Protein Sequence Classification Using Probabilistic Motifs and Neural Networks

Konstantinos Blekas, Dimitrios I. Fotiadis, and Aristidis Likas

Department of Computer Science, University of Ioannina, 45110 Ioannina, Greece
and Biomedical Research Institute, FORTH – Hellas, 45110 Ioannina, Greece
{kblekas,fotiadis,arly}@cs.uoi.gr

Abstract. 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 network. To deal with this problem we propose a method that maps a protein sequence into a numerical feature space using the matching local scores of the sequence to groups of conserved patterns (called *motifs*). We consider two alternative schemes for discovering a group of D motifs within a set of K-class sequences. We also evaluate the impact of the background features (2-grams) to the performance of the neural system. Experimental results on real datasets indicate that the proposed method is superior to other known protein classification approaches.

1 Introduction

Consider a finite set of characters $\Sigma = \{\alpha_1, \ldots, \alpha_\Omega\}$, where $\Omega = |\Sigma|$. Any sequence $S = a_1 a_2 \ldots a_L$, such that $L \geq 1$ and $a_i \in \Sigma$, is called a *sequence* over the alphabet Σ. In the case of proteins, the alphabet Ω is the set of 20 aminoacids. Protein sequence classification constitutes an important problem in biological sciences. It deals with the assignment of sequences to known categories based on homology detection properties (sequence similarity). We 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 this 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 each family of interest. Hidden Markov models (HMMs) are a widely used probabilistic method for protein families [1] that provides a probabilistic measurement (score) of how well an unknown sequence fits to a family. The classification is then made by selecting the class label of the most likely model [1]. Indirect techniques use an *encoding* stage 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 [2].

In biological sequences, *motifs* or patterns can be considered as islands of aminoacids conserved in the same order of a given family [3]. Since they enclose

O. Kaynak et al. (Eds.): ICANN/ICONIP 2003, LNCS 2714, pp. 702–709, 2003.

significant homologous attributes, they can be seen as local features characterizing the sequences. The *background* information also constitutes another source of information for sequence data. A common way to determine background features, also termed as *global* features, is to use the *2-gram* encoding scheme that counts the occurrences of two consecutive characters in sequences [2]. In the case of protein sequences (generated from the alphabet of the 20 aminoacids), there are 400 possible such 2-gram features.

Several neural network schemes have been applied that follow alternative encoding schemes and training methods [4],[2]. 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 [4],[2] only one feature was responsible for carrying local information, while all the others were 2-gram features. Another class of discriminative model used for classifying sequences is the Motif Alignment and Search Tool (MAST) [5]. The MAST algorithm 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 (E-value) can then be used to select the family of the unknown sequence.

In this paper, we focus on building efficient neural classifiers for discriminating multiple protein families by using appropriate local features extracted from efficient probabilistic motif models. As motifs constitute family diagnostic signatures, our aim is to formulate a neural network scheme that exploits motif-based (local) features. It can be considered as a combination of an unsupervised and a supervised learning technique. In the first stage, we identify probabilistic motifs in a training set of multi-class sequences. We assume two alternative ways, depending on whether or not taking into account the class labels. For this purpose we use the MEME algorithm [6] that follows iteratively a two-component mixture model approach. The discovered motifs are then used to convert each sequence to a numerical feature vector that subsequently can be applied to a typical feedforward neural network. Using a Bayesian regularization training technique [7],[8], the neural network parameters are adjusted and therefore a classifier is obtained suitable for predicting the family of an unlabeled sequence. The next section describes the proposed method, while experimental results obtained using real sets of protein sequences are presented in Section 3. Finally, in Section 4 we present our conclusions.

2 The Proposed Method

Consider 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 Σ ($\mathbf{S} \subseteq \Sigma^*$). The proposed protein classification scheme consists of three main stages. A supervised technique is first applied for discovering probabilistic motifs in a set of

K protein families. This follows a feature vector generator that converts protein sequences into feature vectors. Finally, a neural network is used for assigning a protein family to each input vector.

2.1 Discovering Probabilistic Motifs in Sequences

A motif M_j of length W_j can be probabilistically modeled 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 ($l = 1, \ldots, W_j$). The column elements provide the probability $p_{\alpha_\xi, l}$ of each character α_ξ of the alphabet $\Sigma = \{\alpha_\xi, \xi = 1, \ldots, \Omega\}$ to appear in the position l, where $\Omega = 20$ for proteins. Let $s_p = a_{p,1} \ldots a_{p,W_j}$ denote a segment of length W_j beginning at position p and ending at position $p + W_j - 1$ of a sequence S of length L. Totally, there are $L - W_j + 1$ such subsequences. Then, we can define the probability that s_p matches the motif M_j, or has been generated by the model PWM_j corresponding to that motif, using the following equation:

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

Several approaches have been proposed for discovering probabilistic motifs in a set of unaligned biological sequences [3], such as the CONSENSUS, Gibbs sampler and MEME methods. Among these, the MEME algorithm [6] applies a two-component mixture model to discover one motif of length W_j. The first component of the model describes the motif (PWM_j), while the other models the background information, formulated by a probabilistic vector ρ of size Ω. Multiple motifs can be found by sequentially fitting another two-component model to the set of sequences that remain after removing the subsequences that correspond to the occurrences of the already identified motifs[1]. MEME uses the Expectation Maximization (EM) algorithm to maximize the log-likelihood function of the model [6], i.e. to estimate the elements of the corresponding position weight matrix. Furthermore, MEME provides with a strategy for locating efficient initial parameter values in order to prevent the EM algorithm from getting stuck in local optima [6]. 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.

In order to discover a group of motifs from a training set containing sequences of K classes, two alternative approaches can be followed. The first approach is to apply the MEME algorithm K times, one for each protein family, respectively. Then, the union of the discovered groups of motifs D_k ($k = 1, \ldots, K$) can form the final group of D motifs. These will be termed as *class-dependent* motifs. An alternative 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

[1] The model assumes that there are zero or more non-overlapping motifs in each sequence.

algorithm to directly create K protein family profiles, but rather to discover D *class-independent* motifs. During experiments both motif discovery strategies will be considered and evaluated.

Following the probabilistic framework of PWM_j for modeling motifs, we can sequentially 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 ρ_{α_ξ} of aminoacids α_ξ in the family. Given a motif model M_j, the score value $f(s_p|M_j)$ of a subsequence s_p can be defined as:

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

At the sequence level, the score value of a 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. $f(S|M_j) = \max_{1\leq p\leq L-W_j+1} f(s_p|M_j)$. Thus, if we assume that we have discovered a group of D motifs, we can translate each sequence S_i into a D-dimensional feature vector $\mathbf{x}_i$ by calculating the score values $x_{ij} = f(S_i|M_j)$ $(j = 1, \ldots, D)$.

2.2 Construction of the Neural Classifier

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 $\mathbf{x}_i$ into the K protein classes of interest. To construct the neural classifier we use the training set $\mathbf{X} = \{\mathbf{x}_i, \mathbf{t}_i\}$, $i = 1, \ldots, N$. 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 the corresponding sequence S_i belongs to the class k, and 0 otherwise. In an manner analogous, the output of the classifier is represented by a K-dimensional vector $\mathbf{y}_i$. Based on this scheme, the predicted class $h(\mathbf{x}_i)$ of an unlabeled feature vector $\mathbf{x}_i$ is given by the index of the output node with the largest value y_{ik}, i.e. $h(\mathbf{x}_i) = c : \ y_{ic} = \max_{1\leq k\leq K} y_{ik}$. 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:

$$h(\mathbf{x}_i, \theta) = c : \ y_{ic} = \max_{1\leq k\leq K} y_{ik} \ \wedge \ y_{ic} \geq \theta \ . \tag{3}$$

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

In order to train the neural network we use the Gauss-Newton Bayesian Regularization (GNBR) learning algorithm [8]. The GNBR algorithm applies an iterative procedure for Bayesian regularization of the network parameters and implements a Gauss-Newton approximation to the Hessian matrix $\mathbf{H}$ of the regularized objective function [7],[9]:

$$F(\mathbf{w}) = \beta E_X(\mathbf{w}) + \alpha E_W(\mathbf{w}) = \frac{\beta}{2}\sum_{i=1}^{N}\{\mathbf{y}_i - \mathbf{t}_i\}^2 + \frac{\alpha}{2}\sum_{j=1}^{N_W} w_j^2 \ , \tag{4}$$

where $\mathbf{w}$ corresponds to the vector of the network weights and the N_W represent the number of network parameters. The E_X and E_W indicate the sum of the squared errors and the sum of the squares of the network weights, respectively.

At each step, the objective function $F(\mathbf{w})$ is minimized using the Levenberg-Marquardt algorithm to provide a solution $\mathbf{w}_{MP}$. Then, optimal values for parameters α and β at the minimum point $\mathbf{w}_{MP}$ can be computed as follows [7],[9]:

$$\hat{\alpha} = \frac{\gamma}{2E_W(\mathbf{w}_{MP})} \text{ and } \hat{\beta} = \frac{N-\gamma}{2E_X(\mathbf{w}_{MP})} \ , \tag{5}$$

The quantity γ represents the effective number of network parameters $\mathbf{w}$ and can be defined as $\gamma = N_W - 2\hat{\alpha}\mathrm{Tr}\hat{\mathbf{H}}^{-1}$. The GNBR algorithm exploits the approximation of the Hessian provided by the minimization method [8]. 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. 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 $[-1, 1]$.

3 Experimental Results

Several experiments have been conducted to evaluate the proposed method. 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. Experiments have been carried out using the MEME algorithm to discover either groups of $D_k = 5$ *class-dependent* motifs for each family, or a group of $D = 5 \times K$ *class-independent* motifs using the total training dataset (ignoring the class labels). In this way two datasets are created containing D-dimensional feature vectors, denoted by $\mathbf{X}_s$ for the class-dependent case and $\mathbf{X}_g$ for the class-independent case, respectively. 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 θ until 50 false positives are found.

We have selected the two real (public) datasets in our experimental study. The first dataset (nearly 2000 sequences) consists of $K = 6$ families depicted from the PROSITE database, which is a large collection of protein families. The second one (nearly 1800 sequences) contais $K = 7$ subfamilies from the G-protein coupled receptors (GPCR) superfamily. The difficulty of recognizing GPCR subfamilies arises from the fact that their classification has been made based on chemical properties rather than sequence homology.

In the first series of experiments we assessed the impact of using 2-grams (background features). To do this, we constructed a new feature space consisting of only global features. In particular, we defined the feature g_{iq} as the relative frequency of each 2-gram q ($q = 1, \ldots, \Omega^2$) in a sequence S_i. Furthermore, we ignore *redundant* 2-grams and consider only the n_g features g_{iq} that occur frequently (at least half of the N training sequences). Therefore, the new created dataset, called **G**, would contain n_g global features of the input sequences. In summary, we have created five different sets of features: $\mathbf{X_s}$, $\mathbf{X_g}$, $\mathbf{X_s} \cup \mathbf{G}$, $\mathbf{X_g} \cup \mathbf{G}$ and **G** for each problem and we measured their discriminative ability. 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 datasets.

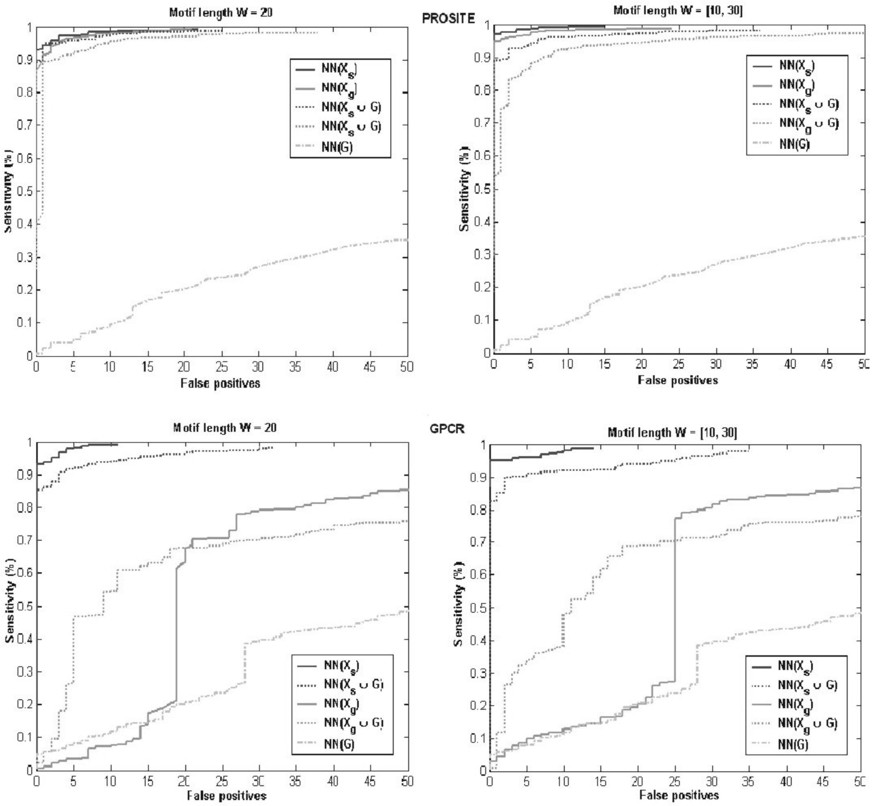

Fig. 1. ROC$_{50}$ curves illustrating the performance of the neural classifier on the two datasets using the five different feature vectors.

Figure 1 displays the ROC$_{50}$ curves obtained after training the five neural classifiers. 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 that lead to the construction of efficient classifiers. In all cases, the neural networks trained by mixed (local and global) 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-gram features alone (case NN(**G**)) do not seem to contain significant discriminant information. The best classification results were obtained with the network NN($\mathbf{X}_s$). This indicates that the class-dependent motifs achieve better allocation among the K families and thus more efficient modeling, in comparison with the class-independent case.

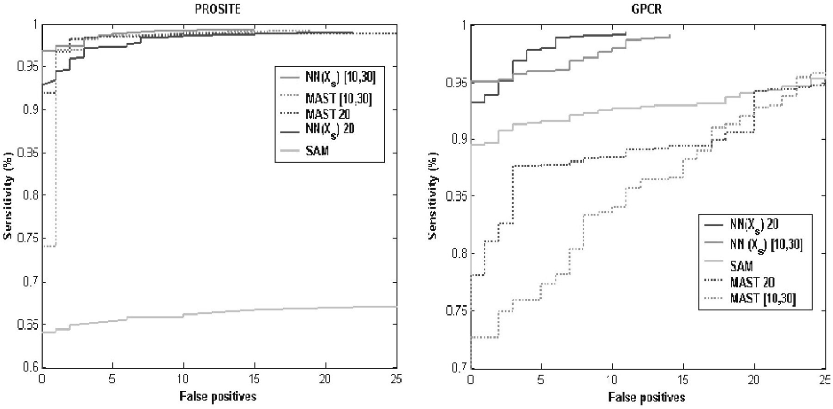

Fig. 2. ROC$_{25}$ curves for the three methods (neural (NN), MAST and SAM) on the two datasets.

During the second series of experiments we have compared the best neural classifier (NN($\mathbf{X}_s$)) with two other protein classification methods, namely the MAST homology detection algorithm [5] and the SAM method based on HMMs [1]. As it has already been discussed, both methods create (indirectly or directly) a probabilistic model-profile for each family and they classify each test sequence into the class with the best score value (minimum E-value). Figure 2 provides comparative results for the two datasets. Five ROC curves are presented until 25 false positives were found (ROC$_{25}$). The performance of the neural classifier and MAST was given by two curves concerning motifs of fixed ($W = 20$) and variable length ($W = [10, 30]$), respectively. In the case of MAST and SAM methods, ROC curves were obtained by setting several E-value thresholds. When the lowest estimated E-value 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 positives). The classification improvement is more clear in the GPCR dataset. 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. A last observation is that, although the MAST approach uses the same groups of motifs, our method seems to offer a more efficient scheme for combining the motif match scores, in comparison with their p-values as suggested by MAST.

4 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. Based on the MEME algorithm, we discover probabilistic motifs in a set of K-class sequences. Two alternative ways have been suggested depending on whether or not the class labels are taken into account. Then, numerical feature vectors are generated by computing the matching score of the sequences to each motif. At the second stage, the extracted feature vectors are used as inputs to a feed-forward neural network trained using a Bayesian Regularization algorithm that provides the class label of a sequence. Experimental results clearly illustrate the superiority of our neural approach in comparison with other probabilistic methods. In addition, we have shown that background features do not provide a useful source of information for the classification task, since they do not lead to performance improvement. Future work is focused on studying alternative methods both in the classification and the motif discovery stage.

References

1. Hughey R. and Krogh A. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, 12(2):95–107, 1996.
2. Wang J.T.L., Ma Q., Shasha D., and Wu C.H. New techniques for extracting features from protein sequences. *IBM: Systems Journal*, 40(2):426–441, 2001.
3. Bréjova B., DiMarco C., Vinař T., Hidalgo S.R., Holguin G., and Patten C. Finding patterns in biological sequences. Project Report for CS798g, University of Waterloo, 2000.
4. Ma Q. and Wang J.T.L. 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.
5. Bailey T.L. and Gribskov M. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, 14:48–54, 1998.
6. Bailey T.L. and Elkan C. 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.
7. MacKay D.J.C. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.
8. Foresse F.D. and Hagan M.T. Gauss-Newton approximation to Bayesian regularization. In *Proceedings of the 1997 International Joint Conference on Neural Network*, pages 1930–1935, 1997.
9. Bishop C.M. *Neural Networks for Pattern Recognition.* Oxford Univ. Press Inc., New York, 1995.