A Novel Method for Protein Fold Recognition Using Sequential Pattern Mining and Optimization Algorithms

Themis P. Exarchos, Costas Papaloukas, Markos G. Tsipouras, Christos Lampros and Dimitrios I. Fotiadis, *Member, IEEE*

Abstract— **Protein classification in terms of fold recognition can be used to determine the structural and functional properties of newly discovered proteins. In this work we propose a method for sequence-based fold recognition which utilizes sequential pattern mining and is implemented using a three stage schema. In the first stage the training set is divided into subsets, each one containing proteins from the same fold only. Then, sequential pattern mining is applied in each of the subsets, generating a set of sequential patterns for every fold under consideration. In the second step, a scoring function evaluates the extracted sequential patterns in order to classify the proteins of the training set. A modification of the Simplex local optimization technique, that takes into account the confusion matrix produced by the training set, is employed to assign a weight factor to each fold, in order to maximize the accuracy on the training set. Finally, in the third step, the test proteins are classified using the sequential patterns extracted from the training set and the scoring function with the optimal fold weights, calculated from the training set. In order to validate the proposed method, an appropriate group of primary protein sequences were taken from the Protein Data Bank. When applying the above method without the use of the optimization step the obtained overall accuracy was 35.9%. When considering the three stage methodology, the overall accuracy was increased to 41.3%.**

I. Introduction

Structure prediction is a challenging task and many different methods have been adopted to address it. Nowadays, we are presented with an exponentially increasing number of protein sequences. However, their structure and biochemical function remains unknown. Structure and function determination is a non-trivial task even for small proteins, so theoretical and computational structure prediction techniques are very useful as they offer a way to relate those proteins to other proteins with known properties. By determining how sequences are related to known proteins we can make predictions of their structural, functional and evolutionary features and therefore classify them to the appropriate structural category [1].

Proteins might have considerable structural similarities even when no evolutionary relationship of their sequences can be detected. This property, when there is similar structure but no obvious homology, is referred to as proteins are sharing the same fold. Methods developed to identify this structural relationship are referred to as fold recognition methods. Finding the fold category where a protein of unknown structure belongs is an indirect way to discover its structure, so fold recognition leads to structure prediction. We could identify two categories of methods in fold recognition, the prediction-based methods [2-3] and the structure-based methods [4-5]. Prediction-based methods estimate the secondary sequence of the target protein as a first step for fold recognition. So these methods use sequence information for both finding the secondary sequence and the correct fold. On the other hand, structure-based methods differ from the first, since they do not use directly any sequence information to detect whether two proteins share a fold or not. Instead they create an energy function describing how well a probe sequence matches a target fold. Besides these two categories it is possible to use purely sequence-based methods [6] or combine different approaches [7].

Sequence-based methods are very common in fold recognition. Several machine learning techniques have been adopted to exploit primary or secondary sequence information, such as genetic algorithms [8], support vector machines [9] and hidden Markov models [10-12]. However, although significant improvement has been made, the accuracy of the existing methods remains low and there is the need for new methods contributing to this field.

In this work, a novel method is presented for protein fold recognition employing data mining techniques and optimization algorithms. Data mining [13] is employed in the form of sequential pattern mining (SPM) [14]. Our method introduces several novelties. The employment of SPM for protein structure analysis offers the potential of discovering new knowledge in the form of patterns. An extracted sequential pattern might correspond to a functionally or structurally important protein region [15]. In

This work was part funded by the European Commission within the NOESIS project: Platform for wide scale integration and visual representation of medical intelligence (IST-2002-507960).

T. P. Exarchos is with the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110. (e-mail: me01238@cc.uoi.gr).

C. Papaloukas is with the Dept. of Biological Applications and Technology, University of Ioannina, Ioannina, Greece, GR 45110. (e-mail: papalouk@cc.uoi.gr).

M. G. Tsipouras is with the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110 (e-mail: markos@cc.uoi.gr).

C. Lampros is with the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110 (e-mail: me00715@cc.uoi.gr).

D. I. Fotiadis is the director of the Unit of Medical Technology and Intelligent Information Systems, Dept. of Computer Science, University of Ioannina, Ioannina, Greece, GR 45110 (+30-26510-98803; fax: +30-26510-97092; e-mail: fotiadis@cs.uoi.gr).

addition, our method uses only the protein's primary structure for classification, whereas other similar approaches make use of the secondary [2], as well as the tertiary structures [16]. Primary structure is easier to be determined and can be easily found in many large databases publicly available (e.g. PDB, Swiss-Prot). Furthermore, the use of the optimization step increases significantly the fold recognition efficiency, compared to the case where only the SPM procedure is employed. As for training and testing we employed a low homology between proteins dataset. The classification results indicate that our method performs more than adequately in terms of accuracy and compares favorably with other similar approaches like the Sequence Alignment and Modeling (SAM) approach [10], which is widely considered as an effective approach for protein classification and fold recognition.

II. Materials and Methods

The proposed method consists of three stages (Figure 1). In the first stage SPM is applied in the training dataset and generates a set of sequential patterns, for every fold under consideration. In the second step, a scoring function is employed that evaluates the extracted sequential patterns and classifies the proteins of the training set. Then, a modification of the Simplex local optimization technique [17], that takes into account the confusion matrix produced by the training set, is employed to assign a weight factor to each fold, in order to maximize the accuracy on the training set. Finally, in the third step, the test proteins are classified using the sequential patterns extracted from the training set and the scoring function with the optimal fold weights calculated from the training set.

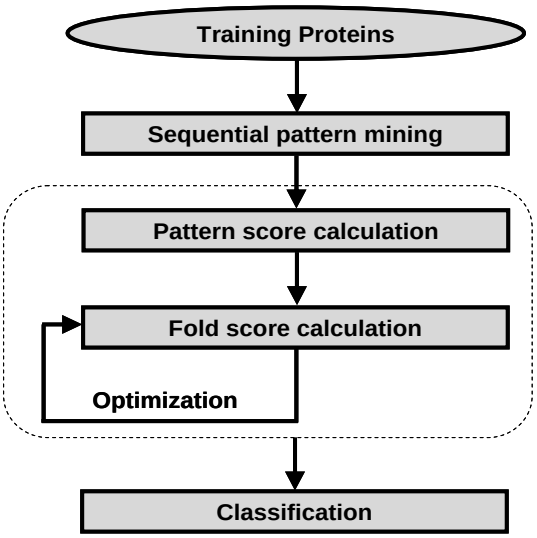

Fig. 1: The flowchart of the proposed method.

A. 1st Stage

In the first stage of the method, the SPM technique is used for protein primary structure analysis. SPM is a common form of local-pattern discovery in unsupervised learning systems, which can be defined as follows [14]: Let $I = \{i_1, i_2, ..., i_n\}$ be a set of items. A subset $X \subseteq I$ is an *itemset* and $|X|$ is the *size* of X. A *sequence* $s = (s_1, s_2, ..., s_m)$ is an ordered list of itemsets, where $s_i \subseteq I, i \in \{1, ..., m\}$. The size, m, of a sequence is the number of itemsets in the sequence, i.e. $|s|$. The length l of a sequence $s = (s_1, s_2, ..., s_m)$ is defined as $l = \sum_{i=1}^{m} |s_i|$. A sequence with length l is called an l-sequence. In our problem the input sequences are the protein primary structures and the set of items I is the 20 amino acids which compose the protein primary structures plus one for the unknown aminoacid. An itemset in a transaction (observation) consists of a single item (one of the 21 letters).

In SPM, a database D is a set of tuples *(sid, tid, X)*, where *sid* is a sequence-*id*, *tid* is a *transaction-id* based on the transaction time and X is an itemset such that $X \subseteq I$. Each tuple in D is referred to as a *transaction*. For a given sequence-*id*, there are no transactions with the same transaction *id*. All the transactions with the same *sid* can be viewed as a sequence of itemsets ordered by increasing *tid*. Thus, an analogous representation for the database is a set of sequences of transactions and we refer to this dual representation of D as its *sequence representation*. In our case, the database D consists of protein primary structures and every one is given a sequence *id*, while the *tid* is the position of the amino acid in the protein primary structure, rather than the time.

A sequence $s_a = (a_1, a_2, ..., a_n)$ is contained in another sequence $s_b = (b_1, b_2, ..., b_m)$ if there exist integers $1 \le i_1 < i_2 < ... < i_n \le m$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, $..., a_n \subseteq b_{i_n}$. If sequence s_a is contained in sequence s_b, then we call s_a a *subsequence* of s_b and s_b a *supersequence* of s_a. The *support* of a sequence s_a in the sequence representation of a database D is defined as the percentage of sequences $s \in D$ containing s_a. The support of s_a in D is denoted by $sup_D(s_a)$. Given a support threshold *minSup*, a sequence s_a is called a *frequent sequential pattern* on D if $sup_D(s_a) \ge minSup$. The problem of mining sequential patterns is to find all frequent sequential patterns for a database D, given a support threshold *sup*.

Several constraints can be incorporated when mining for sequential patterns [18]. One of the simplest constraints applied is the gap constraint. This constraint imposes a limit in the maximum distance between two consecutive itemsets in the sequence. This simple constraint is very useful to reflect the impact of some item on another one, in particular, when each transaction occurs at a specific instant of time. When using gap constraints, the notion of *contained in* is adapted: a sequence $s_a = (a_1, a_2, ..., a_n)$ is a *δ-distance subsequence* of $s_b = (b_1, b_2, ..., b_m)$, if there exist integers

$1 \le i_1 < i_2 < ... < i_n \le m$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, $..., a_n \subseteq b_{i_n}$ and $i_k - i_{k-1} \le \delta$. A sequence s_a is a contiguous subsequence of s_b if s_a is a 1-*distance subsequence* of s_b, i.e. the items of s_a can be mapped to a contiguous segment of s_b. Using δ=1 (i.e. maximum gap=1) the possibility of having gaps between consecutive items is eliminated. Similar to the maximum gap constraint is the minimum gap constraint, which states that the distance between two consecutive items must be more than a specified value ($i_k - i_{k-1} \ge \delta'$).

Several algorithms have been reported in the literature which implement the above described SPM procedure [14,19,20]. However, little work has been done in constrained SPM [21-23]. An algorithm that performs efficient and effective constrained SPM is the cSPADE algorithm [21]. cSPADE is based on the SPADE algorithm [24], and finds the set of all frequent sequences with constraints, such as the minimum and maximum gaps between sequence items. The cSPADE algorithm uses efficient lattice search techniques and simple join operations on *id*-lists. As the length of a frequent sequence increases, the size of its *id*-list decreases, resulting in very fast joins. The performance of the cSPADE algorithm is generally superior, compared to other constrained SPM approaches.

We employed the cSPADE algorithm in order to extract sequential patterns from the training set. The training set is divided into subsets, each one containing proteins from the same fold only. Then, cSPADE generates one set of sequential patterns for every fold under consideration. These patterns constitute the features to be used to classify the unknown proteins. Several experiments were performed, concerning the gap and the support constraints. It should be mentioned that even if SPM is an unsupervised technique, we employed it in a supervised manner, since we generated sequential patterns for each category (fold) separately. A *pattern* extracted from a *fold*, indicates an implication (rule) of the form $pattern \Rightarrow fold$.

B. 2nd Stage

In the second stage of the method, the extracted sequential patterns are employed to classify the proteins of the training set. A scoring function is utilized [25], that takes into account the length of a pattern and the number of patterns extracted from each fold. If a pattern is contained in a protein, the score of this protein with respect to the class (fold) of the pattern is increased by:

$$score_i^j = \left(\frac{length\ of\ the\ pattern_i^j - 1}{\#\ of\ fold\ i\ patterns} \right), \quad (1)$$

where i represents the fold, j represents the pattern of a fold and $pattern_i^j$ denotes the j^{th} pattern of the i^{th} fold. We subtract one from the length of the pattern, in order to assign the minimum score, which is 1, to the minimal pattern, whose length is 2. We calculate the total scores for every fold and for every protein in the training set and we produce the scoring matrix C whose size is $v \times m$, where v is the number of proteins in the training set and m is the number of folds. The (k,i) element of the C matrix denotes the score of the k^{th} protein (training set) received for the j^{th} fold.

Then based on the following pseudocode, we produce the confusion matrix for the training set:

```
for i = 1:m
  for k = 1:v
    if Θ(i)×C(k,i) == max(Θ(i)×C(k,:))
    confusion(i,Class(k)) = confusion(i,Class(k))+1
    END
  END
END.
```

Θ is a vector with m elements containing the fold weights, i.e. $\Theta(i)$ is the weight of the i^{th} fold, with $i=1,2...m$. *Class* (annotation) is a vector with v elements, so $Class(k)$ contains the class of the k^{th} protein for $k=1,..., v$.

Initially all $\Theta(i)$ are set to one and all $confusion(i,j)$, for $i, j = 1 \ldots m$ are set to zero. The goal of the optimization step is to maximize an objective function, with the $\Theta(i)$ as the parameters. The following objective function was employed:

$$F(\Theta(i)) = v - \sum_{i=1}^{m} confusion(i,i). \quad (2)$$

The above function tries to maximize the number of the diagonal elements of the confusion matrix, with respect to the parameters $\Theta(i)$.

For the optimization, we employed the simplex search method [17]. It is a direct search method that does not use numerical or analytic gradients. If $\Theta \in \mathbb{R}^n$, a simplex in the n-dimensional space is characterized by the $n+1$ distinct vectors that are its vertices. In *two*-space, a simplex is a triangle; in three-space, it is a pyramid etc. At each step of the search, a new point in or near the current simplex is generated. The function value at the new point is compared with the function's values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, giving a new simplex. This step is repeated until the diameter of the simplex is less than a specified tolerance.

C. 3rd Stage

In the 3rd stage, our method classifies a protein of unknown class (fold) in only one among all m classes. The classification of the test proteins is similar to the classification of the proteins in the training set. Having found the optimal vector for the fold weights

TABLE I: THE DATASET USED AND THE CORRESPONDING TRAINING AND TEST PROTEINS.

Fold	Index	Train	Test
Globin-like	a1	21	11
Cytochrome c	a3	20	10
DNA-binding 3-helical bundle	a4	103	52
Four-helical up-and-down bundle	a24	28	15
EF-hand	a39	31	15
SAM domain-like	a60	25	12
Alpha-alpha superelix	a118	32	16
All alpha proteins		**260**	**131**
Immunoglobin-like beta sandwich	b1	132	66
Common fold of diphtheria toxin/transcription factors/cytochrome f	b2	20	10
Galactose-binding domain-like	b18	21	10
ConA-like lectins/glucanases	b29	24	12
SH3-like barrel	b34	44	22
OB-fold	b40	61	31
Trypsin-like serine proteases	b47	25	12
PH domain-like	b55	24	12
Double-stranded beta-helix	b82	28	14
Nucleoplasmin-like	b121	27	14
All beta proteins		**406**	**203**
Overall		**666**	**334**

Θ_{opt} from the training set (2nd stage), we employ the scoring function, shown in Eq. (1), in order to produce the scoring matrix C for the test set. Then, using the above mentioned pseudocode and instead of $\Theta(i)$, the $\Theta_{opt}(i)$, the classification of the test sequences is realized.

It should be mentioned that the score of a protein with respect to a fold is calculated based on the number of sequential patterns of this fold contained in the protein. The higher the number of patterns for a fold contained in a protein, the higher the score of the protein for this fold. Some adjustments and weightings are required when calculating the score. The length of the pattern in the nominator causes longer sequential patterns more significant than shorter ones. Also, the score of a protein with respect to a fold is normalized by dividing it with the number of sequential patterns extracted from this fold.

The above scoring function is a heuristic one, selected after a series of experiments. We utilized also the times a sequential pattern is contained in the sequence raised in the power of *n (n=1,2,…)*, the logarithm of the length of the pattern, the length of the pattern raised in the power of n *(n=1,2,…)*, the support of the pattern and others, but all these reported lower classification results.

III. DATASET

In order to validate the proposed classifier, an appropriate group of primary protein sequences was taken from the Protein Data Bank (PDB) [26]. All members of this group correspond to a specific fold at the Structural Classification of Proteins (SCOP) database [27]. As protein members we used those included in the ASTRAL SCOP 1.69 dataset, where no proteins with more than 40% similarity among them are contained. The complete dataset used in the current study is shown in Table I. Specifically the 17 most populated SCOP folds with at least 30 members, from classes A and B (A helixes and B sheets respectively) were used to derive the training and test data. From the 1,000 proteins in total, two thirds from each category were used for training, while the rest for evaluation (Table I).

TABLE II: THE NUMBER OF THE EXTRACTED SEQUENTIAL PATTERNS AND THE OVERALL RESULTS OF OUR METHOD FOR DIFFERENT VALUES OF THE MAXIMUM GAP (MAX_GAP) CONSTRAINT IN THE TRAINING SET.

Max-gap	# of Patterns	Acc1* (%)	Acc2** (%)
1	1568	36.5	40.8
2	3670	38.4	60.8
3	7404	54.4	65.8
4	17542	69.1	77.9
5	38557	67.6	78.1

*The accuracy of the method when only the SPM procedure is used for fold recognition.
**The accuracy of the method when both the SPM procedure and the optimization step are employed for fold recognition.

TABLE III: THE OVERALL RESULTS OF OUR METHOD WHEN APPLIED IN THE TEST SET FOR DIFFERENT VALUES OF THE (MAX_GAP) CONSTRAINT.

max-gap	Acc1* (%)	Acc2** (%)
1	22.5	22.5
2	18.3	32.0
3	27.0	38.0
4	35.9	41.3
5	31.1	39.2

* The accuracy of the method when only the SPM procedure is used for fold recognition.
**The accuracy of the method when the SPM procedure and the optimization step are employed for fold recognition.

IV. RESULTS

Our method has been evaluated in the above described dataset. We set the minimum support to 50%, (i.e. the pattern should be present in at least half of the training proteins), the minimum gap to 1, (which is the minimum value for this type of gap) and we tried several values for the maximum gap (max-gap). In Table II we present the number of the extracted sequential patterns, as well as, the results of our method when the training set is used, for several values of max_gap. In Table III the results concerning the test set are presented. For the different values of max_gap, the overall accuracy increases when the optimization step is employed. Specifically, when only SPM was employed (without the optimization step), the average accuracy was 22.5% using max_gap=1 (i.e. no gaps between the aminoacids), 18.3% with max_gap=2, 27.0% with 3, 35.9% with 4 and 30.5 using max_gap=5. When employing the SPM and the optimization step, the average accuracy increased to 22.5% using max_gap=1, 32.0% with max_gap=2, 38.0% with 3, 41.3% with 4 and 39.2 with 5. The best results were obtained using max_gap=4 and are shown in Table IV. In addition, Table IV shows the number of the extracted sequential patterns and the corresponding performance of the classifier (using the optimal parameters).

As we can see, the number of patterns varies significantly among the folds and this is the reason for using the number

TABLE IV: THE NUMBER OF SEQUENTIAL PATTERNS FOR EVERY FOLD UNDER CONSIDERATION FOR THE OPTIMAL PERFORMANCE CLASSIFIER, THE OVERALL PERFORMANCE ACHIEVED IN THE TRAINING AND THE TEST SETS AND THE TOP1-TOP3 OVERALL ACCURACIES.

	Training set			Test set			
Index	**Patterns**	**Acc1*** **(%)**	**Acc2**(%)**	**Acc1*(%)**	**Acc2**(%)**	**Top2*****	**Top3*****
a1	687	76.2	66.7	36.4	36.4	36.4	45.5
a3	623	80.0	85.0	70.0	60.0	70.0	80.0
a4	500	62.1	70.1	38.5	50.0	73.1	78.9
a24	860	92.9	89.3	20.0	13.3	20.0	26.7
a39	476	83.9	80.7	80.0	66.7	86.7	86.7
a60	559	84.0	92.0	33.3	25.0	33.3	33.3
a118	1381	59.4	90.6	18.8	50.0	62.5	68.8
	5086	**72.3**	**79.2**	**40.5**	**45.0**	**60.3**	**65.7**
b1	742	65.9	74.2	57.6	66.7	71.2	78.8
b2	1364	90.0	90.0	30.0	0.0	10.0	20.0
b18	1384	81.0	95.2	10.0	20.0	20.0	60.0
b29	1525	66.7	91.7	8.3	41.7	66.7	66.7
b34	356	59.1	61.4	31.8	40.9	54.6	63.6
b40	883	55.7	55.7	19.4	12.9	35.5	51.6
b47	1458	92.0	96.0	66.7	66.7	83.3	83.3
b55	695	75.0	91.7	0.0	0.0	16.7	58.3
b82	1884	39.3	85.7	0.0	14.3	28.6	35.7
b121	2165	81.5	88.9	21.4	35.7	71.4	78.6
	12456	**67.0**	**77.1**	**33.0**	**38.9**	**52.7**	**64.5**
	17542	**69.1**	**77.9**	**35.9**	**41.3**	**55.7**	**65.0**

*The accuracy of the method when only the SPM procedure is used for fold recognition.

**The accuracy of the method when both the SPM procedure and the optimization step are employed for fold recognition.

***The Top2 and Top3 accuracies of the method when both the SPM procedure and the optimization step are employed for fold recognition.

of sequential patterns of $fold_i$ in the denominator of the scoring function. Also, in Table IV, the classification results for each fold are presented as Top1, Top2 and Top3 accuracy (Top3 accuracy corresponds almost to the 20% of the total number of classes). Topk accuracy is computed by considering a classification as correct even if the actual (true) fold receives a score between the 1st and k^{th} highest ones. The Topk accuracy provides the k most probable folds that the unknown protein belongs to. In our case Top2 overall accuracy is 55.7% and Top3 overall accuracy is 65.0%. Using the same datasets and in order to compare the efficiency of the proposed approach, we employed also a SAM model for the same task. Our approach reported overall accuracy 41.3%, with max_gap 4, while SAM's overall accuracy was 35.0%.

V. DISCUSSION

We developed a novel method for protein fold recognition that classifies unknown proteins into 17 candidate folds based on sequential pattern mining and optimization algorithms. The SPM technique was employed using the cSPADE algorithm in order to mine the sequential patterns. Using a simple scoring function which utilizes the extracted sequential patterns and an optimization algorithm to compute the optimal class weights, the unknown proteins are classified into the corresponding fold. To evaluate the method, an appropriate group of protein primary structures was acquired from the PDB. Using the same datasets we employed also a SAM model for the same task. Our method exhibited an overall accuracy of 41.3% while SAM's overall accuracy was 35.0%.

The SPM approach employed in this work is suitable for analyzing biosequences like protein primary structures due to their sequential nature and is able to discover strong sequential dependencies (patterns) between aminoacids. Furthermore, the training phase of the method, i.e. the determination of the sequential patterns, is a fast procedure due to the cSPADE algorithm. Generally, SPM is a time consuming process and requires high computational effort which is increased exponentially as longer sequences need to be mined. The lattice search techniques and the simple joins that the cSPADE algorithm employs, handle the two above aspects effectively.

In what concerns the employed optimization procedure for the calculation of the optimal fold weights, the results prove its efficacy. Specifically, using optimization the overall accuracy increased 5.4% when the test set was used.

However, our method imposes two major limitations. When classifying an unknown protein, all the sequential patterns extracted from all the folds in the training phase, should be checked in order to find out if they are contained in the protein. Since the number of the extracted sequential patterns was considerable, a large number of comparisons should be performed in order to reach to the classification decision. Moreover, the utilization of SPM, besides finding valid and causal relationships in the biological data, it will also find all the spurious and particular relationships among the data in the specific dataset. For this reason, results of any SPM procedure should be considered as exploratory and hypothesis-generating.

Further improvement might focus on the utilization of the secondary protein structure in addition to the primary one. This would of course increase the complexity of the method,

but might produce higher classification results. Another issue is the use of global optimization techniques and also the implementation of a more sophisticated objective function.

REFERENCES

[1] C. Branden, *Introduction to protein struct*ure, Garland, Sweeden, 1999.

[2] J. Hargbo and A. Elofsson, "Hidden Markov Models That Use Predicted Secondary Structures For Fold Recognition," *Proteins,* vol. 36, pp. 68-87, 1999.

[3] R. Karchin, M. Cline, Y. Mandel-Gutfreund and K. Karplus, Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry, *Proteins,* vol. 51, pp. 504-514, 2003.

[4] H. Flöckner, F. Domingues and M.J. Sippl, Proteins folds from pair interactions: a blind test in fold recogition. *Proteins: Structure Functions and Genetics,* vol. 1, pp. 129–133, 1997.

[5] J. Xu, "Fold Recognition by Predicted Alignment Accuracy," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,*" vol. 2(2), pp. 157-165, 2005.

[6] K. Karplus, S. Kimmen, C. Barrett, M. Cline, D. Haussler, R. Hughey, L. Holm, and C. Sander, "Predicting protein structure using hidden Markov models," *Proteins: Structure, Function, and Genetics,* Suppl. 1, pp. 134-139, 1997.

[7] A. Elofsson, D. Fischer, D. W. Rice, S. M. LeGrand and David Eisenberg. "A study of combined structure-sequence profiles," *Folding & Design,* vol. 1, pp. 451-461,1996.

[8] T. Dandekar, and P. Argos, "Identifying the tertiary fold of small proteins with different topologies from sequence and secondary structure using the genetic algorithm and extended criteria specific for strand regions," *Journal of Molecular Biology,* vol. 256, pp. 645-660, 1996.

[9] C. Ding, and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks," *Bioinformatics,* vol. 17, pp. 349-358, 2001.

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

[11] E. Lindahl, and A. Elofsson, "Identification of related proteins on family, superfamily and fold level," *J. Mol. Biol.,* vol. 295, pp. 613-625, 2000.

[12] R. Karchin, M. Cline, Y. Mandel-Gutfreund, and K. Karplus, "Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry," *Proteins*, vol. 51, pp. 504-514, 2003.

[13] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, From data mining to knowledge discovery: an overview, in Advances in Knowledge discovery and data mining, AAAI Press/MIT Press (1996) 1-36.

[14] R. Agrawal and R. Srikant, "Mining sequential patterns," In *11th Intl. Conf. on Data Eng.*, 1995, pp. 3-14.

[15] K. Wang, Y. Hu, J. Hu Yu, "Scalable Sequential Pattern Mining for Biological Sequences," *Proceedings of the 13th ACM conference on Information and knowledge Management*, USA, 2004, pp. 178–187.

[16] Z. Aung, and K.L. Tan, "Automatic 3D protein structure classification without structural alignment," Journal of Computational Biology, Mary Ann Liebert, Inc Publishers, June 2005.

[17] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, Vol. 9, Number 1, pp.112-147, 1998.

[18] R. Srikant, and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," In Proc. 5th Int. Conf. Extending Database Technology, EDBT, vol. 1057, Springer-Verlag, 1996, pp. 3-17.

[19] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, "Mining Sequential Patterns by Pattern-Growth: The prefixSpan Approach," IEEE Trans. Knowledge Data Eng., vol. 16, pp. 1424-1440, 2004.

[20] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick, "Sequential pattern Mining Using Bitmaps," In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Canada, July 2002, pp. 429-435.

[21] M. J. Zaki, "Sequence mining in categorical domains: incorporating constraints," In *Proc. of the 9th International Conference on Information and knowledge management*, USA, 2000, pp. 422 – 429.

[22] H. Mannila, H. Toivonen, and I. Verkamo, "Discovery of frequent episodes in event sequences," Data Mining and Knowledge Discovery: An International Journal, vol. 1(3), pp. 259–289, 1997.

[23] M. Garofalakis, R. Rastogi, K. Shim, "SPIRIT: Sequential Pattern Mining with Regular Expression Constraint," In Proceedings of the 25th International Conference on Very Large Databases, 1999, pp. 223-234.

[24] M.J. Zaki, "Efficient enumeration of frequent sequences," In 7th Intl. Conf. Info. and Knowledge Management, Nov 1998, USA, pp. 68 – 75.

[25] V. Shin-Mu Tseng, C.-H Lee, "CBS: A New Classification Method by Using Sequential Patterns," in proc .of SIAM International Data Mining Conference, California, USA, 2005.

[26] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Res*., vol. 28, pp. 235-242, 2000.

[27] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, pp. 536-540, 1995.