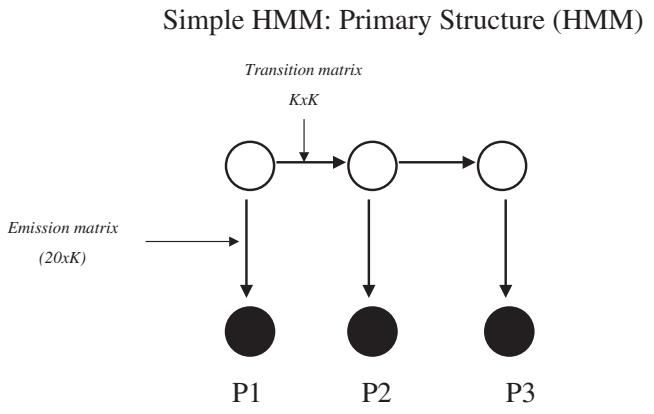

Fig. 2. A Bayesian network specifying conditional independence relations for a HMM showing the hidden states (open circles) and observations (filled circles) for the first three time steps of a sequence.

algorithms for Bayesian networks (Pearl, 1988; Lauritzen and Spiegelhalter, 1988; Smyth *et al.*, 1997).

Graphical diagrams can also be used to represent the transitions allowed in a HMM and, while this form of diagram is easy to confuse with the Bayesian network representation of HMMs, the two have very different semantics. If the variables S_{t-1} and S_t each have K possible discrete values (states), then $P(S_t \mid S_{t-1})$ is a $K \times K$ state transition matrix which we can draw graphically by showing K nodes connected by up to K^2 arrows. If a state is not allowed to transition to some other state, i.e. $P(S_t = k \mid S_{t-1} = m) = 0$, then the corresponding arrow from m to k is missing from such a diagram. For example, in biological sequence modeling using HMMs we usually have states denoted 'match state', 'insert state', etc, and a match state cannot transition to previous match states. Although the left-to-right form of such a diagram may appear similar to a Bayesian network, it is a depiction of the non-zero elements of a transition matrix and does not carry any conditional independence interpretation.

METHODS

In this paper we evaluate the performance of various Bayesian network models which incorporate both sequence, secondary structure and residue accessibility information in comparison to sequence-based HMMs, when modeling diverse superfamilies containing a number of remote homologues. Our goal is to evaluate performance for two scenarios; one in which the actual structure of a protein is known (for instance, as would be the case with a structural genomics experiment) and one in which only the amino acid sequence of a protein is known (for instance, a newly sequenced gene product) and where secondary structure and residue accessibility information must be obtained by prediction. We have compared five models: a hidden Markov model trained and tested on amino acid sequence alone (HMM) a Bayesian network model which simultaneously learns amino acid, secondary structure and residue accessibility symbols, trained and tested using known primary sequence and structural data for proteins of known three-dimensional structure (BN1); a similar model (BN1-JNET) which used predicted secondary structure and residue accessibility in the test data set; a Bayesian network model which incorporates additional elements relating actual secondary structure and accessibility symbols to values obtained by prediction from amino acid sequence alone (BN2), and a model identical to BN1 except trained and tested on known primary but predicted secondary structure and accessibility symbols (BN1-PRED, described in the Discussion section). The differences between these models are summarized in Table 1. We have compared the performance of these models using both posterior probability scores and Fisher kernel discrimination. We describe these models and the scoring methods in more detail below.

IMPLEMENTATION

Training and Test Data

Comparing benchmarks for remote homology detection is difficult, given the constantly evolving nature of sequence and structural databases. One of the most rigorous studies to date has been performed by Chothia and co-workers (Brenner *et al.*, 1998; Park *et al.*, 1998) who developed a structural benchmark for sequence homology methods, based on recognizing superfamily relationships in the Structural Classification of Proteins (SCOP) database (Murzin *et al.*, 1995). SCOP classifies protein domains of similar 3D structure and demonstrated homology (i.e. evolutionarily related proteins) into the same superfamily. Domains which share the same structure but lack evidence for a divergent evolutionary origin are assigned to the same fold. An alternative benchmark for remote homology detection, more suitable for machine learning methods, was used by Jaakkola *et al.* (1999), who describe an alternative way of partitioning training and test data to answer the question 'could the method discover a new (unknown) family of a known superfamily?' In these experiments, the problem of remote homology detection was simulated by withholding all members of one SCOP family from the training set, and then training with the remaining members of the SCOP superfamily. The withheld family (which are known remote homologs of the superfamily sequences) was then used as test data.

To construct our training and test sets we chose protein domains from the SCOP database, version 1.53. Rather

Table 1. Definitions of the various types of Bayesian network models used. The fields in the second row stand for: P = primary structure, SS = secondary structure, RA = residue accessibility. The various model types: HMM, BN1 etc. are then defined by whether the primary structures, secondary structures and residue accessibilities are known or predicted and whether or not a confusion matrix was used. A—in an entry indicates that the corresponding information was not used in that model type

Model type	Training data			Test data		
	P	SS	RA	P	SS	RA
HMM	Known	–	–	Known	–	–
BN1	Known	Known	Known	Known	Known	Known
BN1-JNET	Known	Known	Known	Known	Predicted	Predicted
BN2	Known	Known	Known	Known	Predicted (with plus confusion matrix)	Predicted (with plus confusion matrix)
BN1-PRED	Known	Predicted	Predicted	Known	Predicted	Predicted

than splitting the domains in each superfamily randomly into training and test sets, following Jaakkola *et al.* (1999), we withheld one entire family as a test set and trained on the domains present in the other families (within the same superfamily). The reason for splitting the data in this manner is that it guarantees a low sequence similarity between training and test sets (which a random split does not guarantee—see http://public.kgi.edu/∼wild/BN/table3.htm for sequence identity statistics). To this end, we only chose superfamilies that contain at least three families (so that at least two families could be chosen as training families and one as test). Superfamilies with just one family would not be suitable for this type of cross-validation experiment and superfamilies with two families would give trained models that are likely to be overfitted to the one family on which they are trained. We also filtered the superfamilies chosen so that there were a sufficiently large number of training sequences (greater than or equal to 100 before further filtering). Since the methods we use incorporate secondary strucuture and residue accessibility information (both calculated and predicted), we further filtered all families for domains for which secondary structure and residue accessibility could be calculated as well as predicted. The resulting 25 superfamilies used, along with their SCOP identifiers and descriptions, are given in Table 2. Statistics for the 25 superfamilies, including families withheld as test families, numbers of training and test sequences, and percentage sequence identity between training and test sets, are given at http://public.kgi.edu/∼wild/BN/table3.htm. The percentage sequence identities were calculated using the global Needleman–Wunsch alignment algorithm (GCG program GAP with end gaps penalized).

For each protein domain, secondary structure and relative residue solvent accessibility were calculated from the 3D coordinate files using the DSSP algorithm (Kabsch and Sander, 1983) and converted to a three letter alphabet ({Helix, Sheet, Coil} for secondary structure and a two letter alphabet {Buried, Exposed} for accessibility) according to Rost and Sander (1994). In the work of Hargbo and Elofsson (1999), the secondary structure for each training sequence is inherited from the HSSP database (Sander and Schneider, 1991) and is assumed to be the same for all proteins in a family. However, an analysis of a set of 68 structurally aligned proteins of between 8–30% identity by Zhang *et al.* (1997) showed that the secondary structure, as measured by a three-state model calculated by DSSP, is only 68% conserved on average between pairs of structurally aligned proteins, with a similar figure for residue burial. The assumption made by Hargbo and Elofsson, of a similar secondary structure for every member of a superfamily is not, therefore, justified, which is why we have used the actual (DSSP calculated) secondary structures and residue accessibilities in our model training.

MODEL ARCHITECTURES

An initial model, trained on amino acid sequence alone, was built according to the basic HMM architecture shown as a Bayesian network in Figure 2 (HMM). This model was then extended to simultaneously learn amino acid, secondary structure and accessibility symbols as shown in Figure 3 (BN1). BN1 can be thought of as a regular HMM with vector valued observations, and the figure illustrates the conditional independence relations between different variables (i.e. the Bayesian network). Since the model assumes that the amino acid, secondary structure and accessibility observations are independent, given the hidden state, the total emission probability is obtained by multiplying the emission probabilities for the three different alphabets. This model is similar to that described

Table 2. SCOP descriptions of the 25 superfamilies used in the current experiment. The second column gives the superfamily identifier in SCOP version 1.53 (Class.Fold.Superfamily) and the third column gives the full description (Class/Fold/Superfamily)

No.	SCOP 1.53 ID	**Description** (class/fold/superfamily)
1	1.3.1	All alpha proteins/cytochrome c/cytochrome c
2	1.4.5	All alpha proteins/DNA/RNA binding 3-helical bundle/winged helix DNA-binding domain
3	1.36.1	All alpha proteins/lambda repressor-like DNA binding domains/lambda repressor-like DNA binding domains
4	1.41.1	All alpha proteins/EF hand-like/EF hand
5	2.1.1	All beta proteins/immunoglobulin-like beta sandwich/immunoglobulins
6	2.5.1	All beta proteins/cupredoxins/cupredoxins
7	2.28.1	All beta proteins/concanavalin A-like lectins/glucanases/concanavalin A-like lectins/glucanases
8	2.38.4	All beta proteins/OB-fold/nucleic acid-binding proteins
9	2.44.1	All beta proteins/trypsin-like serine proteases/trypsin-like serine proteases
10	2.47.1	All beta proteins/acid proteases/acid proteases
11	2.56.1	All beta proteins/lipocalins/lipocalins
12	3.1.8	Alpha and beta proteins (a/b)/TIM beta/alpha-barrel/(Trans)glycosidases
13	3.2.1	Alpha and beta proteins (a/b)/NAD(P)-binding Rossmann-fold domains/NAD(P)-binding Rossmann-fold domains
14	3.3.1	Alpha and beta proteins (a/b)/FAD/NAD(P)-binding domains/ FAD/NAD(P)-binding domains
15	3.32.1	Alpha and beta proteins (a/b)/P-loop containing nucleotide triphosphate hydrolysases/P-loop containing nucleotide triphosphate hydrolysases
16	3.42.1	Alpha and beta proteins (a/b)/Thioredoxin fold/Thioredoxin-like
17	3.50.1	Alpha and beta proteins (a/b)/Ribonuclease H-like motif/Actin-like ATPase domain
18	3.50.3	Alpha and beta proteins (a/b)/Ribonuclease H-like motif/Ribonuclease H-like
19	3.62.1	Alpha and beta proteins (a/b)/PLP-dependent transferases/PLP-dependent transferases
20	3.65.1	Alpha and beta proteins (a/b)/alpha/beta-Hydrolases/alpha/beta-Hydrolases
21	4.2.1	Alpha and beta proteins (a+b)/Lysozyme-like/Lysozyme-like
22	4.3.1	Alpha and beta proteins (a+b)/Cysteine proteinases/Cysteine proteinases
23	4.70.1	Alpha and beta proteins (a+b)/Glyceraldehyde-3-phosphate dehydrogenase-like,C-terminal domain/ Glyceraldehyde-3-phosphate dehydrogenase-like,C-terminal domain
24	4.81.1	Alpha and beta proteins (a+b)/Zincin-like/Metalloproteases ('zincins'), catalytic domain
25	4.154.1	Alpha and beta proteins (a+b)/C-type lectin-like/C-type lectin-like

by Hargbo and Elofsson (1999), with the exception that our model also incorporates residue solvent accessibility. In addition, there are important differences in the training and scoring methods, described below. To better model training sets with few sequences, and avoid overfitting, the model was regularized using pseudo-counts for the transition probabilities and emission matrices, according to Laplace's rule (Durbin *et al.*, 1998).

Model lengths (T) were chosen both by comparing validation set likelihoods and by computing modes of sequence length histograms. The number of hidden states in the models shown in Figures 2, 3, and 4 is then K, where $K = 3T$, with the factor of three coming from the fact that there is an insert state and delete state for every match state. Emission probabilities for both match and insert states are learned directly from the training data. To compensate for possible sequence redundancy in the training set, models were trained using a weighted expectation–maximization (EM) algorithm, with similar sequences down-weighted. Training set sequences were multiply aligned using ClustalW (Thompson *et al.*, 1994) and relative weights for each sequence calculated according to the degree of sequence similarity, using the weighting scheme of Vingron and Argos (1989). The model parameters were fitted by the EM algorithm and not obtained via sequence or structure alignment.

BN2 (Figure 4) represents the extension of BN1 to the case where the structure of the probe sequence is unknown and the secondary structure and residue accessibility must be predicted. In this model, the actual secondary structure and residue accessibility are hidden state variables, with the predicted secondary structure and accessibility representing 'pseudo-observations'. Since these quantities can only be predicted with an accuracy of around 70% (Zemla *et al.*, 1997), it is advantageous to incorporate an awareness of the errors inherent in predicted secondary structure and residue accessibility into the model by means of a confusion (or misclassification) matrix, as shown in Figure 4. The experiments described here used the JNET software (Cuff and Barton, 2000), which is reported to give 76.4% average accuracy on a large test set of proteins, for secondary structure and accessibility predictions. In any multi-class prediction problem with K classes, one obtains a $K \times K$ confusion matrix $Z = z_{ij}$ where z_{ij} represents the number of times the observed value is predicted to be in class j whilst belonging to class i (Baldi *et al.* (2000)). For secondary structure prediction with three classes; *helix,*

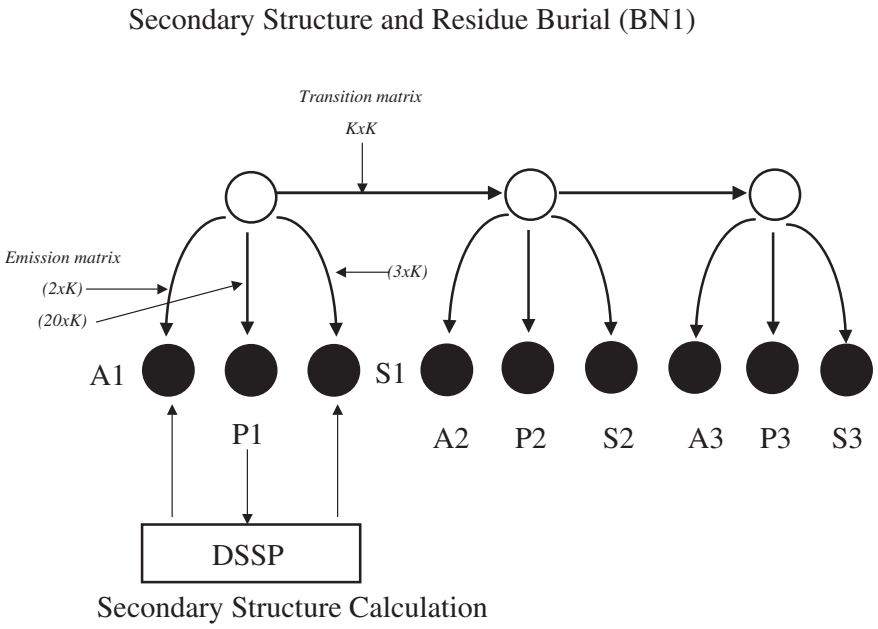

Fig. 3. BN1, incorporating secondary structure and residue burial (solvent accessibility).

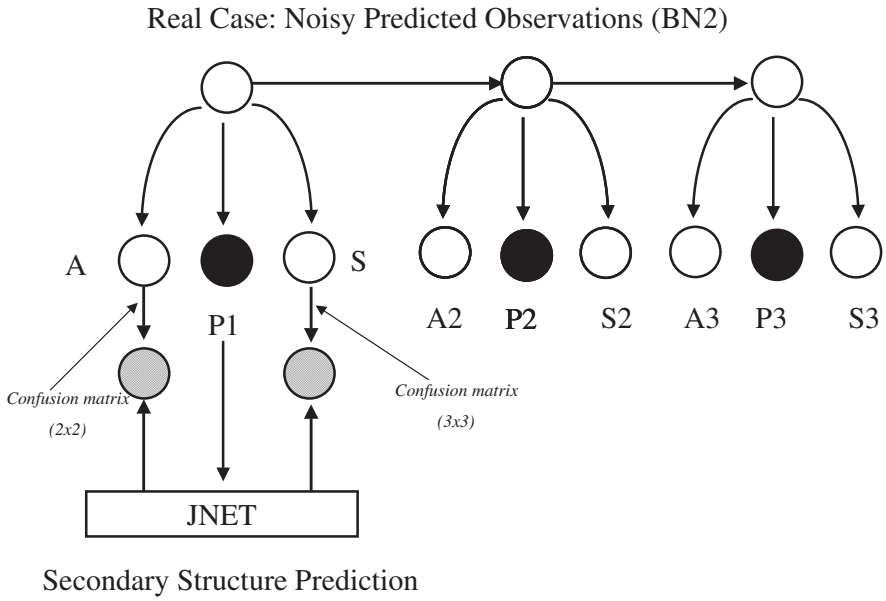

Fig. 4. BN2, incorporating predicted 'pseudo-observations' (gray circles) for secondary structure and residue burial.

sheet and *coil*, we obtain a 3 × 3 confusion matrix, whilst for residue solvent accessibility prediction with two states; *buried* and *exposed*, we obtain a 2 × 2 confusion matrix. The confusion matrix for secondary structure and residue accessibility predictions using JNET was derived for the non-redundant set of 350 high-quality protein structures used by Russell *et al.* (1997) and is shown in Table 3. This inherent uncertainty in secondary structure prediction

Table 3. Confusion (or misclassification) matrix for secondary structure (left) and residue accessibility (right) predictions from JNET, based on a non-redundant set of 350 protein structures. These matrices show the frequency with which a residue in a particular state is predicted to be in a helix, sheet or coil, etc

Predicted/ Observed	Helix	Sheet	Coil	Predicted/ observed	Buried	Exposed
Helix	0.753	0.036	0.211	**Buried**	0.825	0.175
Sheet	0.023	0.838	0.138	**Exposed**	0.329	0.671
Coil	0.079	0.096	0.826			

was not considered in the models of Hargbo and Elofsson (1999) which used predicted secondary structures.

MODEL SCORING USING POSTERIOR PROBABILITIES AND FISHER KERNELS

Log-likelihood scores for the training and test sets were calculated (using the forward algorithm) for each superfamily model and for a default (baseline) model. The baseline model assumes that amino acid, secondary structure and residue accessibility symbols are independent at each position, and assigns fixed emission probabilities based on Dayhoff background frequencies for amino acids (Dayhoff *et al.*, 1978), and the 'stationary probabilities' for secondary structure and residue accessibility derived by Thorne *et al.* (1996). The baseline model is used to classify proteins which cannot be classified by any of the superfamily models, and so represents a 'don't know' classification. Posterior probabilities [Prob(Model | Data)] for each of the structural models plus the baseline model were calculated from Bayes' rule and used to perform Bayesian classification. All proteins in each superfamily were classified according to these posterior probabilities and the results summarized in the form of a *confusion* or *classification table*, similar to that described above for secondary structure prediction. The criterion for selecting the model which best classifies a particular protein is to choose the model with the highest posterior probability for a given pattern of evidence. If N models have been trained, the model selected as the best classification for the protein (X) would be model M_i such that $P(M_i \mid X) > P(M_j \mid X)$ for all $j \neq i$. For each model architecture (HMM, BN1, BN1-JNET and BN2) posterior probabilities were calculated and a Bayesian classification table constructed.

The recently developed Fisher kernel method has been used to obtain more sensitive protein superfamily classification using sequence based HMMs (Jaakkola *et al.*, 1999). This is a variant of support vector machine methods (Boser *et al.*, 1992), using a kernel function derived from a HMM. During the training of a generative model, such as a HMM, parameters are estimated so that positive training examples (proteins belonging to the superfamily being modeled) are likely to be observed under the probability model. In contrast, the parameters of a discriminative model would be estimated using both positive and negative training examples. Jaakkola *et al.* (1999) describe the particular class of discriminative techniques known as Fisher kernel methods. Following their approach, we model the discriminant function $L(X)$, the sign of which determines the classification of a protein X into one of two hypothesis classes (superfamily member/non-member), as

$$\begin{aligned} L(X) &= \log P(H_1 \mid X) - \log P(H_0 \mid X) \\ &= \sum_{i:X_i \in H_1} \lambda_i K(X, X_i) \; - \sum_{i:X_i \in H_0} \lambda_i K(X, X_i). \end{aligned}$$

Here H_1 and H_0 are the positive and negative hypothesis classes, respectively, and the kernel function we use is

$$K(X, Y) \quad = \quad \phi(X) \cdot \phi(Y)$$

where $\phi(X)$ is the Fisher score vector for protein X, given by the derivatives of the log-likelihoods with respect to the parameters of the model. The coefficients λ are computed according to a gradient ascent algorithm that maximizes an objective function of the coefficients λ, as described by Jaakkola *et al.* (1999).

We compute Fisher scores from a 'mixture model', combining all the models for the different superfamilies. In order to compute the Fisher score vectors from the generative models, we start with N training datasets, $D_1, \ldots, D_N$ in N classes, and consider the N generative models $M_1, \ldots, M_N$, plus one baseline model M_0. $P(D \mid M_i)$ is the likelihood of the data given model i. Prior probabilities $\pi_0, \ldots, \pi_N$ are estimated from the frequency of occurrence of protein superfamilies in the SwissProt sequence database, which we consider to be more representative of the distribution of protein sequences than the limited number of sequences of known structure represented in the SCOP database. We then form a global mixture model M^* such that

$$P(X \mid M^*) = \sum_{i=0}^{N} \pi_i P(X \mid \theta_i),$$

θ_i being the parameters of the generative model i. The Fisher score vector for a particular protein X is obtained by evaluating the derivative of the log-likelihood with respect to a vector of parameters. We take these parameters to be the prior probabilities π_k (the mixing proportions of the mixture model). The appropriate derivative is

$$\begin{aligned} \frac{\partial \log P(X \mid M^*)}{\partial \pi_k} &= \frac{\partial}{\partial \pi_k} \log \sum_i \pi_i P(X \mid \theta_i) \\ &= \frac{r_k}{\pi_k} - 1 \end{aligned}$$

where the -1 comes from the sum to 1 constraint on the estimated priors π_k, and the r_k are the previously calculated posterior probabilities of X, i.e. $r_k = P(M_k \mid X)$. Similarly, the Fisher score vector with respect to the model parameters θ_i (emission and transition probabilities) would be

$$\begin{aligned}\frac{\partial \log P(X \mid M^*)}{\partial \theta_i} &= \frac{\pi_i}{P(X \mid M^*)} \frac{\partial P(X \mid \theta_i)}{\partial \theta_i} \\ &= \frac{\pi_i P(X \mid \theta_i)}{P(X \mid M^*)} \frac{\partial \log P(X \mid \theta_i)}{\partial \theta_i} \\ &= r_i \frac{\partial \log P(X \mid \theta_i)}{\partial \theta_i}\end{aligned}$$

From these Fisher score vectors we compute $L_i(X)$ for the protein X to be classified. This would be sufficient for pairwise discrimination (two-class discrimination). However, in order to construct a multi-class classifier we add bias terms a_i that are model dependent but independent of protein. These bias terms are trained to maximize the log-likelihood of the training data D being correctly classified, i.e., we maximize the function

$$C(D) = \sum_{i=1}^{N} \sum_{X_j \in D_i} \log P(i \mid X_j)$$

where

$$P(i \mid X_j) = \exp(L_i(X_j) + a_i) / \left\{ \sum_k \exp(L_k(X_j) + a_k) \right\}$$

is the probability of classifying protein X_j into class i. $C(D)$ is maximized by simple gradient ascent with respect to a_i. The test data are then classified in the model in which $L_i + a_i$ is maximum. If all $L_i(X) + a_i < 0$ then the protein X would be classified as not in any of the N modeled classes (baseline classification).

RESULTS

For all the methods described above (HMM, BN1, BN1-JNET, BN2) a Bayesian (posterior) classification table was constructed for the whole data set, i.e. the number of sequences classified as belonging to each superfamily modeled. This is the usual way of presenting results for K-way multiclass classification. However, for clarity, we present a simplified table showing the total number of correct classifications and misclassifications made by each method for each superfamily (Table 4). The individual detailed classification tables for each method are available at http://public.kgi.edu/~wild/BN/index.htm. The overall numbers of misclassifications made by each method are presented in Figure 5.

In 22 out of the 25 superfamilies studied in this test, the structural Bayesian network model (BN1) trained and tested on real structural data performed as well as or better than the HMM trained only on amino acid information. In particular, the structural BN1 model misclassified far fewer proteins into the default (baseline) category than the sequence-based HMM (Table 4). Although these baseline misclassifications were not strongly correlated with amino acid composition or sequence length, we speculate that these misclassifications are due to the inability of the sequence-based HMM to model diverse superfamilies. Table 4 and Figure 5 also show the classification results obtained using secondary structures and residue accessibilities as predicted by JNET. BN1-JNET shows the performance of BN1 on predicted data when no confusion matrix was used, and this structural model performs as well as or better than the sequence-based HMM on 16 out of the 25 superfamilies studied. Incorporating confusion matrices for secondary structure and residue accessibility (which model inaccuracies in the JNET prediction algorithm) resulted in improved performance over both HMM and BN1-JNET (18 out of 25 superfamilies).

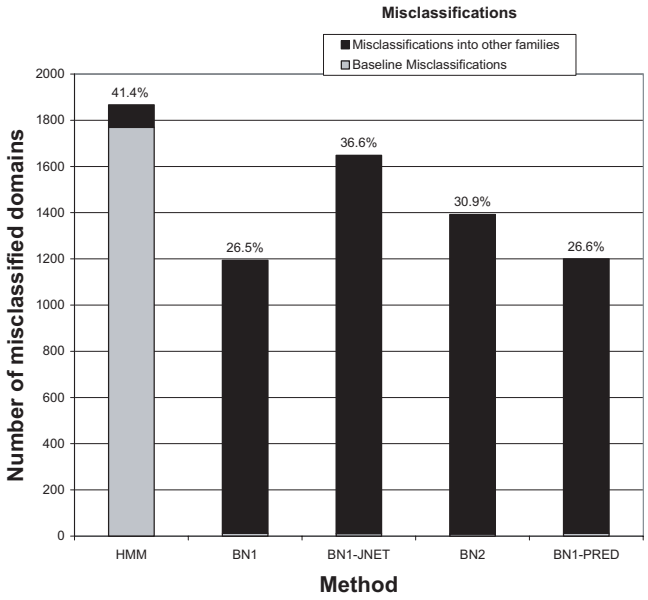

Fig. 5. Graphical representation of Table 4. The height of a bar represents the total number of misclassifications for that method (gray: baseline misclassifications, black: misclassifications into other superfamilies). The percentage number on top of each bar is the overall percentage of misclassified domains (as a fraction of the total number of test domains).

As an additional measure of performance, following Park *et al.* (1998) and Jaakkola *et al.* (1999), we use the rate of false positives (RFP) achieved for each model as a metric. The RFP for a protein is defined as the fraction of negative test proteins (i.e. proteins not belonging to the same superfamily) which score as high, or better, than

Table 4. Overall comparison of the classification performance of all methods. The first column is the superfamily number. For each method, column a gives the number of correct classifications (boldface), column b gives the number of baseline misclassifications, and column c gives the number of misclassifications into other superfamilies

No.	HMM			BN1			BN1-JNET			BN2			BN1-PRED		
	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c
1	**6**	17	0	**12**	0	11	**11**	0	12	**13**	0	10	**21**	0	2
2	**68**	7	0	**75**	0	0	**71**	0	4	**75**	0	0	**75**	0	0
3	**1**	4	0	**2**	0	3	**1**	0	4	**1**	0	4	**1**	0	4
4	**164**	12	0	**176**	0	0	**176**	0	0	**176**	0	0	**176**	0	0
5	**481**	809	19	**875**	0	434	**542**	0	767	**759**	0	550	**925**	0	384
6	**77**	154	0	**91**	0	140	**85**	0	146	**96**	0	135	**95**	0	136
7	**132**	111	8	**166**	0	85	**248**	0	3	**129**	0	122	**119**	0	132
8	**30**	21	1	**45**	0	7	**31**	0	21	**39**	0	13	**41**	0	11
9	**233**	48	10	**236**	0	55	**237**	0	54	**236**	0	55	**235**	0	56
10	**68**	228	0	**68**	0	228	**68**	2	226	**68**	0	228	**68**	0	228
11	**48**	45	0	**48**	0	45	**48**	0	45	**48**	0	45	**48**	0	45
12	**123**	6	18	**109**	0	38	**102**	0	45	**91**	0	56	**106**	0	41
13	**201**	123	0	**324**	0	0	**226**	0	98	**310**	0	14	**322**	0	2
14	**24**	0	1	**23**	0	2	**4**	0	21	**16**	0	9	**24**	0	1
15	**205**	51	20	**265**	0	11	**241**	0	35	**265**	0	11	**262**	0	14
16	**19**	40	0	**27**	0	32	**18**	0	41	**22**	0	37	**22**	0	37
17	**15**	35	2	**20**	0	32	**16**	0	36	**20**	0	32	**17**	0	35
18	**28**	23	0	**28**	0	23	**23**	0	28	**27**	0	24	**28**	0	23
19	**22**	10	8	**32**	0	8	**30**	0	10	**32**	0	8	**32**	0	8
20	**85**	5	1	**79**	12	0	**82**	9	0	**82**	9	0	**79**	12	0
21	**548**	3	0	**548**	0	3	**544**	0	7	**548**	0	3	**548**	0	3
22	**1**	0	10	**1**	0	10	**0**	0	11	**0**	0	11	**0**	0	11
23	**8**	8	0	**8**	0	8	**2**	0	14	**8**	0	8	**8**	0	8
24	**48**	1	0	**48**	0	1	**47**	0	2	**47**	0	2	**48**	0	1
25	**4**	8	0	**6**	0	6	**5**	0	7	**6**	0	6	**6**	0	6

the protein (in terms of posterior probability). In other words, the RFP for each true positive protein is the fraction of incorrect proteins that are found before the correct protein when the proteins are ranked in order of their score (posterior probability). The maximum RFP for a model is the maximum over RFPs of all proteins in the model's superfamily. This measure is zero when there is perfect test set recognition. Since this measure may be dominated by a few outliers, which are hard to recognize, Jaakkola *et al.* (1999) also utilize the median RFP. However, for all the tests conducted here, this quantity was zero for all models, and so is not quoted.

In a two-class classification problem the numbers of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) often depend on how the threshold of the scoring function is selected. Generally there is a trade off between the number of FPs and FNs, and these are usually summarized by a receiver operating characteristics curve (ROC), which displays the *sensitivity* (defined as TP/(TP+FN)) versus the *false positive rate* (FP/(FP+TN)). Alternatively, the sensitivity is often plotted against the *specificity* (TP/(TP+FP)) (Baldi *et al.* (2000)). Following Jaakkola *et al.* (1999), we use an alternative form of the ROC curve more suitable for our multi-class classification problem. The x-axis gives possible values for the maximum rate of false positives (maximum RFP). The y-axis gives, for each method, the number of families with that maximum rate or lower. The higher a curve is on this graph, the better the overall performance of the method. ROC curves are presented in Figures 6 and 7, which give a graphical comparison of the overall performance of the various methods on the 25 test superfamilies. The scoring function used for classification and for the calculation of RFPs is the posterior probability of the model given the data. In the case of support vector machines, the scoring function is the discriminant function for each datum in each model. We also compare all classification methods to the classification performance of BLAST2 (PSI-BLAST version 2.0.10 with one iteration, Altschul *et al.*, 1997). In the BLAST2 case, the RFP for each sequence is computed by searching that sequence against all other sequences in the 25 superfamilies considered. The number of false positives resulting from the search are divided by the total number of unlike sequences (sequences that do not belong to the superfamily of the original sequence) to obtain the RFP for that sequence. Once again, these ROC curves indicate that better performance is obtained

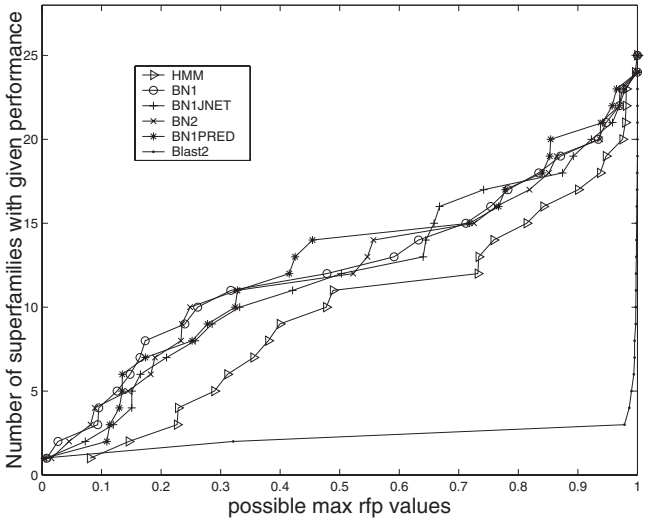

Fig. 6. Maximum rate of false positives (rfp) for each method used, and for Blast2. The rfp of a positive test sequence is defined as the fraction of negative test sequences that have score greater than or equal to the score of the positive test sequence. The maximum is taken over all test sequences in a superfamily. The y-axis gives the number of superfamilies that have less than or equal to the corresponding maximum rfp on the x-axis. The higher a curve is on this graph, the better its overall performance.

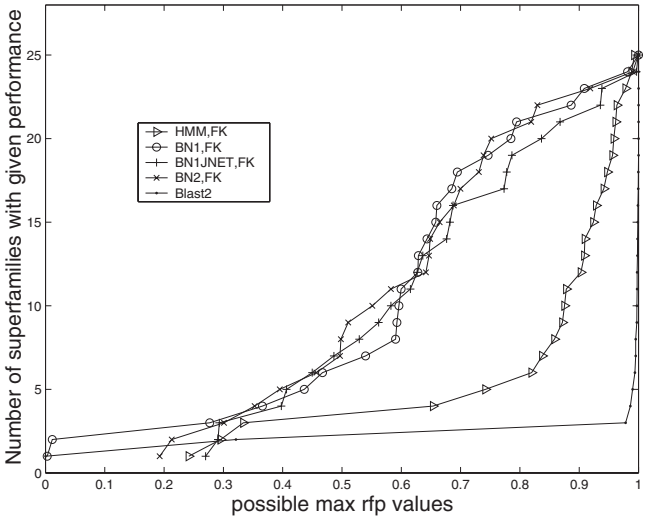

Fig. 7. Maximum rate of false positives (rfp) for each support vector method used, and for Blast2. The rfp of a positive test sequence is defined as the fraction of negative test sequences that have score greater than or equal to the score of the positive test sequence. The maximum is taken over all test sequences in a superfamily. The y-axis gives the number of superfamilies that have less than or equal to the corresponding maximum rfp on the x-axis. The higher a curve is on this graph, the better its overall performance.

by model BN2, which uses real primary, secondary and residue accessibility for *training,* but *predicted* secondary structure and accessibilities, *with their confusion matrices* for *testing*. All of the models considered dramatically outperform BLAST2 in this test of remote homolog recognition.

DISCUSSION AND FUTURE WORK

The cross validation experiments using Bayesian (posterior) classification demonstrate that the Bayesian network model which incorporates structural information outperforms a HMM trained on amino acid sequences alone, when tested with both real and predicted secondary structure and residue accessibilities. A improvement in classification performance when using *predicted* secondary structure and residue accessibilities was obtained by incorporating the confusion matrix for secondary structure and residue accessibility prediction into the Bayesian network model (BN2) (although with the caveat that most of the SCOP 1.53 sequences used in this test have probably been previously 'seen' during the training of the neural network used in the JNET secondary structure prediction method). We find that all methods significantly outperform BLAST2. One possible explanation for the superiority of network BN2 versus BN1-JNET when using known secondary structure and residue accessibility data for training, but predicted structural data for testing, is that model BN2 cannot be lured into putting too much faith in the structural data, because of the confusion matrices which model errors in secondary structure and residue accessibility prediction. For instance, in a region with highly conserved structural features, BN1-JNET will be trained to put high emphasis on these, and so will be vulnerable to mispredictions of these structural features. This suggests an alternative strategy for training our BN1 networks; to use *predicted* secondary structure and residue accessibility data for training, instead of known data. In this case, network BN1 might actually learn which structural features are hard to predict. To test this hypothesis, we re-trained all the BN1 models using JNET *predicted* secondary structure and accessibility data. We call this network model BN1-PRED. The classification results are given in Table 4 and Figure 5, and the ROC curve in Figure 6. These results demonstrate that model BN1-PRED has almost identical performance that obtained using known (ideal) structural data (model BN1). This exciting result suggests that our method is generally applicable to any protein sequence of unknown structure, and we propose to test this hypothesis in future work.

Scoring experiments using an K-way classifier based on a Fisher kernel derived from the Fisher score vector with respect to the posteriors of a mixture model resulted

in poorer overall performance for all models than when using maximum posterior classification, as measured by the maximum RFP (Figure 7). We note that the maximum RFP measure is sensitive to domination by a few outliers (the median RFP continues to be zero for all cases studied). In general, the problem of multiclass discrimination using support vector machines is an open one and largely driven by heuristic techniques, although recent advances (Crammer and Singer, 2002) may provide a useful approach to multiclass generalizations of support vector machines. We are investigating the use of combinations of Fisher score vectors for the posterior probabilities and amino acid and structural model parameters (emission matrices for the secondary structure and residue accessibility). Our present implementation of the Fisher kernel is based on the procedure outlined in Jaakkola *et al.* (1999), using a fixed bias and without adding terms to the diagonal elements of the Fisher kernel matrix that could compensate for imbalances between like and unlike training data (the soft-margin approach; see, for example, Cristianini and Shawe-Taylor (2000)). We believe that this is a possible reason for the poorer performance of the Fisher kernel method, as measured by the maximum RFP, and are currently in the process of implementing a soft-margin version of the Fisher kernel method.

Our Bayesian network models combine features of the 3D profile alignment approach to fold recognition, in that they incorporate both sequence, secondary structure and residue accessibility information, with a HMM formalism. This allows us to use the forward algorithm to sum the probabilities of all possible sequence-to-structure-model alignments as proposed by White *et al.* (1994) and Lathrop *et al.* (1998a,b) rather than relying on the optimal or most probable sequence–structure alignment as produced by the dynamic programming algorithms used in the 3D profile alignment approach. The Bayesian network structures we have explored in this paper can be seen as extensions of HMMs to incorporate multiple observations and confusion matrices. In particular, the parameters of these Bayesian networks can be estimated using a simple modification of the Baum–Welch algorithm for HMMs. One of the main limitations of our models is that the hidden states have a first-order Markov dependency which cannot model the longer range interactions which influence protein folding. Extensions of these Bayesian networks to incorporate multiple hidden variables and long-range interactions would require more sophisticated exact and approximate inference and learning algorithms. In future work we will examine various architectures for the incorporation of longer range interactions into our Bayesian network models, and the use of mean field potentials and a combination of Gibbs sampling and exact inference methods from the Bayesian network community to perform inference and parameter learning in these models.

REFERENCES

Altschul,S.F., Madden,T.L., Schaeffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Baldi,P., Brunak,S., Chauvin,Y., Anderson,C.A.F. and Nielsen,H. (2000) Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, **16**, 412–424.

Bienkowska,J., Yu,L., Zarakhovich,S., Rogers,Jr,R.G. and Smith,T.F. (2000) Protein fold recognition by total alignment probability. *Proteins*, **40**, 451–462.

Boser,B.L., Guyon,I. and Vapnik,V.N. (1992) A training algorithm for optimal margin classifiers. *Fifth Annual Workshop on Computational Learning Theory*. Morgan Kauffman, San Mateo, CA, pp. 144–152.

Bowie,J.U., Luthy,R. and Eisenberg,D. (1991) A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, **253**, 164–170.

Brenner,S.E., Chothia,C. and Hubbard,T.J.P. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl Acad. Sci. USA*, **95**, 6073–6078.

Brenner,S.E., Koehl,P. and Levitt,M. (2000) The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Res.*, **28**, 254–256.

Bryant,S.H. and Lawrence,C.E. (1993) An empirical energy function for threading protein sequence through the folding motif. *Proteins*, **16**, 92–112.

Burley,S.K., Almo,S.C., Bonanno,J.B., Capel,M., Chance,M.R., Gaasterland,T., Lin,D., Sali,A., William,S.F. and Swaminathan,S. (1999) Structural genomics: beyond the Human Genome Project. *Nature Genet.*, **23**, 151–157.

Cowell,R.G., Dawid,A.P., Lauritzen,S.L. and Speigelhalter,D.J. (1999) *Probabilistic Networks and Expert Systems*. Springer, New York.

Crammer,K. and Singer,Y. (2002) On the learnability and design of output codes for multiclass problems. *Machine Learning*, **47**, 201–233.

Cristianini,N. and Shawe-Taylor,J. (2000) *Support Vector Machines and Other Kernel-Based Learning Methods.*. Cambridge University Press, Cambridge, UK.

Cuff,J.A. and Barton,G.J. (2000) Application of multiple sequence alignment profiles to improve secondary structure prediction. *Proteins*, **40**, 502–511.

Dayhoff,M.O., Schwartz,R.M. and Orcutt,B.C. (1978) A model of evolutionary change in proteins. In Dayhoff,M.O. (ed.), *Atlas of Protein Sequence and Structure*. National Biomedical research Foundation, Washington, DC, pp. 345–352.

Di Francesco,V., Geetha,V., Garnier,J. and Munson,P.J. (1997) Fold recognition using predicted secondary structure and hidden Markov models of protein folds. *Proteins*, Suppl. 1, 123–128.

Durbin,R., Eddy,S., Krogh,A. and Mitchison,G. (1998) *Biological Sequence Analysis*. Cambridge University Press, Cambridge, UK.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Fischer,D. and Eisenberg,D. (1996) Protein fold recognition using sequence-derived predictions. *Protein Sci.*, **5**, 947–955.

Fischer,D. and Eisenberg,D. (1997) Assigning folds to the proteins encoded by the genome of *Mycoplasma genitalium*. *J. Mol. Biol.*, **94**, 11929–11934.

Ghahramani,Z. (2001) An introduction to hidden Markov models and Bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, **15**, 9–14.

Godzik,A., Kolinski,A. and Skolnick,J. (1992) Are proteins ideal mixtures of amino acids? Analysis of energy parameter sets. *J. Mol. Biol.*, **227**, 227–238.

Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355–4358.

Hargbo,J. and Elofsson,A. (1999) Hidden Markov models that use predicted secondary structures for fold recognition. *Proteins*, **36**, 68–76.

Holm,L. and Sander,C. (1994) The FSSP database of structurally aligned protein fold families. *Nucleic Acids Res.*, **22**, 3600–3609.

Jaakkola,T., Diekhans,M. and Haussler,D. (1999) Using the Fisher kernel method to detect remote protein homologies. *Proceedings of the Seventh International Conference on Inteligent Systems in Molecular Biology*. AAAI Press, Menlo Park, Calif, Forthcoming.

Jensen,F.V. (1996) *Introduction to Bayesian Networks*. Springer, New York.

Jones,D.T., Taylor,W.R. and Thornton,J.M. (1992) A new approach to protein fold recognition. *Nature*, **358**, 86–89.

Jones,D.T. (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.*, **287**, 797–815.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Karplus,K., Sjolander,K., Barrett,C., Cline,M., Haussler,D., Hugey,R., Holm,L. and Sander,C. (1997) Predicting protein structure using hidden Markov models. *Proteins*, Suppl. 1, 134–139.

Krogh,A., Brown,M., Mian,I.S., Sjolander,K. and Haussler,D. (1994) Hidden Markov models in computational biology—applications to protein modelling. *J. Mol. Biol.*, **235**, 1501–1531.

Lathrop,R.H., Rogers,Jr,R.G., Bienkowska,J., Bryant,B.K.M., Buturovic,L.J., Gaitatzes,C., Nambudripad,R., White,J.V. and Smith,T.F. (1998a) Analysis and algorithms for protein sequence-structure alignment. In Salzberg,S.L., Searls,D.B. and Kasif,S. (eds), *Computational Methods in Molecular Biology*. Elsevier Science, Amsterdam, pp. Chapter 12, pp. 227–283.

Lathrop,R.H., Rogers,Jr,R.G., Smith,T.F. and White,J.V. (1998b) A Bayes-optimal probability theory that unifies protein sequence-structure recognition and alignment. *Bull. Math. Biol.*, **60**, 1039–1071.

Lauritzen,S.L. and Spiegelhalter,D.J. (1988) Local computations with probabilities on graphical structures and their application to expert systems. *J. Roy. Stat. Soc.* B, **50**, 157–224.

Levitt,M. (1997) Competitive assessment of protein fold recognition and alignment accuracy. *Proteins*, (Suppl. 1), 92–104.

Madej,T., Gibrat,J.F. and Bryant,S.H. (1995) Threading a database of protein cores. *Proteins*, **23**, 356–369.

Mueller,A., MacCallum,R.M. and Sternberg,M.J.E. (1999) Benchmarking PSI-BLAST in Genome Annotation. *J. Mol. Biol.*, Forthcoming.

Murzin,A.G., Brenner,S.E., Hubbard,T. and Chothia,C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Park,J., Karplus,K., Barrett,C., Hughey,R., Haussler,D., Hubbard,T. and Chothia,C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.*, **284**, 1201–1210.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California.

Richards,S.-J., Hodgman,C. and Sharp,M. (1995) Reported sequence homology between Alzheimer amyloid-770 and the MRC OX-2 antigen does not predict homology. *Brain Res. Bull.*, **38**, 305–306.

Rost,B. and Sander,C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.*, **232**, 584–599.

Rost,B. and Sander,C. (1994) Conservation and prediction of solvent accessibility in protein families. *Proteins*, **20**, 216–226.

Rost,B. (1995) TOPITS: threading one-dimensional predictions into three-dimensional structures. *Proceedings of the Third International Conference on Intelligent Systems in Molecular Biology*. AAAI Press, Menlo Park, Calif, pp. 314–321.

Rost,B., Schneider,R. and Sander,C. (1997) Protein fold recognition by prediction-based threading. *J. Mol. Biol.*, **270**, 471–480.

Russell,R.B., Copley,R.R. and Barton,G.J. (1996) Protein fold recognition by mapping predicted secondary structures. *J. Mol. Biol.*, **259**, 349–365.

Russell,R.B., Saqi,M.A.S., Sayle,R., Bates,P.A. and Sternberg,M.J.E. (1997) Recognition of analogous and homologous protein folds: analysis of sequence and structure conservation. *J. Mol. Biol.*, **269**, 423–439.

Rychlewski,L., Zhang,B. and Godzik,A. (1998) Fold and function predictions for Mycoplasma genitalium proteins. *Fold. Des.*, **3**, 229–238.

Rychlewski,L., Zhang,B. and Godzik,A. (1999) Functional insights from structure predictions: Analysis of the Escherichia coli genome. *Protein Sci.*, **8**, 614–624.

Sander,C. and Schneider,R. (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, **9**, 56–68.

Shindyalov,I.N. and Bourne,P.E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engg*, **9**, 739–747.

Sippl,M.J. (1990) Calculation of conformational ensembles from potentials of mean force. *J. Mol. Biol.*, **213**, 859–883.

Sippl,M.J. and Weitckus, (1992) Detection of native-like models for amino acid sequences of unknown three-dimensional structure in a data base of known protein conformations. *Proteins*, **13**, 258–271.

Smith,T.F. and Zhang,X. (1997) The challenges of genome sequence annotation or 'the devil is in the details'. *Nat. Biotechnol.*, **15**, 1222–1223.

Smyth,P., Heckerman,D. and Jordan,M.I. (1997) Probabilistic independence networks for hidden Markov probability models. *Neural Comput.*, **9**, 227–269.

Teichmann,S.A., Park,J. and Chothia,C. (1999) Structural assignments to the Mycoplasma genitalium proteins show extensive gene duplications and domain rearrangements. *Proc. Natl Acad. Sci. USA*, **95**, 14658–14663.

Thompson,J.D., Higgins,D.G. and Gibson,T.J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.

Thorne,J.L., Goldman,N. and Jones,D.T. (1996) Combining protein evolution and secondary structure. *Mol. Biol. Evol.*, **13**, 666–673.

Vingron,M. and Argos,P. (1989) A fast and sensitive mutiple sequence alignment algorithm. *Comput. Appl. Biosci.*, **5**, 115–121.

White,J.V., Stultz,C.M. and Smith,T.F. (1994) protein classification by stochastic modeling and optimal filtering of amino acid sequences. *Bull. Math. Biosci.*, **119**, 35–75.

Wild,D. and Ghahramani,Z. (1998) A Bayesian network approach to protein fold recognition. *6th International Conference on Intelligent Systems for Molecular Biology*. Montreal, Canada (abstract).

Yu,L., White,J.V. and Smith,T.F. (1998) A homology identification method that combines protein sequence and structure information. *Protein Sci.*, **7**, 2499–2510.

Zemla,A., Venclovas,C., Reinhardt,A., Fidelis,K. and Hubbard,T.J. (1997) Numerical criteria for the evaluation of ab initio predictions of protein structure. *Proteins*, Suppl. 1, 140–150.

Zhang,B., Jaroszewski,L., Rychlewski,L. and Godzik,A. (1997) Similarities and differences between non-homologous proteins with similar folds: evaluation of threading strategies. *Folding and Design*, **2**, 307–317.