Available online at www.sciencedirect.com
ScienceDirect
Computational Biology and Chemistry 32 (2008) 122–130
Computational Biology and Chemistry
www.elsevier.com/locate/compbiolchem

Subsequence-based feature map for protein function classification

Omer Sinan Sarac [a], Özge Gürsoy-Yüzügüllü [b], Rengul Cetin-Atalay [b], Volkan Atalay [a,*]

[a] Department of Computer Engineering, Middle East Technical University, 06531 Ankara, Turkey
[b] Department of Molecular Biology and Genetics, Faculty of Science, Bilkent University, 06533 Ankara, Turkey

Received 9 August 2007; accepted 30 November 2007

Abstract

Automated classification of proteins is indispensable for further *in vivo* investigation of excessive number of unknown sequences generated by large scale molecular biology techniques. This study describes a discriminative system based on feature space mapping, called subsequence profile map (SPMap) for functional classification of protein sequences. SPMap takes into account the information coming from the subsequences of a protein. A group of protein sequences that belong to the same level of classification is decomposed into fixed-length subsequences and they are clustered to obtain a representative feature space mapping. Mapping is defined as the distribution of the subsequences of a protein sequence over these clusters. The resulting feature space representation is used to train discriminative classifiers for functional families. The aim of this approach is to incorporate information coming from important subregions that are conserved over a family of proteins while avoiding the difficult task of explicit motif identification. The performance of the method was assessed through tests on various protein classification tasks. Our results showed that SPMap is capable of high accuracy classification in most of these tasks. Furthermore SPMap is fast and scalable enough to handle large datasets.

Keywords: Protein function prediction; Subsequence distribution; Function classification

1. Introduction

Along with the recent advances in genome sequencing technologies, the number of protein sequences with missing annotations increases rapidly. Thus, computational classification methods become valuable for providing a road map for the biologist for further investigation of the excessive number of unknown sequences *in vivo*. In general, *in silico* course of action for the classification of a new sequence is to find similar sequences whose functions are experimentally determined. This is usually performed by searching public databases using local alignment search tools such as BLAST or PSI-BLAST and annotations for the highest scoring hits are transferred onto the new sequence (Altschul et al., 1990, 1997). Although this simple method performs well in many cases, it has some important drawbacks such as excessive transfer of annotations, propagation of errors in the source database, threshold relativity and low sensitivity/specificity (Devos and Valencia, 2000; Gilks et al., 2005; Sasson et al., 2006; Friedberg, 2006). It has been shown recently that although inferring homology through sequence similarity generally holds for the 3D structure, it is far less justified for the function. Additional information than just pairwise similarity is needed to find more accurate annotations (Devos and Valencia, 2000).

Existing approaches to the computational classification beyond simple homology-based transfer can be grouped into three classes: *improved homology-based methods*, *feature-based methods*, and *subsequence-based methods* (Pandey et al., 2006). Improved homology-based approach still uses sequence homology, however it incorporates additional information (Andrade et al., 1999; Riley et al., 2005; Martin et al., 2004), such as multiple sequence alignments or classifications of similarity results according to a hierarchical and structured organization of functions like in Gene Ontology (GO) database (Ashburner et al., 2000). On the other hand, both feature-based and subsequence-based approaches pursue discriminative methodology that explicitly models the differences between positive and negative examples. Two approaches differ in the way how they

* Corresponding author. Tel.: +90 312 210 5576; fax: +90 312 210 5544.
E-mail address: volkan@ceng.metu.edu.tr (V. Atalay).

1476-9271/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiolchem.2007.11.004

extract features from sequences. In the feature-based approach, biologically meaningful properties of a protein such as frequency of residues, molecular weight, secondary structure, *n*-gram frequencies, are extracted from the primary sequence. These properties are then arranged as feature vectors and used as input to classification techniques such as artificial neural networks (ANNs) or support vector machines (SVMs) (Duda et al., 2000; King et al., 2000; Pasquier et al., 2001; Jensen et al., 2002; Cai et al., 2003; Karchin et al., 2002; Cheng et al., 2005). On the other hand, conserved subsequences among a class of proteins are employed in subsequence-based methods. The main idea is that, conserved subsequences among different proteins are strong indicators of functional or structural similarity because functionally important regions (catalytic sites, binding sites, structural motifs) are conserved over much wider taxonomic distances than the sequences themselves. Thus, in subsequence-based approach feature vectors are constructed according to the existence of specific motifs or domains in the protein sequences. The critical step in this approach is the extraction and selection of motifs. One possibility is to use motif information from protein databases (Ben-hur and Brutlag, 2003; Wang et al., 2003) in which motifs are assumed to be already available for the family of proteins to be classified. Most of the methods of subsequence-based approach attempt to extract motifs explicitly for the given families (Hannenhalli and Russell, 2000; Wang et al., 2001; Liu and Califano, 2001; Kunik et al., 2005; Blekas et al., 2005). Although motifs are powerful discriminators even in low similarity (remote homology) situations, motif finding is a very difficult task, especially for protein sequences since there are 20 different amino acids and many plausible mutations. Multiple sequence alignments and other computational pattern extraction algorithms are often employed for motif finding. Unfortunately, algorithms that can find optimal solutions in all of these methods have exponential time complexities, hence approximation or heuristic algorithms are used instead. As a consequence, there is always the risk of missing some relatively implicit motifs. Furthermore, classical motif finding algorithms find a specified number of motifs even if there are not that many biological motifs in the family. These insignificant additional motifs might reduce the accuracy of the classification. One other issue is that, depending on the classification task, proteins to be classified might not have a common motif at all. As an example, in the problem of subcellular localization, when discriminating cytosolic proteins, it is not possible to find motifs specific to this class. Methods that consider overall sequence similarity may perform better in such cases.

In this study, we describe a feature space mapping, called subsequence profile map (SPMap), that takes into account the information coming from the subsequences of a protein. Our approach incorporates the information coming from important subregions that are conserved over a family of proteins as well as the overall sequence similarity. Instead of focusing on function specific motifs, SPMap considers all of the subsequences as a distribution over a quantized space by discretizing and reducing the dimension of an otherwise huge space of all possible subsequences.

2. Systems and Methods

The system described in this study is based on a discriminative method which requires positive and negative examples to classify and annotate proteins whose functions are not known. Instead of looking for the overall similarity of protein sequences, we make use of the distribution of short subsequences of a given protein over a subsequence profile map. We generated the profiles using all possible fixed-length subsequences of the protein sequences in the positive training set. Similar subsequences were clustered together and clusters were represent as probabilistic profiles. The major reasoning behind this approach is that, subsequences extracted from the conserved regions are more frequent than any other subsequence extracted from the positive training data. If the frequent subsequences are represented as dimensions of feature vectors, discriminative methods can make use of this information. If there is a conserved motif or a domain in the given sequences or there is an overall similarity between sequences, they would produce similar distributions on the profile map. Classifiers such as support vector machines (SVMs) may then identify these similar distributions and hence improve the classification accuracy.

In order to perform the classification, SVMs were used. We constructed fixed dimensional vectors that represent the subsequence distribution information. There are two critical steps in SPMap as shown in Fig. 1:

A. subsequence profile map construction,
B. feature vector generation and classification.

2.1. Subsequence Profile Map Construction

In SPMap, feature space representation of a protein sequence is the distribution of its subsequences over a map of generative models. General framework for finding this generative feature map is summarized as follows.

- Subsequence Extraction Module: Extract all possible subsequences of a given length from positive training sequences.
- Clustering Module: Cluster similar subsequences by an appropriate clustering method.
- Profile Construction Module: Build a model for each cluster.

The important step here is the clustering of subsequences. Note that the space of all possible subsequences of length l is of size 20^l, since there are 20 possible amino acids. Instead of working in this very high dimensional space, we quantized this space using the clusters of subsequences that are actually existing in the positive training examples. One should note that, as we clustered the subsequences, we were not actually looking for underlying groupings. The aim here was to generate a meaningful quantization of the subsequence space that especially represent groups of frequent and similar subsequences in the positive training data. These subsequences might have been conserved because of their importance for the function of that

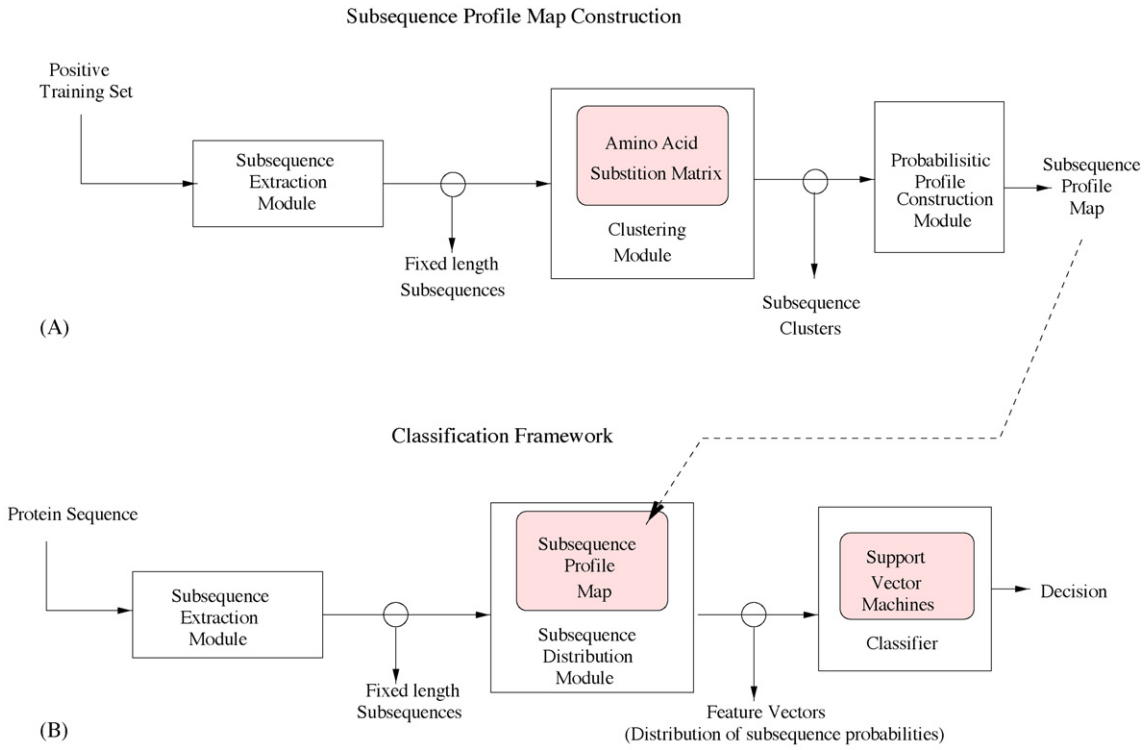

Fig. 1. SPMap flow diagram. (A) Subsequence profile map construction: subsequences of the proteins in positive training set are clustered to construct subsequence profile map. (B) Classification: constructed profile map is utilized to find the feature space representation of the protein sequence to be classified.

class of proteins and we wanted our feature space to take them into account. Clustering algorithm is given in Algorithm 1. It is similar to the average link hierarchical clustering, however it can be implemented very efficiently without calculating all the pairwise distances. Initially the number of clusters is set to 0. Each subsequence is compared against all of the existing clusters and average similarity to the elements of each cluster is calculated. A subsequence is assigned to the cluster, $C_{\max}$, which gives the maximum average similarity value. If the similarity to $C_{\max}$ is less than a threshold, δ, a new cluster is created and the subsequence is assigned to the new cluster. Similarity between two subsequences x and y was calculated by the formula

$$s(x, y) = \sum_{i=1}^{l} M(x(i), y(i)) \quad (1)$$

where l is the length of the subsequences and $M(x(i), y(i))$ is the value in the similarity matrix for the ith elements of x and y. For M, we used an amino acid similarity matrix, since it allows us to incorporate evolutionary information in finding and representing important conserved regions of a family of proteins. The final number of clusters depend on the threshold value δ. If it is set to a high value, clusters will be smaller only allowing very similar subsequences and the total number of clusters will be high. If it is set to a low value, biologically unrelated subsequences might end up in the same cluster.

Algorithm 1. Clustering Algorithm.

Algorithm 1 Clustering Algorithm

$X \leftarrow$ all fixed length subsequences of the positive training set
$C \leftarrow \{\}$
for all $x_i \epsilon X$ **do**
  for all Clusters C_k **do**
    $s_k = \frac{\sum_{x_j \epsilon C_k} s(x_i, x_j)}{|C_k|}$
  end for
  $m = argmax_{k=1..|C|} s_k$
  if $s_m > \delta$ **then**
    Add x_i to C_m
  else
    Create a new cluster $C_{|C|+1}$ and add x_i to $C_{|C|+1}$
  end if
end for

After the clustering step, we generated a probabilistic profile for each cluster. A probabilistic profile PP_k for cluster k, is an $l \times 20$ matrix, where l is the length of a subsequence. Entry $P_k(i, j)$ of this matrix represents the probability of amino acid j to occur at the ith position of the subsequence. Given a cluster C_k, the profile for this cluster is calculated by the following equation:

$$PP_k(i, j) = \log \frac{\phi_k(i, j) + \kappa}{|C_k|} \quad (2)$$

where $\phi_k(i, j)$ represents the count of the amino acid j at position i of the subsequences in C_k. We added a pseudo-count κ for amino acids at each position to avoid over-fitting and zero probabilities. Actually, we took the log of the profiles and worked with log-probabilities in the conversion step.

2.2. Feature Vector Generation

Proteins were represented in the feature space as the distribution of their subsequences over the generated subsequence profile map. All the subsequences of a protein were extracted to construct a feature vector. Each subsequence x was compared with each probabilistic profile PP_k and a probability was calculated as

$$P(x|PP_k) = \sum_{i=0}^{l} PP_k(i, x(i)). \quad (3)$$

The value for the kth dimension of the feature vector V is set to

$$V(k) = \max_{x_i \in S} P(x_i|PP_k), \quad (4)$$

the probability of highest scoring subsequence of protein S on probabilistic profile PP_k. This algorithm is similar to the vector generation algorithm presented in Blekas et al. (2005) with the difference that we set $V(k)$ to 0 if the probability is very small.

2.3. Classification

Once the protein sequences are mapped onto the feature space, any numerical machine learning tool can be employed. Our choice was to use SVMs since they are experimentally proven to be successful for various problems (Cristianini and Shawe-Taylor, 2000). Radial basis function (RBF) was chosen as the kernel for SVM. In all of the experiments, SVM parameter C and RBF kernel parameter γ were fixed to be 2 and 0.05, respectively. SVM-light software was used for learning and classification steps (Joachims, 1999).

2.4. Experimental Setup

In all of the experiments, BLOSUM62 matrix was employed to calculate the similarity between subsequences (Henikoff and Henikoff, 1992) although it is possible to use different similarity matrices depending on the sequence divergence or the taxonomic distance between the proteins to be classified (Atalay and Cetin-Atalay, 2005; Tomii and Kanehisa, 1996). BLOSUM62 is shown to be useful for a wide range of problems and is the default selection for most of the alignment tools (Altschul et al., 1990, 1997). Length of the subsequences was set to 5. Setting the subsequence length to 5 did not mean that we sought for motifs of 5 amino acid length. In SPMap, motifs were the overall distribution of the subsequences over the profiles constructed from resulting 5 length subsequence clusters. Hence subsequence length 5 allowed us to capture longer motifs as a distribution over more than one profile. We tested the performance of SPMap by changing the subsequence length in the interval [5,12] on selected sample sets of data. We observed that although there were differences in the performance with respect to the change in the subsequence length, 5 was the optimal in the sense of performance versus computational complexity. Threshold similarity score δ in Algorithm 1 was fixed to 8 where the expected similarity score of two random subsequences of length 5 using BLOSUM62 matrix is −5.325. Compared to the expected value, 8 is high enough to disallow random similarities. Extensive tests with different threshold values showed that 8 performed better in most of the test cases and it was set as default in all of the experiments.

Table 1
Average ROC scores and standard deviations for subcellular localization predictions

Localization	Data size	Mean ROC	S.D.
ER targeted	3115	0.97	0.006
Cytoplasmic	1789	0.95	0.005
Mitochondrial	1148	0.96	0.006
Nuclear	2225	0.96	0.005

3. Results

3.1. Subcellular Localization

The idea of subsequence distribution was first proposed in P2SL (Atalay and Cetin-Atalay, 2005). However, we developed more robust, reliable and efficient method for this idea. In order to be able to show the improvement, we first performed tests on the subcellular localization dataset on which P2SL was trained and tested. Dataset was composed of four different classes, namely ER targeted (ER), cytoplasmic (C), mitochondrial (M) and nuclear (N) (Atalay and Cetin-Atalay, 2005). ER targeted and mitochondrial proteins have signal peptides of length 25 and 35 amino acids, respectively, at the N-terminal of the proteins. While extracting subsequences for feature map construction we used first 30 amino acids for ER targeted proteins and first 40 amino acids for mitochondrial proteins. Two types of tests were performed. First, in a one-versus-all setting, ROC scores were calculated for each localization and results are given in Table 1.

In the second test case, classifiers for each localization were combined using the winner-take-all principle. Each test sample was assigned to the location whose classifier produced the highest SVM score. The confusion matrix obtained by averaging fourfold cross-validation tests and their comparison with P2SL results are given in Table 2 (Atalay and Cetin-Atalay, 2005).

3.2. G-protein-Coupled Receptor Subfamily Classification

Tests are subsequently carried on G-protein-coupled receptor (GPCR) subfamily classification problem that was extensively studied in the literature. Consequently, GPCR subfamily classification constitutes a good benchmark dataset for comparing with other methods. For GPCR subfamily classification, we used the dataset presented in Karchin et al. (2002) to compare with the results of various classifiers presented in Karchin et al. (2002) and Cheng et al. (2005). Same train and test splits were used for twofold cross validation for fairness of comparison. SPMap was tested on level I and level II subfamily classification of GPCR proteins. In level I subfamily classification, there

Table 2
Confusion matrix representing average percentage results of fourfold prediction tests compared with P2SL results

Actual	Predicted label			
	N (%)	C (%)	M (%)	ER (%)
N				
SPMap	**89.83**	7.5	1.1	1.58
P2SL	**75.34**	19.94	3.29	1.43
C				
SPMap	7.14	**89.05**	1.8	2.02
P2SL	14.66	**79.33**	3.65	2.36
M				
SPMap	2.09	5.4	**89.29**	3.22
P2SL	3.31	7.23	**83.80**	5.66
ER				
SPMap	2.07	2.5	1.41	**94.03**
P2SL	4.89	6.19	3.29	**85.63**

were 1269 sequences from 19 subfamilies within classes A and C in addition to 149 non-GPCR sequences. In level II subfamily classification, there were 1170 GPCR sequences from 70 different level II subfamilies. Some of the sequences in level I subfamily classification have no level II subfamily classification and some of the level II subfamilies only have one protein so they are grouped as other sequences with non-GPCR sequences. Datasets and train and test splits are available at http://www.soe.ucsc.edu/research/compbio/gpcr/subfamily_seqs.

The comparison of accuracy of various classifiers and SPMap is presented in Table 3. Fisher-SVM, BLAST, SAM-T2K HMM, and kernNN methods were presented in Karchin et al. (2002) and Decision Tree and Naïve Bayes methods were presented in Cheng et al. (2005).

3.3. Enzyme Class Classification

Finally we evaluated the performance of SPMap on enzyme class classification. Enzymes play a central role in many of the biological functions in a cell. They are indispensable for understanding the molecular systems in a cell and are important drug targets. Hence accurate classification is very important in enzyme research.

Dataset for enzyme classification is extracted from BRENDA database (Schomburg et al., 2002). International Union of Biochemistry and Molecular Biology defines the numerical classification scheme for enzymes based on the chemical reactions they catalyze. Each enzyme is described by a sequence of four numbers (EC numbers) resulting from a four-level hierarchy where first number specifies the most general class and the last one specifies the most specific. At the highest level there are six major classes of enzymes. Automated prediction methods are successfully applied to enzyme classification according to the first (Lu et al., 2007) and second level of EC numbers (Cai et al., 2003). We also performed tests according to the first and second EC numbers. On the first level there are six major classes of enzymes. Dataset used for this level is presented in Lu et al. (2007). Each class is filtered so that there are no pair of proteins with more than 25% sequence identity. The success rates for various methods and SPMap for six classes with leave-one-out cross-validation is presented in Table 4.

Table 3
Comparison of accuracy of various classifiers at GPCR levels I and II subfamily classification

Classifier	Level I accuracy	Level II accuracy
BLAST	83.3	74.5
Decision Tree	77.3	70.8
Fisher-SVM	88.4	86.3
kernNN	64.0	51.0
Naïve Bayes	93.0	92.4
SAM-T2K HMM	69.9	70.0
SPMap	**95.4**	**93.8**

Table 4
Comparison of success rates of various classifiers on six major enzyme classes calculated with leave-one-out cross-validation

Classes	Total	Success (%)				
		Lu et al.	Blast	Psi-Blast	SVM-Prot	SPMap
Oxidoreductase	436	**93.53**	89.68	91.06	73.62	80.73
Transferase	832	**93.63**	88.46	87.98	82.45	66.23
Hydrolase	741	**94.20**	86.10	86.77	77.33	71.93
Lyase	170	75.29	75.29	70.59	68.82	**94.12**
Isomerase	114	74.56	73.68	73.68	68.42	**96.49**
Ligase	150	89.33	**90.00**	88.67	37.33	88.00

We also classified proteins according to their first two EC numbers, resulting in 56 classes. We omitted classes with very few members. Sensitivity and specificity values calculated over fourfold cross validation are presented in Table 5. This classifier for 56 enzyme classes is available as an online service at http://gen.ceng.metu.edu.tr/spmap/cgi-bin/enzyme.cgi.

4. Discussion

4.1. Computational Complexity

SPMap is composed of two main parts. First part is the subsequence profile map construction. It is only performed once for a new classifier to be trained. Hence, its efficiency does not affect the performance during the classification of new sequences. The most expensive part of the map construction is the clustering of subsequences. Most of the standard clustering algorithms require numerical vectors to work on. More specifically, they require a metric to calculate the distance between the cluster representations and data points and a method to update these cluster representations throughout the course of the algorithm. These methods usually perform $O(nk)$ distance calculations where n is the number of data points and k is the number of clusters. They require the number of clusters k to be given at the start. There are also clustering algorithms that use only pairwise distances between data points. They do not require the number of clusters k as a parameter but they have

Table 5
Sensitivity ($TP/(TP + FN)$) and specificity ($TN/(TN + FP)$) values for 56 enzyme class classifiers calculated over fourfold cross validation

Enzyme class	Data size	Sensitivity	Specificity
EC 1.1 Acting on the CH–OH group of donors	8878	95.33	85.05
EC 1.2 Acting on the aldehyde or oxo group of donors	4099	91.63	97.17
EC 1.3 Acting on the CH–CH group of donors	2455	85.75	98.09
EC 1.4 Acting on the CH–NH_2 group of donors	1573	88.64	99.74
EC 1.5 Acting on the CH–NH group of donors	1244	81.35	99.72
EC 1.6 Acting on NADH or NADPH	5572	94.54	95.85
EC 1.7 Acting on other nitrogenous compounds as donors	802	83.67	99.93
EC 1.8 Acting on a sulfur group of donors	1699	89.94	99.82
EC 1.9 Acting on a heme group of donors	1620	93.99	98.51
EC 1.10 Acting on diphenols and related substances as donors	813	86.86	99.98
EC 1.11 Acting on a peroxide as acceptor	1267	91.56	99.97
EC 1.12 Acting on hydrogen as donor	243	68.89	99.97
EC 1.13 Acting on single donors/with incorporation of molecular oxygen (oxygenases)	1048	87.66	99.97
EC 1.14 Acting on paired donors, with incorporation/or reduction of molecular oxygen	1909	83.3	98.42
EC 1.15 Acting on superoxide radicals as acceptor	935	93.56	99.99
EC 1.16 Oxidising metal ions	142	65.71	99.96
EC 1.17 Acting on CH or CH_2 groups	1063	90.31	99.92
EC 1.18 Acting on iron–sulfur proteins as donors	745	91.94	99.97
EC 1.20 Acting on phosphorus or arsenic in donors	66	66.67	99.99
EC 1.21Acting on X–H and Y–H to form an X–Y bond	60	88.89	100
EC 1.97 Other oxidoreductases	169	80.95	99.99
EC 2.1 Transferring one-carbon groups	6061	92.28	90.97
EC 2.2 Transferring aldehyde or ketonic groups	1058	94.32	99.94
EC 2.3 Acyltransferases	6149	92.52	91.55
EC 2.4 Glycosyltransferases	6004	92.65	89.54
EC 2.5 Transferring alkyl or aryl groups, other than methyl groups	5188	93.94	96.73
EC 2.6 Transferring nitrogenous groups	2011	95.22	99.85
EC 2.7 Transferring phosphorus-containing groups	23424	89.78	91.08
EC 2.8 Transferring sulfur-containing groups	982	87.35	99.91
EC 2.9 Transferring selenium-containing groups	72	88.89	100
EC 3.1 Acting on ester bonds	9879	74.79	96.05
EC 3.2 Glycosylases	4789	93.76	91.98
EC 3.3 Acting on peptide bonds (peptidases)	5945	93.4	87.48
EC 3.5 Acting on carbon–nitrogen bonds, other than peptide bonds	5942	90.28	88.25
EC 3.6 Acting on acid anhydrides	7430	96.23	88.22
EC 3.7 Acting on carbon–carbon bonds	66	81.25	100
EC 3.8 Acting on halide bonds	101	49.33	99.98
EC 4.1 Carbon–carbon lyases	7606	93.77	87.95
EC 4.2 Carbon–oxygen lyases	7211	93.23	87.46
EC 4.3 Carbon–nitrogen lyases	1264	91.14	99.89
EC 4.4 Carbon–sulfur lyases	626	82.91	99.8
EC 4.6 Phosphorus–oxygen lyases	614	91.28	99.9
EC 4.99 Other lyases	297	90.99	99.98
EC 5.1 Racemases and epimerases	2030	92.18	99.66
EC 5.2 *cis–trans*-Isomerases	1232	92.86	99.92
EC 5.3 Intramolecular isomerases	2910	90.65	99.18
EC 5.4 Intramolecular transferases (mutases)	2195	88.57	99.37
EC 5.5 Intramolecular lyases	135	71.72	99.98
EC 5.99 Other isomerases	1418	95.57	99.96
EC 6.1 Forming carbon–oxygen bonds	6285	97.05	98.39
EC 6.2 Forming carbon–sulfur bonds	1112	93.17	99.91
EC 6.3 Forming carbon–nitrogen bonds	6784	94.53	95.25
EC 6.4 Forming carbon–carbon bonds	785	94.9	99.87
EC 6.5 Forming phosphoric ester bonds	433	89.2	99.97
EC 6.6 Forming nitrogen-metal bonds	118	90.81	99.97

to perform $O(n^2)$ pairwise distance calculations and that might be very inefficient in terms of time and memory for large n. Note that n in this case is the total number of subsequences extracted from all of the positive training examples, which is roughly the number of amino acids in the positive training examples. However, Algorithm 1 can be implemented in $O(nk)$. The critical step is the calculation of the average distance of subsequence x_i to the cluster u given in the following equa-

tion:

$$s_u = \frac{\sum_{x_j \in C_u} s(x_i, x_j)}{|C_u|} \quad (5)$$

With this definition, Algorithm 1 requires n^2 pairwise subsequence similarity calculations. Combining Eqs. (1) and (5) and rearranging the formula, s_u can be written as given in the following equation:

$$s_u = \sum_{t=1}^{l} \sum_{j=1}^{20} f_u^t(a_j) M(x_i(t), a_j) \quad (6)$$

where $x_i(t)$ denotes the amino acid appearing at the tth position of the subsequence x_i and $M(x_i(t), a_j)$ is the entry of similarity matrix for amino acids $x_i(t)$ and a_j. $f_u^t(a_j)$ represents the frequency of amino acid a_j at the tth position of subsequences in cluster u. The complexity of Algorithm 1 becomes $O(nkl)$ where l is the length of the subsequences, k is the number of clusters, and n is the total length of all of the proteins in positive training set. Since l, is an arbitrary but fixed parameter, it can be said that it is $O(nk)$ with respect to the size of the input sequences. k is dependent on the threshold value δ given in Algorithm 1; but it is around 1800 for the default δ value, 8. It is almost constant or varying very slowly with the data size. The second part of the presented method is construction of the feature vectors. Since the probability of each subsequence of the protein against all of the subsequence profiles must be calculated, it again can be implemented in $O(nk)$time. In this case, n represents the length of the given protein to be mapped and k is the number of subsequence profiles. SPMap is linear in the size of the input data. It is very efficient and scalable to handle large datasets.

4.2. Performance Test Results

SPMap has a significant improvement over P2SL for subcellular localization classification. The improvement is both in terms of accuracy and computational efficiency. In order to discretize the subsequence space, P2SL uses self-organizing maps (SOMs) which are hard to train because of the necessity of large training data and convergence problems. As a result different runs on SOM might result in different feature spaces. P2SL is prone to missing some important subsequences since it does not consider all possible subsequences. Since SOM requires numerical vectors, P2SL encodes amino acids as 20 dimensional vectors which causes a 5 length subsequence to be represented as a 100 dimensional vector further complicating the SOM training. SPMap uses clusters of all possible subsequences for discretization of subsequence space instead of SOM in P2SL. Similarity between subsequences are calculated using an amino acid similarity matrix and standard string similarity calculation methods, avoiding high dimensional encoding of subsequences. One of the advantages of SPMap is that it works well on wide range of different classification tasks with the default parameter values. This makes it easier to use without expertise and optimization. Furthermore, our feature space mapping algorithm have only one parameter, the threshold value δ, which has a well performing default value in general.

We also investigated the performance of SPMap on functional classification tasks other than subcellular localization. In order to assess and compare the capabilities of SPMap, we performed tests on G-protein-coupled receptor subfamily level classification. GPCRs are very important targets in drug design but known to be hard to classify, because they have highly diverse family at the sequence level (Moriyama and Kim, 2006). It can be seen that SPMap outperformed other classifiers in both level I and level II GPCR subfamily classification. To our knowledge, at the time of writing this paper, Naïve Bayes approach of Cheng et al. (2005) was the best performing method on the benchmark dataset presented in Karchin et al. (2002).

The application of SPMap on enzyme class classification demonstrated that our method too generates comparable or better results to those obtained by previous studies. The dataset used for the test on 6 major enzyme classes was filtered so that there are no pair of proteins with more than 25% sequence identity. This makes the classification task more difficult especially for the methods that only use sequence or subsequence similarity. Furthermore, SPMap depends solely on the available training data to generate the subsequence feature map, where the method presented in Lu et al. (2007) uses domains that are already available in the databases. Nevertheless, results were interestingly complementary. SPMap achieved very high accuracy when the other methods performed poorly and vice versa. For the second level of enzyme hierarchy SPMap achieved high sensitivity in most of the classes. We used all the available data in fourfold cross validation. As a result, a few classes with comparably large data sizes were biased towards false positives, hence relatively low specificity. Selecting a representative training subset for large classes might enhance the specificity of the classifier.

4.3. Perspectives

Since supervised discriminative methods model the differences between families of positive and negative examples explicitly, they provide better solutions for most of the problems of function classification. Most widely used discriminative method is the support vector machines (SVMs) combined with an appropriate kernel or feature space mapping (Cristianini and Shawe-Taylor, 2000). The main issue in classification of proteins according to their primary sequences is to find a kernel or a feature mapping that captures the information hidden in the important discriminative regions of the given sequences. Since, functionally important regions (catalytic sites, binding sites, structural motifs) are conserved over much wider taxonomic distances than the sequences themselves, conserved subsequences among different proteins are strong indicators of functional or structural similarity. Hence, SPMap pursued a new approach based on distribution of subsequences over a map constructed using the actual protein sequences in the positive training set.

The idea of constructing similarity graphs of subsequences and extracting motifs from the clusters of these graphs was already exploited for DNA sequences (Fratkin et al., 2006). In

SPMap, we did not try to identify the motifs explicitly. We just let the classification algorithm learn which subsequence distributions are in fact discriminative. One advantage of SPMap is that it allows further investigation of these constructed profiles to identify motifs of positive training family. As a feature study, constructed profiles can be investigated to see how similar or different they are, compared to the aligned regions resulting from a multiple sequence alignment of that family of proteins.

One further step may be identifying disordered regions and extracting subsequences from these regions. Most of the active sites, catalytic sites, etc. lies along disordered regions (Dunker et al., 2002; Wright and Dyson, 1999). This would reduce the number of unrelated subsequences hence the noise during the feature map construction.

One reason the discriminative methods do not receive as much attention among the biologists compared to the standard sequence alignment methods is the requirement of handling large number of functional classes. It is almost prohibitive if one wants to perform the classification in a one-versus-one scheme. In this study we preferred to use one-versus-all classification. If the number of classes is large, it would be infeasible to use all of the proteins in the negative classes. One-class classifiers might provide a good solution for this problem.

The use of discriminative classifiers is confined to selecting the correct function among a small set of functional classes. In order to develop a general annotation system with a discriminative approach, one might define a hierarchical classification system over a function ontology structure. Examples of two such annotation systems are Gene Ontology (GO) and Mips Functional Catalogue (FunCat) (Ashburner et al., 2000; Ruepp et al., 2004). Although GO is an intensively used annotation system, implementing such a discriminative framework over GO hierarchy might pose (present) some problems. First, GO describes gene products with fine granularity resulting in thousands of terms. As a result many terms have none or very few gene products. One should carefully filter and generate relevant classes for the classification system. Secondly, GO allows directed acyclic graphs in its hierarchy, further complicating the selection of terms to generate classes for the discriminative system. Being a tree hierarchy with especially relevant terms, FunCat might provide an easier framework to develop a general discriminative annotation framework. Once such a framework is established, each classifier might be extended to incorporate useful information other than the primary sequence, such as structural motifs or structural alignments (Can and Wang, 2004; Sacan et al., 2007).

5. Conclusion

We described a discriminative system for functional classification of protein sequences. It uses a subsequence similarity based feature space mapping, SPMap, to convert protein sequences into vector representations. The main idea was to consider the distribution of the subsequences of a given protein over a set of subsequence profiles as its feature representation. SPMap outperformed P2SL tool in subcellular localization and various well known methods in GPCR subfamily classification. In enzyme class classification SPMap produced better or at least comparable results to some of the existing methods.

Our results showed that using subsequence distributions over a quantized space as a feature space for classification of proteins is an effective method in wide range of different classification problems. Furthermore, the proposed method is computationally efficient and capable of handling large datasets.

Acknowledgement

This study is partially supported by TUBITAK under EEEAG-105E035.

References

Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J., 1990. A basic local alignment search tool. J. Mol. Biol. 215, 403–410.

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

Andrade, M.A., Brown, N.P., Leroy, C., Hoersch, S., De Daruvar, A., Reich, C., Franchini, A., Tamames, J., Valencia, A., Ouzounis, C., Sander, C., 1999. Automated genome sequence analysis and classification. Bioinformatics 15 (5), 391–412.

Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al., 2000. Gene Ontology: tool for the unification of biology. Nat. Genet. 25, 25–29.

Atalay, V., Cetin-Atalay, R., 2005. Implicit motif distribution based hybrid computational kernel for sequence classification. Bioinformatics 21 (8), 1429–1436.

Ben-hur, A., Brutlag, D., 2003. Remote homology detection: a motif based approach. Bioinformatics 19, 26–33.

Blekas, K., Fotiadis, D.I., Likas, A., 2005. Motif-based protein sequence classification using neural networks. J. Comput. Biol. 12 (1), 64–82.

Cai, C., Han, L., Ji, Z., Chen, X., Chen, Y., 2003. SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Res. 31 (13), 3692–3697.

Can, T., Wang, Y.-F., 2004. Protein structure alignment and fast similarity search using local shape signatures. J. Bioinformatics Comp. Biol. 2, 215–239.

Cheng, B.Y.M., Carbonell, J.G., Klein-Seetharaman, J., 2005. Protein classification based on text document classification techniques. Proteins 58 (4), 955–970.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press.

Devos, D., Valencia, A., 2000. Practical limits of function prediction. PROTEINS: Struct. Function Genet. 41, 98–107.

Duda, R.O., Hart, P.E., Stork, D.G., 2000. Pattern Classification, 2nd ed. Wiley–Interscience.

Dunker, A.K., Brown, C.J., Lawson, J.D., Iakoucheva, L.M., Obradovic, Z., 2002. Intrinsic disorder and protein function. Biochemistry 41, 6573–6582.

Fratkin, E., Naughton, B.T., Brutlag, D.L., Batzoglou, S., 2006. MotifCut: regulatory motifs finding with maximum density subgraphs. Bioinformatics 22 (14), e150–e157.

Friedberg, I., 2006. Automated protein function prediction—the genomic challenge. Briefings Bioinformatics 7, 225–242.

Gilks, W.R., Audit, B., de Angelis, D., Tsoka, S., Ouzounis, C.A., 2005. Percolation of classification errors through hierarchically structured protein sequence databases. Math Biosci. 193, 223–234.

Hannenhalli, S.S., Russell, R.B., 2000. Analysis and prediction of functional sub-types from protein sequence alignments. J. Mol. Biol. 303 (1), 61–76.

Henikoff, S., Henikoff, J.G., 1992. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. U.S.A. 89, 10915–10919.

Jensen, L.J., Gupta, R., Blom, N., Devos, D., Tamames, J., Kesmir, C., Nielsen, H., Staerfeldt, H.H., Rapacki, K., Workman, C., Andersen, C.A.F., Knudsen,

S., Krogh, A., Valencia, A., Brunak, S., 2002. Prediction of human protein function from post-translational modifications and localization features. J. Mol. Biol. 319 (5), 1257–1265.

Joachims, T., 1999. Making large-Scale SVM Learning Practical (Book Chapter). Advances in Kernel Methods—Support Vector Learning, MIT Press.

King, R.D., Karwath, A., Clare, A., Dehaspe, L., 2000. Accurate prediction of protein functional class from sequence in the *Mycobacterium tuberculosis* and *Escherichia coli* genomes using data mining. Yeast 17 (4), 283–293.

Karchin, R., Karplus, K., Haussler, D., 2002. Classifying G-protein coupled receptors with support vector machines. Bioinformatics 18 (1), 147–159.

Kunik, V., Solan, Z., Edelman, S., Ruppin, E., Horn, D., 2005. Motif extraction and protein classification, In: Proceedings of the Computational Systems Bioinformatics (CSB), pp. 80–85.

Liu, A.H., Califano, A., 2001. Functional classification of proteins by pattern discovery and top-down clustering of primary sequences. IBM Syst. J. 40 (2), 379–393.

Lu, L., Qian, Z., Cai, Y., Li, Y., 2007. ECS: an automatic enzyme classifier based on functional domain composition. Comput. Biol. Chem. 31, 226–232.

Martin, D.M.A., Berriman, M., Barton, G.J., 2004. GOtcha: a new method for prediction of protein function assessed by the classification of seven genomes. BMC Bioinformatics 5, 178.

Moriyama, E.N., Kim, J., 2006. Protein family classification with discriminant function analysis. In: Gustafson, J.P. (Ed.), Genome Exploitation: Data Mining the Genome. Springer.

Pandey, G., Kumar, V., Steinbach, M., 2006. Computational Approaches for Protein Function Prediction. TR 06–028, Department of Computer Science and Engineering, University of Minnesota, Twin Cities.

Pasquier, C., Promponas, V.J., Hamodrakas, S.J., 2001. PRED-CLASS: cascading neural networks for generalized protein classification and genome-wide applications. Proteins 44 (3), 361–369.

Riley, M.L., Schmidt, T., Wagner, C., Mewes, H.W., Frishman, D., 2005. The PEDANT genome database in 2005. Nucleic Acids Res., Database issue 33, D308–D310.

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Guldener, U., Mannhaupt, G., Munsterkotter, M., Mewes, H.W., 2004. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Res. 32 (18), 5539–5545.

Sacan, A., Ozturk, O., Ferhatosmanoglu, H., Wang, Y., 2007. LFM-Pro: a tool for detecting significant local structural sites in proteins. Bioinformatics 23 (6), 709–716.

Sasson, O., Kaplan, N., Linial, M., 2006. Functional classification prediction: All for one and one for all. Protein Sci. 15, 1–16.

Schomburg, I., Chang, A., Schomburg, D., 2002. BRENDA, enzyme data and metabolic information. Nucleic Acids Res. 30 (1), 47–49.

Tomii, K., Kanehisa, M., 1996. Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. Protein Eng. 9, 27–36.

Wang, J.T.L., Ma, Q., Shasha, D., Wu, C.H., 2001. New techniques for extracting features from protein sequences. IBM Syst. J. 40 (2), 426–441.

Wang, X., Schroeder, D., Dobbs, D., Honavar, V.G., 2003. Automated data-driven discovery of motif-based protein function classifiers. Inf. Sci. 155 (1–2), 1–18.

Wright, P.E., Dyson, H.J., 1999. Intrinsically unstructured proteins: re-assessing the protein structure–function paradigm. J. Mol. Biol. 293, 321–331.