MULTI-CLASS PROTEIN SEQUENCE CLASSIFICATION USING NEURAL NETWORKS

K. Blekas, D.I. Fotiadis and A. Likas

23– 2002

Preprint, no 23 – 02 / 2002

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece

Multi-class protein sequence classification using neural networks

Konstantinos Blekas, Dimitrios I. Fotiadis and Aristidis Likas
Department of Computer Science and Biomedical Research Institute - FORTH
University of Ioannina
45110 Ioannina, Greece

Abstract

We present a system for multi-class protein classification based on neural networks. The basic issue concerning the construction of neural network systems for protein classification is the sequence encoding scheme that must be used in order to feed the neural network. To deal with this problem we propose a method that maps a protein sequence into a numerical feature space using the matching scores of the sequence to groups of conserved patterns (called motifs) into protein families. We consider two alternative ways for identifying the motifs to be used for feature generation and provide a comparative evaluation of the two schemes. We also evaluate the impact of the incorporation of background features (2-grams) on the performance of the neural system. Experimental results on real datasets indicate that the proposed method is highly efficient and is superior to other well-known methods for protein classification.

1 Introduction

Protein sequence classification constitutes an important problem in biological sciences for annotating new protein sequences and detecting close evolutionary relationships among sequences. It deals with the assignment of sequences to known categories based on homology detection properties (sequence similarity). In several studies, protein classification has been examined at various levels, according to a top-down hierarchy in molecular taxonomy, consisting of superfamilies, families and subfamilies [7]. In this paper we will use the term *family* or class to denote any collection of sequences that are presumed to share common characteristics.

Various approaches have been developed for solving the protein classification problem. Most of them are based on appropriately modeling protein families, either directly or indirectly. Direct modeling techniques use a training a set of sequences to build a model that characterizes the family of interest. Hidden Markov models (HMMs) are a widely used probabilistic modeling method for protein families [9] that provides a probabilistic measurement (score) of how well an unknown sequence fits to a family. Indirect techniques use direct models as a preprocessing tool in order to extract useful sequence features. In this way, sequences of variable length are transformed into fixed-length input vectors that are subsequently used for training discriminative models, such as neural networks.

In protein sequences, *motifs* or *patterns* enclose significant homologous attributes, since they correspond to conserved regions in protein families holding useful structural and functional biological properties. They can be considered as islands of aminoacids conserved in the same order of a given family. Therefore they can be seen as local features characterizing the sequences. Motifs can be either deterministic or probabilistic [6, 23]. Deterministic motifs follow grammatical inference properties in order to syntactically describe conserved regions of homologous sequences. The PROSITE database [13] represents a large collection of such motifs used to identify protein families. On the other hand, probabilistic motifs are more flexible models and they provide a probabilistic matching score of a sequence to a motif. The BLOCKS database [11] is an example of ungapped probabilistic motifs. In any case, motif-models are suitable for constructing efficient similarity score functions that can be subsequently used as local features for protein classification. An example is presented in [21, 25] where motif-based local features are produced based on the minimum description length (MDL) principle for the case of deterministic motif models.

The *background* information also constitutes another source for extracting features from sequence data. The determination of the background features, also defined as *global* features, is usually made by using the *2-gram* encoding scheme that counts the occurrences of two consecutive aminoacids in protein sequences [25]. In the case of protein sequences (generated from the alphabet of the 20 aminoacids), there are 400 possible 2-grams, that produce a large feature space. A recent approach [1] proposes a scheme for globally encoding sequences, where each aminoacid character is initially represented as a unique binary number with n bits ($n = 5$ for the 20 aminoacids) and then each sequence is mapped into a position inside the n-dimensional hypercube.

In this paper, we focus on building efficient neural classifiers for discriminating multiple protein families by using appropriate local features that have been extracted by efficient probabilistic motif models. As motifs constitute family diagnostic signatures, our aim is to exploit this information by constructing a neural network scheme that exploits motif-based (local) features.

The proposed method can be considered as combining an unsupervised with a supervised learning technique. Starting by applying a motif discovery (unsupervised) algorithm, we identify probabilistic motifs in a training set of multi-class sequences. This can be achieved in two alternative ways: applying the algorithm for motif discovery either to each family training set separately (*class-dependent* motifs), or to the whole dataset of training sequences (*class-independent* motifs). The discovered motifs are then used to convert each sequence to a numerical input vector that subsequently can be applied to a typical feedforward neural

network. Using a Bayesian regularization training technique, the neural network parameters are adjusted and therefore a classifier is obtained suitable for predicting the family of an unlabeled sequence.

The next section provides a brief overview of statistical and neural techniques proposed for classifying biological sequences, while Section 3 describes the proposed method. Experimental results obtained using several sets of protein families are presented in Section 4, along with a comparison with other known protein classification approaches. Finally, Section 5 summarizes the proposed classification scheme and specifies directions for future research.

2 Protein Classification Methods

One class of methods for protein sequence classification work directly with sequence information and establish classification criteria based on sequence homology properties. In the general scheme, a representative set of sequences is selected for each protein family and used to build an appropriate model for each family. The classification of an unknown sequence is made by selecting the family that best matches according to the model homology mechanism. This can be considered as a simple *nearest neighbor* scheme that ranks sequence similarities and selects the best ranking.

The popular BLAST tool [2] represents the simplest nearest neighbor approach and exploits pairwise local alignments to measure sequence similarity. The BLAST technique compares protein queries with a protein database of labeled sequences and produces normalized alignment scores for each comparison by calculating the corresponding expectation values (E-values). The classification procedure is based on the selection of the label of the sequence that produces the best pairwise alignment score (i.e. minimum E-value).

Another type of direct modeling methods is based on hidden Markov models (HMMs) [9, 18]. After constructing an HMM for each family, protein queries can be easily scored against all established HMMs by calculating the log-likelihood of each model for the unknown sequence and then selecting the class label of the most likely model.

The Motif Alignment and Search Tool (MAST) [4] is based on the combination of multiple motif-based statistical score values. According to this scheme, groups of probabilistic motifs discovered by the MEME motif discovery algorithm [3], are used to construct protein profiles for the families of interest. The MAST algorithm successively estimates the significance of the match of a query sequence to a family model as the product of the p-values of each motif match score. This measure (called E-value) can then be used to select the family of the unknown sequence.

Neural network schemes for protein classification consist of two stages: the *encoding* stage,

where discriminative numerical features are computed for each training sequence and the *decision* stage where the created feature vectors are used as input vectors to a neural network classifier. Various encoding schemes have been proposed in the literature that produce numerical features in the encoding stage based on the calculation of background features (global sequence homology) and local features (locally conserved family information) embedded in motifs. In the decision stage feedforward neural networks have been used trained either through backpropagation [26] or using Bayesian regularization [21, 25]. These approaches are characterized by the enormous size of the extracted input vectors, the imbalance between global and local features (more emphasis on global features) and the need for large training sets (since the number of network inputs is very large). For example in [21, 25] only one feature was responsible for carrying local information, while all the others were produced by the 2-grams encoding scheme (background features).

Support vector machines (SVMs) [24] have been also applied to protein homology detection problems. Such an approach, which has been introduced in [20], feeds probabilistic score values from all motifs available (nearly 10000) in the BLOCKS database [11] into an SVM classifier. Obviously, this scheme uses only local features but the dimensionality of the input space is extremely high. Another method has been proposed in [16, 17] that combines hidden Markov models (HMMs) and SVMs for detecting remote protein homologies. In particular, an HMM is first trained to model a protein family, and then the observed probabilities (in the log space) of each sequence with respect to each parameter of the HMM are calculated. The obtained gradient-log-probability vectors are applied to an SVM to identify the decision boundary between the family and the rest of the protein universe.

3 The proposed method

This paper studies the problem of classifying a set of N protein sequences $\mathbf{S} = \{S_i, i = 1, \ldots, N\}$ into K classes. The set $\mathbf{S}$ is a union of positive example datasets $\mathcal{S}_k$ from K different classes, i.e. $\mathbf{S} = \{ \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_K \}$, and can be seen as a subset of the complete set of all possible sequences over the aminoacid alphabet ($\mathbf{S} \subseteq \Sigma^*$).

Figure 1 illustrates the architecture of the proposed protein classification scheme. It consists of a search tool (unsupervised learning) for discovering probabilistic motifs in a set of K protein families, a feature vector generator that converts protein sequences into feature vectors, and a decision module (neural network) for assigning a protein family to each input sequence. The following subsections describe in detail the major building blocks of the proposed architecture.

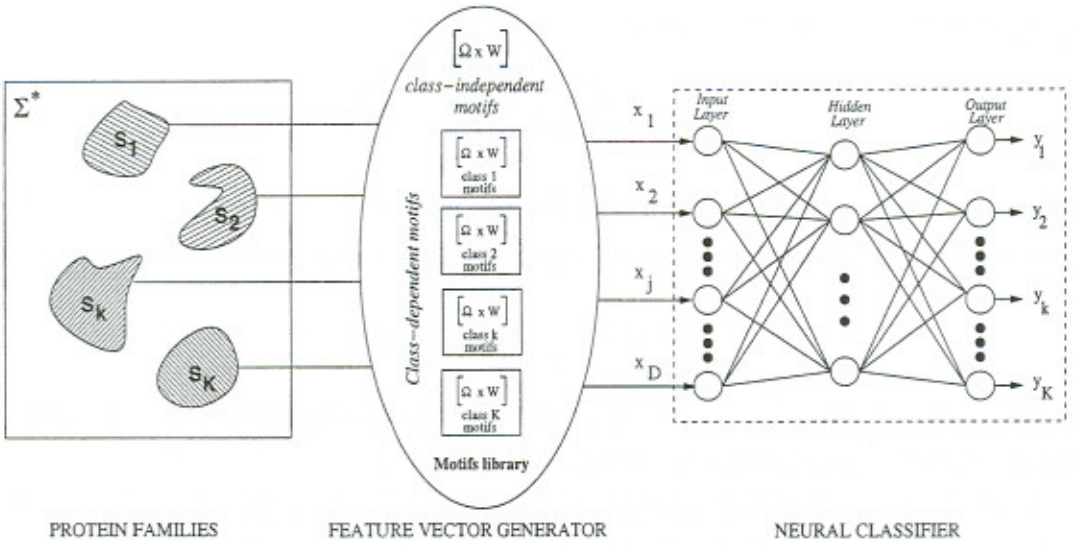

Figure 1: The architecture of the proposed classification scheme.

3.1 Using motifs for feature generation

Consider a finite alphabet consisting of set of characters $\Sigma = \{\alpha_1, \ldots, \alpha_\Omega\}$ ($\Omega = 20$ for protein sequences). We can probabilistically model a contiguous (ungapped) motif M_j of length W_j using a position weight matrix (PWM_j) that follows a multinomial character distribution. Each column (l) of the matrix corresponds to a position l in the motif sequence ($l = 1, \ldots, W_j$), where the column elements provide the probability of each character of the alphabet $p_{\alpha_\xi, l}$ ($\xi = 1, \ldots, \Omega$) to appear in that position.

Let $s_p = a_{p,1} \ldots a_{p,W_j}$ denote a segment of a sequence S beginning at position p and ending at position $p + W_j - 1$. This represents a subsequence of length W_j. Totally, there are $L - W_j + 1$ such subsequences for a sequence S of length L. Then, we can define the probability that s_p matches the motif M_j, or alternatively, has been generated by the model PWM_j corresponding to that motif, using the following equation:

$$P(s_p|M_j) = \prod_{l=1}^{W_j} p_{a_{p,l},l} \ . \tag{1}$$

A major advantage of using the probabilistic matrix PWM_j is the ability to compute the corresponding position-specific score matrix ($PSSM_j$) in order to score a sequence. The $PSSM_j$ is a log-odds matrix calculating the logarithmic ratio $r_{\alpha_\xi, l}$ of the probabilities $p_{\alpha_\xi, l}$ suggested by the PWM_j and the corresponding general relative frequencies of aminoacids ρ_{α_ξ} in the family[1]. According to the definition of $PSSM_j$, the score value $f_j(s_p)$ of a subsequence

[1]The general relative frequencies of aminoacids indicate the background information in a protein family and can be presented as a probabilistic vector ρ of size $\Omega = 20$.

s_p of a sequence S can be defined as:

$$f_j(s_p) = \sum_{l=1}^{W_j} \log(\frac{p_{a_{p,l},l}}{\rho_{a_{p,l}}}) = \sum_{l=1}^{W_j} r_{a_{p,l},l} \ . \tag{2}$$

At the sequence level, the score value of a protein sequence S against a motif M_j can be determined as the maximum value among all scores of the possible subsequences of S, i.e.:

$$f_j(S) = \max_{1 \leq p \leq L - W_j + 1} f_j(s_p) \ . \tag{3}$$

It must be noted that it is possible to adopt other definitions for scoring a sequence, such as setting scores below a certain threshold equal to zero [4].

If we assume that we have discovered a group of D motifs in the set of sequences $\mathbf{S}$, we can generate a D-dimensional numerical feature space and map each sequence S_i into a vector $\mathbf{x}_i$ in the D-dimensional feature space by calculating the score values $x_{ij} = f_j(S_i)$ $(j = 1, \ldots, D)$ for each of the D motif models.

3.2 Finding probabilistic motifs in protein sequences

Several approaches have been proposed for discovering probabilistic motifs in a set of unaligned biological sequences. The CONSENSUS [12], Gibbs sampler [19] and MEME [3] are examples of such methods that identify multiple shared motifs in protein families. We have selected the MEME approach for the motif identification component of our strategy, since it has been widely used in biological applications and directly extracts position specific score matrices. Below we briefly describe this algorithm and propose two ways to integrate it in our classification system.

The MEME algorithm follows an iterative procedure, which applies at each iteration a two-component mixture model to discover one motif of length W. In the two-component model, one component describes the motif (ungapped common subsequences of length W) while the other component models the background information. Multiple motifs can be found by sequentially fitting the two-component model to the set of sequences that remain after removing the sequences containing occurrences of the already identified motifs.

In particular, MEME uses the Expectation Maximization (EM) algorithm [8] to maximize the log-likelihood function of the two-component mixture model [3], i.e. to estimate the elements of the corresponding position weight matrix[2]. Furthermore, MEME provides with a strategy for locating efficient initial parameter values in order to prevent the EM algorithm

[2]The model used in our experiments assumes that there are zero or more non-overlapping occurrences of the motif in each sequence of the dataset. Alternative models that can be used are the exactly one occurrence per sequence and the zero or one occurrence per sequence.

from getting stuck in local optima [3]. The D motif models PWM_j $(j = 1, \ldots, D)$ discovered by MEME can be of either fixed or variable length W_j. In our experimental studies both types of motifs will be examined to evaluate the impact of this decision on the performance of the neural classifier.

In order to discover a group of motifs from a multi-class training set of sequences (containing sequences of K classes) two alternative approaches can be followed. The first approach is to apply the MEME algorithm K times, *seperately* to the training sequences of each protein family. Then, putting all the discovered K family profiles together we can form the final group of D motifs. An alterantive approach is to apply the motif discovery algorithm only once to the total training set $\mathbf{S}$ ignoring class labels. In this way, we do not allow the algorithm to directly create K protein family profiles, but rather to discover D *class-independent* motifs.

The advantage of the second approach is the ability of taking into account local similarity measurements in the whole training set, without restricting the search procedure to a single class. Therefore, possible partial homologies among sequences from different families can be defined that may prove helpful for the classification task. On the other hand, a disandvantage of the class-independent approach is that the D discovered motifs may not be equally distributed among the K families. This may result in insufficient modeling of some families, thus leading to performance deterioration. During expriments both motif discovery strategies will be considered and evaluated.

3.3 Construction of a neural classifier

After dicovering D motifs and constructing the D-dimensional feature space, the last stage in our methodology is to implement and train a feed-forward neural network that will be able to map the input vectors into the protein classes of interest. A typical network architecture is illustrated in Figure 1. To construct the neural classifier we use the training set $\mathbf{X} = \{\mathbf{x}_i, \mathbf{t}_i\}$, $i = 1, \ldots, N$ consisting of positive examples $\mathbf{x}_i$ from the set of K protein families. The target vector $\mathbf{t}_i$ is a binary vector of size K indicating the class label of input $\mathbf{x}_i$, i.e. $t_{ik} = 1$ if $\mathbf{x}_i$ corresponds to a sequence S_i belonging to class k, and 0 otherwise. The output of the classifier is represented by the K-dimensional vector $\mathbf{y}_i$ where component y_{ik} corresponds to class k. Based on this scheme, the predicted class $h(\mathbf{x}_i)$ of an unlabeled feature vector $\mathbf{x}_i$ corresponding to a query sequence S_i is given by the index of the output node with the largest value y_{ic}, i.e.

$$h(\mathbf{x}_i) = c : \; y_{ic} = \max_{1 \le k \le K} y_{ik} \; . \tag{4}$$

Setting a threshold value θ $(\in [0, 1])$, we can restrict the classifiers' decision only to those input vectors whose maximum output value surpasses this threshold. In this case we can

write:

$$h(\mathbf{x}_i, \theta) = c : \; y_{ic} = \max_{1 \le k \le K} y_{ik} \; \wedge \; y_{ic} \ge \theta \; . \tag{5}$$

Parameter θ can be used to specify the sensitivity of the classifier.

In order to train the neural network we used the Gauss-Newton Bayesian Regularization (GNBR) learning algorithm [10]. This algorithm applies Bayesian regularization and implements a Gauss-Newton approximation to the Hessian matrix of the objective function.

In the Bayesian regularization framework the objective function is formulated as the weighted sum of two terms: the sum of the squared errors (E_X) and the sum of squares of the network weights (E_W). Using Bayes' rule, the posterior probability distribution for the weights $\mathbf{w}$ of the network given a training set $\mathbf{X}$ can be written as follows:

$$P(\mathbf{w}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{X})} \; . \tag{6}$$

By properly choosing the prior distribution $P(\mathbf{w})$ and the likelihood function $P(\mathbf{X}|\mathbf{w})$, we can obtain the following expression for the posterior distribution [5, 10]:

$$P(\mathbf{w}|\mathbf{X}) = \frac{1}{Z_F} \exp(-\beta E_X - \alpha E_W) = \frac{1}{Z_F} \exp(-F(\mathbf{w})), \tag{7}$$

where the Z_F corresponds to the normalizing factor that is independent of the weights.

Maximizing the above posterior distribution is equivalent to minimizing the regularized objective function $F(\mathbf{w})$:

$$F(\mathbf{w}) = \frac{\beta}{2} \sum_{i=1}^{N_X} \{\mathbf{y}_i - \mathbf{t}_i\}^2 + \frac{\alpha}{2} \sum_{j=1}^{N_W} w_j^2 \; , \tag{8}$$

where N_X and N_W represent the number of input vectors and network parameters, respectively. In order to estimate the normalizing factor Z_F a Gaussian approximation can be used for the posterior distribution [22] as obtained by the Taylor expansion of function $F(\mathbf{w})$ around the minimum value of the posterior, $\mathbf{w}_{MP}$. This gives the following estimation: [5]:

$$Z_F^*(\alpha, \beta) = \exp(-F(\mathbf{w}_{MP}))(2\pi)^{N_W/2} |\mathbf{H}|^{-1/2} \; , \tag{9}$$

where $\mathbf{H}$ corresponds to the Hessian matrix of the regularized objective function and, therefore, optimal values for parameters α and β at the minimum point $\mathbf{w}_{MP}$ can be computed as follows:

$$\hat{\alpha} = \frac{\gamma}{2E_W(\mathbf{w}_{MP})} \text{ and } \hat{\beta} = \frac{\gamma N_X}{2E_X(\mathbf{w}_{MP})} \; . \tag{10}$$

The quantity γ represents the effective number of network parameters $\mathbf{w}$ and can be defined using the eigenvalues of H^{-1} as $\gamma = N_W - 2\alpha \mathrm{Tr}\mathbf{H}^{-1}$. In cases where the number of effective parameters is equal to the actual ones ($\gamma \approx N_W$), more hidden units must be added to

the network. Furthermore, the GNBR algorithm follows a Gauss-Newton approximation method [10] for calculating the Hessian matrix of $F(\mathbf{w})$ at the minimum point $\mathbf{w}_{MP}$, using the Levenberg-Marquardt optimization algorithm [5]. It must be noted that in our experiments, the best results for the GNBR algorithm were obtained by scaling the network inputs in the range $[-1, 1]$.

4 Experimental results

Several experiments were conducted to evaluate the proposed method. The classification accuracy was measured by counting the sensitivity and specificity rates. In all K-class classification problems, each protein family $\mathcal{S}_k$ $(k = 1, \ldots, K)$ was randomly partitioned into training and test sequences, with the training set being only a small percentage (5 - 10%) of the family dataset. Using the training datasets experiments have been carried out using the MEME algorithm to discover groups of motifs. Two cases were considered: in the first case, the MEME algorithm has been applied seperately to each training set providing a group of $D_k = 5$ *class-dependent* motifs for each family $\mathcal{S}_k$[3]. In the second case the MEME algorithm was applied only once to the total training dataset (ignoring the class labels) to provide a group of $D = 5 \times K$ *class-independent* motifs.

In any case, the obtained final group of D motifs were used to transform each sequence of the dataset into a dataset with numerical D-dimensional feature vectors, denoted $\mathbf{X}_s$ for the class-dependent case and $\mathbf{X}_g$ for the class-independent case. Furthermore, we also experimented with the effect of the length W of the discovered motifs to the performance of the proposed classifier, by applying the MEME algorithm with either fixed or variable motif length. We selected $W = 20$ for the first case and the range $[10, 30]$ for the second case. In summary, we have considered four distinct cases considering the application of MEME: discovering either class-dependent or class-independent motifs with either fixed or variable motif length. Therefore, for each classification problem four distinct neural classifiers will be constructed and tested.

To evaluate classification performance ROC (Receiver Operating Characteristic) analysis was used. More specifically, we used the ROC_{50} curve which is a plot of the sensitivity as a function of false positives for various decision threshold values until 50 false positives are found.

For our experimental study three real datasets were selected. In particular we have used protein families from the PROSITE database [13], which is a large collection of protein families together with their characteristic (deterministic) motifs. Two datasets with $K = 6$

[3]Experiments with greater number of motifs did not yield better classification performance.

Problem: PROSITE 1 ($K = 6$)			*Problem: PROSITE 2 ($K = 7$)*		
PROSITE family	*Positive data*	*Training set (avg length of seqs)*	*PROSITE family*	*Positive data*	*Training set (avg length of seqs)*
PS00030	302	20 (370)	PS00070	129	15 (558)
PS00038	289	20 (359)	PS00077	155	15 (502)
PS00061	317	20 (299)	PS00118	168	15 (127)
PS00198	300	20 (284)	PS00180	123	15 (408)
PS00211	574	30 (478)	PS00215	123	15 (321)
PS00301	386	20 (517)	PS00217	148	15 (490)
			PS00338	173	15 (212)

Table 1: The two PROSITE families used in the experimental study.

Problem: GPCR ($K = 7$)		
GPCR Class A subfamily	*Positive data*	*Training set (avg length of seqs)*
Amine	306	20 (485)
Peptide	654	30 (383)
Hormone	325	20 (317)
Rhodopsin	270	20 (358)
Olfactory	58	10 (348)
Prostanoid	43	10 (721)
Nucleotide-like	43	10 (378)

Table 2: Seven families from the GPCR class A used in the experimental study.

and $K = 7$ classes from the PROSITE database [13] were selected, summarized in Table 1. Moreover, experiments have also been conducted on a dataset of G-protein coupled receptors (GPCR) [14], that is a superfamily of cell membrane proteins. The GPCR database is hierarchically classified into five major classes and their subfamilies [14]. We studied the problem of classifying subfamilies within the class A, since it dominates the whole GPCR database. As indicated in [17], the difficulty of recognizing GPCR subfamilies arises from the fact that the classification of the subfamilies has been made based on chemical properties rather than sequence homology. Therefore, members from different subfamilies may share strong homology thus making their discrimination hard. Among 15 subfamilies consisting class A, seven of them have been selected in our experimental study described in Table 2. The remaining eight subfamilies are of very small size and it is difficult to construct an effective system for their discrimination.

Problem	N_g *2-gram features*	D *motif-based features*
PROSITE 1	174	$5 \times 6 = 30$
PROSITE 2	285	$5 \times 7 = 35$
GPCR	152	$5 \times 7 = 35$

Table 3: The number of the extracted motif-based (D) and 2-gram (N_g) features that corresponds to each dataset.

4.1 Local versus global features

In this series of experiments we assessed the impact of using 2-grams (background features) on the performance of the proposed classification scheme. For a sequence S_i with length L_i we define the feature value g_{iq} for each 2-gram q with respect to this sequence as:

$$g_{iq} = \frac{\mathcal{N}(q|S_i)}{L_i - 1} \ , \tag{11}$$

where $\mathcal{N}(q|S_i)$ denotes the number of occurrence of the 2-gram feature q in the sequence S_i. As it is obvious, the above equation gives the relative frequency of a 2-gram feature in a sequence. In a training set $\mathbf{S} = \{S_1, S_2, \ldots, S_N\}$ of N sequences we can ignore *redundant* 2-grams and consider only the N_g features g_{iq} that correspond to the most frequently occurring 2-grams. We select the N_g 2-grams occurring in at least half of the training sequences and by computing the corresponding g_{iq} ($q = 1, \ldots, N_g$) values for each sequence S_i, we construct the corresponding feature vectors to be fed in the neural classifier.

Table 3 presents the dimensionality of the feature spaces obtained using 2-grams and motifs for each dataset used in the experiments. It must be noted that we can further reduce the dimensionality of the 2-gram feature vectors using standard dimension reduction techniques, such as principal component analysis (PCA). To assess the impact of the several feature types on the performance of the classification system we have considered five different datasets:

- $\mathbf{X_s}$: D motif-based features separately identified for each family (class-dependent),
- $\mathbf{X_g}$: D motif-based class-independent features,
- $\mathbf{X_s} \cup \mathbf{G}$: D motif-based class-dependent features along with N_g 2-gram features,
- $\mathbf{X_g} \cup \mathbf{G}$: D motif-based class-independent features, along with N_g 2-gram features
- $\mathbf{G}$: N_g 2-gram features.

The neural network architecture had one hidden layer of either 10 (for the cases $\mathbf{X}_s$ and $\mathbf{X}_g$) or 20 nodes for the other three cases[4].

[4]We have used the *trainbr* network training function of the Neural Network Toolbox of Matlab v. 6.1.

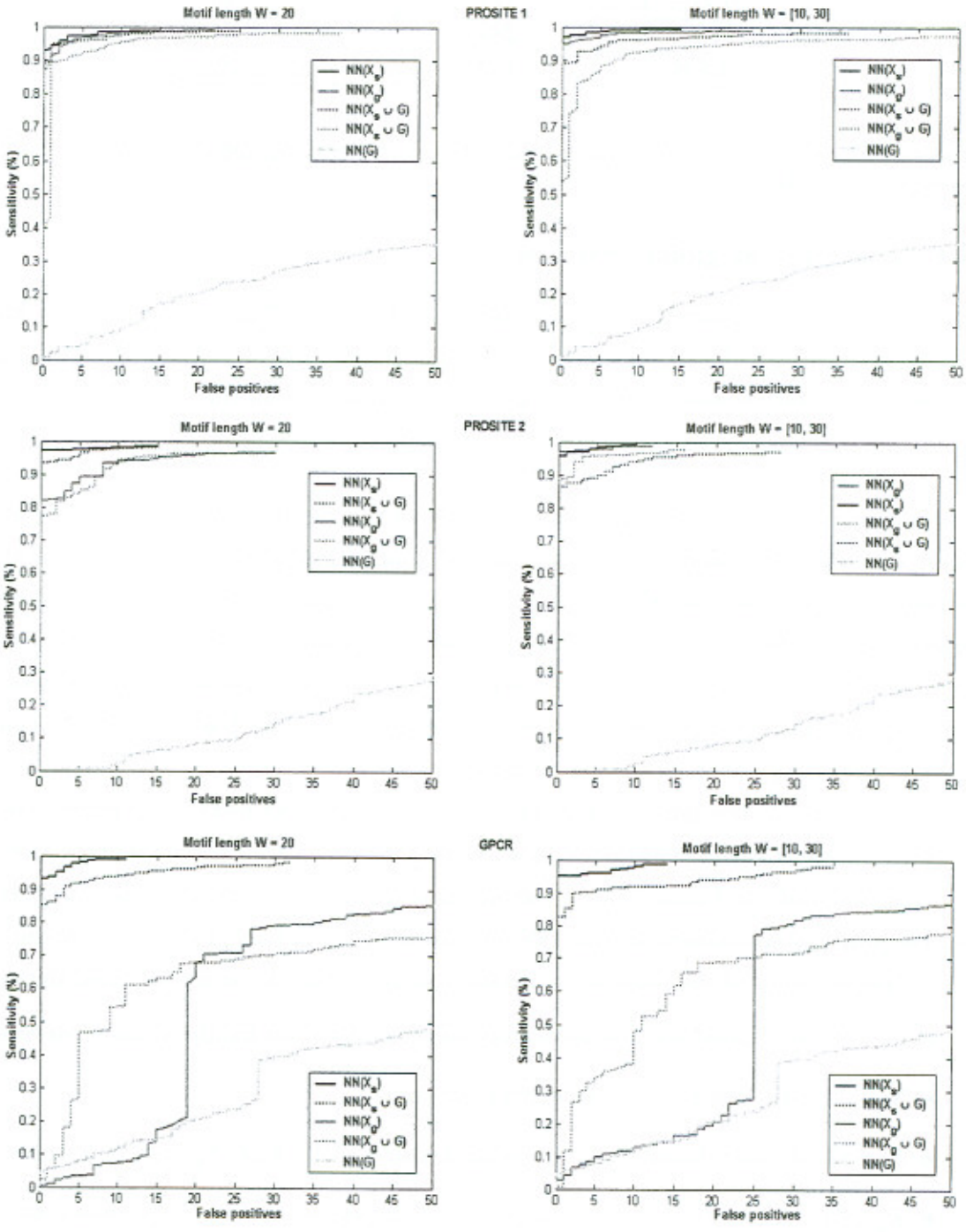

Figure 2: ROC_{50} curves illustrating the performance of the neural classifier on the three datasets using the five different feature vectors.

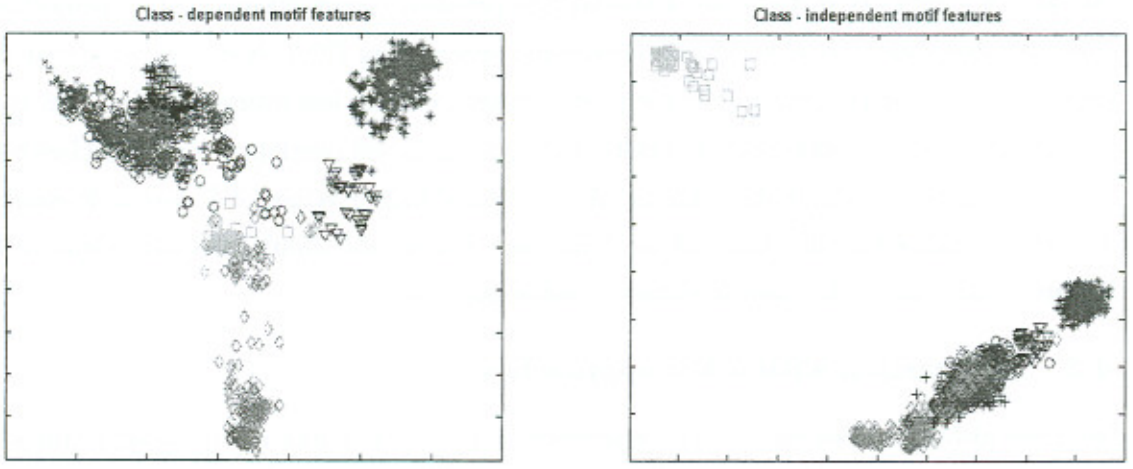

Figure 3: The seven class regions in the GPCR dataset in the case of class-dependent and class-independent features. The data have been projected in two dimensions using PCA.

Figure 2 displays the ROC_{50} curves obtained after training the five neural classifiers in each of the three classification problems respectively. For each problem two different graphs are presented concerning motifs of fixed length ($W = 20$) and of variable length $W \in [10, 30]$. As it is obvious, motif-based features itself constitute an excellent source of information able to generate significant features and lead to the construction of efficient classifiers. In all cases, the neural networks trained by mixed features (e.g. NN($\mathbf{X}_s \cup \mathbf{G}$)) exhibit lower classification accuracy compared to the corresponding classifier trained with only motif-based features (e.g. NN($\mathbf{X}_s$)). Furthermore, the 2-grams features alone (case NN($\mathbf{G}$)) do not seem to contain significant discriminant information.

Another observation that can be made from the ROC_{50} curves in Figure 2 is related to the performance of the neural classifier with class-dependent motifs (network NN($\mathbf{X}_s$)) compared to that obtained with class-independent motifs (network NN($\mathbf{X}_g$)). In almost all cases we obtained better classification results with the network NN($\mathbf{X}_s$). One explanation for this behaviour is that, when searching for a specific number D of motifs in the whole training set (ignoring class labels) the algorithm may focus on some of the of families and leave the other families explored only partially. This possibly affects the satisfactory modeling of some families, since the discovered class-independent motifs may not be sufficient for describing them (only a few individual motifs dedicated to this family). Experiments in the $\mathbf{X}_g$ datasets with MEME have shown that the allocation of motifs in most cases was not equal for all the K families.

An example is shown in Figure 3 that illustrates the constructed feature space of the

$\mathbf{X}_s$ and $\mathbf{X}_g$ datasets in the case of the GPCR problem (seven classes), after projecting the 35-dimensional numerical to a two-dimensional space using PCA. It can be observed that in the case of class-dependent motifs the protein classes exhibit less overlap while in the reduced feature space of class-independent motifs there is a significant overlapping among class regions, thus making the discrimination harder. A selection of higher values of D probably would lead to better results for the class-independent case, but would simultaneously result in larger feature spaces or to the overestimation of some families.

4.2 Comparison with other approaches

We have also compared the neural classifier (with class-dependent motif-based features) with two other protein classification methods, namely the MAST homology detection algorithm [4] and the profile HMMs built using SAM [15]. In both MAST and SAM each protein family is transformed (indirectly or directly) into a probabilistic model-profile and the test sequences are classified using the class of the profile with the best score value.

More specifically, the MAST procedure [4] initially uses the MEME algorithm to discover groups of motifs separately for each one of the K protein families. For each sequence in the testing set, the MAST algorithm combines the calculated p-values and estimates the significance of the observed match (called E-value) of the sequence to each of the K groups of motifs[5]. Then the query sequence is assigned to the class with the minimum E-value. The SAM method [15] works in a similar way by building an HMM for each one of the K protein families instead of discovering groups of motifs[6].

Figure 4 provides comparative results from the application of the proposed neural classifier, MAST and SAM to the three datasets. We have created five ROC curves for each method (number of false positives versus sensitivity for several threshold values) until 25 false positives were found (ROC_{25}). The performance of the neural classifier and MAST was given by two curves respectively[7] concerning motifs of fixed ($W = 20$) and variable length ($W = [10, 30]$), while the last one corresponds to SAM performance. In the case of MAST and SAM methods, ROC curves were obtained by setting several E-value thresholds. When the lowest estimated E-value for a query sequence was greater than the threshold then the test sequence was considered unclassified.

The superior classification of the proposed neural approach is obvious from the plotted curves in all problems, offering greater sensitivity rates with perfect specificity (zero false

[5] We use the *meme* and *mast* commands from the available MEME package v.3.0.4.

[6] We used the *buildmodel* and *hmmscore* commands from the available SAM package v.3.3.1.

[7] The curves for the neural classifier performance were the best plots from the corresponding ROC_{50} diagrams in Figure 2.

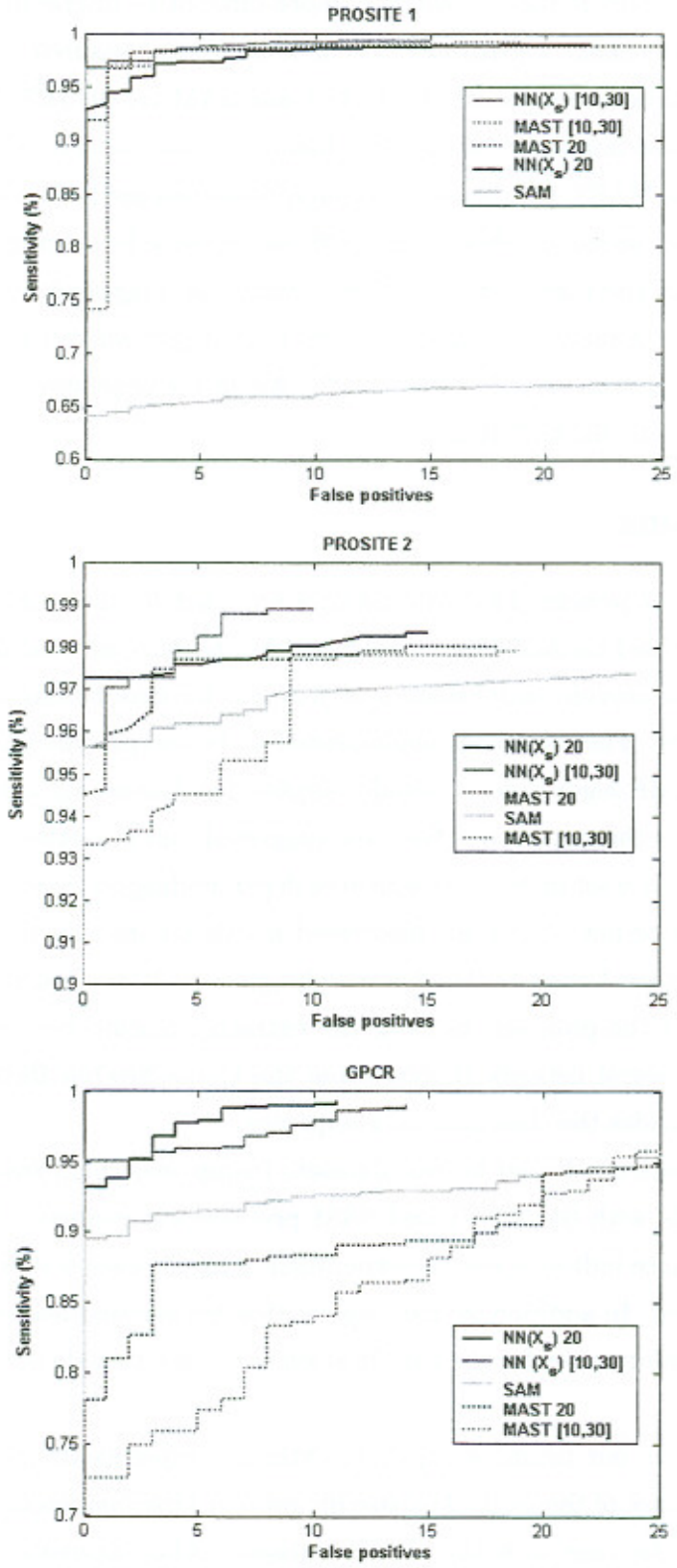

Figure 4: ROC_{25} curves for the three methods (neural (NN), MAST and SAM) on the three datasets.

positives). For the GPCR dataset which is more difficult to discriminate, the classification improvement is more clear: a sensitivity rate of 99.30% was measured with only 11 false positives, while the corresponding results for MAST and SAM are $(95.76\%, 25)$ and $(95.38\%, 25)$, respectively. It is also important to stress the higher accuracy that the neural scheme achieves compared with the MAST (dot lines). Although these two methods use the same groups of motifs, our method seems to offer a more efficient scheme for combining the motif match scores compared to the combination of their p-values as suggested by MAST. In addition, the neural classifier achieves less false positives with higher sensitivity rates in all datasets concerning either fixed or variable motif length. Again the improvement is more clear in the plots corresponding to the GPCR dataset.

5 Conclusions

In this paper we have presented a neural network approach for the classification of protein sequences. The proposed methodology is motivated by the principle that in biological sequence analysis motifs can provide major diagnostic features for determining the class label of the unknown sequences. The method is implemented in two steps, where a pre-processing step (based on the MEME algorithm) is initially applied for discovering a group of probabilistic motifs appearing in the sequences. We have suggested and evaluated two alternative ways for motif discovery in a set of K-class sequences dependinding on whether the class labels are taken into account or not. Using the discovered motifs a numerical feature vector is generated for each sequence by computing the matching score of the sequence to each motif. At the second stage of the proposed method, the extracted feature vectors are used as inputs to a feed-forward neural network trained using the Gauss-Newton Bayesian Regularization algorithm that provides the class label of a sequence.

Experiments were conducted on real datasets (using very small training sets) and comparisons were made with the MAST and SAM probabilistic methods. ROC diagrams were used as a performance indicator and the experimental results clearly illustrate the superiority of the neural system. In addition we have shown that background features do not constitute a useful source of information for the classification task since they do not lead to performance improvement.

In what concerns our future work, more extensive experiments could be conducted to assess the performance of the method on specific protein superfamilies of important biological functions, as was the case with the GPCR dataset. Also, alternative methods could be implemented and tested, both in the classification stage (mixture models, SVMs etc) and in the motif discovery stage.

References

[1] J. S. Almeida and S. Vinga. Universal sequence map (USM) of arbitrary discrete sequences. *BMC Bioinformatics*, 3(6), 2002.

[2] S. F. Altshul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[3] T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, Menlo Park, California, 1994. AAAI Press.

[4] T. L. Bailey and M. Gribskov. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, 14:48–54, 1998.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford Univ. Press Inc., New York, 1995.

[6] A. Brāzma, I. Jonasses, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5(2):277–303, 1998.

[7] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. *Atlas of protein sequence and structure.* Natl. Biomed. Res. Found., Washington, Dc., Vol. 5., 1978.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B*, 39:1–38, 1977.

[9] R. Durbin, S. Eddy, A. Krough, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Protein and Nucleic Acid.* Cambridge University Press, New York, NY, 1998.

[10] F. D. Foresse and M. T. Hagan. Gauss-Newton approximation to Bayesian regularization. In *Proceedings of the 1997 International Joint Conference on Neural Network*, pages 1930–1935, 1997.

[11] S. S. Henikoff and J. G. Henikoff. Protein family classification based on searching a database of blocks. *Genomics*, 19:97–107, 1994.

[12] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7/8):563–577, 1999.

[13] K. Hofmann, P. Bucher, L. Falquet, and A. Bairoch. The PROSITE database, its status in 1999. *Nucleic Acids Research*, 27:215–219, 1999.

[14] F. Horn, J. Weare, M. W. Beukers, S. Hörsch, A. Bairoch, W. Chen, O. Edvardsen, F. Campagne, and G. Vriend. GPCRDB: an information system for G protein-coupled receptors. *Nucleic Acids Research*, 21(1):227–281, 1998.

[15] R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, 12(2):95–107, 1996.

[16] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1,2):95–114, 2000.

[17] R. Karchin, K. Karplus, and D. Haussler. Classifying G-protein coupled receptors with support vector mahcines. *Bioinformatics*, 18(1):147–159, 2002.

[18] K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

[19] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwland, and J. C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 226:208–214, 1993.

[20] B. Logan, P. Moreno, B. Suzek, Z. Weng, and S. Kasif. A study of remote homology detection. Technical Report CRL 2001/05, Cambridge Research Laboratory, 2001.

[21] Q. Ma and J. T. L. Wang. Application of Bayesian neural networks to protein sequence classification. In *ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 305–309, Boston, MA, USA, Aug 2000.

[22] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[23] I. Rigoutsos, A. Floratos, L. Parida, Y. Gao, and D. Platt. The Emergency of Pattern Discovery Techniques in Computational Biology. *Metabolic Engineering*, 2:159–177, 2000.

[24] V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data.* Nauka, Birmingham, AL, 1979.

[25] J. T. L. Wang, Q. Ma, D. Shasha, and C. H. Wu. New techniques for extracting features from protein sequences. *IBM: Systems Journal*, 40(2):426–441, 2001.

[26] C. H. Wu, S. Zhap, H. L. Chen, C. J. Lo, and J. McLarty. Motif identification neural design for rapid and sensitive protein family search. *CABIOS*, 12(2):109–118, 1996.