Control and Cybernetics

vol. **36** (2007) No. 1

New machine learning methods for prediction of protein secondary structures[1]

by

Jacek Błażewicz[1,2], Piotr Łukasiak[1,2] and Szymon Wilk[1]

[1]Institute of Computing Sciences, Poznań University of Technology
ul. Piotrowo 3a, 60-965 Poznań, Poland

[2]Institute of Bioorganic Chemistry, Polish Academy of Sciences
ul. Noskowskiego 12/14, 61-704 Poznań, Poland

Abstract: The first and probably the most important step in predicting the tertiary structure of proteins from its primary structure is to predict as many as possible secondary structures in a protein chain. Secondary structure prediction problem has been known for almost a quarter of century. In this paper, new machine learning methods such as LAD, LEM2, and MODLEM have been used for secondary protein structure prediction with the aim to choose the best among them which will be later parallelized in order to handle a huge amount of data sets.

Keywords: machine learning, prediction of protein structures, LAD, LEM2, MODLEM

1. Introduction

Proteins represent one of the most exciting and complex products of nature. They are the machinery of life, in the sense that they are involved in all the processes, which regulate the functional cycles of living organisms. We can observe proteins guiding the catalysis of biochemical reactions, signal transduction and transmission or the correct expression of genetic information. Basically, proteins are the result of genetic code translation and the function of a protein is closely related to its structure. It is therefore clear that being able to predict structural features of a proteome could have a strong positive impact on any attempt to discover unknown characteristics of a genome. Unfortunately, finding the structure in laboratory is extremely difficult, cost prohibitive and can take months or years in some cases. The ability to provide effective computational tools for protein structure prediction is a key to overcome these experimental problems and to guide a part of the future scientific effort in molecular biology.

[1]Partially supported by KBN grant 3T11F00227

The growth of interest in 'non conventional computer algorithms' (in particular those offered by AI and machine learning) applied to biological domains relies on the so called "data revolution". Since the beginning of the first genomic and sequencing projects, the amount of information available concerning biological molecules such as DNA, RNA and proteins (primarily in the form of sequences) continues to grow at an exponential rate. Following this situation, bioinformatics (the fusion of computer science and biology comprising many algorithmic ideas coming from different disciplines and applied to molecular biology problems) has emerged as a key discipline for performing inference tasks on biological sequences such as automatic analysis, interpretation and prediction (Baldi and Brunak, 2001). Considering the variety of protein structure prediction problems, machine learning plays an increasingly significant role. Its techniques have already been applied in a large number of important applications like computational gene finding (Pevzner, 2000), prediction of DNA active sites, sequence clustering, analysis of gene expression data (Baldi and Brunak, 2001) and knowledge discovery in biological domains. This paper is intended to show a possibility of use of a new machine learning algorithm in secondary protein structure prediction. Three approaches used are: LAD, LEM2 and MODLEM. The aim of the study was to choose the best among them for the problem in question. The winning approach will be parallelized in further studies, in order to handle huge data sets (available protein chains).

The paper is organized as follows. Section 2 describes basic elements of structural biology and the protein folding problem. Section 3 reviews the known methods used for solving the problem. Section 4 lists machine learning algorithms considered in our study, describes data used for experiments and presents transformations of the data required by the algorithms, while Section 5 reports the results of a set of experiments. Section 6 presents a discussion and conclusions.

2. Description of the problem

Proteins are macromolecules built of 20 basic units, called *amino acids*. All amino acids share the same generic chemical properties. There is a central *carbon atom* (C_α) attached to a hydrogen atom, an *amino group* (NH_2), a *carboxy group* ($COOH$) and a lateral chain or *residue* (R), which distinguishes one amino acid from the others. Every residue is assigned a 3-letter or 1-letter code. During DNA transcription-translation phases, proteins are assembled through the formation of peptide bonds, where the carboxy group of one amino acid is joined with the amino group of another to release water. So, we can talk of a protein as a polypeptide formed by a backbone (the sequence of peptide bonds) and a side chain (the sequence of residues).

The structure of a protein may be represented hierarchically at three structural levels. The primary structure is the sequence of amino acids in the polypeptide chain and it can be described as a string on the finite alphabet Σ_{aa}, with

$|\Sigma_{aa}| = 20$. The secondary structure refers to the set of local conformation motifs of the protein and schematizes the path followed by the backbone in the space. The secondary structure is built from three main classes of motifs: *α-helix (H)*, *β-strand (E)* and *loop* or *coil (C)*. An α-helix is built up from one continuous region in the sequence through the formation of hydrogen bonds between residues in position i and $i+4$. A β-strand does not represent an isolated structural element by itself, because it interacts with one or more β-strands (which can be distant in a sequence) to form a pleated sheet called β-sheet. Strands in a β-sheet are aligned adjacent to each other such that their amino acids have the same biochemical direction (parallel β-sheet) or alternating directions (anti-parallel β-sheet). The α-helices and β-strands are often connected by loop regions, which can significantly vary in length and structure, having no fixed regular shape as the other two elements. Every amino acid in the sequence belongs to one of the three structural motifs, so the secondary structure can be reduced to a string over the alphabet Σ_{ss}= {H; E; C}, being of the same length as the primary structure. The most important description level and main objective of experimental and prediction efforts is the so-called tertiary structure. It describes the
3-dimensional organization of polypeptide chain atoms (both backbone and side chain atoms). It is the result of the combinations of secondary structure elements due to interactions between the amino acids and the solvent (the environment of the protein, being water). Some proteins allow also for another level of description, called quaternary structure. It is introduced to describe the complex spatial conformation of a protein composed of many distinct polypeptide chains (multimeric protein). Chains of a multimeric protein are often called protein sub-units.

Protein chains are subject to a folding process. Starting from the primary structure, they are capable of organizing themselves into a unique 3-dimensional stable (native) conformation which is responsible of their biological functions. According to the experimentally confirmed Anfinsen's hypothesis (Anfinsen, 1973), the tertiary structure depends only on lower order structures, plus the native solution environment. This means that the primary sequence contains all the information needed to reach the final stable conformation (a corollary could involve also the functional dependence of the secondary structure on the primary one).

This gave rise to the folding problem, the prediction of protein's tertiary structure from its amino acid sequence. Finding a solution to the folding problem is one of the most difficult and challenging open problem in structural genomics. Despite many decades of intensive research efforts, the problem has not a general solution yet. An evidence for this claim is the rapidly increasing sequence-structure gap. The number of proteins for which the sequences are known is about a half a million (Bairoch and Apweiler, 2000), whereas the number of protein structures deposited in public databases is less than twenty thousands (Berenstein et al., 1997). Excluding experimental difficulties, the

reason for this impressive difference is largely due to the lack of a comprehensive theory of the folding. At present, we only know a few reliable facts about the folding mechanism. The most accurate tools for doing protein folding are knowledge-based methods, i.e. comparative (homology) modeling and fold recognition. The next section overviews the homology based methods used so far for solving the protein structure prediction problem.

3. The overview of homology based methods

Homology modeling techniques (Sanchez and Sali, 1997) assign a 3D structure to a novel protein by searching for proteins with similar sequences. The basic assumption is that proteins with similar sequences adopt similar folds and functions. The unknown fold is then modeled using the structure of homologs as a template. However, it is frequently found that two proteins with low sequence identity have similar functions and three-dimensional structures. In this case, fold recognition (threading) techniques can be applied (Fisher and Eisenberg, 1996; Jones, 1999), which select the target structure as the most compatible one among those present in a library. Overall, it is estimated that knowledge based methods can be applied to only about 20-30% of novel proteins. In the majority of cases, the structure of a novel protein must be assigned *ab initio*, not relying on the protein having a fold similar to the known one. Typical *ab initio* approaches compute 3D structure by doing searches on the space of the protein allowable conformations. Simulations employ the physical laws of motion in carefully devised potential fields (molecular dynamics). Unfortunately, exact numerical calculations are beyond the possibilities of the present and near future computers and results can be obtained only for small proteins. Another obstacle is represented by the extremely small differences in energy between native and unfolded structures (about 1 kcal/mol).

Secondary structure prediction was one of the first and most important problem faced by computer learning techniques. Its significance can be understood by looking at the variety of prediction systems that were developed over time. Roughly, one can distinguish between the 1st, 2nd and 3rd generation methods. First generation methods are those making predictions only on the basis of information coming from a single residue, either in the form of statistical tendency to appear in an α-helix, β-strand and coil region (Garnier et al., 1978), or in the form of explicit biological expert rules (Lim, 1974). Average accuracy of these methods (known as Q_3 index) was limited to 55%.

Second generation methods basically apply the connection architecture, taking into account local interactions by means of an input-sliding window with encoding. Values in the output layer discriminate each residue as belonging to one of the three states α-helix (H), β-strand (E) and coil (C). The first original work (Qian and Sejnowski, 1988) reports $Q_3 = 62.7\%$, and in Riis and Krogh (1996), using techniques to reduce over fitting and incorporate prior knowledge, the authors achieve an accuracy of 66.3%. One of the major difficulty of these

methods is a correct location of β-strands, because they are predominantly determined by long-range interactions. It is generally assumed that 65% of a secondary structure depends on local interactions.

The 3rd generation methods started exploiting the information coming from homologous sequences. The basic observation is that the secondary structure within a family of evolutionary related proteins is more conserved than the sequences. The evolutionary pressure to conserve function has favored mutations preserving structural characteristics. This information is processed first by doing a multiple alignment between a set of similar sequences and extracting a matrix of profiles (PSSM). Each matrix column represents the input given to the network for the corresponding sequence position. PHD (Rost, 1996) was the first method incorporating profile-based inputs and going beyond 70% in accuracy. The system is composed of cascading networks. A final stage takes a jury decision averaging the outputs from independently trained models. Other well-known profile based methods are PSI-PRED (Jones, 1999), which uses two neural networks to analyze profiles generated from a PSI-BLAST search, JNet (Cuff and Barton, 2000) and SecPred. An alternative adaptive model is employed in Baldi et al. (1999). It employs bi-directional recurrent neural networks, a non-causal architecture that exploits upstream and downstream dependencies in the form of the contextual knowledge. This method is potentially capable of capturing long-range information stored in hidden state variables. At present, almost all 3rd generation methods lie in the accuracy range of 76-78%. Generally, it is believed that the hypothetical performance limit of SS predictors is not at 100% but at around 90%. This belief stems from noise and inconsistencies that are present in available sequence archives.

There are other predictors, which are not strictly based on neural network implementations. NNSP (Salamov and Solovyev, 1995) uses a nearest-neighbor algorithm where the SS state of the residue test segment is assigned scoring information coming from different templates according to their similarities. Template segments are those of proteins with known 3D structures. The web-server JPred (Cuff et al., 1998) integrates six different structure prediction methods and returns a consensus based on the majority rule. The program DSC (King and Sternberg, 1996) combines several explicit parameters in order to get a meaningful prediction. It runs the GOR3 algorithm (Garnier et al., 1978) (an evolution of GOR1 based on information theory applied to local interactions) for every sequence, to provide mean potentials for the three states. A linear combination of the different attributes gives an output, which is subsequently filtered. The program PREDATOR (Frishman and Argos, 1997) is based on the calculated propensities of the 400 amino acid pairs to interact inside an α-helix or one upon three types of β-bridges. It then incorporates non-local interaction statistics and propensities for α-helix, β-strand and coil derived from a nearest-neighbor approach. To use information coming from homologous proteins, PREDATOR relies on local pairwise alignments. Accuracy is claimed to be at 75%.

In principle, Hidden Markov Models (HMM) could be effectively used for SS prediction allowing for the incorporation of syntactic restraints on the form of the output strings. The method of profiles continues to improve as more and more sequences are becoming available (Przybylski and Rost, 2002). The reason for the success of the profile based method seems to be that it captures the fact that protein structures conserve more information than sequences. Because only mutations that do not disrupt the three-dimensional structure of a protein will survive the evolution, sequence divergence under the structural constraints reflects the interactions between amino acid residues of a protein, where the interactions could be either short range or long range in sequence. Now, most secondary structure prediction methods achieving high average performance with Q_3 measure near 80% (Riis and Krogh, 1996; Baldi et al., 1999; Cuff and Barton, 1999; Jones, 1999; Ouali and King, 2000; Pollastri et al., 2002) make use of PSI-BLAST profiles (Altschul et al., 1997) in combination with improvement of prediction algorithms. New machine learning methods such as support vector machine (Hua and Sun, 2001) should also benefit from PSI-BLAST profiles.

Lee (2005) proposed a new kind of HMMs, so called Hidden Markov models with states depending on observations (HMMSDO). HMMSDO may have advantages over HMM in some cases such as prediction of protein secondary structures. When using a HMM to predict protein secondary structure, the observations are regarded as amino acid residues, and the states are regarded as tokens of secondary structure (Asai et al., 1993). According to the basic assumption of biochemistry, i.e. protein secondary structure depends on primary structure, it may be theoretically better to use a HMMSDO than a HMM in this case. At present, no HMM based method is able to outperform neural networks; not surprisingly, the literature in this case reports an improvement not on accuracy, but on the length distribution of predicted segments. In the next Section three new algorithms for the protein secondary structures will be briefly described.

4. New approaches to the protein folding problem

In this paper three rule generation algorithms: LEM2 (Learning from Examples Module ver. 2) used in LERS (Learning from Examples based on Rough Sets) system (Stefanowski, 1998; Grzymała-Busse, 1992), MODLEM (Grzymała-Busse and Stefanowski, 2001) and LAD (Logical Analysis of Data) (Boros et al., 1996; Hammer 1986; Mayoraz, 1995), were used for the prediction of secondary protein structures.

LEM2 and MODLEM are rule induction algorithms that generate a minimal set of rules given a set of positive examples and a set of negative examples. A minimal set of rules is the smallest set of rules that cover all positive examples and do not cover any negative examples. Traditionally, both algorithms have been used together with data analysis methods based on rough set theory to

deal with inconsistent data sets. Rough set techniques have been used to identify approximations of decision classes (either lower or upper); then a selected algorithm has been iteratively invoked for each decision class, with a positive set including examples from a rough approximation of a currently processed class, and a negative set including all the remaining examples.

LEM2 generates rules with conditions in the form of 'attribute = value', so it is suited for symbolic data and may offer poorer performance (in terms of a number of rules and their strength) for numerical data (generated rules may be weak and their number may be high in comparison with the number of examples). To overcome this limitation, numerical data sets are usually discretized before running LEM2. MODLEM is an extension of LEM2 that deals with this limitation. It generates rules with extended syntax of conditions, i.e., 'attribute >= value', 'attribute < value' and 'attribute in set of values' (the algorithm uses entropy or Laplace estimate to evaluate conditions when constructing them). Therefore no prior discretization is required – MODLEM induces rules directly from numerical data. Moreover it can induce more general rules for symbolic data, therefore, resulting sets of rules are better (considering number of rules and their strengths) than rules generated by LEM2 from the same data.

Both algorithms work in a similar way – the main difference is how possible conditions are identified (MODLEM uses much more advanced technique) – and they both follow the "sequential covering" approach, where the positive examples covered by the already induced rules are removed from a considered set. LEM2 and MODLEM are built with two loops. Each iteration of an outer loop starts from finding a working set of positive examples (a set of positive examples with all examples covered by already constructed rules removed). If the working set is not empty, an algorithm identifies possible conditions and in an inner loop it constructs a rule that (possibly) covers the largest number of examples from the working set and that covers no negative examples. The rule is induced in a greedy fashion – in each iteration of the inner loop best available conditions are added one by one. Then the rule is "optimized" (redundant conditions are removed) and added to the set of constructed rules. If the working set is empty (i.e., constructed rules cover all positive examples), the outer loop is finished and the set of rules is "optimized" (redundant rules are removed). Finally, the "optimized" set of rules is returned by the algorithm. Considering the way in which the data used by the algorithms were prepared and preprocessed (described in detail below), they can be treated as belonging to the second generation of prediction algorithms but obtained results are comparable to the ones produced by methods from the third generation. Additionally, rules generated by these algorithms can be easily interpreted by domain experts.

Logical Analysis of Data algorithm consists of three stages: binarization, pattern generation and rule classification. The data binarization stage is needed only if data are in numerical or nominal formats (e.g. color, shape, etc.). To make such problems useful for LAD one has to transform all data into a bi-

nary format. The simplest non-binary attributes are the so-called nominal (or descriptive) ones.

As a result of this stage, all attributes for each observation are changed into binary attributes. After the binarization phase all of the observations that belonged to different classes are still different when binary attributes are taken into account.

In pattern generation stage every pattern is defined by a set of conditions, each of which involves only one of the variables.

The precise definition of a pattern P1 involves two requirements. First, there should be no observation belonging to other classes, satisfying the conditions describing P1, and on the other hand, a vast number of observations belonging to class H should satisfy the conditions describing P1.

Clearly, the satisfaction of condition describing P1 can be interpreted as a sufficient condition for an observation to belong to class H.

The observation is covered by a pattern if it satisfies all conditions describing P1.

The description of a pattern can be given in several forms. Pattern P1 given above is specified by its minimal description, which gives an essential set of conditions identifying it. If any of these conditions is softened or entirely eliminated the characteristic property of the pattern to cover only observations from class H, disappears. On the other hand, all observations covered by pattern P1 satisfy some additional conditions.

Symmetric definitions of positive and of negative patterns lead to symmetric generation procedures. Based on this assumption only a procedure for generating positive patterns is described here. The generation of negative patterns proceeds in a similar way (see Boros et al., 1996).

For the pattern generation stage it is important not to miss any of the "best" patterns. Pattern generation procedure is based on the use of combinatorial enumeration techniques which can follow a "top-down" or a "bottom-up" approach.

The top-down approach starts by associating to every positive observation its characteristic term. Such a term is obviously a pattern, and it is possible that even after the removal of some literals the resulting term will remain a pattern. The top-down procedure systematically removes literals one by one until arriving at a prime pattern.

The bottom-up approach starts with the term that covers some positive observations. If such a term does not cover any negative observation, it is a pattern. Otherwise, literals are added to the term one by one as long as necessary, i.e. until generating a pattern.

In the original LAD method (Boros et al., 1994, 1996, 1997) this stage has been called twice. The first time for positive patterns generated for observations belonging to class A, and the second time for negative patterns generated for observations belonging to class B. In the discussed experiments, one has three sets of secondary structures, thus, this stage had to be modified and patterns have been generated in the first version six times. *Each time an observation from*

one set of a secondary structure played a role of positive examples, the other sets played roles of negative ones. A call for positive observations is repeated three times, each time a different set of secondary structures playing a role of a positive observation. A call for negative observations is also repeated three times but now negative observations consist of the other two sets, respectively.

In classifier construction stage for any particular class there are numerous patterns, which cover only observations belonging to that class. The list of these patterns is too long to be used in practice. Therefore, we restricted our attention to a subset of these patterns, called [class_indicator] model (e.g. *H model*). Similarly, if one studied those observations which do not belong to the particular class, one can consider the *not H model*.

Before this stage is performed, every positive (negative) observation point is covered by at least one positive (negative) pattern, and it is not covered by any negative (positive) patterns that have been generated. Based on that it can be expected that an adequately chosen collection of patterns can be used for construction of a general classification rule. This rule is an extension of a partially defined Boolean function, and will be called below a theory.

A good classification rule should capture all the significant aspects of the phenomenon.

Technical parameters for this stage remain unchanged during the experiment as compared with the original approach (Hammer, 1986), but one had to call this stage three times (each time for a different set of secondary structures). In every call one tried to construct the best classifier for a particular structure.

The same rule as in the original method: winner takes all, is applied to calculate weights of the three functions describing a structure, that each observation belongs to.

To implement the three analyzed methods and extract the basic properties of proteins, examples were obtained from the Dictionary of Secondary Structures of Proteins (DSSP) (Kabsch and Sander, 1983). DSSP contains a description of secondary structures for entries from the Brookhaven Protein Data Base (PDB) (Berenstein et al., 1997). Moreover, it contains data calculated from protein tertiary structures obtained by NMR or X-ray experiments and maintained in PDB.

There are many ways to divide secondary structures into classes. Here we used the most popular one, based on information obtained from DSSP.

Data retrieved from the DSSP set consist of eight types of secondary protein structures. Usually one can reduce them into three main secondary structures and this assumption has been made in this study. The following sets of secondary structures have been created:

- helix (H) consisting of: α-helix (structure denoted by H in DSSP), 3_{10}-helix (G) and π-helix (I);
- $\beta-$strand (E) consisting of E structure in DSSP;
- the rest (X) consisting of structures belonging neither to set H nor to set E.

The first step one has to do, is to prepare a set of observations (based on a protein sequence) to be acceptable by algorithms. When making a transformation from a protein sequence to the set of observations one has to assume that the main influence on the secondary structure have amino acids situated in the neighbourhood of the observed amino acid. We also took into account that some n-mers are known always to occur in the same structure in many proteins, while others do not. Certain 4-mers and 5-mers are known to have different secondary structures in different proteins. To fulfill this assumption and avoid naive mistakes, a concept of windows (King and Sternberg, 1990) of length equal to 7 was used.

Below, an example is presented, illustrating the way a protein chain is changed into a set of observations. Let us consider a protein chain called *4gr1* (in PDB). The first and the last fifteen amino acids in the sequence are shown below:

MKRIGVLTSGGDSPG ... TIDQRMYALSKELSI

For every amino acid the corresponding secondary structure in DSSP is given as follows:

__EEEEEEESS__TT ... ___HHHHHHHHHH__

One may change this structure into secondary structures, involving three main secondary structures only, in the way depicted below:

XXEEEEEEEXXXXXX ... XXXHHHHHHHHHHXX

At the end, a chain consisting of n amino acids is transformed into set consisting of n observations as shown in Table 1.

A window of length 7 generates an observation with 7 attributes ($a_{-3}, a_{-2}, a_{-1}, a_0, a_{+1}, a_{+2}, a_{+3}$) representing a secondary structure corresponding to the amino acid located in place a_0. Of course, at this moment all values of attributes are symbols of amino acids.

As one can see, secondary structures on the boundaries have been omitted. Amino acids situated from the (i-3rd) to (i+3rd) position in the protein sequence, where the considered secondary structure is relevant to the i-th amino acid, create the smallest number of attributes to be used to change protein chain (assumed in experiments) into a unique set of observations without loosing more than 1% of observations from the considered data set. By unique, we mean here the fact that there are no two identical observations belonging to different sets of secondary structures.

All observations were used to create a learning subset or a testing subset. During creation of a learning subset one has to exclude the first three observations and the last three ones (one has not enough information to learn anything). In the testing set, this exclusion is not important because in such a situation one can get a decision for an observation without a complete set of attributes, treating missing values as values playing against him.

Table 1. An example of a transformation from a sequence to a set of observations

#	Condition attributes $a_{-3}a_{-2}a_{-1}a_0a_{+1}a_{+2}a_{+3}$	Code in DSSP	Codes of the three sets of the main secondary structures
1	* * * M K R I		-
2	* * M K R I G		-
3	* M K R I G V	E	-
4	M K R I G V L	E	E
5	K R I G V L T	E	E
6	R I G V L T S	E	E
	...		
313	M Y A L S K E	H	H
314	Y L S K E L	H	H
315	A L S K E L S	H	H
316	L S K E L S I	H	H
317	S K E L S I *	H	-
318	K E L S I * *		-
319	E L S I * * *		-

The last step of the preprocessing is to replace in each observation, symbols of amino acids (treated as attributes) with numbers representing relevant properties of amino acids. All properties are received from ProtScale service at http://expasy.hcuge.ch/cgi-bin/protscale.pl.

During experiment only the physical and chemical properties of the amino acids offered by ProtScale have been taken into account. We considered here the best ten properties obtained form previous experiments (Błażewicz et al., 2001a,b, 2005), which had the most important influence on the created secondary structures (Table 2).

LEM2 required an extra step of preprocessing. As it is not well suited for handling continuous data (it then usually generates a large number of weak rules with poor classification abilities), values obtained in the last phase were discretized using recursive minimal entropy partitioning (Catlett, 1991). The other algorithms could work directly on observations described with real-valued properties.

After the preprocessing stage, the three new algorithms have been used to the problem of protein secondary structure prediction. A comparison of the methods is described in the next section.

5. Results and discussion of the computational experiments

During experiments nine protein chains have been chosen for consideration randomly from the representative set of proteins defined in the benchmark set

Table 2. Properties of amino acids considered in experiments

#	**Description**	**Author(s)**	**Reference**
1	Mobilities of amino acids on chromatography paper (RF).	Aboderin A.A.	*Int. J. Biochem.* 2:537-544 (1971).
2	Normalized consensus hydrophobicity scale.	Eisenberg D., Schwarz E., Komarony M., Wall R.	*J. Mol. Biol.* 179:125-142 (1984).
3	Hydration potential (kcal/mole) at 25 ° C.	Wolfenden R.V., Andersson L., Cullis P.M., Southgate C.C.F.	*Biochemistry* 20:849-855 (1981).
4	Hydrophobicity indices at ph 7.5 determined by HPLC.	Cowan R., Whittaker R.G.	*Peptide Research* 3:75-80 (1990).
5	Hydrophobicity indices at ph 3.4 determined by HPLC.	Cowan R., Whittaker R.G.	*Peptide Research* 3:75-80 (1990).
6	Average surrounding hydrophobicity.	Manavalan P., Ponnuswamy P.K.	*Nature* 275:673-674 (1978).
7	Hydrophobicity scale based on free energy of transfer (kcal/mole).	Guy H.R.	*Biophys J.* 47:61-70 (1985).
8	Retention coefficient in HPLC, pH 7.4.	Meek J.L.	*Proc. Natl. Acad. Sci. USA* 77:1632-1636 (1980).
9	Retention coefficient in TFA.	Browne C.A., Bennett H.P.J., Solomon S.	*Anal. Biochem.* 124:201-208 (1982).

RS126 (Rost, 1993) (accession number given in Appendix). Based on these nine protein chains 2100 observations have been created using the procedure described above and ten-fold cross validation test was applied. In the first series of experiments the partition into folds was done manually. First, the observations were ordered based on their positions in protein chains (i.e., observations belonging to the first protein were put at the beginning of the set, then observations belonging to the second protein were considered etc.). Then the set of 2100 objects was divided into ten subsets (the first 210 objects were selected to the first subset, the second 210 – to the second one, etc.). Results obtained from experiments for each of three classes are presented in Figs. 1 through 3. The Fig. 4 shows overall accuracy of prediction of LEM2, MODLEM and LAD.

As one can see, for class H (Fig. 1) the best average results were obtained using LAD. For each protein chain prediction accuracies using LAD ranged from 38% to 45%. In this case LAD was better than MODLEM, but for some proteins the highest accuracy of prediction were obtained using LEM2 (50%). Unfortunately, these impresive results were obtained only for some subsets of

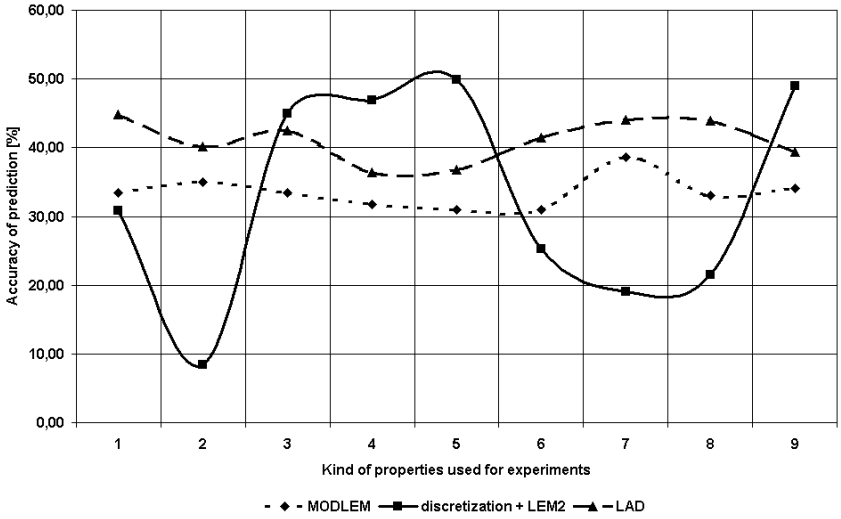

Figure 1. Accuracy of prediction for class H.

Figure 2. Accuracy of prediction for class E.

observations. In other cases performance of this algorithm was very poor (accuracy below 10%).

For class E (Fig. 2) the best results were obtained by LEM2. For some subsets average accuracy of prediction was over 80 %. LAD and MODLEM were worse than LEM2. LAD was better than MODLEM by about 5%.

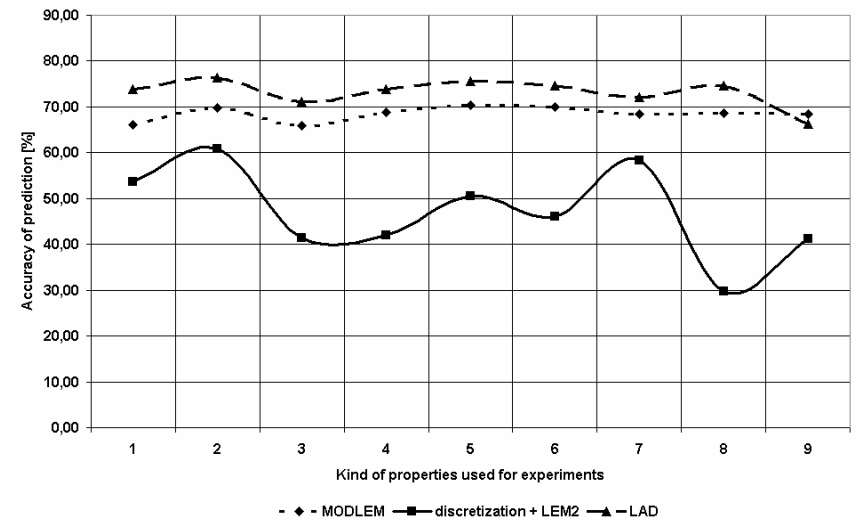

Figure 3. Accuracy of prediction for class X.

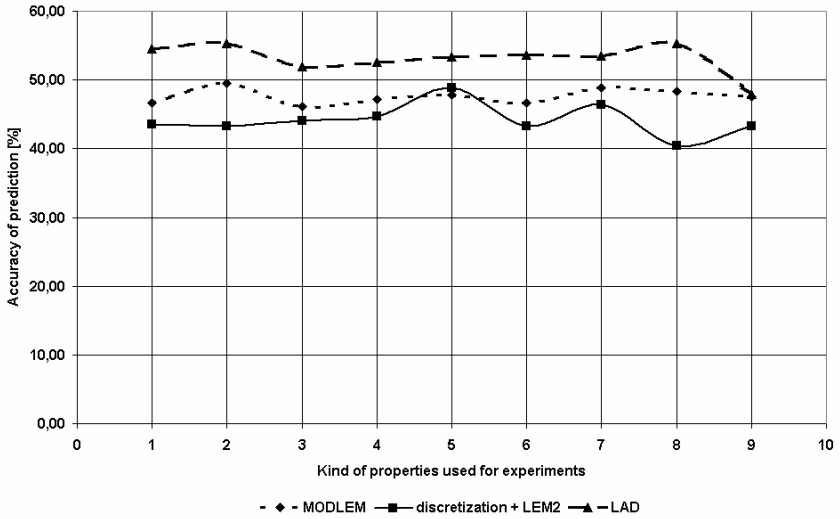

Figure 4. Accuracy of the overall prediction for three classes.

The advantage of LEM2 disappeared for class X (Fig. 3), where it recognized only about 45% of observations. For that class the best results were obtained using LAD (about 75%). MODLEM also was better than LEM2 (about 70%).

In Fig. 4 one can see overall prediction for three classes. The best results were produced by LAD (about 55%). MODLEM did not pass the limit of 50%. The worst was LEM2 (about 45%).

In the next series of experiments we checked if random selection of observations would have an impact on prediction accuracy. In previous approach all observations of the same type might land in the same subset. Thus, if this subset was used as a testing set in the validation test, the algorithms could not generate any rules describing such observations (they were not present in learning set) and prediction accuracy for them was poor. For these experiments one used only the best two algorithms from the previous part, i.e. LAD and MODLEM.

It has been shown (Fig. 5) that random selection resulted in an increase of prediction power by about 5%. The accuracy of LAD was in the range of 63% to 70% (60% before). MODLEM, similarly to the first set of experiments, was worse than LAD, but its accuracy was better than before (55%) and oscillated now around 60%.

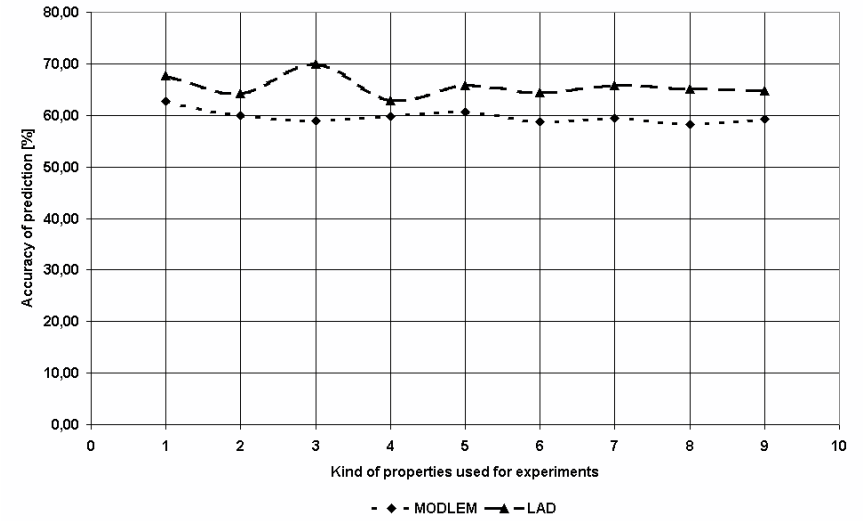

Figure 5. Accuracy of prediction for 3 classes (stratified folds).

The analyzed set of proteins is limited to nine proteins and this set has been used for evaluation of the obtained results. From these nine proteins 2100 residues have been chosen for construction of the observations. With these 2100 residues, the ten-fold cross-validation test was performed. It is hard to treat the obtained results as representative for the whole set of proteins but the aim at this stage of analysis was to examine different machine learning techniques for prediction of secondary structures of proteins problem and to decide which of the proposed method is more suitable for the considered problem, while taking into consideration a compromise between computational time and size of the data set. The reason for such a methodology was that many of the machine learning methods have problems in dealing with large data sets because of the exponential

complexity of algorithms. Thus, the ability to choose a proper method at the first stage is crucial for a success of a further analysis. To analyze the whole available set of proteins one should use about 30 000 protein chains what would generate at least 4 000 000 observations. It is impossible to analyze this set with a sequential version of the algorithms.

The amount of data analyzed in the paper is not impressive but it was caused by the complexity of the problem. LEM2 and MODLEM used to have good performance when someone deals with a huge number of observations with a small number of attributes but LAD used to give good result for a reasonable number of observations with a huge amount of attributes. The aim of the study was to check which of the method behaves better in the considered environment.

The procedure used in the presented analysis considered two approaches: in the first approach 2100 residues have been ordered in the dataset, in the second approach the order has been randomized. The aim of these approaches was to check whether or not whole protein can be used as a training set for the prediction or it is better to combine information from different proteins. For the analyzed set of observations the second approach has been more successful by about 10% of prediction accuracy and it can give helpful directions for future experiments with much more complicated data sets.

The obtained accuracy of predictions is not very impressive in comparison with the currently used methods for prediction of protein secondary structures and it is hard to conclude, based on the small set of observations (in comparison with a number of available protein chains), that these methods can be successfully/unsuccessfully applied for solving the problem of prediction of secondary structures. However, the obtained results based on the considered set of proteins, are promising for future considerations of the analyzed approaches.

Nine physical and chemical properties of aminoacids, used during experiments, have been chosen based on the previous experiments. The influence of some of them as e.g. "mobility of amino acid on chromatography paper" seems strange from the chemical and physical point of view, but this situation can be interpreted in such a way that some characteristic is still hidden behind, and a combination of numbers coming from such a property by chance gave a good description of this something uncovered.

6. Conclusions

The paper presented an application of new machine learning algorithms which have been used to solve the problem of prediction of the secondary structure of proteins problem. The aim was to identify which of the considered methods is more suitable for the analyzed problem and to find rules that would predict the protein secondary structure, based on its primary structure. The best average results were obtained using the LAD algorithm, but one has to mention that very interesting results for class E were obtained by algorithm LEM2 (80%). The results obtained from the experiments show that these methods are promising

but to say definitely whether such an approach is successful/unsuccessful, it is necessary to analyze the problem with a bigger amount of data. A recognition of long-reach interactions is the weakest point in that kind of prediction. Unfortunately, weak overall accuracy for that method means that rules for class E classified too many observations from classes X and H. An overall prediction accuracy (for the considered data set) for LAD was better by about 5% as compared to MODLEM and by more than 10% as compared to LEM2. Standard deviation for overall accuracy for each algorithm was not higher than 6%, so one can expect results in the range of 60-70% from LAD if considered protein chains were representative for all proteins. The result of this study will be used for further research, where the best of the analyzed methods, i.e. LAD, will be parallelized and tested against the set of proteins of a considerably higher dimension.

Appendix

Accession numbers of proteins used in the computational experiments: *1cbh*, *1fdx*, *1fkf*, *1hip*, *1mrt*, *1pyp*, *2cyp*, *2fnr*, *4gr1*, *8adh*.

References

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

Anfinsen, C.B. (1973) Principles that govern the folding of protein chains. *Science* **181**, 223-230.

Bairoch, A. and Apweiler, R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Res.*, **28**, 45-48.

Baldi, P. and Brunak, S. (2001) *Bioinformatics. The Machine Learning approach.* The MIT Press.

Baldi, P., Brunak, S., Frasconi, P., Soda, G. and Pollastri., G. (1999) Exploiting the past and the future in protein secondary structure predicton. *Bioinformatics* **15**, 937-946.

Berenstein, F.C., Koeyzle, T.F., Williams, G.J.B., Meyer, J.E.F., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi, T. and Tatsumi, M. (1997) The protein Data Bank: A computer based archival file for macromolecular structures. *Journal of Molecular Biology* **112**, 525-542.

Błażewicz, J., Hammer, P.L. and Łukasiak, P. (2001a) Prediction of protein secondary structure using Logical Analysis of Data algorithm. *Computational Methods in Science and Technology* **7**(1), 7-26.

Błażewicz, J., Hammer, P.L. and Łukasiak, P. (2001b) Protein Secondary Structure Prediction using Logical Analysis of Data. *Mathematics and*

Simulation with Biological, Economical and Musicoacoustical Applications, WSES Press, 202-207.

Błażewicz, J., Hammer, P.L. and Łukasiak, P. (2005) Predicting Secondary structures of Proteins. *IEEE Engineering in Medicine and Biology* **24** (3), 88-94.

Boros, E., Hammer, P.L., Ibaraki, T., Kogan, A., Mayoraz, E., and Muchnik, I. (1996) An implementation of logical analysis of data. *Rutcor Research Report,* 22-96.

Catlett, J. (1991) On changing continuous attributes into ordered discrete attributes. In: Y. Kodratoff, ed., *Proceedings of the European Working Session on Learning.* SpringerVerlag, Berlin, 164–178.

Cuff, J.A. and Barton, G.J. (1999) Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins* **34**, 508–519.

Cuff, J. and Barton, G. (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* **40**, 502-511.

Cuff, J. et al. (1998) Jpred: A consensus secondary structure prediction server. *Bioinformatics* **14**, 892-893.

Fisher, D. and Eisenberg, D. (1996) Fold recognition using sequence derived properties. *Prot. Sci.* **5**, 947-955.

Frishman, D. and Argos, P. (1997) Seventy-five percent accuracy in protein secondary structure prediction. *Proteins*, 329-335.

Garnier, J., Osguthorpe, D. and Robson, B. (1978) Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J. Mol. Biol.* **120**, 97-120.

Grzymała-Busse, J.W. (1992) LERS – A system for learning from examples based on rough sets. In: R. Słowiński, ed. *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Set Theory.* Kluwer Academic Publishers, Dordrecht/Boston, 3-18.

Grzymała-Busse, J.W. and Stefanowski, J. (2001) Three discretization methods for rule induction. *International Journal of Intelligent Systems,* January.

Hammer, P.L. (1986) Partially Defined Boolean Functions and Cause-Effect Relationships. Presented at the *International Conference on Multi-Attribute Decision Making Via OR-Based Expert Systems*, University of Passau, Germany, April 1986.

Hua, S., Sun, Z. (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J. Mol. Biol.* **308**, 397–407.

Jones, D. (1999a) GenThreader: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* **287**, 797-815.

Jones, D. (1999b) Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol* . **292**, 195-202.

Kabsch, W. and Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. *Biopolymers* **22**, 2577-2637.

King, R. and Sternberg, M. (1996) Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Prot. Sci.* **5**, 2298-2310.

Lee, Y. (2005) Hidden Markov models with states depending on observations. *Pattern Recognition Letters* **26**, 977–984.

Lim, V. (1974) Algorithms for prediction of α-helical and β-structural regions in globular proteins. *J. Mol. Biol.* **88**, 873-894.

Mayoraz, E. (1995) C++ Tools for Logical Analysis of Data. *Rutcor Research Raport*, 1-95.

Ouali M. and King, R.D. (2000) Cascaded multiple classifiers for secondary structure prediction. *Protein Sci.* **9**, 1162–1176.

Pevzner, A. (2000) *Computational Molecular Biology. An algorithmic approach*. The MIT Press.

Pollastri, G., Przybylski, D., Rost B. and Baldi, P. (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins* **47**, 228–235.

Qian, N. and Sejnowski, T. (1988) Predicting the secondary structure of globular proteins using neural networks models. *J. Mol. Biol.* **202**, 865-884.

Przybylski, D., Rost, B. (2002) Alignments grow, secondary structure prediction improves. *Proteins* **46**, 197–205.

Riis, S. and Krogh, A. (1996) Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J. Comp. Biol.* **3**, 163-183.

Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70 % accuracy. *J. Mol. Biol.* **232**, 584-599.

Rost, B. (1996) PHD: predicting one-dimensional protein structure by profile based neural networks. *Meth. Enzymol.* **266**, 525-539.

Salamov, A. and Solovyev, V. (1995) Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignment. *J. Mol. Biol.* **247**, 11-15.

Sanchez, R. and Sali, A. (1997) Evaluation of comparative protein structures modeling by MODELLER-3. *Proteins Suppl.* **1**, 50-58.

Stefanowski, J. (1998) The rough set based rule induction technique for classification problems. In: *Proceedings of 6th European Conference on Intelligent Techniques and Soft Computing EUFIT'98*, 109-113.