TEL-AVIV UNIVERSITY
RAYMOND AND BEVERLY SACKLER
FACULTY OF EXACT SCIENCES

Motif Extraction and Protein Classification

Thesis submitted in partial fulfillment of the requirements
for the degree of M.Sc.
in
Tel -Aviv University
The Department of Computer Science

by

Vered Kunik

The research work for this thesis
has been carried out at Tel-Aviv University
under the supervision of Prof. David Horn and Prof. Eytan Ruppin

April 2006

To my beloved grandparents

Abstract

One of the central problems in computational biology is the classification of proteins into functional classes given their amino acid sequence. In this thesis, we address the challenge of predicting the function of enzymes by distilling biologically meaningful motifs from their amino acid sequence. The motifs are obtained by applying a novel unsupervised motif extraction algorithm to a set of the oxidoreductases class of enzymes extracted from the Swiss-Prot database. This motif extraction (MEX) algorithm is data driven, using the statistical information present in the raw sequential data to identify significant segments that are not necessarily over-represented in the data.

The application of MEX to the dataset of oxidoreductases enzymes yielded motifs of various lengths. The space spanned by the extracted MEX motifs of length 6 and longer, serves as the basis for functional classification of the oxidoreductases enzymes by an SVM classifier.

The performance of the SVM classification based on our MEX motifs is compared to those of two other methods:

- SVM-Prot, an SVM classification method based on the analysis of physical-chemical properties of a protein generated from its sequence of amino acids
- SVM-pairwise, an SVM classification based on the space spanned by the p-values obtained by applying the Smith-Waterman pairwise sequence similarity algorithm to all pairs of enzymes in the training set

The classification tasks carried out in our work were repeated on matched random partitions of the data into a train and a test set to accumulate statistics. The differences in classification performance obtained by the various methods on the matched sets were examined statistically. The classification results based on our method surpassed that of the two other methods.

Our findings indicate that the motifs extracted by MEX form an excellent basis for classifying the oxidoreductases enzymes into small classes known to have different functional roles. Thus, MEX motifs support a successful sequence-to-function classification. Moreover, our work provides additional insights regarding the existence of short peptides that hold biological information regarding the functionality of enzymes, supporting the usage of sequence motifs for sequence analysis.

Acknowledgments

First and foremost I would like to thank my supervisors, Prof. David Horn and Prof. Eytan Ruppin, for their guidance and support throughout my thesis work. I have learned tremendously from their brilliance, high standards and passion for knowledge and science. I feel privileged to have had the opportunity to work closely and learn so much from them. Apart from my supervisors, I would like to express my gratitude to Zach Solan whom without this thesis topic would not exist. I owe him for both his professional and moral support through very long nights of work. I would like to thank Ben Sandbank for fruitful discussions, helpful insights and for being a friend. Furthermore, I would like to thank Eddie Aronovich for his patience, good will and for supplying the required computing resources.

Second, I would like to thank the members of the lab: Zach, Roy, Assaf, Yaron, Liat and Yasmine for their help, interesting conversations and insights, friendship and most of all for distracting me now and then in an attempt to try and keep me sane. Lots of thanks and appreciation to Michal Finkelman-Reuven for her constant care and help. I thank my lovely friends for being there for me and especially for their will to keep on considering me as their friend despite my constant lack of time. I could never do without you.

Special thanks to Hillel for granting me the peace and quiet of his home during the final stages of writing this work.

Last, but not least, I am eternally grateful to my parents for sticking to me through one of the hardest and most challenging times I experienced in my life. I thank you for allowing me to fulfill my dream. To my uncle, Roman, for allowing me to be a part of his very special life and for being an inspiration. To my grandparents, who are no longer among the living, for setting an example and for being a ray of light in my life. I especially thank my sister for her love, devotion and for Tom - you mean the world to me.

List of Figures

Contents

1 Introduction

In this work, we explore the ability to predict the function of enzymes from their sequence of amino acids by applying a novel unsupervised motif extraction (MEX) algorithm, coupled with a machine learning classification method. We construct a machine that learns the motif composition of enzyme sequences, according to which it generates the prediction about the function of the enzymes. We first introduce the incentive to our work (section 1.1). We then provide (section 1.2) a review of related work in the field of protein function prediction from raw sequential data, showing the novelty and superiority of our method (section 1.3). Finally, we overview the content of the following chapters.

1.1 Proteins function prediction

The complex functions of a living cell are carried out through the synchronous activity of many genes and gene products, i.e., proteins. The ability to determine the function of the proteins encoded in the genome is one of the fundamental elements in understanding biological processes. Despite the rapid development in experimental techniques, such as DNA microarrays [24, 20], yeast two-hybrid system [7], RNA interference (RNAi) [12, 18] and many others, experimental characterization of proteins for the purpose of elucidating their function lags far behind the availability of new sequences. This rapid growth in available biological sequence information creates new opportunities for studying the function of proteins and mechanisms underlying complex biological processes. Deciphering these enormous amounts of data into biological information is a challenging task.

One of the central problems in computational biology is the classification of proteins into functional classes given their amino acid sequence. The earliest and most straightforward computational method for assigning function to a protein is based on the detection of homologs with known function. Homologous proteins are proteins derived from a common ancestral sequence [13]. They have a similar three dimensional (3D) structure and are likely to perform a similar function [30], at least at the molecular level. This is the basis of homology-based function prediction, in which one infers the function of a protein by extrapolating the knowledge from its experimentally characterized homologs.

Most computational methods can detect homology when the protein at hand shares a high level of sequence similarity to other proteins. In some cases, the sequence of an unknown protein is too distantly related to any protein of known function or structure to detect its resemblance by using sequence similarity methods. Furthermore, homologous sequences may share similarity only in a sub-region of the sequence. Therefore, detecting very subtle sequence similarities, known as patterns or motifs, is of high importance for predicting the function of proteins.

Sequence motifs arise due to particular requirements on the structure and amino acid composition of specific regions of a protein which may be important, for example, for their binding properties or for their enzymatic activity. These requirements impose very tight constraints on the evolution of those limited (in size) but highly important portions of a protein sequence. Appropriately chosen sequence motifs may be expected to reduce noise in the data and indicate structural and active regions of the protein, hence improving predictability of its function.

The use of protein sequence motifs to determine functions of proteins is rapidly becoming one of the essential tools for sequence analysis. Our work fits well into the rapidly growing efforts made to annotate newly sequenced genomes using computational methods.

1.2 Related work in protein function prediction

It is commonly accepted that high sequence similarity guarantees functional similarity of proteins. A contemporary analysis of enzyme function conservation by Tian and Skolnick [31] suggests that 40% pairwise sequence identity can be used as a threshold to certify functional similarity (i.e., the first three digits of the Enzyme Commission number (EC) are identical, see section 2.1 for a detailed explanation). The golden standard of pairwise sequence similarity methods is the Smith-Waterman algorithm [26] and its heuristic, faster version BLAST [1]. Using pairwise sequence similarity scores, obtained by applying the Smith-Waterman algorithm, combined with the Support Vector Machine (SVM) classification method [25, 14], Liao and Noble [19] have argued that their **SVM-pairwise** method obtains a siginificantly improved remote homology detection relative to existing state-of-the-art algorithms such as the SVM-Fisher method [17] (i.e., an SVM trained on features extracted from Hidden Markov Models).

Another sequence-based approach to the task of protein classification is a method based on the general characteristics of the sequence, such as the number of specific amino-acids within it, as described in [9]. A recent variation of this approach represents the amino-acid sequence as a sequence of physical-chemical features [5, 6], such as hydrophobicity, normalized Van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility. Cai et al. [5, 6] have applied SVM to these feature vectors and reported that their **SVM-Prot** technique reaches a high degree of accuracy at the subclass level (i.e., two digits of the EC number hierarchy), on various enzyme subclasses.

An alternative to the straightforward sequence similarity approach is the usage of motifs.

A protein can then be represented as a bag of motifs [2] (i.e., neglecting their particular order on the linear sequence), or a vector in a space spanned by these motifs. A recent work by Ben-Hur and Brutlag [3], based on the **eMOTIF** approach [15, 23], led to very good results. They based their analysis on ∼60,000 regular expression eMOTIFs, and by using an appropriate feature selection method [33] they obtained success rates well over 90% for a variety of classifiers.

1.3 Our Approach to protein function prediction

In this thesis we study the problem of enzymes function prediction from their sequence of amino acids. Our approach to protein function classification is motif based. Appropriately chosen sequence motifs may be expected to reduce noise in the data and indicate structural and active regions of the protein, hence improving the predictability of its function. We attempt to identify biologically meaningful sequence motifs that reflect various functional aspects of enzymes and exploit the space spanned by these motifs as the basis for functional classification by an SVM classifier. The novelty of our approach is in the employed motif extraction algorithm (MEX). Conventional approaches [11, 15] construct motifs from multiple sequence alignments of related sequences and express them as position specific weight matrices or regular expressions. Other methods use hidden Markov models and Bayesian networks [17],hence are supervised to some extent. MEX extracts motifs from proteins sequential data in an **unsupervised** manner, without requiring over-representation of its amino-acid motifs in the data set. In contradistinction to position-specific weight matrices or regular expressions, MEX motifs are explicit strings.

Due to the fact that the Smith-Waterman algorithm is the most established pairwise sequence similarity method, we have chosen to replicate the experimental procedure described in [19] and use the obtained results. For the SVM-Prot method, the results are obtained from their published results [5, 6]. The classifications performance based on MEX motifs, outperform that of the two other SVM based methods (i.e., SVM-Prot and SVM-pairwise).

1.4 Outline

The outline of this thesis is as follows. In chapter 3 we provide information about the datasets we have used in our work, the preprocessing procedure for obtaining the required representations for the SVM classifier (i.e., the vectorization step) and information about the various classification tasks.

Chapter 4 is intended to provide a thorough explanation of the motif extraction (MEX) algorithm. We then provide an explanation of the Smith-Waterman local alignment algorithm employed in this work.

Finally, we briefly describe the SVM algorithm we have chosen to use as our machine learning classification tool. Chapter 5 contains the results of performing the classification tasks and the comparison between the results obtained by the different methods. Finally, chapter 6 contains the concluding remarks as well as a discussion regarding missing aspects in our work, presenting the challenges awaiting to be addressed.

2 Learning the function of enzymes

This chapter describes how is the function of an enzyme defined throughout this work. Furthermore, it describes the preprocessing procedures applied to the dataset for SVM classification.

2.1 Determining the function of an enzyme

The function of an enzyme is specified by a name and a number given to it by the Enzyme Commission (EC) classification hierarchy [32]. The EC number consists of four numbers, n_1:n_2:n_3:n_4, corresponding to the four levels of classification hierarchy. The first level of the classification hierarchy (n_1) includes six classes: Oxidoreductases, Transferases, Hydrolases, Lyases, Isomerases and Ligases. The remaining levels have divisions that are unique to each of the six classes.

The oxidoreductases class of enzymes, discussed in this work, catalyze the transfer of electrons from one molecule (i.e., the electron donor) to another molecule (i.e., the electron acceptor). For this class, n_1=1, n_2 (subclass) specifies the electron donors, n_3 (sub-subclass) specifies the electron acceptors and n_4 indicates the enzyme's substrate.

2.2 Data preprocessing for SVM classification

Each of the methods presented in our work (i.e., SVM classification based on MEX motifs, SVM-pairwise and SVM-Prot) consist of two steps:

- converting the given set of enzyme sequences into vectors in some feature space
- training the SVM classifier on the vectorized enzymes

The methods differ only in the vectorization step, i.e., the SVM training and testing procedure is identical. The following subsections describe the process of converting the enzyme sequences into the vectors the SVM classifier is given as an input.

2.2.1 MEX vectorization step

The oxidoreductases sequences are converted to MEX motifs vectors by representing each sequence as a ”bag of motifs”, neglecting the motifs particular order on the linear sequence. Hence, each enzyme sequence is represented by the MEX motifs the enzyme holds. Figure 2.1 illustrates MEX’s vectorization process.

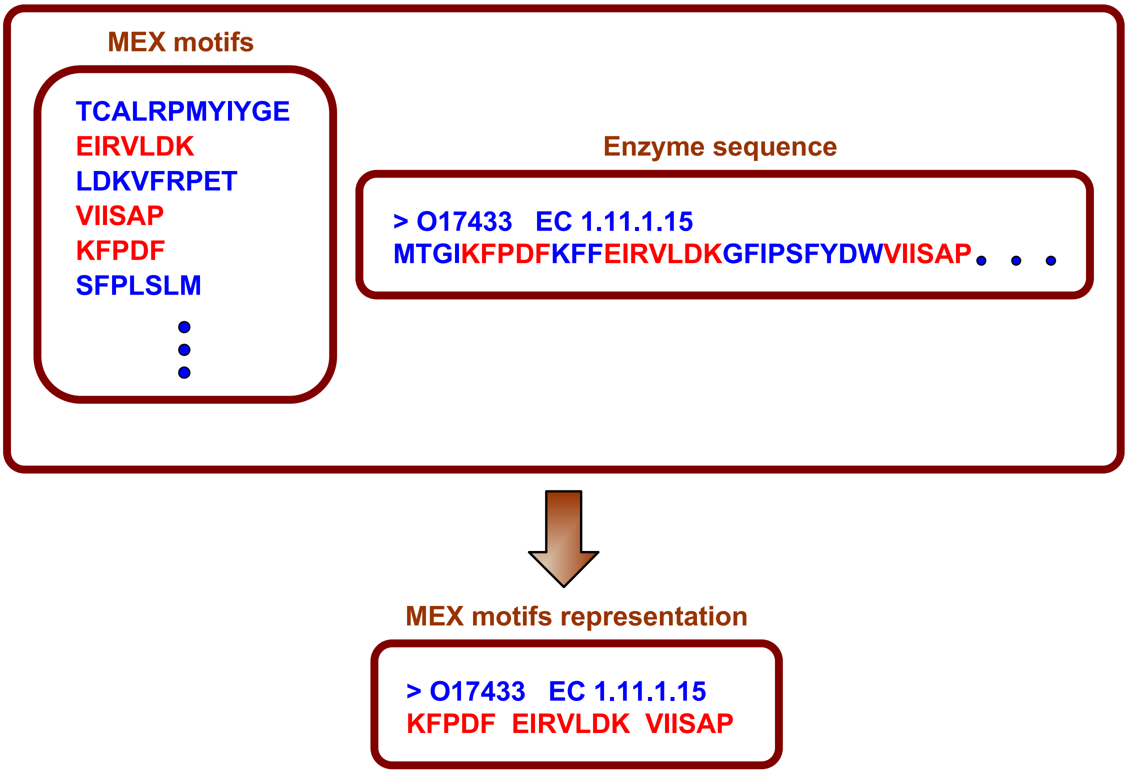


Figure 2.1: Representation of an enzyme sequence by its MEX motifs content. The left upper pane shows the list of obtained MEX motifs. The red motifs are motifs contained on the enzyme sequence appearing in the right upper pane. The lower pane shows the representation of the enzyme by its MEX motifs content.

2.2.2 Smith-Waterman vectorization step

The SVM-pairwise method, utilizes the Smith-Waterman algorithm [26] to perform a one-versus-all sequence similarity comparison. The Smith-Waterman algorithm has been applied to the oxidoreductases dataset. The ariadne tool [21] has been used (available online at http://www.well.ox.ac.uk/ariadne) in order to obtain the p-values distances matrix, SWM, defining the feature space of the SVM classifier. A minimal p-value threshold of 10^{-6} was imposed to allow usage of p-values logarithm. The SWM matrix is normalized so that each vector has the length of 1 in the feature space, i.e., $K(X,Y) = \frac{X \cdot Y}{\sqrt{(X \cdot X)(Y \cdot Y)}}$. Let us denote the obtained normalized distances matrix as SWD. The obtained SWD matrix is given as an input to the SVM classifier.

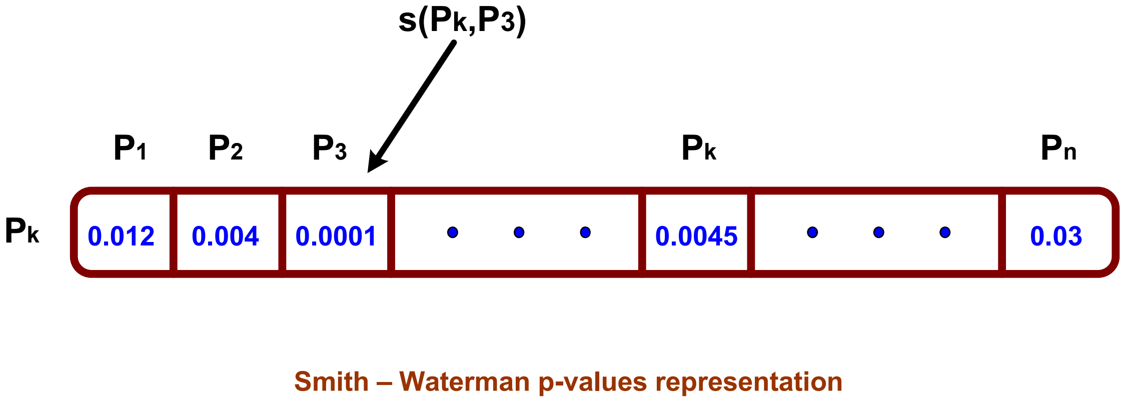


Figure 2.2: Representation of an enzyme as a vector of pairwise similarity scores. The enzyme P_k is represented as a vector if scores. The score function $S(\cdot,\cdot)$ is computed using the Smith-Waterman local alignment algorithm. The vectorization set P_1, P_2, P_3, ..., P_k, ..., P_n is the set of enzymes that appear in the training set.

2.2.3 SVM-Prot vectorization step

The SVM-Prot method represents each enzyme sequence by a specific feature vector assembled from encoded representations of tabulated residue properties: amino acid composition, hydrophobicity, normalized Van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility. For each residue in the sequence, Three descriptors, composition (C), transition (T) and distribution (D), are used to describe global composition of each of these properties [10] (see [5, 6] for further details). C is the number of amino acids of a particular property (for instance: hydrophobicity) divided by the total number of amino acids. T characterizes the percent frequency with which amino acids of a particular property is followed by amino acids of a different property. D measures the chain length within which the first, 25%, 50%, 75% and 100% of the amino acids of a particular property is located respectively.

3 The Data

This chapter describes the source of the datasets to which we have applied our research methods.

3.1 The oxidoreductases dataset

We have concentrated our analysis on the oxidoreductases class of enzymes. To achieve our research goal, a high-quality, well-defined dataset of annotated enzyme sequences was required. Enzyme sequences (i.e., annotated with EC numbers, where n_1=1) were extracted from the UniProt/Swiss-Prot database Release 48.3 25-Oct-2005.

The sequences were strictly screened according to remove the following sequences:

- sequences shorter than 100 amino acids or sequences longer than 1200 amino acids (i.e., indicated as 'Fragment');
- sequences with imprecise annotation (i.e., indicated as 'Probable' or 'Hypothetical' or 'Putative' or partially specified EC number)
- enzymes that catalyze more than one reaction (i.e., indicated as 'Bifunctional' or 'bifunctional' or annotated with more than one EC number)

Following the aforementioned screening procedure, the dataset contained 9437 enzyme sequences. Motifs were extracted by applying MEX to the obtained dataset.

Table 1 summarizes the subclasses & sub-subclasses data of Swiss-Prot release 48.3. 7.

EC class	**# of subclasses**	**# of sub-subclasses**
Swiss-Prot release 48.3	21	81

Table 1: Number of subclasses & sub-subclasses in the dataset

3.2 The Prosite dataset

To test whether the obtained MEX motifs carry relevant biological information we used the PROSITE [16] database. We extracted the PROSITE patterns from Release 19.2 of 24-May-2005 (available online at: ftp://ftp.expasy.org/databases/prosite/).

To extract the set of protein sequence signatures that appear on the enzyme sequences MEX was applied to we used the ps_scan tool, provided by PROSITE (available online at: ftp://ftp.expasy.org/databases/prosite/tools/)

4 Methods

The goal of our research was to explore the ability to predict the function of enzyme's from their sequence of amino acids using our motif extraction (MEX) algorithm. In order to examine whether the extracted motifs convey information regarding the functionality of the enzyme's, each enzyme was represented by its MEX motifs content and the capability of the classifier to predict the function of the enzyme, based on the described representation, was tested.

Due to computational and comparative considerations we have chosen to use the SVM algorithm as our classifier.

The following chapter describes in detail the motif extraction algorithm and provides a concise intuitive description of the Smith-Waterman pairwise sequence similarity algorithm and the SVM algorithm.

4.1 The motif extraction (MEX) algorithm

MEX is a motif extraction algorithm that serves as the basic unit of ADIOS [27, 28, 29], an unsupervised algorithm for the extraction of syntax from linguistic corpora. It discovers structure in any sequence data, on the basis of the minimal assumption that the dataset at hand contains partially overlapping segments.

4.1.1 The intuition behind MEX

To explain the intuition behind MEX let us suppose that our dataset is generated by randomly sampling a set of short predefined sequences. Consider a MEX scan that starts at the beginning of such a sequence, $s = e_1 e_2 \dots e_n$, and proceeds through the first i symbols (i.e., $e_1 \dots e_i$). As i approaches n, the probability of observing the first i symbols by chance (i.e., not as a part of the sequence s) approaches zero, and the conditional probability $P(e_{i+1}|e_1 \dots e_i)$ correspondingly increases. When $i = n$, though, the observed history places no constraints on the next symbol, and the conditional probability for observing the next symbol is likely to be significantly smaller than $P(e_n|e_1 \dots e_{n-1})$. Hence, segmentation is performed at the edge of determinism, where the observed history ceases to be a good predictor of the next symbol. Figure 4.1 shows a partially aligned segment which we consider a MEX motif.

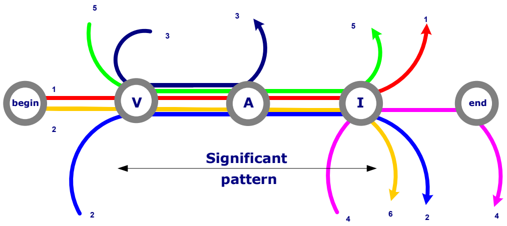


Figure 4.1: A MEX motif.

4.1.2 Description of the algorithm

Consider a dataset of sequences of variable length. Each sequence is expressed in terms of an alphabet of finite size N (i.e., N=20 amino-acids in proteins). The N letters form the vertices of a directed multi graph (i.e., a non-simple graph in which both loops and multiple edges are permitted). The algorithm starts by loading the dataset sequences onto the graph. Each sequence in the dataset defines an ordered path over the graph (i.e., the sequence is indexed by the order of its appearance in the dataset).

In terms of conditional probabilities (i.e., all $p(e_j|e_i)$) the graph defines a variable order Markov model up to order k, where k is the length of the ordered path being placed on the graph.

Loading is followed by a right and a left scan of the graph in search for candidate motifs.

For each ordered path, $(e_1;e_k) := e_1 e_2 \cdots e_k$, we define a right-moving probability function, it's value $\forall$ $1 \leq i,j \leq k$ being:

$$P_R(e_i;e_j) = p(e_j|e_i e_{i+1} e_{i+2} ... e_{j-1}) = \frac{l(e_i;e_j)}{l(e_i;e_{j-1})} \tag{4.1}$$

where $l(e_i;e_j)$ is the number of occurrences of the sub-path $e_i, e_{i+1}, \ldots, e_j$ in the graph.

A drop in the right-moving probability function is defined by:

$$D_R(e_i;e_j) = P_R(e_i;e_j)/P_R(e_i;e_{j-1}) \tag{4.2}$$

The drop in the right-moving probability function is bounded by a threshold parameter η. Therefore, if $D_R(e_i;e_j) < \eta$ then e_{j-1} is defined as the ending edge of the right candidate motif, $C_R = e_i \ldots e_{j-1}$. P_R is calculated from all possible starting points (i.e., $\forall i$, $i = 1, \ldots, k-1$), traversing the paths from left to right.

Once all right candidate motifs are located, the graph is analogously scanned from left to right. Hence, we define a left-moving probability function:

$$P_L(e_j; e_i) = p(e_i | e_{i+1} e_{i+2} ... e_{j-1} e_j) = \frac{l(e_j; e_i)}{l(e_j; e_{i+1})}. \tag{4.3}$$

where $l(e_j; e_i)$ is the number of occurrences of the sub-path $e_j, e_{j-1}, \ldots, e_i$ in the graph.

A drop in the left-moving probability function is defined by:

$$D_L(e_j; e_i) = P_L(e_j; e_i) / P_L(e_j; e_{i+1}) \tag{4.4}$$

The drop in the left-moving probability function is, analogously, bounded by the threshold parameter η. Therefore, if $D_L(e_j; e_i) < \eta$ then e_{i+1} is defined as the ending edge of the left candidate motif, $C_L = e_j \ldots e_{i+1}$. P_L is calculated from all possible starting points (i.e., $\forall j$, $j = k, \ldots, 2$), traversing the paths from right to left.

Since the experimental probabilities, $P_R(e_i; e_j)$ and $P_L(e_j; e_i)$, are determined by a finite number of paths, a statistical measure is introduced to avoid erroneous results. Hence, we calculate the statistical significance of both $D_R(e_i; e_j)$ and $D_L(e_j; e_i)$ by requiring them to be smaller than a predefined parameter $\alpha < 1$.

Therefore, if

$$B(e_i, e_j) = \sum_{x=0}^{l(e_i, e_j)} Bin(l(e_i, e_{j-1}), \eta P_R(e_i; e_{j-1})) < \alpha \tag{4.5}$$

then $e_i \ldots e_{j-1}$ is a significant right motif. Analogously, if

$$B(e_j, e_i) = \sum_{x=0}^{l(e_j, e_i)} Bin(l(e_j, e_{i+1}), \eta P_L(e_j; e_{i+1})) < \alpha \tag{4.6}$$

then $e_j \ldots e_{i+1}$ is a significant left motif.

Once all right and left significant motifs are located, we search for MEX motifs. Figure 1 demonstrates the type of structures we expect to find in our graph. A MEX motif is shown as the assimilation of paths over a subsequence. The criteria for motif selection are defined by local maxima of P_L and P_R signifying, respectively, the beginning and ending of a motif.

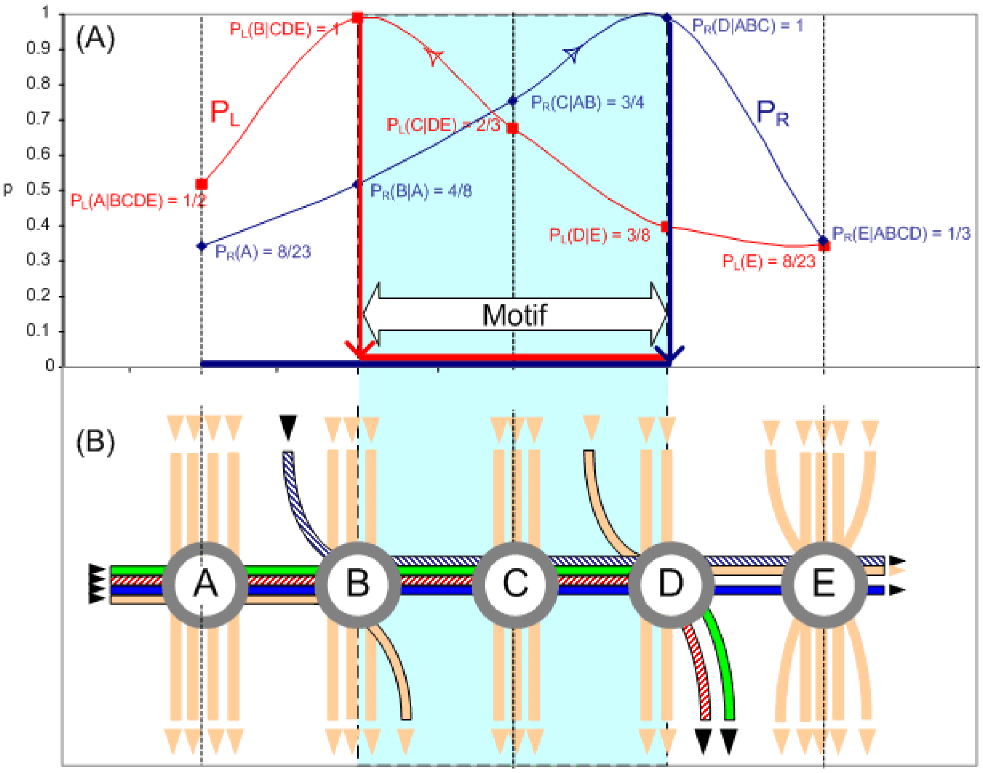


Figure 4.2: The definition of a motif within the MEX algorithm. Descents in P_L and P_R, following the maxima, signify the divergence of paths. Hence, defining the boundaries of the right and left motifs.

Let $C_R = e_i \ldots e_{j-1}$, $i < j$ denote a right significant motif and let e_{R_i} denote the edge at which the increase in P_R has begun. Analogously, Let $C_L = e_j \ldots e_{i+1}$, $i < j$ denote a left significant motif and let C_{L_i} denote the edge at which the increase in P_L has begun. For each such pair of right significant motif and left significant motif, C_R and C_L, we say that the right and left motifs intersect if the maximal index between the index of edge at which the increase in P_R has begun (i.e., $arg(e_{R_i})$) and the index of the edge at which C_L ended (i.e., $i+1$) is smaller then the minimal index between the index of the edge at which the increase in P_L has begun (i.e., $arg(e_{L_i})$) and the index of the edge at which C_R ended (i.e., $j-1$). Hence, if $\max(arg(e_{R_i}), i+1) < \min(arg(e_{L_i}), j-1)$ then $e_{i+1} \ldots e_{j-1}$ is a MEX motif.

Once the algorithm reaches the stop criteria (i.e., ceases to locate new motifs) the sequences are "rewired" according to their MEX motifs content. Hence, the obtained motifs are sorted in a length-significance descending order, by which their loci are identified and placed on the original sequences.

In our work we applied MEX to the set of Oxidoreductases enzymes using the following parameters: $\eta = 0.9$, $\alpha = 0.01$.

4.2 The Smith-Waterman local alignment algorithm

The Smith-Waterman algorithm [26] is a well-known algorithm for performing local sequence alignment (i.e., determining similar regions between two nucleotide or protein sequences). It is a dynamic programming algorithm and as such, it guarantees to find the optimal local alignment with respect to the scoring system being used (i.e., a substitution matrix and a gap-scoring scheme).

As opposed to global sequence alignment algorithms, which look at the entire sequence, the Smith-Waterman algorithm compares segments of all possible lengths and chooses whichever maximizes the similarity measure.

4.2.1 Description of the algorithm

In order to find the optimal local alignment of two sequences, a two-dimensional matrix, containing information about every position of the sequence is filled. This is done with a recurrence relation that is specific to the Smith-Waterman algorithm:

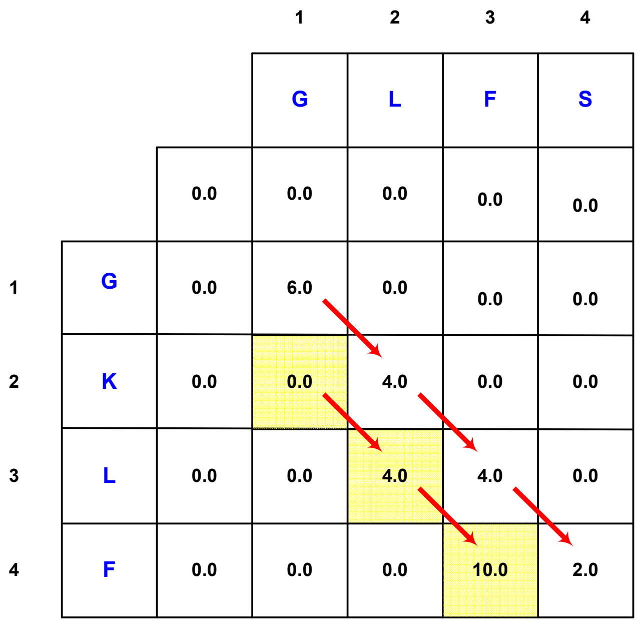
$$F_{i,j} = \begin{cases} 0 \\ F_{i-1,j-1} + s(x_i, y_i) \\ F_{i-1,j} + d \\ F_{i,j-1} + d \end{cases} \tag{4.7}$$

Where:

- $F_{i,j}$ is the value at the (i, j) position in the matrix.
- $s(x_i, y_j)$ is the value obtained from the substitution matrix for the amino acids (x, y) corresponding to the (i, j) position in the matrix.
- d is the opening gap penalty. The affine gap penalty function is usually given by: $f(n) = d + e(n - 1)$, where n = length of the gap, d = opening gap penalty, e = extension gap penalty.

Figure 4.3 is an example of filling up the matrix with two short protein sequences using the BLOSUM62 substitution matrix with d = -12 and e = -2. The position (i=1,j=1) is obtained as follows:

$$F(1, 1) = \max\{0, F(0, 0) + s(G, G), F(0, 1) - 12, F(1, 0) - 12\} = \max\{0, 6, -12, -12\} = 6 \qquad (4.8)$$



			1	2	3	4
			G	L	F	S
		0.0	0.0	0.0	0.0	0.0
1	G	0.0	6.0	0.0	0.0	0.0
2	K	0.0	0.0	4.0	0.0	0.0
3	L	0.0	0.0	4.0	4.0	0.0
4	F	0.0	0.0	0.0	10.0	2.0

Figure 4.3: The dynamic programming matrix generated by aligning GLFS with GKLF. The red arrows indicate where the values come from and the yellow squares indicate the optimal alignment. In order to find the optimal alignment(s), the maximum value(s) must be found and traced back until the value of $F(i, j)$ is equal to 0.

To construct the alignment graphically, these three rules must be followed:

- If the value comes from the position (i-1,j-1), the amino acids (x_i,y_j) are aligned together
- If the value comes from the position (i,j-1), the amino acid y_j is aligned with a gap '_'.
- If the value comes from the position (i-1,j), the amino acid x_i is aligned with a gap '_'.

The optimal alignment is obtained by starting from the position that has the maximum value (in this case, F(i = 4,j = 4) = 10) and by following back the appropriate red arrows. Hence, the optimal alignment is LF.

In our work we applied the Smith-Waterman algorithm to the set of oxidoreductases using the ariadne [21] tool (available online at: http://www.well.ox.ac.uk/ariadne/onepage.html). We have used BLOSUM62 as our substitution matrix and the following gap-scoring scheme: gap opening penalty = 11, gap extension penalty = 1.

4.3 SVM - Support Vector Machines

This section is meant to provide an introduction to the basic concepts of SVMs (Support Vectors Machines). SVM [4, 14, 22, 8] is a known supervised learning method used for classification and regression. In our work, we have used SVM for classification, therefore only the former case will be discussed.

The original formulation of SVM was for the problem of binary classification. Let us start by defining the notion of a classification task. Given a set of data points labeled as $+1$ if they belong to the target class (i.e., positive examples) and -1 if they do not belong to the target class (i.e., negative examples). The task is to ”teach” the classifier, using the given data points, to predict whether a new data point does or does not belong to the target class. SVMs operate by trying to find a hyperplane in the space of possible inputs (i.e., training data) that separates the positive examples from the negative examples.

4.3.1 Maximum Margin Linear SVMs

The simplest kind of SVM (Linear SVM) suits the case of linearly separable data (i.e., if all the data points can be correctly classified by a linear hyperplane). Figure 4.4 illustrates linearly separable data.

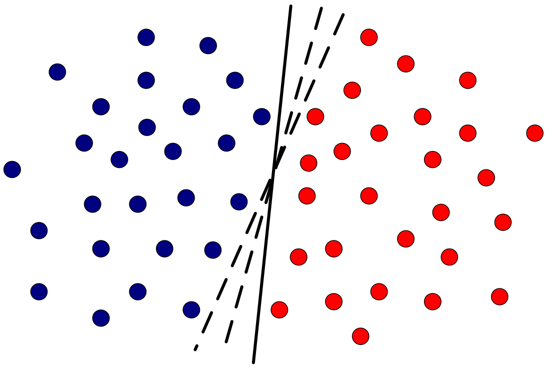


Figure 4.4: Linearly separable data. The red dots denote class1 and blue dots denote class2.

Figure 4.5 illustrates non linearly separable data.

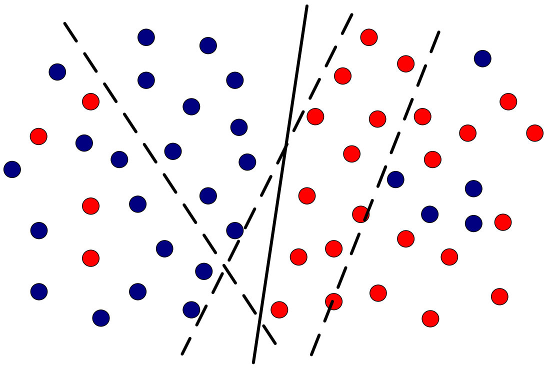


Figure 4.5: Non linearly separable data. The red dots denote class1 and blue dots denote class2.

As demonstrated in figure 4.4, there is more than one possible separating hyperplane. Naturally, we are interested in the hyperplane that provides us with the ”best” separation (i.e., leads to the minimal number of training errors). This is obtained by selecting the hyperplane with the maximal *margin*, where the margin is defined as the width that the separating hyperplane could be increased by before hitting a data point. Hence, we disregard points that are within some fixed positive margin of the hyperplane. This guarantees that a slight misplacement of the separating hyperplane will cause a minimal misclassification.

The *Support Vectors* are the data points lying on the edge of the margin, as demonstrated in figure 4.6.

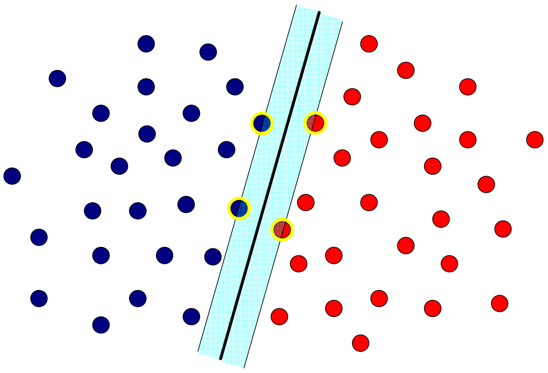


Figure 4.6: The support vectors are the data points lying on the edge of the margin.

The solution for the problem of finding the optimal separating hyperplane (i.e., computing the margin) is based on the concepts of Structural Risk Minimization (SRM) and VC-dimension, and is obtained by representing the problem as a quadratic optimization problem, using standard optimization algorithms.

4.3.2 Soft Margin Linear SVMs

To make it a generally applicable tool, an SVM should be able to handle cases in which the classes overlap. The Soft Margin method chooses a hyperplane that separates the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly separated examples. If the data has noise and outliers, a smooth decision boundary that ignores a few data points is better than one that loops around the outliers. To allow a soft margin, additional constraints (i.e., slack variables) are added to the equation used to compute the margin. The slack variables allow a point to be a small distance on the wrong side of the hyperplane without violating the constraints of the equation used to compute the margin. To assure that the deviations from the ”correct” location are indeed small, large slacks are penalized in the equation used to compute the margin.

4.3.3 Non Linear SVMs

We would like to use the same methodology in cases where the separating hyperplane is not a linear function of the data. The non linearly separable data points are transformed to a different space by using a tractable kernel-based technique, then a linear separating hyperplane is found in the new feature space.

The original data points are mapped to some other high dimensional Euclidean space using a kernel method. Kernel methods exploit information about the inner products between the data points. The data is processed using $\Phi : X \to H$, $x \to \phi(x)$ where H is a dot product space, and the mapping of $\phi(x)$ is learned from the label of the original data point (i.e., $+1$ / -1). Thus, the kernel is a non-linear similarity measure.

Examples of common kernels:

- Polynomial: $k(x, x^{'}) = (< x, x^{'} > +c)^{d}$
- Radial Basis Function (RBF / Gaussian): $k(x, x^{'}) = \exp(-\frac{||x-x^{'}||}{2\sigma^2})$
- Sigmoid : $k(x, x^{'}) = \tanh(k < x, x^{'} > +\Theta)$, for $k > 0$ and $\Theta < 0$

Figure 4.7 demonstrates an example of non linearly separable data in R^2.

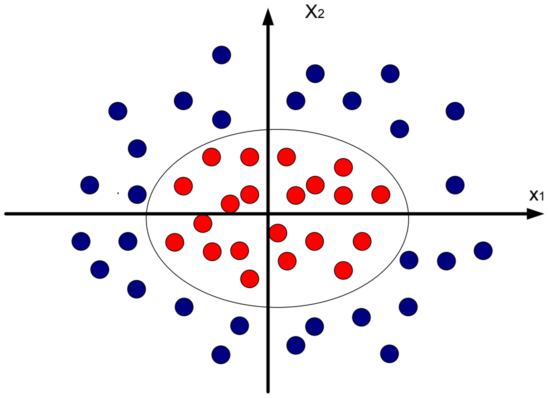


Figure 4.7: The non linearly separable data points in the original R^2 space. Image is taken from [22].

When applying the kernel function $\Phi : (x_1, x_2) \rightarrow (x_1^2, \sqrt{2}x_1x_2, x_2^2)$, one obtains linearly separable data in R^3 as shown in figure 4.8.

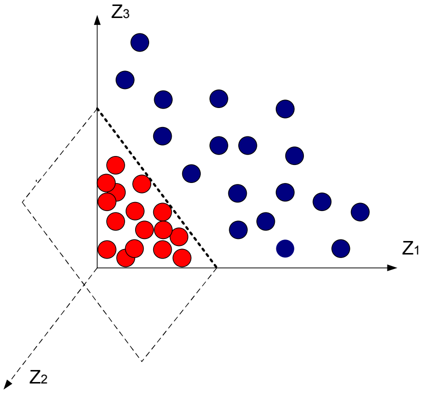


Figure 4.8: Original data is mapped to a higher dimensional Euclidean space where the data is linearly separable. Image is taken from [22].

5 Results

5.1 The classification procedure

The various classification tasks correspond to subclasses and sub-subclasses that contain more than 20 enzyme sequences. (see Appendix C, Tables 7 and 8 for the full list). As disclosed in Table 7, there are 20 classification tasks at the subclass level, corresponding to 20 subclasses that contain more than 20 enzyme sequences. Correspondingly, Table 8 indicates that at the sub-subclass level there are 54 classification tasks, corresponding to 54 sub-subclasses that contain more than 20 enzyme sequences. Hence, enzymes function prediction is defined in terms of predicting the first two digits of the EC number (subclass level) and the first three digits of the EC number (sub-subclass level).

The oxidoreductases dataset has been preprocessed in order to produce an appropriate input file for the learning tasks. A random 75% : 25% partition of the data into a training set and a testing set, respectively, has been used for each learning task. The train - test procedure was repeated on 45 different random partitions of the dataset in order to accumulate statistics.

As our SVM classifier we have used the SVM-Light software
(available online at http://svmlight.joachims.org/). For each of the classification tasks we have used a soft margin linear SVM.

5.2 Performance measurement

As our classification performance measurement we have chosen to use the Jaccard score, defined as follows:

$$J = \frac{TP}{TP + FP + FN} \tag{5.1}$$

where TP, TN, FP and FN denote the number of true positive, true negative, false positive, and false negative outcomes respectively. There are other performance criteria, such as the one used by Cai *et al.* for the SVM-Prot method [5, 6]:

$$Q = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.2}$$

Since the large negative set used in each classification task quickly yields a high Q value, the Jaccard score is a more discriminative performance measurement.

5.3 Motif selection

Applying MEX to the set of oxidoreductases gave rise to 27,471 MEX motifs of various lengths. Figure 5.1 shows the distribution of the entire set of MEX motifs. Note the peak at motifs of length 6 and the drop that follows at motifs of lengths 7 and 8.

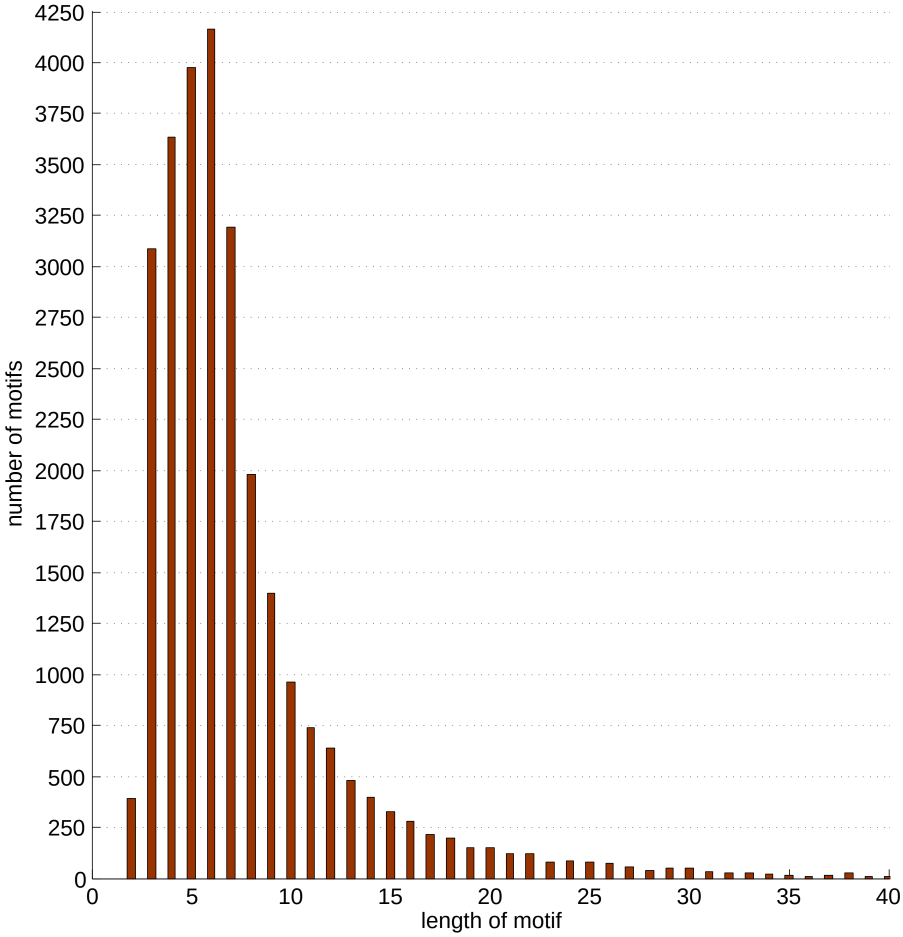


Figure 5.1: Distribution of MEX motifs.

Obviously, MEX is not perfect and returns spurious motifs. Therefore, we used feature selection process. A series of experiments was conducted in order to evaluate the influence of length dependent sets of MEX motifs on the data coverage (i.e., number of enzyme sequences represented by MEX motifs) and on the performance of the SVM classifier on the various classification tasks. It is important to note that the dataset coverage varies according to the subset selection of MEX motifs.

The classification performance was tested using various length dependent subsets of MEX motifs. Table 2 summarizes the results of the classification tasks (for both subclass level and sub-subclass level) and the corresponding dataset coverage.

motifs subset	**# of motifs**	**Avg. 2nd level**	**Avg. 3rd level**	**coverage (%)**
motifs length $\geq$ 4	23994	0.84 ± 0.02	0.83 ± 0.04	100% (9437)
motifs length $\geq$ 5	20360	0.89 ± 0.02	0.85 ± 0.07	99% (9333)
motifs length = 6	4165	0.90 ± 0.03	0.80 ± 0.05	80% (7533)
motifs length $\geq$ 6	**16382**	**0.92 ± 0.02**	**0.90 ± 0.04**	**95% (8933)**
motifs length $\geq$ 7	12217	0.92 ± 0.02	0.91 ± 0.04	90% (8496)
motifs length $\geq$ 8	9024	0.93 ± 0.02	0.89 ± 0.1	85% (8011)
motifs length $\geq$ 9	7041	0.93 ± 0.02	0.89 ± 0.1	79% (7512)

Table 2: Classification performance and coverage percentage for various length dependent subsets of MEX motifs.

5.3.1 Summary

The peak at MEX motifs of length 6 in figure 5.1 accounts for the finding that MEX motifs of length 6 yields both a high classification result (i.e., average Jaccard score is: 0.90 ± 0.03) and a high data coverage (i.e., 80% of the sequences in the dataset).

The classification results, using various length dependent subsets of MEX motifs, show that the classification task performed by the subset of 16,382 motifs of length 6 and longer yields a 0.92 Jaccard score on average for subclass level and a 0.90 Jaccard score on average for sub-subclass level. Furthermore, this subset of MEX motifs covers 95% (8933) of the oxidoreductases enzyme sequences to which MEX was applied. This subset of MEX motifs give rise to the best average Jaccard score - coverage ratio, therefore we chose to perform the classification using this subset of MEX motifs.

5.4 SVM functional classification based on MEX motifs of length 6 and longer

5.4.1 SVM functional classification - subclass level

Figure 5.2 presents the subclass level classification results obtained using the set of MEX motifs of length 6 and longer. Note that there is no correlation between the size of the subclass (i.e., number of sequences in the subclass) and the accuracy of the classification. Correlation was tested using the Spearman correlation at a 0.05 level of confidence. The weighted average Jaccard score obtained at the subclass level is: 0.92 ± 0.02.

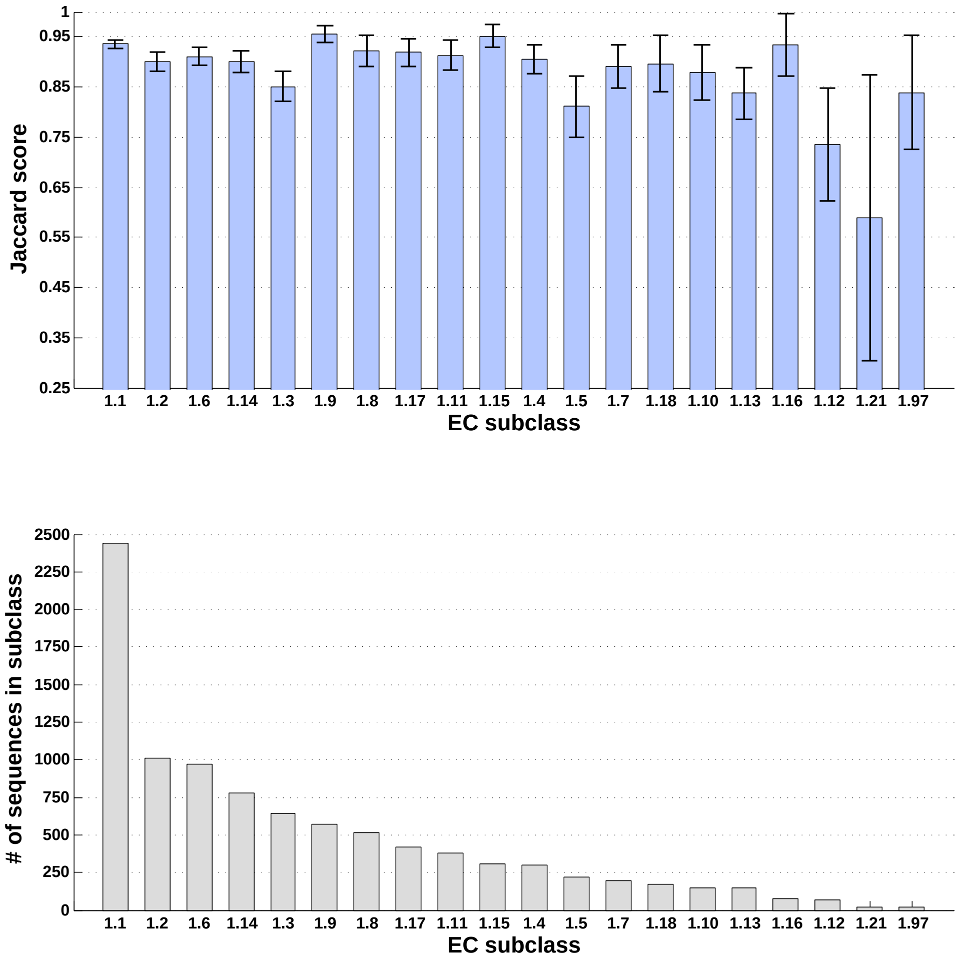


Figure 5.2: Subclass level MEX based classification results.

Table 3 summarizes the classification results for the various methods at the subclass level. We have used SVM-Prot published results, since their published results do not contain standard deviations none are indicated. Furthermore, their published results do not include classification performances for each subclass in our set, hence empty cells indicate such cases. Note that ** denotes the subclasses for which our MEX based classification results were significantly higher than those obtained by the Smith-Waterman based classification. Analogously, * denotes the subclasses for which the Smith-Waterman based classification results were significantly higher than those obtained by our MEX based classification.
Significance was tested using the Wilcoxon signed rand test at a 0.01 level of confidence.

EC subclass	**# of sequences**	**MEX**	**SW**	**SVM-Prot**
1.1	2444	0.94 ± 0.008	0.93 ± 0.012	0.80
1.2	1013	0.90 ± 0.018	0.89 ± 0.022	0.78
** **1.6**	970	0.91 ± 0.017	0.83 ± 0.03	0.88
** **1.14**	777	0.90 ± 0.021	0.77 ± 0.155	0.77
** **1.3**	641	0.82 ± 0.030	0.70 ± 0.038	0.61
1.9	575	0.96 ± 0.016	0.94 ± 0.016	0.95
** **1.8**	513	0.92 ± 0.031	0.82 ± 0.035	0.64
* **1.17**	423	0.92 ± 0.027	0.98 ± 0.013	0.77
* **1.11**	379	0.91 ± 0.03	0.93 ± 0.018	0.78
1.15	310	0.95 ± 0.022	0.97 ± 0.013	0.89
** **1.4**	299	0.90 ± 0.028	0.73 ± 0.055	0.66
** **1.5**	218	0.81 ± 0.06	0.40 ± 0.061	0.39
** **1.7**	196	0.89 ± 0.042	0.64 ± 0.06	0.73
** **1.18**	171	0.90 ± 0.056	0.70 ± 0.07	0.79
** **1.10**	148	0.88 ± 0.055	0.72 ± 0.077	0.69
** **1.13**	147	0.84 ± 0.052	0.58 ± 0.01	0.77
** **1.16**	76	0.93 ± 0.062	0.74 ± 0.065	-
** **1.12**	67	0.73 ± 0.112	0.67 ± 0.117	-
** **1.97**	34	0.84 ± 0.114	0.67 ± 0.168	-
** **1.21**	20	0.59 ± 0.285	0	-
average	-	0.91	0.86	0.74

Table 3: Subclass level classification results. (using MEX motifs of length 6 and longer)

5.4.2 SVM functional classification - sub-subclass level

Table 4 summarizes the classification results at the sub-subclass level. Since the SVM-Prot published results contained only classifications at the subclass level, the comparison at the sub-subclass level is between our MEX motifs based method and the Smith-Waterman based method. Note that ** denotes the sub-subclasses for which our MEX based classification results were significantly higher than those obtained by the Smith-Waterman based classification. Analogously, * denotes the sub-subclasses for which the Smith-Waterman based classification results were significantly higher than those obtained by our MEX based classification. Significance was tested using the Wilcoxon signed rand test at a 0.01 level of confidence.

EC sub-subclass	# of sequences	MEX	Smith-Waterman
1.1.1	2309	0.94 ± 0.011	0.94 ± 0.01
1.2.1	825	0.93 ± 0.022	0.95 ± 0.017
1.6.5	713	0.89 ± 0.021	0.88 ± 0.026
** **1.9.3**	575	0.96 ± 0.013	0.93 ± 0.001
1.11.1	379	0.90 ± 0.027	0.93 ± 0.014
1.15.1	310	0.96 ± 0.025	0.98 ± 0.01
* **1.17.4**	271	0.90 ± 0.03	0.98 ± 0.016
1.3.3	267	0.91 ± 0.027	0.91 ± 0.04
1.14.14	248	0.91 ± 0.032	0.89 ± 0.02
1.8.1	239	0.90 ± 0.037	0.89 ± 0.044
** **1.3.1**	233	0.80 ± 0.043	0.64 ± 0.05
* **1.8.4**	224	0.94 ± 0.037	1
** **1.6.99**	208	0.69 ± 0.076	0.24 ± 0.067
** **1.5.1**	168	0.83 ± 0.055	0.56 ± 0.01
** **1.14.13**	163	0.81 ± 0.057	0.44 ± 0.09
* **1.17.1**	151	0.96 ± 0.04	0.98 ± 0.014
** **1.13.11**	128	0.86 ± 0.05	0.62 ± 0.06
* **1.18.6**	125	0.92 ± 0.046	0.95 ± 0.03
** **1.4.3**	104	0.82 ± 0.084	0.40
1.4.1	99	0.93 ± 0.053	0.92 ± 0.07
** **1.7.1**	94	0.93 ± 0.048	0.87 ± 0.048
** **1.3.99**	93	0.76 ± 0.073	0.14 ± 0.034
** **1.14.99**	90	0.86 ± 0.071	0.31 ± 0.060
* **1.2.4**	86	0.86 ± 0.084	0.90 ± 0.078
** **1.1.99**	75	0.92 ± 0.07	0.52 ± 0.065
** **1.10.2**	69	0.80 ± 0.086	0.62 ± 0.01
** **1.14.11**	64	0.85 ± 0.102	0.31 ± 0.09
** **1.14.12**	63	0.89 ± 0.082	0.84 ± 0.07
** **1.2.7**	56	0.77 ± 0.112	0.26 ± 0.131
* **1.16.3**	55	0.95 ± 0.073	1
* **1.4.4**	50	0.98 ± 0.04	1
** **1.18.1**	46	0.80 ± 0.123	0.21 ± 0.067
** **1.4.99**	46	0.93 ± 0.068	0.73 ± 0.061
** **1.1.3**	44	0.66 ± 0.16	0
** **1.10.3**	42	0.87 ± 0.11	0.72 ± 0.09
* **1.2.99**	39	0.54 ± 0.16	0.61 ± 0.109
** **1.7.99**	38	0.75 ± 0.15	0
1.10.99	37	0.88 ± 0.082	0.86 ± 0.07

EC sub-subclass	**# of sequences**	**MEX**	**Smith-Waterman**
** **1.7.2**	35	0.93 ±	0.39 ± 0.15
** **1.97.1**	34	0.78 ± 0.15	0.68 ± 0.114
** **1.14.15**	32	0.85 ± 0.116	0
** **1.14.17**	32	0.96 ± 0.087	0.64 ± 0.086
** **1.12.99**	30	0.90 ± 0.09	0.54 ± 0.29
1.14.19	29	0.88 ± 0.105	0.92 ± 0.07
** **1.5.99**	29	0.79 ± 0.17	0
* **1.14.16**	27	0.93 ± 0.01	1
** **1.8.98**	27	0.90 ± 0.14	0
** **1.14.18**	26	0.83 ± 0.185	0.31 ± 0.246
** **1.6.1**	26	0.84 ± 0.15	0.20 ± 0.164
* **1.3.7**	25	0.72 ± 0.23	0.96 ± 0.07
** **1.7.3**	24	0.91 ± 0.117	1
** **1.3.5**	23	0.87 ± 0.127	0.20 ± 0.163
** **1.6.2**	23	0.90 ± 0.116	0.16 ± 0.137
** **1.16.1**	21	0.74 ± 0.253	0
average	-	0.90	0.82

Table 4: Sub-subclass level classification results.

5.4.3 Summary

At the subclass level, for 4 subclasses (1.1, 1.2, 1.9, 1.15) there was no significant difference between the Jaccard scores obtained by our method and the results obtained by the Smith-Waterman based method. For 2 subclasses (1.17, 1.11) the Jaccarad scores obtained by the Smith-Waterman based method were significantly higher than those obtained by our method. For the rest **14** subclasses, the Jaccard scores obtained by our method were significantly better than those obtained by the Smith-Waterman based method. Furthermore, for all subclass classification tasks, the results obtained by our method surpassed those obtained by the SVM-Prot method. At the sub-subclass level, for 11 sub-subclasses there was no significant difference between our method and the Smith-Waterman based method. For 10 sub-subclasses the results obtained by the Smith-Waterman method were significantly higher than those obtained by our method. For the rest **33** sub-subclasses, the results obtained by our method were significantly higher than those obtained by the Smith-Waterman based method.

5.5 Unique MEX motifs

To gain a more thorough understanding of the information carried in the extracted motifs, we analyzed which of the motifs appear at a single EC subclass or sub-subclass. These motifs are termed **unique motifs**. The statistics is presented in Figure 5.3.

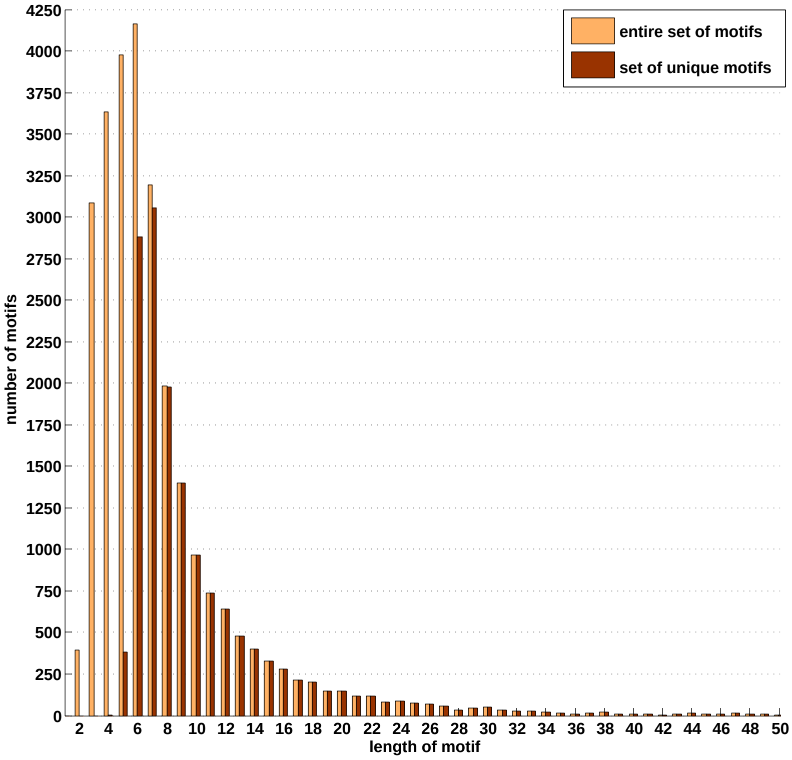


Figure 5.3: Distribution of unique MEX motifs.

5.6 SVM functional classification based on unique MEX motifs

The 14,028 unique MEX motifs cover 90% of the oxidoreductases dataset (8565 sequences). We used this subset of MEX motifs to train the SVM classifier. Since the uniqueness of MEX motifs was obtained in a supervised manner, we used only the unique motifs that were found on the training set sequences to represent the train set and test set enzyme sequences. Figure 5.4 shows the results for the subclass level. Sub-subclass level results are presented in table 5.

5.6.1 SVM functional classification - subclass level

Figure 5.4 presents the subclass level classification results obtained by using the set of unique MEX motifs. Note that there is no correlation between the size of the subclass (i.e., number of sequences in the subclass) and the accuracy of the classification. Correlation was tested using the Spearman correlation at a 0.05 level of confidence. The average Jaccard score obtained at the subclass level is: 0.93 ± 0.018.

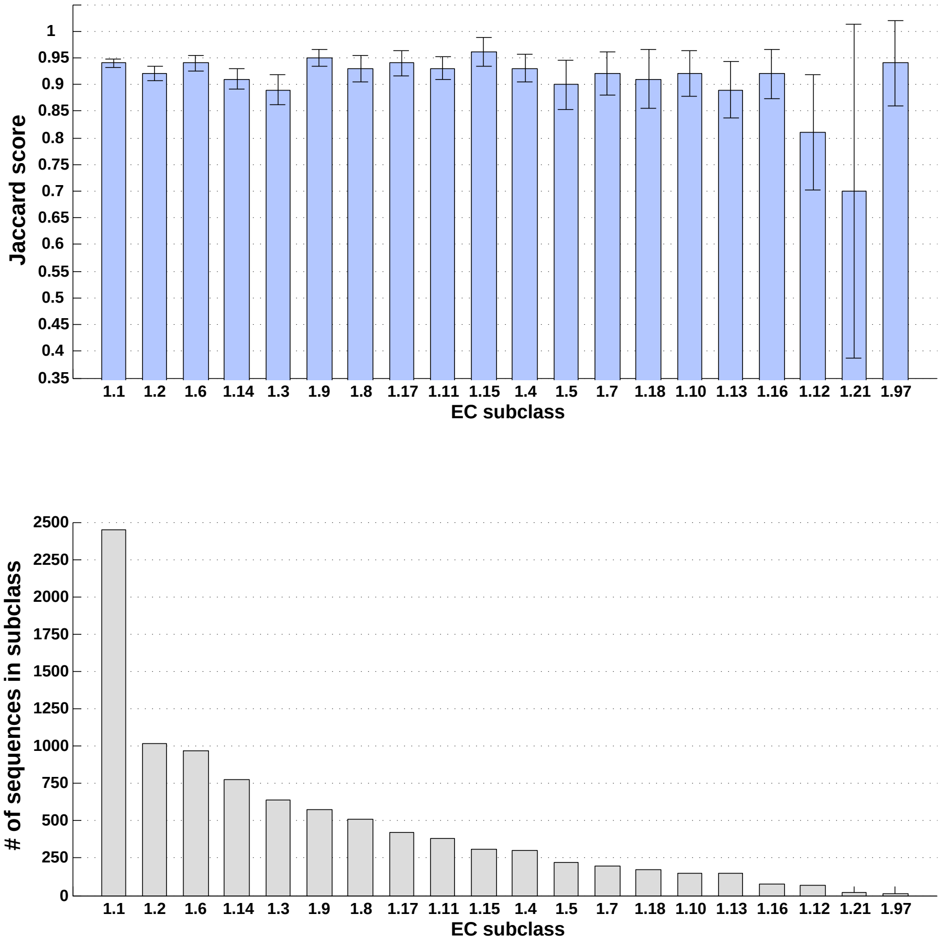


Figure 5.4: Subclass level classification performance based on unique MEX motifs.

5.6.2 SVM functional classification - sub-subclass level

EC sub-subclass	# of sequences	MEX
1.1.1	2309	0.94 ± 0.010
1.2.1	825	0.94 ± 0.019
1.6.5	713	0.89 ± 0.023
1.9.3	575	0.96 ± 0.0173
1.11.1	379	0.92 ± 0.026
1.15.1	310	0.95 ± 0.027
1.17.4	271	0.91 ± 0.027
1.3.3	267	0.92 ± 0.026
1.14.14	248	0.91 ± 0.031
1.8.1	239	0.92 ± 0.039
1.3.1	233	0.83 ± 0.044
1.8.4	224	0.95 ± 0.037
1.6.99	208	0.73 ± 0.081
1.5.1	168	0.89 ± 0.049
1.14.13	163	0.85 ± 0.062
1.17.1	151	0.96 ± 0.033
1.13.11	128	0.89 ± 0.074
1.18.6	125	0.92 ± 0.041
1.4.3	104	0.86 ± 0.075
1.4.1	99	0.95 ± 0.045
1.7.1	94	0.97 ± 0.034
1.3.99	93	0.83 ± 0.077
1.14.99	90	0.91 ± 0.053
1.2.4	86	0.87 ± 0.075
1.1.99	75	0.93 ± 0.068
1.10.2	69	0.86 ± 0.079
1.14.11	64	0.88 ± 0.1
1.14.12	63	0.88 ± 0.078
1.2.7	56	0.88 ± 0.098
1.16.3	55	0.92 ± 0.065
1.4.4	50	0.98 ± 0.039
1.18.1	46	0.86 ± 0.116
1.4.99	46	0.96 ± 0.047
1.1.3	44	0.66 ± 0.151
1.10.3	42	0.92 ± 0.078
1.2.99	39	0.65 ± 0.160

EC sub-subclass	**# of sequences**	**MEX**
1.7.99	38	0.87 ± 0.129
1.10.99	37	0.90 ± 0.070
1.7.2	35	0.93 ± 0.102
1.97.1	34	0.89 ± 0.116
1.14.15	32	0.89 ± 0.103
1.14.17	32	0.96 ± 0.087
1.12.99	30	0.86 ± 0.114
1.14.19	29	0.92 ± 0.102
1.5.99	29	0.86 ± 0.160
1.14.16	27	0.93 ± 0.099
1.8.98	27	0.92 ± 0.141
1.14.18	26	0.84 ± 0.158
1.6.1	26	0.90 ± 0.126
1.3.7	25	0.90 ± 0.018
1.7.3	24	0.90 ± 0.119
1.3.5	23	0.88 ± 0.129
1.6.2	23	0.90 ± 0.116
1.16.1	21	0.80 ± 0.208
average	-	**0.91 ± 0.037**

Table 5: Sub-subclass level classifications performance based on unique MEX motifs.

5.6.3 Summary

Evidently, motifs of length 6 are both abundant and, concomitantly, comprise a large fraction of motifs unique of a single subclass or sub-subclass. This finding provides a good explanation for the classification results obtained by using the subset of motifs of length 6 and longer. As expected, enzyme classification based on unique MEX motifs yields better classification results than those obtained by using the subset of MEX motifs of length 6 and longer. Note that for the smaller subclasses and sub-subclasses the standard deviations are larger than the standard deviations for the larger subclasses and sub-subclasses. These larger deviations are the result of dataset partitioning into a train set and a test set. Since only the motifs found on the sequences of the training set are used, the SVM classifier generalization ability is highly correlated with the motifs found on the training set sequences.

Using the subset of unique motifs for classification, it has been shown that the average Jaccard score at the subclass level is 0.93 ± 0.018 as opposed to 0.91 when using the subset of motifs of length 6 and longer. The weighted average Jaccard score at the sub-subclass level is 0.91 ± 0.037 as opposed to 0.90 ± 0.082 when using the subset of motifs of length 6 and longer. The classification tasks were performed using the set of enzyme sequences that the unique MEX motifs appeared on. In addition, the classification tasks were performed using the entire set of Oxidoreductases enzyme sequences. The results are presented in 11.

5.7 Assessing MEX motifs biological relevance

To further analyze biological information carried by MEX motifs, we examined the abundance of the motifs within PROSITE patterns. As described in section 3.2, we have used the ps_scan tool to extract from the PROSITE database the set of protein sequence signatures that appear on the enzyme sequences MEX was applied to. This yielded 42,789 distinct protein sequence signatures.

We tested the number of MEX motifs that are fully contained in the set of extracted PROSITE patterns. As a background model we randomly selected 16,382 k-mers (i.e., equivalent to the number of MEX motifs of length 6 and longer) out of the set of k-mers that have at lease 4 occurrences in the oxidoreductases dataset. The k-mers were selected according to the underlying distribution of the set of MEX motifs. We generated a 100 random samples of k-mers to gather sufficient statistics and for each random sample we tested the number k-mers that are fully contained in the set of PROSITE patterns that appear on our dataset sequences. To test whether the difference between the number of PROSITE matches obtained by MEX motifs and by the background model k-mers is statistically significant, we computed the significance of the z-score
(i.e., $\frac{MEX-MEAN_{random}}{STD_{random}}$), using the MATLAB normal cumulative distribution function at a 0.01 level of confidence. The results are summarized in table 6.

# of MEX motifs matches	**Mean and std of background model matches**	**significance**
8630	4484 ± 46	$\mathbf{p} < 10^{-300}$

Table 6: MEX motifs matches with PROSITE patterns versus matches of background model.

5.7.1 Summary

As demonstrated in table 6, the number of MEX motifs matches is significantly higher than the number of matches obtained by the background model. The fact that the randomly chosen k-mers are chosen according to the underlying distribution of MEX motifs and yet do not match PROSITE patterns as much as MEX motifs do, attests that MEX motifs do carry relevant biological information.

6 Discussion

Applying the MEX algorithm to the set of 9437 oxidoreductases enzymes, we demonstrated that the extracted motifs form an excellent basis for classifying these enzymes into small classes known to have different functional roles. In particular, the classification from sequence to function based on the motifs of this class of enzymes was shown to outperform any of the alternative methods.

Our results are compared with those of two other approaches: (i) Classification based on pairwise sequence similarity, analogous to the one employed by [19], using the same SVM procedure that was employed for MEX. As demonstrated, MEX derived motifs form a better basis for classification at the subclass and sub-subclass level of the EC number hierarchy, indicating that MEX selected motifs improve the signal to noise ratio inherent in the original sequences. The drawback of the SVM-pairwise method is its efficiency. The vectorization step for the whole dataset requires pre-computing of all pairwise sequence comparison p-values in the training set, where each such computation requires O(n^2), n = length of the protein sequences. MEX's vectorization procedure (section 2.2.1) yields a sparse representation as opposed to the Smith-Waterman representation. This compressed representation reduces the run time of the SVM classifier. Using state of the art machines, obtaining the matched classification results for the Smith-Waterman based method required two orders of magnitudes more time than was required for obtaining the classification results for our method. (ii) The SVM-Prot method introduced by [5, 6] on subclass level data (using their published results). Despite the fact their method is based on semantic information, i.e. physical and chemical properties of the sequence of amino-acids, the results obtained by MEX are better, again indicating that MEX selected motifs carry relevant information.

Moreover, using the subset of unique motifs for classification, we showed that the average Jaccard score at both subclass and sub-subclass levels are even higher than the classification results obtained by using the subset of MEX motifs of length 6 and longer.

It should be noted that the classification based on MEX motifs is accomplished by using only 16,382 motifs of length 6 or longer. Considering the 74 classification tasks for approximately 10,000 enzymes, the number of features allowing a successful classification is surprisingly small.

MEX motifs are all precise, consecutive amino-acid sequences, as opposed to the regular expression motifs used by other methods. Such regular-expression motifs approach was presented by [3]. They have used regular-expression motifs of average length of 21 amino-acids (termed eMOTIFs) derived in a supervised manner. Applying a feature-selection procedure to select approximately 1000 eMOTIFs out of their original very large set of eMOTIFs, they have achieved impressive classification results.

Despite the fact that the number of MEX motifs of length 6 and longer used by our approach is an order of magnitude larger than the number of selected eMOTIFs, it should be noted that the sequences space spanned by our precise, consecutive MEX motifs is much smaller than the one spanned by the eMOTIFs, yet, achieving successful sequence to function classification. Unfortunately, a direct comparison with this work could not be done due to insufficient data.

Furthermore, we demonstrated that the number of MEX motifs matches with PROSITE patterns was significantly higher than the number of matches obtained by the background model, again indicating that the selected MEX motifs carry relevant biological information.

The drawback of our method is its reduced coverage (i.e., number of enzyme sequences represented by MEX motifs). Since not all enzyme sequences contain motifs of length 6 and longer or unique motifs, there are enzyme sequences that are left out, while this is not the case for the two other methods.

In general, MEX was found to be a useful tool for deriving sequence motifs that support a sequence to function classification of enzymes. The ”bag of motifs” representation compressed the information carried by the enzyme sequences, enabling better classification performance and reducing the time required to train the SVM classifier.

6.1 Future challenges

As demonstrated, MEX motifs carry relevant biological information concerning the functionality of enzymes. Therefore, the major future challenge is deciphering the biological meaning of the motifs.

The original use of MEX is as a text segmenting component of the unsupervised grammar induction algorithm, ADIOS [27, 28, 29]. MEX is intended to extract the basic grammatic building blocks, hence words. As opposed to words, where the substitution of a single letter will, most probably, change its meaning, this is not necessarily the case for biological patterns. Peptides (i.e., short amino acid sequences) may bare the same functional role despite the substitution of some amino acids. Furthermore, since the algorithm traverses the graph (see section 4.1) from all possible starting points, there is redundancy in the obtained motifs. Hence, despite the high classification performance achieved by using MEX motifs, the algorithm is not perfect for the purpose of deriving motifs from biological sequential data. The algorithm should be extended to enable the incorporation of statistical information concerning amino acids substitutions, such as substitution matrices. This will reduce redundancy and might enlarge data coverage, hence render the algorithm more suitable for biological data.

7 Appendix A

To test whether the differences between the classification performances of the various methods are statistically significant, we used the non-parametric Wilcoxon signed rank test. As our software we extended the Matlab signrank.m code to include a one-tailed test.

7.1 The Wilcoxon signed rank test

The Wilcoxon signed rank test is a non-parametric method for testing whether two populations have the same continuous distribution. It is used when the data does not meet the strict requirements associated with the corresponding parametric paired t-test (i.e., a ”distribution free” test). The assumption in this test is that there is information both in the magnitudes of the differences between paired observations, as well as in the ”direction” of the differences between the matched observations. We have used in our analysis a one-tailed Wilcoxon’s signed rank test to examine both the significance and ”direction” of the differences between the Jaccard scores obtained by MEX and the corresponding Jaccard scores obtained by Smith-Waterman. In our experiments we have independent pairs of sample data from the populations (i.e., $\{(mex_1, sw_1), (mex_2, sw_2), ..., (mex_n, sw_n)\}$).

The test is performed by ranking the absolute value of the differences between paired observations in an ascending order. Let T_+ denote the sum of all ranks associated with positive differences. Correspondingly, let T_- denotes the sum of all ranks associated with negative differences. The test statistic is either T_+ or T_-.
The critical regions are: $T_+ \leq T_{critical}$ or $T_- \geq \frac{n(n+1)}{2} - T_{critical}$.
The P-value associated with the test statistic (either T_+ or T_-) is extracted from a Wilcoxon T-table according to the sample size, level of significance and whether the test is one or two tailed.

For the classification tasks in which the average Jaccard score obtained by MEX was higher than the average Jaccard score obtained by Smith-Waterman the hypotheses are:

$H_0 : MEX_{score} = SW_{score}$
$H_1 : MEX_{score} > SW_{score}$

Thus,

$$T_+ = \sum_i |mex_i - sw_i|, \; (mex_i - sw_i) > 0$$
$$T_- = \sum_i |mex_i - sw_i|, \; (mex_i - sw_i) < 0$$

For the classification tasks in which the average Jaccard score obtained by Smith-Waterman was higher than the average Jaccard score obtained by MEX the hypotheses are:

$H_0 : MEX_{score} = SW_{score}$
$H_1 : SW_{score} > MEX_{score}$

Thus,

$$\begin{aligned} T_+ &= \sum_i |sw_i - mex_i| \ , \ (sw_i - mex_i) > 0 \\ T_- &= \sum_i |sw_i - mex_i| \ , \ (sw_i - mex_i) < 0 \end{aligned}$$

8 Appedix B - MEX

8.1 Pseudocode of MEX

Algorithm 1 Pseudocode for the MEX algorithm. Only the rightward scan is shown (leftward scanning is symmetric).

```
PROCEDURE $MEX(G)$
$G$ is a MULTI GRAPH
for all node $n_c \in G$ do
  for all node $n_d \in G$ do
    $E_{n_c}$ = the set of all edges leaving $n_c$ (i.e., the degree of $n_c$);
    $E_{n_d|n_c}$ = the set of all edges leading from $n_c$ to $n_d$ (i.e., the degree of $n_d$ given $n_c$);
    $P = |E_{n_c}|/|E_{n_d|n_c}| = P_R(n_d|n_c)$;
    $ScanRight(E_{n_c}, P)$;
  end for
end for
for all path $p \in G$ do
  retrieve any pattern of $p$;
end for
```

Algorithm 2 Pseudocode for the ScanRight algorithm. It is assumed that each edge e is stored in such a way that *e.nextEdge* and *e.toNode* (the edge's successor on the path and the node to which it points) are easily retrieved. The *ScanRight* procedure calls *SignificanceCriterion* (on line 8), which returns true $\iff$ it detects a significant pattern.

```
PROCEDURE $ScanRight(E, P)$
        $processedNodes = \{\}$;
    2:  for all $e \in E$ do
          $e_2 = e.nextEdge$;
    4:    if $e_2.toNode \notin processedNodes$ then
            add $e_2.toNode$ to $processedNodes$;
    6:      $E_2 = \{e.nextEdge \mid e \in E \& e.toNode = e_2.toNode\}$
            $P_2 = |E_2|/|E|$ {probability for $e_2.toNode$ given search history}
    8:      if $SignificanceCriterion(P, P_2, |E|, |E_2|)$ then
              mark significant drop on all relevant paths;
   10:      end if
            if $|E_2| > 1$ then
   12:        $ScanRight(E_2, P_2)$;
            end if
   14:    end if
        end for
```

*** Pseudocode of quickMEX algorithm by Ben Sandbank**

9 Appendix C

This appendix lists the details of the oxidoreductases subclasses and sub-subclasses.

9.1 List of Oxidoreductases subclasses

Table 7 lists the oxidoreductases subclasses in the dataset.

EC subclass	# of sequences
1.1	2444
1.2	1013
1.6	970
1.14	777
1.3	641
1.9	575
1.8	513
1.17	423
1.11	379
1.15	310
1.4	299
1.5	218
1.7	196
1.18	171
1.10	148
1.13	147
1.16	76
1.12	67
1.97	34
1.21	20
* **1.20**	16

Table 7: List of oxidoreductases subclasses. * denotes subclasses in which there are less than 20 enzyme sequences.

9.2 List of Oxidoreductases sub-subclasses

Table 8 lists the oxidoreductases sub-subclasses in the dataset.

EC sub-subclass	# of sequences		EC sub-subclass	# of sequences
1.1.1	2309		1.14.17	32
1.2.1	825		1.12.99	30
1.6.5	713		1.14.19	29
1.9.3	575		1.5.99	29
1.11.1	379		1.14.16	27
1.15.1	310		1.8.98	27
1.17.4	271		1.14.18	26
1.3.3	267		1.6.1	26
1.14.14	248		1.3.7	25
1.8.1	239		1.7.3	24
1.3.1	233		1.3.5	23
1.8.4	224		1.6.2	23
1.6.99	208		1.16.1	21
1.5.1	168		** 1.12.98	17
1.14.13	163		** 1.13.12	15
1.17.1	151		** 1.5.3	15
1.13.11	128		** 1.21.3	13
1.18.6	125		** 1.20.4	11
1.4.3	104		** 1.8.3	11
1.4.1	99		** 1.12.2	9
1.7.1	94		** 1.1.2	8
1.3.99	93		** 1.21.4	7
1.14.99	90		** 1.2.3	6
1.2.4	86		** 1.8.7	6
1.1.99	75		** 1.8.99	6
1.10.2	69		** 1.12.7	5
1.14.11	64		** 1.7.7	5
1.14.12	63		** 1.1.4	4
1.2.7	56		** 1.1.5	4
1.16.3	55		** 1.12.1	4
1.4.4	50		** 1.13.99	4
1.18.1	46		** 1.20.98	4
1.4.99	46		** 1.5.5	4
1.1.3	44		** 1.12.5	2
1.10.3	42		** 1.14.20	2
1.2.99	39		** 1.5.8	2
1.7.99	38		** 1.14.21	1
1.10.99	37		** 1.17.99	1
1.7.2	35		** 1.2.2	1
1.97.1	34		** 1.20.1	1
1.14.15	32			

Table 8: List of oxidoreductases sub-subclasses. ** denotes sub-subclasses in which there are less than 20 enzyme sequences.

10 Appendix D

This appendix lists the details of the subclass level and sub-subclass level classification results obtained using various length dependent subsets of MEX motifs. subclasses and sub-subclasses.

10.1 Subclass level classification results based on length dependent subsets of MEX motifs

Table 9 lists the classification results we obtained on various length dependent subsets of MEX motifs.

subclass	**length $\geq$ 4**	**length $\geq$ 5**	**length = 6**	**length $\geq$ 6**	**length $\geq$ 7**	**length $\geq$ 8**	**length $\geq$ 9**
1.1	0.88 ±0.02	0.92±0.01	0.92±0.04	0.94±0.01	0.94±0.01	0.93±0.01	0.94±0.01
1.2	0.86±0.01	0.88±0.02	0.89±0.02	0.89±0.02	0.91±0.01	0.93±0.01	0.93±0.02
1.6	0.85±0.02	0.87±0.01	0.92±0.02	0.92±0.02	0.94±0.02	0.95±0.01	0.96±0.01
1.14	0.82±0.02	0.84±0.03	0.89±0.04	0.90±0.02	0.91±0.01	0.94±0.02	0.94±0.02
1.3	0.76±0.03	0.81±0.03	0.85±0.04	0.85±0.04	0.90±0.02	0.91±0.03	0.93±0.03
1.9	0.88±0.02	0.93±0.01	0.95±0.02	0.95±0.02	0.94±0.02	0.95±0.02	0.96±0.02
1.8	0.87±0.02	0.93±0.02	0.93±0.02	0.93±0.02	0.90±0.03	0.91±0.03	0.94±0.02
1.17	0.87±0.04	0.90±0.03	0.92±0.04	0.92±0.04	0.91±0.03	0.92±0.02	0.93±0.02
1.11	0.87±0.02	0.89±0.03	0.90±0.03	0.92±0.03	0.93±0.02	0.93±0.03	0.93±0.02
1.15	0.92±0.03	0.96±0.02	0.95±0.02	0.96±0.02	0.96±0.02	0.95±0.02	0.94±0.02
1.4	0.82±0.05	0.85±0.04	0.90±0.03	0.90±0.03	0.93±0.03	0.94±0.03	0.93±0.04
1.5	0.58±0.06	0.75±0.04	0.84±0.05	0.84±0.05	0.87±0.04	0.85±0.06	0.85±0.04
1.7	0.74±0.04	0.81±0.03	0.88±0.04	0.88±0.04	0.92±0.03	0.90±0.03	0.88±0.03
1.18	0.73±0.05	0.82±0.06	0.86±0.06	0.86±0.06	0.89±0.06	0.94±0.04	0.93±0.04
1.10	0.75±0.05	0.85±0.07	0.85±0.07	0.88±0.04	0.89±0.04	0.92±0.04	0.94±0.04
1.13	0.59±0.7	0.73±0.04	0.81±0.03	0.82±0.05	0.90±0.05	0.91±0.05	0.92±0.05
1.16	0.70±0.01	0.79±0.01	0.90±0.07	0.92±0.07	0.93±0.04	0.92±0.06	0.86±0.07
1.12	0.55±0.12	0.70±0.12	0.74±0.16	0.73±0.13	0.80±0.08	0.88±0.06	0.90±0.1
1.97	0.69±0.11	0.78±0.01	0.80±0.1	0.80±0.1	0.94±0.07	0.95±0.07	0.98±0.04
1.21	0.62±0.2	0.64±0.15	0.62±0.25	0.62±0.25	0.82±0.22	0.93±0.1	0.93±0.1
average	**0.84±0.02**	**0.89±0.02**	**0.90±0.03**	**0.92±0.02**	**0.92±0.02**	**0.93±0.02**	**0.93±0.02**

Table 9: Subclass level classification results based on length dependent subsets of MEX motifs.

10.2 Sub-subclass level classification results based on length dependent subsets of MEX motifs

Table 10 lists the sub-subclass level classification results obtained using various length dependent subsets of MEX motifs.

subclass	length ≥ 4	length ≥ 5	length = 6	length ≥ 6	length ≥ 7	length ≥ 8	length ≥ 9
1.1.1	0.87±0.01	0.92±0.01	0.83±0.01	0.93±0.01	0.93±0.01	0.94±0	0.95±0.01
1.2.1	0.9±0.02	0.9±0.08	0.83±0.03	0.92±0.02	0.92±0.02	0.93±0.02	0.92±0.02
1.6.5	0.88±0.02	0.83±0.03	0.79±0.02	0.89±0.01	0.9±0.01	0.82±0.27	0.83±0.28
1.9.3	0.89±0.02	0.9±0.06	0.88±0.03	0.95±0.01	0.93±0.01	0.75±0.37	0.76±0.38
1.11.1	0.86±0.02	0.87±0.08	0.83±0.03	0.92±0.01	0.92±0.01	0.93±0.02	0.93±0.02
1.15.1	0.9±0.03	0.91±0.1	0.9±0.03	0.95±0.01	0.94±0.02	0.94±0.01	0.94±0.01
1.17.4	0.84±0.03	0.86±0.09	0.8±0.03	0.9±0.02	0.9±0.04	0.92±0.02	0.93±0.03
1.3.3	0.86±0.03	0.86±0.13	0.82±0.03	0.91±0.03	0.91±0.03	0.91±0.02	0.92±0.04
1.14.14	0.87±0.04	0.86±0.06	0.85±0.03	0.9±0.02	0.9±0.03	0.9±0.02	0.9±0.05
1.8.1	0.86±0.03	0.86±0.11	0.85±0.05	0.9±0.03	0.88±0.02	0.78±0.26	0.84±0.28
1.3.1	0.73±0.06	0.72±0.09	0.72±0.05	0.79±0.04	0.86±0.08	0.9±0.03	0.9±0.05
1.8.4	0.9±0.03	0.92±0.07	0.91±0.04	0.94±0.03	0.9±0.03	0.73±0.37	0.76±0.38
1.6.99	0.51±0.05	0.63±0.08	0.6±0.1	0.71±0.07	0.73±0.04	0.66±0.23	0.66±0.23
1.5.1	0.61±0.07	0.73±0.09	0.77±0.09	0.83±0.05	0.82±0.06	0.81±0.08	0.85±0.08
1.14.13	0.66±0.04	0.71±0.05	0.7±0.08	0.8±0.05	0.86±0.06	0.89±0.04	0.89±0.04
1.17.1	0.89±0.04	0.91±0.09	0.89±0.05	0.96±0.04	0.94±0.02	0.94±0.03	0.92±0.03
1.13.11	0.59±0.08	0.72±0.07	0.78±0.04	0.85±0.05	0.93±0.06	0.93±0.03	0.95±0.05
1.18.6	0.82±0.06	0.83±0.12	0.83±0.05	0.9±0.04	0.91±0.07	0.94±0.03	0.93±0.05
1.4.3	0.69±0.06	0.65±0.11	0.65±0.11	0.8±0.09	0.83±0.08	0.88±0.07	0.88±0.06
1.4.1	0.84±0.06	0.85±0.12	0.81±0.06	0.9±0.05	0.87±0.09	0.89±0.1	0.87±0.08
1.7.1	0.87±0.06	0.89±0.14	0.88±0.06	0.94±0.06	0.93±0.05	0.82±0.28	0.8±0.27
1.3.99	0.64±0.08	0.71±0.17	0.67±0.09	0.78±0.05	0.78±0.09	0.73±0.08	0.79±0.14
1.14.99	0.72±0.07	0.79±0.11	0.73±0.12	0.85±0.08	0.9±0.05	0.94±0.03	0.94±0.03
1.2.4	0.8±0.09	0.84±0.12	0.83±0.08	0.86±0.05	0.88±0.06	0.93±0.03	0.91±0.07
1.1.99	0.71±0.09	0.82±0.11	0.62±0.11	0.9±0.08	0.92±0.09	0.92±0.09	0.88±0.11
1.10.2	0.55±0.1	0.7±0.11	0.69±0.15	0.78±0.07	0.88±0.08	0.91±0.07	1±0
1.14.11	0.72±0.12	0.76±0.21	0.77±0.11	0.83±0.09	0.85±0.09	1±0	0.98±0.04
1.14.12	0.7±0.13	0.69±0.24	0.84±0.09	0.88±0.06	0.92±0.07	0.87±0.07	0.94±0.06
1.2.7	0.56±0.12	0.65±0.12	0.76±0.17	0.79±0.1	0.84±0.17	0.95±0.09	0.95±0.09
1.16.3	0.8±0.09	0.89±0.07	0.88±0.07	0.93±0.08	0.93±0.05	0.91±0.08	1±0
1.4.4	0.98±0.03	0.98±0.03	0.98±0.03	0.98±0.03	0.98±0.03	0.98±0.03	0.95±0.07
1.18.1	0.63±0.09	0.68±0.2	0.82±0.12	0.82±0.11	0.86±0.1	0.87±0.09	0.98±0.03
1.4.99	0.82±0.09	0.82±0.1	0.83±0.14	0.93±0.06	0.93±0.07	0.93±0.05	0.98±0.04
1.1.3	0.56±0.11	0.67±0.14	0.22±0.2	0.66±0.16	0.77±0.2	0.85±0.11	0.92±0.12
1.10.3	0.78±0.09	0.9±0.1	0.73±0.09	0.86±0.12	0.86±0.12	0.88±0.12	0.88±0.06
1.2.99	0.35±0.15	0.45±0.19	0.43±0.19	0.55±0.11	0.71±0.14	0.88±0.21	1±0
1.7.99	0.47±0.19	0.61±0.21	0.59±0.21	0.76±0.17	0.8±0.18	0.73±0.27	0.69±0.26
1.10.99	0.86±0.09	0.9±0.07	0.62±0.15	0.86±0.09	0.9±0.08	0.89±0.11	0.88±0.15

1.7.2	0.73±0.15	0.91±0.11	0.8±0.23	0.89±0.12	0.91±0.12	0.78±0.29	0.78±0.28
1.97.1	0.67±0.2	0.72±0.24	0.57±0.28	0.83±0.15	0.93±0.1	0.74±0.38	0.78±0.39
1.14.15	0.72±0.12	0.83±0.11	0.85±0.1	0.83±0.11	0.88±0.12	0.86±0.07	0.92±0.1
1.14.17	0.94±0.06	0.92±0.11	0.91±0.09	0.98±0.03	0.97±0.05	0.97±0.07	1±0
1.12.99	0.76±0.14	0.75±0.15	0.79±0.12	0.82±0.14	0.85±0.12	0.84±0.12	0.93±0.12
1.14.19	0.78±0.13	0.83±0.13	0.79±0.13	0.84±0.11	0.89±0.11	0.89±0.11	0.83±0.29
1.5.99	0.53±0.22	0.8±0.28	0±0	0.8±0.15	0.83±0.29	0.92±0.09	0.82±0.29
1.14.16	0.66±0.24	0.8±0.21	0.89±0.14	0.92±0.08	0.9±0.14	0.89±0.15	0.89±0.15
1.8.98	0.69±0.23	0.88±0.21	0.28±0.25	0.93±0.11	0.94±0.1	0.74±0.38	0.7±0.45
1.14.18	0.54±0.14	0.66±0.24	0.6±0.15	0.81±0.18	0.72±0.21	0.72±0.21	0.58±0.34
1.6.1	0.78±0.16	0.79±0.16	0.52±0.24	0.89±0.1	0.98±0.06	1±0	1±0
1.3.7	0.29±0.14	0.51±0.29	0.16±0.21	0.67±0.23	0.9±0.3	1±0	0.9±0.3
1.7.3	0.67±0.13	0.59±0.17	0.35±0.45	0.88±0.12	0.94±0.09	0.8±0.29	0.85±0.29
1.3.5	0.9±0.1	0.86±0.11	0.86±0.21	0.9±0.1	0.88±0.13	0.86±0.12	0.77±0.17
1.6.2	0.8±0.08	0.9±0.18	0.87±0.13	0.92±0.13	0.84±0.12	0.89±0.18	0.76±0.3
1.16.1	0.54±0.12	0.54±0.16	0.7±0.22	0.71±0.23	0.85±0.19	0.88±0.16	0.68±0.2
average	**0.82±0.04**	**0.85±0.07**	**0.80±0.05**	**0.90±0.04**	**0.91±0.04**	**0.89±0.1**	**0.90±0.1**

Table 10: Sub-subclass level classification results based on length dependent subsets of MEX motifs.

11 Appendix E

Figure 5.4 presents the subclass level classification results using the set of enzyme sequences that the unique motifs appeared on (8565 sequences). Figure 11.1 presents the subclass level classification results using the entire set of enzyme sequences (9437 sequences).

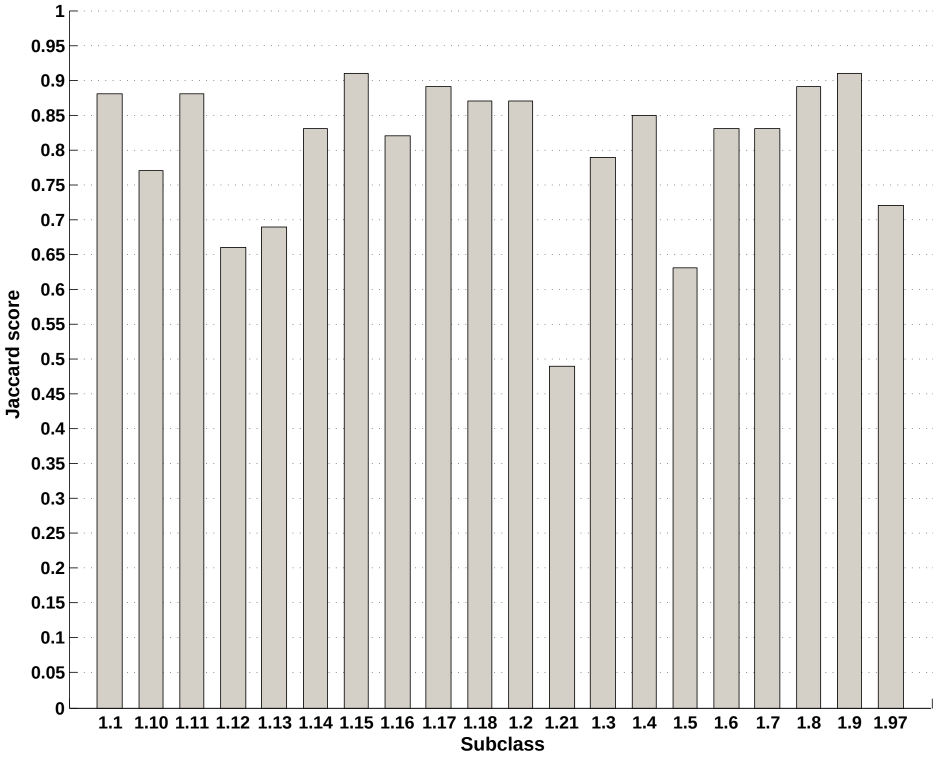


Figure 11.1: Subclass level classification performance based on unique motifs (using the entire set of Oxidoreductases enzyme sequences).

Table 5 presents the sub-subclass level classification results using the set of enzyme sequences that the unique motifs appeared on (8565 sequences). Table 11 presents the sub-subclass level classification results using the entire set of enzyme sequences (9437 sequences).

EC sub-subclass	Jaccard score
1.1.1	0.88 ± 0.012
1.2.1	0.92 ± 0.018
1.6.5	0.84 ± 0.029
1.9.3	0.90 ± 0.023
1.11.1	0.89 ± 0.028
1.15.1	0.92 ± 0.031
1.17.4	0.91 ± 0.027
1.3.3	0.88 ± 0.041
1.14.14	0.91 ± 0.034
1.8.1	0.90 ± 0.045
1.3.1	0.72 ± 0.048
1.8.4	0.92 ± 0.032
1.6.99	0.59 ± 0.07
1.5.1	0.66 ± 0.0639
1.14.13	0.67 ± 0.072
1.17.1	0.90 ± 0.045
1.13.11	0.69 ± 0.076
1.18.6	0.89 ± 0.049
1.4.3	0.69 ± 0.095
1.4.1	0.91 ± 0.051
1.7.1	0.92 ± 0.044
1.3.99	0.72 ± 0.082
1.14.99	0.83 ± 0.094
1.2.4	0.84 ± 0.089
1.1.99	0.87 ± 0.086
1.10.2	0.59 ± 0.094
1.14.11	0.75 ± 0.094
1.14.12	0.70 ± 0.114
1.2.7	0.57 ± 0.125
1.16.3	0.87 ± 0.088
1.4.4	0.98 ± 0.050
1.18.1	0.80 ± 0.107
1.4.99	0.92 ± 0.071
1.1.3	0.50 ± 0.13
1.10.3	0.85 ± 0.116
1.2.99	0.36 ± 0.136

EC sub-subclass	**Jaccard score**
1.7.99	0.87 ± 0.129
1.10.99	0.90 ± 0.088
1.7.2	0.93 ± 0.1
1.97.1	0.76 ± 0.128
1.14.15	0.84 ± 0.12
1.14.17	0.93 ± 0.070
1.12.99	0.88 ± 0.128
1.14.19	0.88 ± 0.113
1.5.99	0.68 ± 0.184
1.14.16	0.89 ± 0.116
1.8.98	0.94 ± 0.149
1.14.18	0.76 ± 0.19
1.6.1	0.70 ± 0.161
1.3.7	0.54 ± 0.185
1.7.3	0.70 ± 0.16
1.3.5	0.89 ± 0.141
1.6.2	0.89 ± 0.117
1.16.1	0.66 ± 0.155

Table 11: Sub-subclass level classifications performance based on unique MEX motifs (using the entire set of Oxidoreductases enzyme sequences).

12 Glossary

- A **substitution matrix** estimates the rate at which each possible residue in a amino acid or DNA sequence changes to each other residue over time
- A **gap scoring system** is used to chose scores for 'gaps'
- A gap is one or more adjacent nulls in one sequence aligned with letters in the other sequence. Ideally, the gap scoring system charges a large initial penalty for the existence of a gap, and smaller penalties for each individual residue. This takes into account that each mutational event can insert or delete multiple residues at a time - the bulk of the gap cost penalty is for the existence of the mutation itself, not the length.
- **Similarity Scores:** Identical or similar residues have positive scores while dissimilar residues can have 0 or even negative scores
- **homology** two or more elements are said to be homologous if they are alike due to shared ancestry. This could be evolutionary ancestry, meaning that the elements evolved from some element in a common ancestor or developmental ancestry, meaning that the elements arose from the same tissue in embryonal development.

References

[1] S.F. et al. Altschul. Basic local alignmnet search tool. *J. of Mol. Biology*, 215:403–410, 1990.

[2] A. Ben-Hur and D. Brutlag. Remote homology detection: a motif based approach. *Bioinformatics*, 19 Suppl. 1:i26–i33, 2003.

[3] A. Ben-Hur and D. Brutlag. Sequence motifs: highly predictive features of protein function. *Neural Information Processing Systems*, 2004.

[4] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):955–974, 1998.

[5] C. Z. Cai, L. Y. Han, Z. L. Ji, and Y. Z. Chen. Svm-prot: web-based support vector machine software for functional classification of a protein from its primary sequence. *Nuclear Acids Research*, 31:3692–3697, 2003.

[6] C. Z. Cai, L. Y. Han, Z. L. Ji, and Y. Z. Chen. Enzyme family classification by support vector machines. *PROTEINS: Structure, Function and Bioinformatics*, 55:66–76, 2004.

[7] C. T. Chien, P. L. Bartel, R. Sternglanz, and S. Fields. The two-hybrid system: a method to identify and clone genes for proteins that interact with a protein of interest. *Proc. Natl. Acad. Sci., USA*, 88:9578–9582, 1991.

[8] N. Cristianini and J. Shawe-Taylor. An introduction to support vector machines. *Cambridge University Press*, 2000.

[9] M. des Jardin, 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. *Proceedings of ISMB*, 1997.

[10] I. Dubchak, I. Muchnik, S.R. Holbrook, and K. Sung-Hou. Prediction of protein folding class using global description of amino acid sequence. *Proced. Natl. Sci USA*, 92:8700–8704, 1995.

[11] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. Biological sequence analysis probabilistic models of proteins and nucleic acids. *Cambridge University Press.*, 1998.

[12] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391:806–811, 1998.

[13] W. M. Fitch. Distinguishing homologous from analogous proteins. *Syst. Zool.*, 19:99–113, 1970.

[14] S.R. Gunn. Support vector machines for classification and regression. *Technical Report ISIS-1-98, Department of Electrics and Computer Science, University of Southampton*, 1998.

[15] J. Y. Huang and D. L. Brutlag. The emotif database. *Nuclear Acids research*, 29:202–204, 2001.

[16] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P.S. Langendijk-Genevaux, M. Pagni, and C.J.A. Sigrist. The prosite database. *Nucleic Acids Res.*, 34:D227–D230, 2006.

[17] T. Jaakkola, M. Diekhans, and D. Haussler. Basic local alignmnet search tool. *Proc. 7th Int. Conf. Intelligent Systems for Molecular Biology*, pages 149–158, 1999.

[18] R. S. Kamath and J. Ahringer. Genome-wide rnai screening in caenorhabditis elegans. *Methods*, 30:313–321, 2003.

[19] L. Liao and W. S. Noble. Combining pairwise sequence analysis and support vector machines for detecting remote protein evolutionary and structural relationships. *J. of Comp. Biology*, 10:857–868, 2003.

[20] H. Lockhart, D. J. abd Dong, M. C. Byrne, M. T. Follettie, M. V Gallo, and M. S. et al. Chee. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat. Biotechnol*, 14:1675–1680, 1996.

[21] R. Mott. Accurate formula for p-values of gapped local sequence and profile alignments. *J. Mol Biol.*, 300:649–659, 2000.

[22] K.R. Müller, S. Mika, K. Rätsch, and B. Scölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Networks*, 12(2):181–201, 2001.

[23] C. G. Neville-Manning, T. D. Wu, and D. L. Brutlag. Highly specific protein sequence motifs for genome analysis. *Proc. Natl. Acad. Sci. USA*, 95:5865–5871, 1998.

[24] M. Schena, D. Shalon, R. W. Davis, and Brown P. O. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470, 1995.

[25] B. Schölkopf. Support vector learning. *R. Oldenburg Verlag, Munich.*, 1997.

[26] T. Smith and M. Waterman. Identification of common molecular subsequences. *J. of Mol. Biology*, 147:195–197, 1981.

[27] Z. Solan, Ruppin, E., D. Horn, and S. Edelman. Automatic acquisition and efficient representation of syntactic structures. *In S. Becker, S. Thrun and K. Obermayer, editors, Advance in Neural Information Processing Systems*, 15:91–98, MIT Press, Cambridge, MA., 2002.

[28] Z. Solan, Ruppin, E., D. Horn, and S. Edelman. Unsupervised context sensitive language acquisition from a large corpus. *In Sebastian Thrun and Lawrence Saul and Bernhard Schölkopf, editors, Advances in Neural Information Processing Systems*, 16, 2003.

[29] Z. Solan, Ruppin, E., D. Horn, and S. Edelman. Unsupervised learning of natural languages. *Proc. Natl. Acad. Sci*, 102:11629–11634, 2004.

[30] S. A. Teichmann. The constraints protein-protein interactions place on sequence divergence. *J. Mol. Biol.*, 324:399–407, 2002.

[31] W. Tian and J. Skolnick. How well is enzyme function conserved as a function of pairwise sequence identity? *J. Mol. Biol.*, 333:863–882, 2003.

[32] E. C. Webb. Enzyme nomenclature, 1992: Recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzymes. *Academic Press*, 1992.

[33] J. Weston, Elisseeff A., B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, 3:1439–1461, 2003.