Sequence motifs: highly predictive features of protein function

Asa Ben-Hur[1] and Douglas Brutlag[2]

[1] Department of Genome Sciences, University of Washington
asa@gs.washington.edu
[2] Department of Biochemistry, Stanford University brutlag@stanford.edu

Summary. Protein function prediction, i.e. classification of protein sequences according to their biological function is an important task in bioinformatics. In this chapter we illustrate that the presence of sequence motifs – elements that are conserved across different proteins – are highly discriminative features for predicting the function of a protein. This is in agreement with the biological thinking that considers motifs to be the building blocks of protein sequences. We focus on proteins annotated as enzymes, and show that despite the fact that motif composition is a very high dimensional representation of a sequence, that most classes of enzymes can be classified using a handful of motifs, yielding accurate and interpretable classifiers. The enzyme data falls into a large number of classes; we find that the one-against-the-rest multi-class method works better than the one-against-one method on this data.

1 Introduction

Advances in DNA sequencing are yielding a wealth of sequenced genomes. And yet, understanding the function of the proteins coded by a specific genome is still lagging. The determination of the function of genes and gene products is performed mainly on the basis of sequence similarity (homology) [1]. This leaves the function of a large percentage of genes undetermined: close to 40% of the known human genes do not have a functional classification by sequence similarity [2, 3].

The most commonly used methods for measuring sequence similarity are the Smith-Waterman algorithm [4], and BLAST [5]. These assign a similarity by aligning a pair of sequences. Other commonly used methods measure similarity to a family of proteins: PSI-BLAST, profiles, or HMM methods [5, 6, 7]. Motif methods on the other hand, represent short, highly conserved *regions* of proteins [8, 9, 10]. Sequence motifs often correspond to functional regions of a protein – catalytic sites, binding sites, structural motifs etc [8]. The presence of such protein motifs often reveals important clues to a protein's role even if it is not globally similar to any known protein. The motifs for most catalytic sites and binding sites are conserved over much larger taxonomic distances and evolutionary time than the rest of the sequence. However, a single motif is often not sufficient to determine the function of a protein. The catalytic site or binding site of a protein might be composed of several regions that are not contiguous in sequence, but are close in the folded protein structure (for example in serine proteases). In addition, a motif representing a binding site might be common to several protein families that bind the same substrate. Therefore, a pattern of motifs is required in general to classify a protein into a certain family of proteins. Manually constructed fingerprints are provided by the PRINTS database [11]. We suggest an automatic method for the construction of such fingerprints by representing a protein sequence in a feature space of motif counts, and performing feature selection in this feature space. Our experiments show that motifs are highly predictive of enzyme function; using feature selection we find small sets of motifs that characterize each class of enzymes; classifiers trained on those feature sets have reduced error rates compared to SVM classifiers trained on all the features. Representing protein sequences using a "bag of motifs" representation is analogous to the bag of words representation used in text categorization. This type of approach was suggested in the context of remote homology detection [12] (see also the unpublished manuscript [13]).

Our work should be compared with several other approaches for protein classification. Leslie and co-authors have focused on various flavors of kernels that represent sequences in the space of k-mers, allowing gaps and mismatches; these include the spectrum and mismatch kernels [14, 15]. The k-mers used by these methods are analogous to the discrete motifs used here. k-mers are less flexible than motifs, but can provide a result in cases when a sequence does not contain known motifs. When it comes to remote homology detection, discriminative approaches based on these kernels and kernels based on HMM models of sequence families (Fisher kernels) [16, 17] yield state of the art performance.

An alternative approach is to represent a sequence by a set of high-level descriptors such as amino acid counts (1-mers), predicted secondary structure content, molecular weight, average hydrophobicity, as well as annotations of the sequence that document its cellular location, tissue specificity etc. [18, 19]. These approaches are complementary to sequence-similarity based approaches such as our motif-based approach.

SVMs are typically used for multi-class problems with either the one-against-the-rest or one-against-one methods [20]. The large number of classes in the data considered in this chapter makes the one-against-one method infeasible. Moreover, we find that the accuracy of the one-against-the rest method is better, which we attribute to the large number of classes. Other studies of multi-class classification using SVMs (see [21] and references therein) have not addressed datasets with such a large number of classes.

In this chapter we consider the problem of classifying proteins according to their enzymatic activity using a motif-based representation. In Section 2 we introduce the Enzyme Commission (EC) numbering system used to classify enzymes. In Section 3 we describe sequence motifs and the classification and feature selection methods used in this chapter. Finally, we show results of these methods, illustrating that SVM-based feature selection methods yield accurate low dimensional predictors of enzyme function.

2 Enzyme Classification

Enzymes represent about a third of the proteins in the SwissProt database [22], and have a well established system of annotation. The function of an enzyme is specified by a name given to it by the Enzyme Commission (EC) [23]. The name corresponds to an *EC number*, which is of the form: `n1.n2.n3.n4`, e.g. `1.1.3.13` for alcohol oxidase. The first number is between 1 and 6, and indicates the general type of chemical reaction catalyzed by the enzyme; the main categories are oxidoreductases, transferases, hydrolases, lyases, isomerases and ligases. The remaining numbers have meanings that are particular to each category. Consider for example, the oxidoreductases (EC number starting with 1), which involve reactions in which hydrogen or oxygen atoms or electrons are transferred between molecules. In these enzymes, `n2` specifies the chemical group of the (electron) donor molecule, `n3` specifies the (electron) acceptor, and `n4` specifies the substrate. The EC classification system specifies over 750 enzyme names; a particular protein can have several enzymatic activities. Therefore, at first glance, this is not a standard multi-class problem, since each pattern can have more than one class label; this type of problem is sometimes called a *multi-label* problem [24]. In order to reduce this multi-label problem into a multi-class problem consider the biological scenarios in which an enzyme has multiple functions:

1. The enzyme can catalyze different reactions using the same catalytic site.
2. The enzyme is a multi-enzyme, an enzyme with multiple catalytic functions that are contributed by distinct subunits/domains of the protein [25].

In both cases it is reasonable to consider an enzyme that catalyzes more than one reaction as distinct from enzymes that catalyze only one reaction. This is clear for multi-enzymes; in the other case, a catalytic site that can catalyze more than one reaction might have different sequence characteristics than a catalytic site that only catalyzes one of the reactions. We found that this multi-label problem can be reduced to a regular multi-class problem by considering a group of enzyme that have several activities as a class by itself; for example, there are 22 enzymes that have EC numbers `1.1.1.1` and `1.2.1.1`, and these can be perfectly distinguished from enzymes with the single EC number `1.1.1.1` using a classifier that uses the motif composition of the proteins. When looking at annotations in the SwissProt database we then found that these two groups are indeed recognized as distinct.

3 Methods

We propose to use the motif composition of a protein to define a similarity measure or *kernel* function that can be used with various kernel based classification methods such as Support Vector Machines (SVMs).

3.1 The motif composition kernel

In this chapter we use discrete sequence motifs extracted using the eMOTIF method [9, 10], which is described here briefly. A motif is a simple regular expression specifying the allowed amino acids in each position. Take for example the motif [`as`].`dkf`[`filmv`]..[`filmv`]...`l`[`ast`]. A sequence will match (or contain) the above motif if it has either an `a` or an `s` in some position, followed by any amino acid, then `d`, `k`, `f` and so on, matching until the end of the motif. A group of amino acids in brackets is called a *substitution group.* A formal definition is as follows:

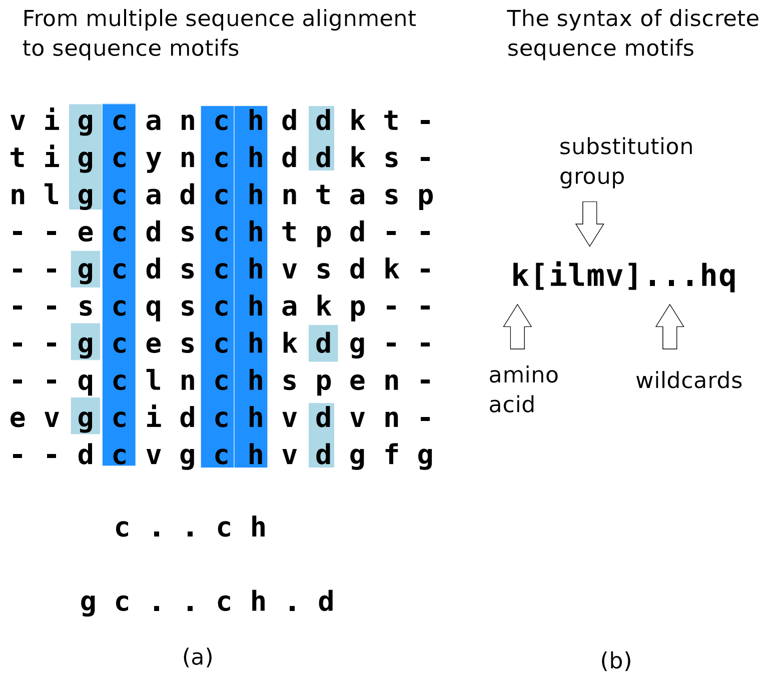
Definition 1. *Denote by $\mathcal{A}$ the alphabet of amino acids. A* substitution group *$\mathcal{S} = \{s_1, \ldots, s_k\}$ is a subset of $\mathcal{A}$, written as $[s_1 \ldots s_k]$. Let $\bar{\mathcal{S}}$ be a set of substitution groups, and let '.' denote the wildcard character.*
A motif *m is a sequence over $\mathcal{A} \cup \bar{\mathcal{S}} \cup \{.\}$.*
A sequence $s = s_1 s_2 \ldots s_{|s|} \in \mathcal{A}^*$ is said to contain *a motif m at position i if for $j = 1, \ldots, |m|$, if $m_j \in \mathcal{A}$ then $s_{i+j-1} = m_j$; if m_j is a substitution group $\mathcal{S}$ then $s_{i+j-1} \in \mathcal{S}$; if m_j is the wildcard character, then s_{i+j-1} can be any character. A sequence s contains a motif m, if s contains m at some position.*

Protein sequence motifs are typically extracted from an ungapped regions (blocks) of a multiple sequence alignment (see Figure 1 for an illustration of the process). Each position in the motif represents the variability in a column of the block. A substitution group such as [`filmv`] denotes the appearance of several amino acids in a particular column in a block. Motifs generated by the eMOTIF method contain only a limited number of substitution groups that reflect chemical and physical properties of amino acids and their tendency to co-occur in multiple sequence alignments. If the pattern of amino acids that appear in a column of a block does not match any substitution group, then the motif contains the wildcard symbol, '.'.

Motifs can often be associated with specific functional sites of a protein: catalytic sites, DNA binding sites, protein-protein interactions sites, small molecule binding sites etc. We give a few examples that illustrate this in the context of the enzyme data.

Example 1. The motif `k[kr][iv]a[iv][iv]g.g.sgl..[ilv][kr]` appears in 19 out of 19 enzymes belonging to the enzyme class `1.14.13.11`. It characterizes a binding site for an FAD molecule.

Example 2. In many cases a binding site motif is not specific to one class of enzymes, but characterizes a similar functional site in several enzyme classes that


From multiple sequence alignment to sequence motifs
The syntax of discrete sequence motifs
substitution group
k[ilmv]...hq
amino acid
wildcards
(a)
(b)

```
v i g c a n c h d d k t -
t i g c y n c h d d k s -
n l g c a d c h n t a s p
- - e c d s c h t p d - -
- - g c d s c h v s d k -
- - s c q s c h a k p - -
- - g c e s c h k d g - -
- - q c l n c h s p e n -
e v g c i d c h v d v n -
- - d c v g c h v d g f g

    c . . c h

  g c . . c h . d
```

Fig. 1. (a) The pipeline from multiple sequence alignment to the construction of sequence motifs: we use discrete motifs that are simple regular expressions that represent the variability in conserved columns in ungapped regions of a multiple sequence alignment. (b) The syntax of a discrete motif: each position in the motif is either an amino acid, a substitution group (a set of amino acids) or the wildcard symbol. Our motif databases use only a limited set of substitution groups; substitution groups are sets of amino acids that tend to substitute for each other in column of a multiple sequence alignment. Considering a limited set of substitution groups helps avoid overfitting.

constitute a broader family of proteins: The motif `dp.f...h.....[ilmv]...[fwy]` has 57 hits in the SwissProt database, and is specific to the EC classes `1.14.18.1`, `1.10.3.1` and `5.3.3.12`. The histidine residue binds one of two copper atoms that act as co-factors for these enzymes.

We are currently undertaking the task of automatically characterizing the function of motifs in our database based on annotations available in the SwissProt database.

A sequence s can be represented in a vector space indexed by a set of motifs $\mathcal{M}$:

$$\Phi(s) = (\phi_m(s))_{m \in \mathcal{M}} \, , \tag{1}$$

where $\phi_m(s)$ is the number of occurrences of the motif m in s. Now define the *motif kernel* as:

$$K(s, s') = \Phi(s) \cdot \Phi(s')\,. \qquad (2)$$

Since in most cases a motif appears only once in a sequence, this kernel essentially counts the number of motifs that are common to both sequences. The computation of the kernel can be performed efficiently by representing the motif database in a TRIE structure: Let m be a motif over the alphabet $\mathcal{A} \cup \bar{\mathcal{S}} \cup \{.\}$. Every prefix of m has a node; let m_1 and m_2 be prefixes of m; there is an edge from m_1 to m_2 if $|m_2| = |m_1| + 1$. The motifs are stored in the leaf nodes of the TRIE. To find all motifs that are contained in a sequence x at a certain position, traverse the TRIE using DFS and record all the leaf nodes encountered during the traversal (see Figure 2 for an illustration).

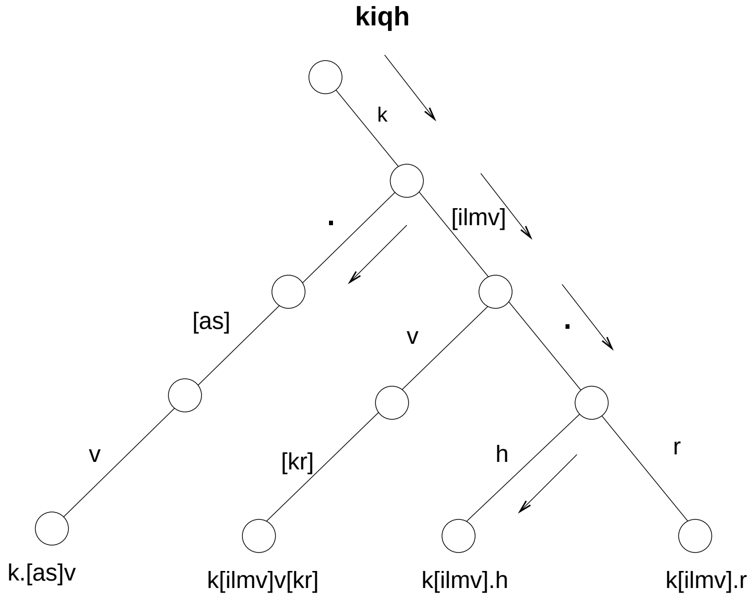


Fig. 2. Motifs are stored in the leaves of a TRIE. The figure shows a TRIE storing the motifs `k.[as]v`, `k[ilmv]v[kr]`, `k[ilmv].h` and `k[ilmv].r`. To find the motif content, the tree is traversed, matching at each position a letter from the sequence with the same letter, a substitution group containing the letter, or the wildcard symbol. Traversing the tree shows that the sequence `kiqh` contains the motif `k[ilmv].h`.

To find all motifs that are contained in a sequence s at any position, this search is started at each position of s. Thus the computation time of the motif content of a sequence is linear in its length. Unlike a standard TRIE searching the motif TRIE has a complexity that depends on the size of the database; this is the result of the presence of wildcards and substitution groups. A trivial upper bound is linear in the size of the database; numerical experiments indicate that the complexity is sub-linear in practice.

The motif kernel is analogous to the "bag of words" representation that is commonly used in information retrieval, where a document is represented as a vector of (weighted) counts of the number of occurrences of each word in the document [26, 27]. In a recent study we found that this "bag of motifs" representation of a protein sequence provides state of the art performance in detecting remote homologs [12]. Like the bag of words representation, our motif composition vector is both high dimensional and sparse: the eBlocks database of motifs [28] used in this work contains close to 500,000 motifs, while a sequence typically contains only a handful of conserved regions. Motifs are often very specific as features: We found that using feature selection we could reduce the number of motifs to a few tens at the most, while keeping the same level of classification accuracy (see Section 4 for details).

3.2 Classification methods

In what follows we assume that our data are vectors $\mathbf{x}_i$ representing the motif content of the input sequences. In this chapter we report results using two classification methods: SVMs and k-Nearest-Neighbors (kNN). A linear SVM is a two-class classifiers with a decision function of the form

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b\,, \tag{3}$$

where $\mathbf{w}$ is a weight vector, and b is a constant, and a pattern $\mathbf{x}$ is classified according to the sign of $f(\mathbf{x})$. The vector $\mathbf{w}$ and the bias, b, are chosen to maximize the margin between the decision surface (hyperplane) and the positive examples on one side, and negative examples on the other side, in the case of linearly separable data; in the case of non-separable data some slack is introduced [29, 20, 30]. As a consequence of the optimization process, the weight vector can be expressed as a weighted sum of the Support Vectors (SV):

$$\mathbf{w} = \sum_{i \in SV} \beta_i \mathbf{x}_i\,. \tag{4}$$

The decision function is now written as:

$$f(\mathbf{x}) = \sum_{i \in SV} \beta_i \mathbf{x}_i \cdot \mathbf{x} + b\,. \tag{5}$$

To extend the usefulness of SVMs to include nonlinear decision functions, and non-vector data one proceeds by mapping the data into a feature space, typically high dimensional, using a map Φ, and then considering a linear SVM in the high dimensional feature space [20, 30]. Since the SVM optimization problem can be expressed in terms of dot products, this This approach is practical if the so called *kernel function*, $K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$, can be computed efficiently. In terms of the kernel function, the decision function is expressed as:

$$f(\mathbf{x}) = \sum_{i \in SV} \beta_i K(\mathbf{x}_i, \mathbf{x}) + b \,. \tag{6}$$

A kNN classifier classifies a pattern according to the class label of the training set patterns that are most similar to it. We use a kNN classifier with a continuous valued decision function that assigns a score for class j defined by:

$$f_j(\mathbf{x}) = \sum_{i \in \mathrm{kNN}_j(\mathbf{x})} K(\mathbf{x}_i, \mathbf{x}) \,, \tag{7}$$

where $\mathrm{kNN}_j(\mathbf{x})$ is the set of k nearest neighbors of $\mathbf{x}$ in class j; a pattern $\mathbf{x}$ is classified to the highest scoring class.

3.3 Feature Scoring and Selection

In order to show that motifs are highly predictive of the class of an enzyme we compute for each motif feature the following statistics. The Positive-Predictive Value (PPV) of a feature is the fraction of the predictions made on the basis of a motif m that are correct, namely

$$\mathrm{ppv}(m) = \frac{\mathrm{count}(m|C)}{\mathrm{count}(m)} \,, \tag{8}$$

where $\mathrm{count}(m)$ ($\mathrm{count}(m|C)$) is the number of occurrences of the motif m (in class C). Note that this is referred to as *precision* in information retrieval. A motif has PPV which is equal to 1 in a class C if it occurs only in proteins from class C. On the other end of the spectrum we consider the sensitivity (or *recall* in information retrieval terms) of a motif m in picking members of a class C:

$$\mathrm{sens}(m) = \frac{\mathrm{count}(m|C)}{|C|} \,, \tag{9}$$

where $|C|$ is the size (number of members) of class C. The motifs in the database we use are often highly redundant in their pattern of occurrence in a group of proteins. Due to this level of redundancy, feature selection methods that are based on ranking individual features, will not yield a small subset of non-redundant motifs.

In this chapter we focus on SVM-based feature selection methods and show their effectiveness. Note that The weight vector of an SVM ($\mathbf{w}$) is a weighted sum of a subset of the motif composition vectors (the support vectors). In most cases the number of support-vectors was rather small when training a classifier to distinguish one enzyme class from all the others, so the weight vector is typically very sparse; discarding features that are not represented in the weight vector would already yield a significant reduction in the number of features, without modifying the decision function. The idea of using the magnitude of the weight vector to perform feature selection is implemented in the RFE method [31], which alternates between training an SVM and discarding a subset of the features with small components of the weight vector. The RFE method requires a halting condition. A halting condition based on cross-validation is too expensive in our case so we we use a halting condition that is based on the observed monotonicity of the the number of support vectors in successive iterations of RFE: initially, most of the features removed are noise; when these are eliminated the data is simpler to describe, requiring less support vectors. At a later stage essential features are removed, making the features insufficient to describe the data, so many data points will be misclassified, making them bounded support vectors. We choose the smallest number of features for which the number of support vectors is minimal.

A related method to RFE is the zero-norm method of Weston et al. [32]. They formulate the feature selection problem as a search for the smallest set of features such that a dataset is still linearly separable (with slack variables added for the case of data that is not linearly separable). In other words, minimizing the zero norm of the weight vector of a linear decision boundary, subject to the constraints that the decision boundary separates the two classes (the zero norm of a vector is its number of nonzero coefficients). They show that this difficult combinatorial problem can be relaxed into a problem that is solved by an algorithm similar to RFE: alternate between training an SVM and multiplying the data by the absolute value of the weight vector (feature i of each pattern is multiplied by $|w_i|$). This is iterated until convergence.

In general, minimizing the zero norm might not be an optimal strategy: the method minimizes the number of variables that separate the two classes, without considering the margin of the separation. However, the data under consideration is discrete, so if a set of motifs separates the data, they do so with large margin. This can explain the good performance obtained with this method on the motif data: our results show that the accuracy of classifiers trained on features selected using the zero-norm method is higher than that of classifiers trained on all the features. For comparison we tested the zero-norm method on the problem of predicting protein function on the basis of gene expression data (we used the data analyzed in [33] and considered the five functional classes shown to be predictable using SVMs). In this case of continuous data in 79 dimensions, SVMs trained on all features outperformed SVMs trained using the zero-norm method (data not shown).

3.4 Multi-class classification

Protein function prediction is a classification problem with a large number of classes (hundreds of classes in the EC classification scheme alone). This poses a computational challenge when using a two-class classifier such as SVM. The two standard approaches for using a two-class classifier for multi-class data are the one-against-one method and one-against-the-rest method [20]. The one-against-the-rest method trains c classifiers, where c is the number of classes in the data, and classifier i is trained on class i against the rest of the data. A test pattern is then classified to the class that receives the highest value of the decision function. The one-against-one method requires training $c(c-1)/2$ classifiers on all pairs of classes; an unseen pattern is tested by all these classifiers and is classified to the class that receives the highest number of votes. In our case this amounts to training 651 * 650/2 = 2,111,575 classifiers, which makes this too computationally intensive. Moreover, the data in Table 3 shows that the one-against-the-rest method works better on the enzyme function prediction task. It can be argued that the large number of "irrelevant" tests performed by the one-against-one method may be the cause of this.

A recent paper by Rifkin and Klautau [21] argues that the one-against-the-rest method should work as well as other multi-class methods, and presents experimental results to support their arguments, including a critical analysis of previous studies. The datasets they considered are UCI datasets that have a small number of classes compared to the enzyme data. Yeang *et al.* [34] studied a gene expression dataset with 14 classes corresponding to patients with various types of cancer; they also obtained higher accuracy with one-against-the-rest than with one-against-one. Our results further support their findings, considering the issue of multi-class for data with a much larger number of classes.

3.5 Assessing classifier performance

In some of our analyses we will consider two-class problems that are highly unbalanced, i.e. one class is much larger than the other; in such cases the standard error rate is not a good measure of classifier performance. Therefore we consider two alternative metrics for assessing the performance of a classifier: the area under the Receiver Operator Characteristic (ROC) curve [35], and the balanced success rate. The balanced success rate is:

$$1 - \sum_i P(err|C_i)\,, \tag{10}$$

where $P(err|C)$ is a shorthand for a classifier's error on patterns that belong to class C. The ROC curve describes the trade-off between sensitivity and specificity; it is a plot of the true positive rate as a function of the false positive rate for varying classification thresholds [35]. The area under the ROC curve (AUC) is commonly used to summarize the ROC curve. The AUC is a measure of how well the classifier works at *ranking* patterns: it quantifies the extent to which positive examples are ranked above the negative examples. The AUC is a useful metric for assessing a classifier used in the context of protein classification: a user will typically be interested in the most promising patterns, i.e. patterns that are most likely to belong to a class of interest. The AUC score however, can be problematic for highly unbalanced data: one can obtain a very high AUC even when the ranking produced by the classifier is almost useless from the point of view of the user if for example 500 out of 30,000 patterns from the negative class are ranked above the real members of the class. Therefore we consider the ROC50 curve, which counts true positives only up to the first 50 false positives [15]. A classifier that correctly classifies all the data has an ROC50 score (AUC50) equal to 1, while if the top 50 values of the decision function are false positives, the AUC50 is 0.

4 Results

We extracted protein sequences annotated with EC numbers from the SwissProt database Release 40.0 [22]. EC numbers were taken from the description lines; we removed sequence fragments, and sequences where the assigned EC number was designated as "putative" or assigned by homology. Sequences with an incompletely specified EC number were discarded as well. Enzyme classes with a small number of representatives (less than 10) were not considered in our analysis. The resulting dataset has 31117 enzymes in 651 classes. In some cases we focus on oxidoreductases – enzymes that have an EC number starting with 1. The statistics of the two datasets are summarized in Table 1.

	number of sequences	number of classes	number of motifs
Oxidoreductases	5911	129	59783
All enzymes	31117	651	178450

Table 1. The enzyme sequence data; oxidoreductases are enzymes with EC number that starts with 1.

In order to illustrate that the eBLOCKS database contains many motifs that are predictive of enzyme function we consider their positive predictive value (PPV) and sensitivity in picking members of each enzyme class. Out of the 651 enzyme classes, 600 classes had a motif that was perfectly specific to that class, i.e. had a PPV equal to 1. To see the sensitivity of such perfectly specific motifs, for each class we find the set of motifs with maximum PPV and find the one with maximum sensitivity. The distribution of the sensitivity of these motifs is shown in Figure 3. We observe that 89 enzyme classes have a motif that covers all its proteins and has no hits outside the class. In general we do not expect to find motifs that cover all members of an enzyme class, since it might be heterogenous, composed of several clusters in sequence space. We considered aggregating motifs with perfect PPV to form predictors of EC classes, but only a limited number of classes had sets of motifs that cover the entire class, so a less strict form of feature selection is required.

Next, we report experiments using motifs as features for predicting the EC number of an enzyme. In these experiments we used the `PyML` package (see section 4.3). We used a linear kernel in motif space in view of the high dimensionality. SVM performance was not affected by normalizing the patterns to unit vectors, but was critical for the kNN classifier. On other datasets we observed that classification accuracy did not vary much when changing the

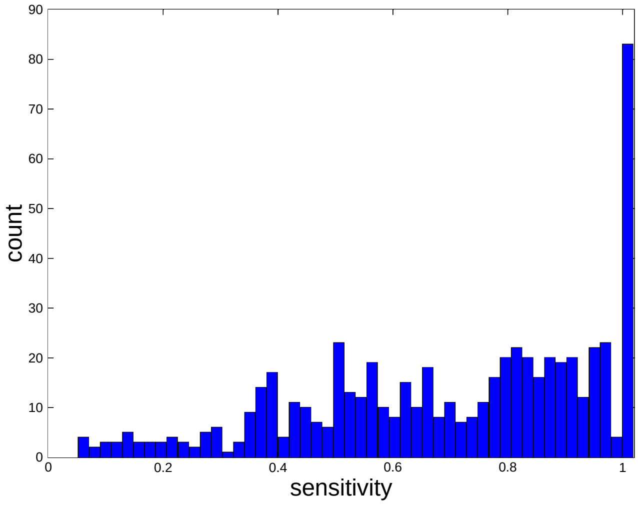


Fig. 3. The motif database contains motifs whose occurrences are highly correlated with the EC classes. For each class we computed the highest PPV, and the highest sensitivity of a motif with that value of PPV. 600 out of 651 classes had a motif with PPV equal to 1. The distribution of the sensitivity of these motifs is shown. There are 89 EC classes that have a "perfect motif": a motif that cover all enzymes of the class, and appears only in that class, i.e. the class can be predicted on the basis of a single motif.

SVM soft margin constant, so we kept it at its default value. In order to reduce the computational effort in comparing multiple approaches we focus on enzymes whose EC number starts with 1 (oxidoreductases); this yields a dataset with 129 classes and 5911 enzymes. We compare classifiers trained on all 129 one-against-the-rest problems; Figure 4 shows the number of classes which have a given level of performance for two metrics: AUC50 (area under ROC50 curve), and the balanced success rate. The results are for the following methods: SVM and kNN trained on all features and SVM with RFE and zero-norm feature selection.

The kNN classifier works well, outperforming the SVM on the balanced success rate. The fact that kNN works so well despite the high dimensionality of the data is the result of the presence of highly informative features, coupled with the sparsity of the data and the discreteness of the representation. In another set of experiments we compared kNN classifiers trained on features derived from scores with respect to Position Specific Scoring Matrices (PSSMs). The PSSM score reflects how well the best hit in the sequence fits the model of the PSSM, and not just the presence or absence of a motif. Scores that were not statistically significant were set to zero, so the representation

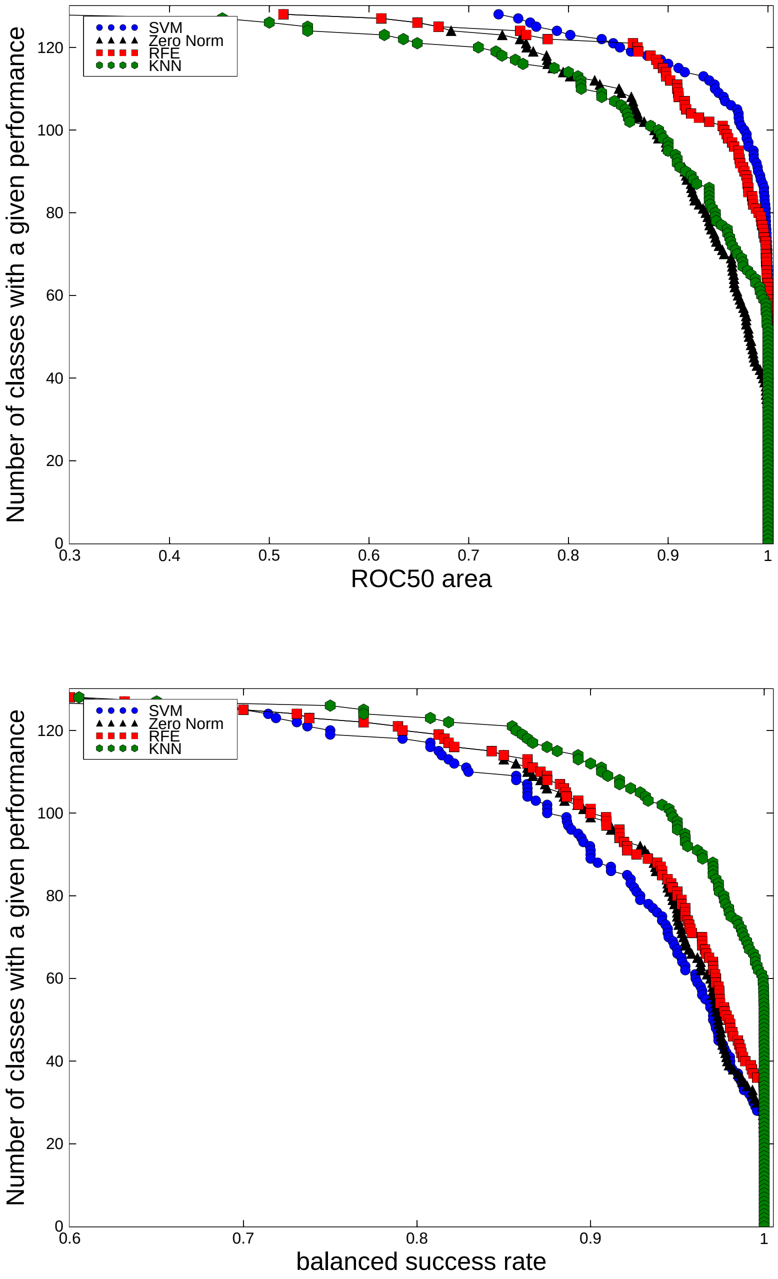


Fig. 4. Performance in 5 fold cross-validation for the 129 enzyme classes in the oxidoreductase data. The number of classes with a given level of performance is plotted for the various methods: Top: Area under ROC50 curve. Bottom: balanced success rate.

remained sparse. In this case SVMs performed significantly better than kNN (data not shown). The SVM's better performance with respect to the AUC50 metric can be attributed to the fact that SVM training explicitly optimizes the decision function while for the kNN classifier a continuous valued decision function is an after-thought, rather than an integral part of the design.

Feature selection using the RFE and zero-norm method lead to an improvement over the SVM method with respect to the balanced success rate. An intuitive explanation is provided in Section 3.3. The performance with respect to the AUC50, however, deteriorated as a result of feature selection. This can be understood by the fact that the feature selection process is governed by the objective of the underlying SVM, which is good performance under the balanced success rate (since we account for the class imbalance by introducing a misclassification cost that is inversely proportional to the class size). The improvement in the balanced success rate comes at the expense of the AUC50. We also note that the RFE method performed better than the zero-norm method. We attribute this to the difference in the halting conditions used, and the resulting number of features — the zero-norm method yielded 10 features on average, whereas RFE yielded 77 features on average over the 129 classes. Using a polynomial kernel after the feature selection stage using RFE yielded worse results than using a linear kernel (data not shown). All the differences seen in Figure 4 are statistically significant other than the difference between kNN and the zero-norm method in the AUC50 metric, and between the zero-norm method and RFE in the balanced success rate metric.

We also ran an experiment with a filter method that ranks a feature m according to abs $\left(\frac{\text{count}(m|C)}{|C|} - \frac{\text{count}(m|\bar{C})}{|\bar{C}|}\right)$, where $\bar{C}$ is the set of patterns outside of class C. Features whose score was less two standard deviations above the average score obtained on randomly labeled datasets were discarded. Due to the high redundancy of the motif features, the method is not successful in reducing the dimensionality in a significant way – over 5000 features on average over the 129 enzyme classes were chosen, and the performance was almost identical to an SVM trained on all the features.

Method	success rate	balanced success rate
kNN-normalized	0.94	0.92
SVM-one-against-rest	0.96	0.94
BLAST	0.96	0.93

Table 2. Success rate in Multi-class classification of the enzyme data, estimated by 5-fold CV. The differences are significant: the standard deviation in 10 repeats of the experiment was 0.002 and below. All methods other than the BLAST method use motifs; a normalized kernel is generated by dividing each motif composition vector by its L_1 norm.

Number of classes	one-against-rest	one-against-one
10	0.99	0.97
20	0.98	0.96
40	0.98	0.96
60	0.98	0.94
80	0.98	0.95

Table 3. Success rate in multi-class classification measured using 5-fold cross-validation for the SVM-based methods when varying the number of classes.

4.1 Enzymes with multiple functions

In our analysis we considered a set of enzymes with multiple functionalities as a unique class; for example enzymes with EC numbers 1.1.1.1 and 1.2.1.1 were considered a distinct class. The data contains 27 classes with multiple functionalities. In order to quantify the degree of success in predicting a class with multiple functionalities, for each multiple-functionality class we assessed the accuracy of a classifier trained to distinguish between the multiple-functionality class and the classes with which it shares a function. The average balanced success rate in this experiment was 0.95, and the average AUC50 was 0.95. These results support our reduction of the multi-label problem to a multi-class problem.

4.2 Multi-class classification

The results of multi-class experiments appear in Table 2. The BLAST method assigns the class label according to the class of the enzyme with which an input sequence has the largest BLAST score (a nearest neighbor BLAST). The one-against-the-rest SVM motif method worked as well as the BLAST-based method and better than the nearest-neighbor motif method. When using a nearest neighbor method normalization of the kernel was critical: normalizing the kernel by dividing each pattern by its L_1-norm improved the results significantly. Most of the misclassifications in the case of the non-normalized kernel occurred by classifying a pattern into a class of large proteins that contain many motifs; such classes "attract" members of other classes. A comparison of the SVM-based methods is provided in Table 3 for an increasing number of classes. It shows an advantage for the one-against-the-rest method that increases as the number of classes increases, that may be explained by the large number of "irrelevant" comparisons made by the one-against-one method.

4.3 Data and Software

A license for the motif database used in this work is freely available for academic users; see `http://motif.stanford.edu`. The machine learning experiments were performed using `PyML`, which is an object oriented environment for performing machine learning experiments. PyML is available at http://pyml.sourceforge.net.

5 Discussion

Several databases of conserved regions reviewed in the introduction are constructed by experts that use known annotations to group protein sequences that are then modeled by motifs, profiles, or HMMs, namely PROSITE, BLOCKS+ and Pfam [8, 36, 7]. The use of such patterns as features to train classifiers that predict protein function can lead to biased results since knowledge about function is often incorporated by the experts constructing these databases. The eBLOCKs database used in this study on the other hand, is constructed in an unsupervised way by aligning clusters of similar sequences in SwissProt, so our results are free from such bias. We are currently developing an motif database that is constructed on the basis of enzyme annotations rather than on the basis of clustering by sequence similarity. In this case we cannot test on sequences that were used in creating the motifs since the labels are used in motif construction. To avoid bias in the test results we are are evaluating the method by constructing several versions of the database, each on a different training set, allowing the use of cross-validation. Such an evaluation is not necessary in this case since no information about the labels is used in creating the motifs.

Although results of classification using motifs did not offer a significant advantage over BLAST in terms of accuracy, we have shown several examples where motifs offer greater interpretability. Since manually curating the function of motifs is infeasible, we are working on automating the process to produce annotations for as many motifs as possible using sequence annotations available in the SwissProt database. Performing a motif search on a motif set selected by feature selection is also more efficient than performing a BLAST search against an enzyme database: computing the motif content of a sequence is sub-linear in the number of motifs in the database since only a small fraction of the nodes are traversed in the TRIE search method, compared with a BLAST search whose complexity is linear in the number of sequences in the database (although here one can perform feature selection as well to find a smaller subset of non-redundant sequences that provide adequate representation of the variability in the database).

Despite the higher accuracy of the SVM-based multi-class methods over kNN, the SVM-based methods are expensive to train in view of the large number of classes. The sequence data is very sparse in sequence space: most classes are well separated. One can take advantage of this property in many ways. We performed experiments using a method that uses one-class SVMs to filter a small number of "candidate" classes, and then deciding among that smaller number of classes using a one-against-one approach. Since the number of classes that are not well separated is small, this resulted in the need to train only a small number of classifiers, with accuracy similar to the one-against-the-rest method.

In this work we represented a motif by a simple regular expression. Position Specific Scoring Matrices (PSSMs) offer greater flexibility in describing a pattern of conservation; therefore it is of interest to see if better performance can be obtained using PSSMs, since relevant information may be lost in the process of extracting discrete motifs (note however that the eMOTIF method generates multiple motifs out of a conserved sequence block, compensating for the loss in expressive power of a single motif). Preliminary results indicate a slight improvement in accuracy for PSSM-based classifiers. On the other hand, the advantage of using discrete motifs over PSSMs is the efficient search methods for computing motif hits. Using a hybrid approach – choosing PSSMs over motifs for classes that are not as well described using discrete motifs, could offer the best of both worlds.

6 Conclusion

In this chapter we have illustrated that the motif composition of a sequence is a very “clean” representation of a protein; since a motif compactly captures the features from a sequence that are essential for its function, we could obtain accurate classifiers for predicting enzyme function using a small number of motifs. We plan to develop the motif-based classifiers as a useful resource that can help in understanding protein function.

References

1. F.S. Domingues and T. Lengauer. Protein function from sequence and structure. *Applied Bioinformatics*, 2(1):3–12, 2003.
2. E.S. Lander, L.M. Linton, and B. Birren. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.
3. J.C. Venter, M.D. Adams, E.W. Myers, and P.W. Li. The sequence of the human genome. *Science*, 2901(16):1304–1351, 2001.
4. T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
5. S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.
6. M. Gribskov, A.D. McLachlan, and D. Eisenberg. Profile analysis: Dectection of distantly related proteins. *Proc. Natl. Acad. Sci. USA*, 84:4355–4358, 1987.
7. E.L. Sonnhammer, S.R. Eddy, and E. Birney. Pfam: multiple sequence alignments and hmm-profiles of protein domains. *Nucleic Acids Research*, 26(1):320–322, 1998.
8. L. Falquet, M. Pagni, P. Bucher, N. Hulo, C.J. Sigrist, K. Hofmann, and A. Bairoch. The PROSITE database, its status in 2002. *Nucliec Acids Research*, 30:235–238, 2002.

9. C.G. Nevill-Manning, T.D. Wu, and D.L. Brutlag. Highly specific protein sequence motifs for genome analysis. *Proc. Natl. Acad. Sci. USA*, 95(11):5865–5871, 1998.
10. J.Y. Huang and D.L. Brutlag. The eMOTIF database. *Nucleic Acids Research*, 29(1):202–204, 2001.
11. T.K. Attwood, M. Blythe, D.R. Flower, A. Gaulton, J.E. Mabey, N. Maudling, L. McGregor, A. Mitchell, G. Moulton, K. Paine, and P. Scordis. PRINTS and PRINTS-S shed light on protein ancestry. *Nucleic Acids Research*, 30(1):239–241, 2002.
12. A. Ben-Hur and D. Brutlag. Remote homology detection: A motif based approach. In *Proceedings, eleventh international conference on intelligent systems for molecular biology*, volume 19 suppl 1 of *Bioinformatics*, pages i26–i33, 2003.
13. B. Logan, P. Moreno, B. Suzek, Z. Weng, and S. Kasif. A study of remote homology detection. Technical report, Cambridge Research Laboratory, June 2001. http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-2001-5.html.
14. C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 564–575. World Scientific, 2002.
15. C. Leslie, E. Eskin, J. Weston, and W. Stafford Noble. Mismatch string kernels for svm protein classification. In *Advances in Neural Information Processing Systems*, 2002.
16. T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, 1999.
17. T. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158, Menlo Park, CA, 1999. AAAI Press.
18. U. Syed and G. Yona. Using a mixture of probabilistic decision trees for direct prediction of protein function. In *RECOMB*, 2003.
19. M. des Jardins, P.D. Karp, M. Krummenacker, T.J. Lee, and C.A. Ouzounis. Prediction of enzyme classification from protein sequence without the use of sequence similarity. In *Intelligent Systems for Molecular Biology*, pages 92–99, 1997.
20. B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* MIT Press, Cambridge, MA, 2002.
21. R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
22. C. O'Donovan, M.J. Martin, A. Gattiker, E. Gasteiger, A. Bairoch A., and R. Apweiler. High-quality protein knowledge resource: SWISS-PROT and TrEMBL. *Brief. Bioinform.*, 3:275–284, 2002.
23. Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). *Enzyme Nomenclature. Recommendations 1992.* Academic Press, 1992.
24. A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, 2001.
25. A.D. McNaught and A. Wilkinson. *IUPAC Compendium of Chemical Terminology.* Royal Society of Chemistry, Cambridge, UK, 1997.
26. T. Joachims. *Learning to Classify Text using Support Vector Machines.* Kluwer Academic Publishers, 2002.

27. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, pages 137–142, Berlin, 1998. Springer.
28. Q. Su, S. Saxonov, and D.L. Brutlag. eBLOCKS: an automated database of protein conserved regions maximizing sensitivity and specificity, 2003.
29. B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.
30. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge UP, 2000.
31. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.
32. J. Weston, A. Elisseeff, M. Tipping, and B. Schölkopf. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3(7-8):1439–1461, 2003.
33. Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terence S. Furey, Jr. Manuel Ares, and David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA*, 97:262–267, 2000.
34. C.H. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T. Golub. Molecular classification of multiple tumor types. In *Proceedings, eleventh international conference on intelligent systems for molecular biology*, volume 17 suppl 1 of *Bioinformatics*, pages S316–S322, 2001.
35. J.P. Egan. *Signal detection theory and ROC analysis.* Series in Cognition and Perception. Academic Press, New York, 1975.
36. S. Henikoff, J.G. Henikoff, and S. Pietrokovski. Blocks+: A non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinformatics*, 15(6):471–479, 1999.