CMST 7(1) 7-25 (2001) DOI:10.12921/cmst.2001.07.01.07-25

COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY 7 (1), 7-25 (2001)

PREDICTION OF PROTEIN SECONDARY STRUCTURE USING LOGICAL ANALYSIS OF DATA ALGORITHM

J. BŁAŻEWICZ[1], P. HAMMER[2], P. ŁUKASIAK[1]

[1] *Institute of Computing Sciences, Poznań University of Technology & Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznań, Poland*
[2] *RUTCOR, Rutgers University, USA*

Abstract: In the paper, the problem of a secondary structure prediction, has been considered. The Logical Analysis of Data has been used as a method for this prediction. The approach has led to relatively high prediction accuracy for certain protein structures, as indicated by the experiments constructed.

1. INTRODUCTION

Although it is possibly true in theory that given a protein sequence one can infer its properties, current state of the art in biology falls far short of being able to implement this in practice. Current sequence analysis is a painful compromise between what is desired and what is possible. To help to solve this problem, biologists have divided the structural features of proteins into levels. The first level of the protein structure, termed primary structure, refers just to the sequence of amino acids in the protein. Decades ago it was found that polypeptide chains can sometimes fold into regular structures; that is, structures which are the same in shape for different polypeptides. These structures create the second level of protein structure. The secondary structures are very simple and regular (e.g. the loop of an alpha helix structure or the back and forth of a beta sheet structure). When one looks at an actual polypeptide chain, the final shape is made up of secondary structures, perhaps super-secondary structural features, and some apparently random conformations. This overall structure is referred to as the tertiary structure. Finally, many biological proteins are constructed of multiple polypeptide chains. The way these chains fit together is referred to as the quaternary structure of the protein. The reason that this complex nomenclature for protein structure has been developed, is that, the problem of understanding protein structure is so important and so difficult. The importance of understanding protein structure comes from two factors working together. The first of these is that the function of the protein is absolutely dependent on its structure. In fact, one of the most common ways for proteins to loose their function is to have their structure disrupted, for example by heat or mechanical stress. Only completely and properly folded proteins "work". The second factor is that it is extremely difficult to determine the structure of a protein experimentally. To date, the primary structures of many sequences have been determined (about 30 000). In contrast, the tertiary structures of many fewer (about 1000) have been determined. Obviously, it would be of a great value to determine a tertiary structure from the primary protein structure. It is not an exaggeration to say that the ability to exactly predict protein structures and, from that, protein functions would revolutionize medicine, pharmacology, chemistry and ecology.

The first and probably the most important step to predict tertiary structure from its primary structure is to predict as many as possible secondary structures. Secondary structure prediction has been around for almost a quarter of century. The early methods suffered from a lack of data. Predictions were performed on single sequences rather than families of homologous sequences, and there were relatively few known 3D structures from which to derive parameters. Probably the most famous early methods are those of Chou and Fasman [7], Garnier, Osguthrobe & Robson (GOR) [11] and Lim. Although the authors originally claimed quite high accuracies (70-80%), under careful examination, the method were shown to be only between 56 and 60% accurate [14]. An early problem in secondary structure prediction had been the inclusion of structures used to derive parameters in the set of structures used to access the accuracy of the method.

Nowadays, the best method for protein secondary prediction is a method based on the neural networks and evolutionary information [21, 22]. It gives prediction accuracy over 70% for the three state prediction. Unfortunately, it requires the existence of similar proteins with known structures - a feature which is not always available. The other popular solution is Monte Carlo method [16,23] trying to determine the structure which minimizes free energy.

Trying to solve protein structure prediction problem, scientists use many methods and algorithms [6,10,12,17,19,24]. The most important of them is machine learning approach [15, 20] giving prediction accuracy about 65%. It is interesting because differs from the methods described above in that it emphasizes both: acquiring humanly comprehensible prediction rules and maximizing prediction accuracy.

Such tools as machine learning are needed because it is often difficult for humans to perceive patterns in data, even though strong patterns exist. The idea to create a tool to aid working molecular biologists was the main reason to choose new rule-based method - Logical Analysis of Data [4] with its high accuracy [1]. It generates simple and strong rules which could be easy interpreted by the domain expert. Logical Analysis of Data gives impressive results in many fields of science, so it seemed possible that the same accuracy for the problem in question, is obtained. This paper is devoted to a preliminary study of the above approach to the protein structure prediction problem.

An organization of the paper is as follows. Section 2 formulates the problem to be solved. Section 3 describes the basic ideas of the Logical Analysis of Data method and elaborates on the details of its implementation in the context of the protein structure prediction. Finally section 4 describes the results of a computational experiment showing high accuracy of the approach considered.

2. PROBLEM FORMULATION

The goal of the analysis described in this paper is to create a system which allows to receive as the output the protein secondary structure, based on its primary structure being an input, and to find rules responsible for this effect.

Proteins are chains in the three dimensional space built from smaller chemical molecules called amino acids. There are 20 different amino acids. Each of them is denoted by a different letter in the Latin alphabet as shown below.

Table 1. The codes of 20 amino acids

#	Amino acid	Chemical code	Code in Latin alphabet
1	Alanine	Ala	A
2	Arginine	Arg	R
3	Asparagine	Asn	N
4	Aspartic acid	Asp	D
5	Cysteine	Cys	C
6	Glutamine	Gln	Q
7	Glutamic acid	Glu	E
8	Glycine	Gly	G
9	Histidine	His	H
10	Isoleucine	Ile	I
11	Leucine	Leu	L
12	Lysine	Lys	K
13	Methionine	Met	M
14	Phenylalanine	Phe	F
15	Proline	Pro	P
16	Serine	Ser	S
17	Threonine	Thr	T
18	Tryptophan	Trp	W
19	Tyrosine	Tyr	Y
20	Valine	Val	V

Based on the protein chain it is easy to create its relevant sequence of amino acids replacing an amino acid in chain by its code in Latin alphabet. As a result a word on the amino acids' alphabet is received. This word can be called a protein primary structure on the condition that letters in this word are in the same order as amino acids in the protein chain are.

A secondary structure of a protein is a subsequence of amino acids coming from the relevant protein. These subchains form in the three dimensional space regular structures which are the same in shape for different polypeptides (proteins). In the analysis, a similar representation for the secondary structures as for the primary ones, has been used. A secondary structure is represented by a word on the relevant alphabet of secondary structures - each kind of a secondary structure has its own unique letter. An alphabet of secondary structures consisting of three different secondary structures has been considered in the analysis.

The Logical Analysis of Data is the one of machine learning algorithms. For this reason some examples of a primary and the corresponding secondary structure as a training set are needed to generate rules used for a prediction. These examples were obtained from the Dictionary of Secondary Structures of Proteins (DSSP) [14], DSSP contains a description of secondary structures for entries from the Brookhaven Protein Data Base [2], Moreover, it contains data

calculated from protein tertiary structures obtained by NMR or X-ray experiments and maintained in PDB.

Data gained from DSSP set consist of eight types of protein secondary structures. Usually one can reduce them into three main secondary structures and the same 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);
- β-strand (E) consisting of E structure in DSSP;
- the rest (X) consisting of structures belonging neither to set H nor to set E.

3. THE METHOD

As it has been already said Logical Analysis of Data [13] has been widely applied to the analysis of a variety of real life data sets. Making this paper more understandable it is necessary to recall some terms and definitions relevant to LAD approach.

Observation is a n-dimensional vector having as components the values of n attributes.

Each observation is accompanied by its "classification", i.e. by the indication of the particular class (e.g. positive or negative) this observation belongs to.

Cut point is a critical value along non binary attributes needed for a binarization stage to binarize this attribute.

Pattern can be treated as a m-dimensional vector consisting of m binarized attributes. A pattern generated for a particular class is a vector having as components only these attributes of observations for which their values are the same as for at least one observation belonging to the relevant class. One can say that such an observation is *covered* by a pattern. On the other hand there is not possible to find any observation belonging to the other class for which the value of any relevant attribute is the same as for the considered pattern.

Degree of a pattern is a number of dimensions the pattern consists of.

It is not possible to use the original method [3, 5, 8, 13, 18] directly for this experiment. The first problem lies in input data representation. Here one has a sequence of amino acids but to use the logical analysis of data approach one should have a set of observations. Each observation has to consist of a set of attributes and all of them should be in a number format. If all of them are written in binary one can resign from the binarization stage but this is not the case here. The next question is connected with a number of decision classes. The original method is designed only for two classes. In this experiment three sets of protein secondary structures have been designed.

Because of a complexity of the algorithm of Logical Analysis of Data [9] it is hard to present all aspects of this method. All important phases one can see in Fig. 1. Below only the main stages of this algorithm and here the most important changes that have been necessary for the use of the logical analysis of data for the protein prediction problem, are described.

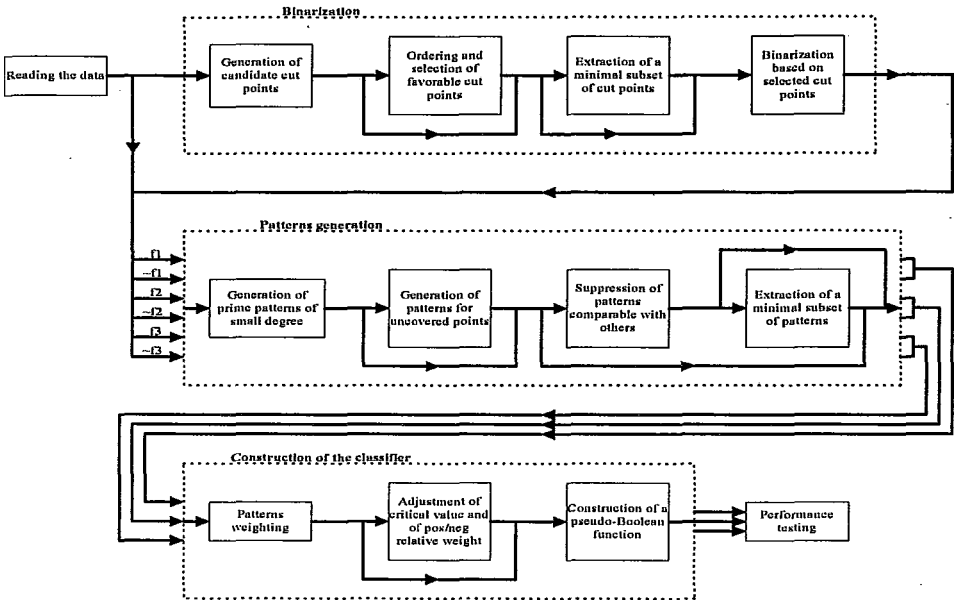

Fig. 1. Modified Logical Analysis of Data method's stages and phases

3.1. Preparation of data

As we mentioned above the first step one has to do, is to prepare a set of observations (based on a protein sequence) to be acceptable by the logical analysis of data algorithm. 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. To fulfill this assumption a concept of windows [15,21] of length equal to 6 (from the (i - 1st) to (i + 4th) amino acid, where the considered secondary structure is relevant to the z'-th amino acid), is used. This is 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. While by unique, we mean here the fact, that there are no two identical observations belonging to different sets of secondary structures.

Below an example is presented, that illustrates the way a protein chain is changed into a set of observations. Let us consider a protein chain called $4pf$ (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 . . ._____HHHHHHHHHHT_

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

XXEEEEEEEEXXXXXX . . . XXXHHHHHHHHHHHXX

At the end of a chain consisting of n amino acids one obtains a set consisting of n observations as illustrated in Table 2

Table 2. An example transformation from a sequence to a set of observations

#	Condition attributes						Code in DSSP	Codes of the three sets of the main secondary structures
	a_{-1}	a_0	a_{+1}	a_{+2}	a_{+3}	a_{+4}		
1	*	M	K	R	I	G		X
2	M	K	R	I	G	V		X
3	K	R	I	G	V	L	E	E
4	R	I	G	V	L	T	E	E
5	I	G	V	L	T	S	E	E
6	G	V	L	T	S	G	E	E
			.	.	.			
314	L	S	K	E	L	S	H	H
315	S	K	E	L	S	I	H	H
316	K	E	L	S	I	*	H	H
317	E	L	S	I	*	*	H	H
318	L	S	I	*	*	*	T	X
319	S	I	*	*	*	*		X

A window of length 6 generates an observation with 6 attributes $(a_{-1}\ a_0,\ a_{+1}\ a_{+2},\ a_{+3},\ a_{+4})$ 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.

All observations are used to create a learning subset or a testing subset. During a creation of a learning subset one has to exclude the first observation and the last four ones (one has not enough information to learn anything). In the testing set, this exclusion is not important because in a 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.

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. Originally we considered 54 properties, but after a discussion with domain experts 28 of them have been chosen for the experiment. All of them are listed below. The first 28 properties were used in this study.

Table 3. Properties of amino acids used in the approach

#	File Id	Description	Author(s)	Reference
1	BULKIN~1.TXT	Bulkiness	Zimmerman J.M., Eliezer N., Simha R.	J. Theor. Biol. 21:170-201(1968)
2	HP11F5~1.TXT	Antigenicity value X 10	Welling G. W., Weijer W. J., Van der Zee R., Welling-Wester S.	FEBS Lett. 188:215-218(1985)
3	HP21D1~1.TXT	Hydrophilicity scale derived from HPLC peptide retention times	Parker J. M. R., Guo D., Hodges R. S.	Biochemistry 25:5425-5431(1986)
4	HP34F9~1.TXT	Hydrophobic constants derived from HPLC peptide retention times	Wilson K. J., Honegger A., Stotzel R. P., Hughes G. J.	Biochem. J. 199:31-41(1981)
5	HP5440~1.TXT	Mobilities of amino acids on chromatography paper (RF)	Aboderin A.A.	Int. J. Biochem. 2:537-544(1971)
6	HP9780~1.TXT	Normalized consensus hydrophobicity scale	Eisenberg D., Schwarz E., Komarony M., Wall R.	J. Mol. Biol. 179:125-142(1984)
7	HP999E~1.TXT	Hydration potential (kcal/mole) at 25°C	Wolfenden R. V., Andersson L., Cullis P. M., Southgate C.C.F.	Biochemistry 20:849-855(1981)
8	HPC2A5~1.TXT	Hydrophobicity indices at ph 7.5 determined by HPLC	Cowan R., Whittaker R.G.	Peptide Research 3:75-80(1990)
9	HPC695~1.TXT	Hydrophobicity indices at ph 3.4 determined by HPLC	Cowan R., Whittaker R.G.	Peptide Research 3:75-80(1990)
10	HPE27B~1.TXT	Hydrophobicity scale (contact energy derived from 3D data)	Miyazawa S., Jernigen R.L.	Macromolecules 18:534-552(1985)
11	HPF0E4~1.TXT	Hydrophobicity (free energy of transfer to surface in kcal/mole)	Bull H.B., Breese K.	Arch. Biochem. Biophys. 161:665-670(1974).
12	HPF1F3~1.TXT	Hydrophilicity	Hopp T.P., Woods K.R.	Proc. Natl. Acad. Sci. U.S.A. 78:3824-3828(1981)
13	HPF8B7~1.TXT	Average surrounding hydrophobicity	Manavalan P., Ponnuswamy P.K.	Nature 275:673-674(1978).
14	HPFD65~1.TXT	Hydrophobicity scale based on free energy of transfer (kcal/mole)	Guy H.R.	Biophys J. 47:61-70(1985)
15	HPHOB_~1.TXT	Optimized matching hydrophobicity (OMH)	Sweet R.M., Eisenberg D.	J. Mol. Biol. 171:479-488(1983)
16	HPHOB_~2.TXT	Hydropathicity	Kyte J., Doolittle R.F.	J. Mol. Biol. 157:105-132(1982)
17	HPHOB_~3.TXT	Hydrophobicity (delta G1/2 cal)	Abraham D.J., Leo A.J.	Proteins: Structure, Function and Genetics 2:130-152(1987)
18	HPHOB_~4.TXT	Hydrophobicity scale (pi-r)	Fauchere J.-L., Pliska V.E.	Eur. J. Med. Chem. 18:369-375(1983)

Table 3. Continued

#	File Id	Description	Authors	Reference
19	HPLC2_1.TXT	Retention coefficient in HPLC, pH 2.1	Meek J.L.	Proc. Natl. Acad. Sci. USA, 77:1632-1636(1980)
20	HPLC7_4.TXT	Retention coefficient in HPLC, pH 7.4	Meek J.L.	Proc. Natl. Acad. Sci. USA 77:1632-1636(1980)
21	HPLCHFBA.TXT	Retention coefficient in HFBA	Browne C.A., Bennett H.P.J., Solomon S.	Anal. Biochem. 124:201-208(1982)
22	HPLCTFA.TXT	Retention coefficient in TFA	Browne C.A., Bennett H.P.J., Solomon S.	Anal. Biochem. 124:201-208(1982)
23	MOLECU~1.TXT	Molecular weight of each amino acid	–	Most textbooks
24	POLARI~1.TXT	Polarity (p)	Grantham R.	Science 185:862-864(1974)
25	POLARI~2.TXT	Polarity	Zimmerman J.M., Eliezer N., Simha R.	J. Theor. Biol. 21:170-201(1968)
26	RECOGN~1.TXT	Recognition factors	Fraga S.	Can. J. Chem. 60:2606-2610(1982)
27	REFRAC~1.TXT	Refractivity	Jones. D.D.	J. Theor. Biol. 50:167-184(1975)
28	RELATI~1.TXT	Relative mutability of amino acids (Ala=100)	Dayhoff M.O., Schwartz R.M., Orcutt B.C.	In "Atlas of Protein Sequence and Structure", Vol.5, Suppl.3 (1978)
29	**ACCESS~1.TXT**	**Molar fraction (%) of 3220 accessible residues**	**Janin J.**	**Nature 277:491-492(1979)**
30	**ALPHA~~1.TXT**	**Conformational parameter for alpha helix (computed from 29 proteins)**	**Chou P.Y., Fasman G.D.**	**Adv. Enzym. 47:45-148(1978)**
31	**ALPHA~~2.TXT**	**Conformational parameter for alpha helix**	**Deleage G., Roux B.**	**Protein Engineering 1:289-294(1987)**
32	**ALPHA~~3.TXT**	**Normalized frequency for alpha helix**	**Levitt M.**	**Biochemistry 17:4277-4285(1978)**
33	**ANTIPA~1.TXT**	**Conformational preference for antiparallel beta strand**	**Lifson S., Sander C.**	**Nature 282:109-111(1979)**
34	**AVERAG~1.TXT**	**Average area buried on transfer from standard state to folded protein**	**Rose G.D., Geselowitz A.R., Lesser G.J., Lee R.H., Zehfus M.H.**	**Science 229:834-838(1985)**
35	**AVERAG~2.TXT**	**Average flexibility index**	**Bhaskaran R., Ponnuswamy P.K.**	**Int. J. Pept. Protein. Res. 32:242-255(1988)**
36	**A_A_CO~1.TXT**	**Overall amino acid composition (%)**	**McCaldon P., Argos P.**	**Proteins: Structure, Function and Genetics 4:99-122(1988)**
37	**A_A_PG~1.TXT**	**Amino acid composition (%) in the PGtrans Protein Sequence data bank**	**Claverie J.-M., Sauvaget I., Bougueleret L.**	**Biochimie 67:437-443(1985)**

Table 3. Continued

#	File Id	Description	Authors	Reference
38	A_A_SW~1.TXT	Amino acid composition (%) in the SWISS-PROT Protein Sequence data bank	Bairoch A.	Release notes for SWISS-PROT release 31 - February 1995
39	BETA-S~1.TXT	Conformational parameter for beta-sheet	Deleage G., Roux B.	Protein Engineering 1:289-294(1987)
40	BETA-S~2.TXT	Normalized frequency for beta-sheet	Levitt M.	Biochemistry 17:4277-4285(1978)
41	BETA-S~3.TXT	Conformational parameter for beta-sheet (computed from 29 proteins)	Chou P.Y., Fasman G.D.	Adv. Enzym. 47:45-148(1978)
42	BETA-T~1.TXT	Conformational parameter for beta-turn (computed from 29 proteins)	Chou P.Y., Fasman G.D.	Adv. Enzym. 47:45-148(1978)
43	BETA-T~2.TXT	Conformational parameter for beta-turn	Deleage G., Roux B.	Protein Engineering 1:289-294(1987)
44	BETA-T~3.TXT	Normalized frequency for beta-turn	Levitt M.	Biochemistry 17:4277-4285(1978)
45	BURIED~1.TXT	Molar fraction (%) of 2001 buried residues	Janin J.	Nature 277:491-492(1979)
46	COILROUX.TXT	Conformational parameter for coil	Deleage G., Roux B.	Protein Engineering 1:289-294(1987)
47	HP1058~1.TXT	Proportion of residues 95% buried (in 12 proteins)	Chothia C.	J. Mol. Biol. 105:1-14(1976)
48	HP7912~1.TXT	Hydrophobicity scale (pi-r)	Roseman M.A.	J. Mol. Biol. 200:513-522(1988)
49	HPE383~1.TXT	Mean fractional area loss (f) [average area buried/standard state area]	Rose G.D., Geselowitz A.R., Lesser G.J., Lee R.H., Zehfus M.H.	Science 229:834-838(1985)
50	HPFEC7~1.TXT	Free energy of transfer from inside to outside of a globular protein	Janin J.	Nature 277:491-492(1979)
51	NUMBER~1.TXT	Number of codon(s) coding for each amino acid in universal genetic code	–	Most textbooks
52	PARALL~1.TXT	Conformational preference for parallel beta strand	Lifson S., Sander C.	Nature 282:109-111(1979)
53	RATIOS~1.TXT	Atomic weight ratio of hetero elements in end group to C in side chain	Grantham R.	Science 185:862-864(1974)
54	TOTALB~1.TXT	Conformational preference for total beta strand (antiparallel+parallel)	Lifson S., Sander C.	Nature 282:109-111(1979)

3. 2. Binarization stage

A 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 binary format. The simplest non-binary attributes are the so-called nominal (or descriptive) ones. The binarization of such an attribute is accomplished by associating with each value v_s of the attribute x a Boolean variable $b(x, v_s)$ such that:

$$b(x, v_s) = \begin{cases} 1 & \text{if } x = v_s \\ 0 & \text{otherwise} \end{cases}.$$

In case all variables are numerical, one can distinguish two types of Boolean variables. The first type called *the level variable,* $b(x, t)$ is introduced for every attribute x and cut-point t in the following way:

$$b(x, t) = \begin{cases} 1 & \text{if } x \geq t_s \\ 0 & \text{if } x < t \end{cases}.$$

The second type, called *the interval variable,* $b(x,t',t'')$ is introduced for every attribute x and each pair of cut-points t', t'' ($t' < t''$) in the following way:

$$b(x, t', t'') = \begin{cases} 1 & \text{if } t' \leq x < t'' \\ 0 & \text{otherwise} \end{cases}.$$

Since the number of observation points in the training set is finite, each ordered attribute x takes only a finite number of different values in the training set. Let this values be

$$v_{s-1} < v_2 < \ldots < v_s .$$

Since two cut-points t' and t'' such that

$$v_{s-1} < t', t'', v_s .$$

produce identical Boolean variables (on the training set), it is not necessary to consider any cut-points outside the set $\{v_i, v_2, \ldots, v_q\}$. It is sufficient to use in the binarization procedure only cut-points for which there exist both a true and a false observation points in the training set, such that one of them has $x = v_s$ while the other has $x = v_{s-1}$.

The next main problem in the binarization stage is to reduce the size of a binary archive by eliminating as many redundant attributes as possible. One can introduce a term *the support set* to name a set of binary attributes in case the archive obtained by the elimination of all the other attributes will not contain simultaneously true and false observations. A support set is called *irredundant* if it contains no support set as its proper subset. One associates a Boolean variable y_i with attribute b_i in such a way that $y_i = 1$ means that the attribute b_i is retained in a support set, and $y_i = 0$ means that it is removed from it. Given the archive with Boolean attributes $b_1, \ldots b_q$ and variables $y_1, \ldots, y_q$ associated with them as above, one can define for every pair of true and

false vectors p' and p'', the subset $I(p', p'')$ of those indices where p' and p'' differ. It is easy to see that $(y_1, \ldots, y_q)$ is the characteristic vector of a support set if and only if it satisfies the following system of inequalities:

$$\sum_{i \in I(p', p'')} y_i \geq 1 \quad \forall p' \in S^+, \quad \forall p'' \in S^-,$$

where S^+ and S^- are the sets of positive and negative observations respectively.

A smallest support set can be obtained by solving the set covering problem.

$$\min \sum_{i=1} y_i$$

$$\text{s.t.} \sum_{i \in I(p', p'')} y_i \geq 1 \quad \forall p' \in S^+, \quad \forall p'' \in S^-,$$

$$y_i \in \{0, 1\} \quad i = 1, \ldots, q$$

As a result of this stage, all attributes for each observation are changed into binary attributes. Each property gives 20 real numbers usually unique for every amino acid. An example of these values for the property called bulkiness is shown below.

```
ProtScale Tool
Amino acid scale: Bulkiness
Author(s): J. M. Zimmerman, N. Eliezer, R. Simha
Reference: J. Theor. Biol. 21: 170-201(1968).
Amino acid scale values:
Ala: 11.500
Arg: 14.280
Asn: 12.820
Asp: 11.680
Cys: 13.460
Gin: 14.450
Glu: 13.570
Gly:  3.400
His: 13.690
He:  21.400
Leu: 21.400
Lys: 15.710
Met: 16.250
Phe: 19.800
Pro: 17.430
Ser:  9.470
Thr: 15.770
Trp: 21.670
Tyr: 18.030
Val: 21.570
```

There are 19 cut points that are generated for each attribute. This gives 114 possible cut points used for binarization. They have been used to extract a minimal set of cut points which allowed to binarize all attributes without loosing any observation. It means that after the binarization phase

all of observations that belonged to different classes are still different when binary attributes are taken into account. On average, 17 cut points were enough to make classes still unique.

3. 3. Pattern generation stage

A symmetric definition of positive and of negative patterns leads 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.

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 to 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 observations, it is a pattern. Otherwise, literals are added to the term one by one as long as necessary, i.e. until generating a pattern.

Pattern generation used in the experiment described in this paper is achieved by a hybrid bottom-up - top-down approach. This strategy uses the bottom-up approach to generate all the patterns of very small degrees, and then uses a top-down approach to cover those positive observations that remained uncovered after the bottom-up step.

The number of terms of degree d over n Boolean variables is $2^d\binom{n}{d}$. Even for n fixed at a moderate value, this is a very rapidly growing function of d. Therefore, the term enumeration method used for pattern generation must be extremaly selective.

During the experiment a breadth-first search technique was used. It produces at each stage d all the positive prime patterns of degree d, as well as the list of the so-called "candidate" terms to be examined at stage $d+1$ of the algorithm. A *candidate term* is any term that covers at least one negative and at least one positive observation. The terms of degree d examined by the algorithm at stage d are all those from which one gets a candidate term of degree $d-1$ (generated at stage $d-1$) by eliminating any of its literals. The terms of degree d examined by algorithm are then partitioned into following three sets:

P_d, consisting of those terms which cover at least one positive and no negative observations;

C_d, consisting of those terms which cover at least one positive and at least one negative observations;

G_d, consisting of all the remaining terms.

Set P_d consists of all positive patterns of degree d, and set C_d consists of all candidate terms of degree d. Set G_d is eliminated from any further considerations.

Additional reductions are obtained by examining terms C_d in the lexicographic order induced by the linear order

$$x_1 \prec \bar{x}_1 \prec x_2 \prec \bar{x}_2 \prec \ldots$$

of the literals. Since it is sufficient to generate each term only once, the term of degree $d + 1$ are generated from C_d by adding in all possible ways to term $M \in C_d$ a literal which is larger (in this order) than any literal in M. Indeed, let the indices of the literals in M be $i_1 < i_2 < \ldots < i_d$. Suppose that term M' is obtained by adding to M a literal of index $i < i_d$. Let M'' be the term whose literals have the indices $i_1, i_2, \ldots, i, \ldots, i_{d-1}$. Clearly M' can also be obtained by adding to M'' the literal of index i_d. If $M'' \notin C_d$, M' does not have to be examined, because term M' is then neither a prime pattern, nor a candidate term. If $M'' \in C_d$, then M' was already considered, because M'' is lexicographically smaller than M.

Other important property of datasets, that has to be taken into account in pattern generation stage, consists in the presence of some so-called *monotone* variables. Given Boolean function f, the variable x_i is called *positive* in f if for any Boolean vector $p \in \{0, 1\}^n$,

$$f(p_1 \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_n) \leq f(p_1 \ldots, p_{i-1}, 1, p_{i+1}, \ldots, p_n).$$

Similarly, the variable x_i is called *negative* in f if for any Boolean vector $p \in \{0, 1\}^n$,

$$f(p_1 \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_n) \geq f(p_1 \ldots, p_{i-1}, 1, p_{i+1}, \ldots, p_n).$$

Finally, the variable x_i is called *monotone* in f if it is either positive or negative in f.

A priori knowledge of the monotone character of some variables immediately disqualifies certain patterns from the consideration.

In the original method 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 six times. Each time an observation from one set of secondary structures 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. The reason for calling this stage for a class consisting of two sets of secondary structures was to check whether better accuracy for the secondary structure prediction could be obtained if one knew where the relevant secondary structure cannot appear.

During the experiments it was not allowed to cover an observation belonging to an opposite class. Patterns have been generated until the whole set of all observations has been covered by at least one pattern.

All patterns have been generated using breadth first search strategy (for the patterns of up to degree 4) and depth first search strategy (for other patterns).

3. 4. Classifier construction stage

Before this stage is performed every positive (negative) observation point is covered by at least one positive (negative) pattern, and 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 a 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.

The simplest method of building a theory consists of defining a weighted sum of positive and negative patterns and classifying new observations according to the value of the following weighted sum:

$$\Delta = \sum_{k=1}^{r} \omega_k^+ P_k + \sum_{l=1}^{s} \omega_l^- N_l$$

Such a weighted sum will be called a *discriminant.* The weights of the patterns are chosen in such a way that large positive (negative) values of the discriminant will be indicative of the positive (negative) character of the new observation.

The selection process of the subset of the patterns generated for the case of positive patterns is described below.

One can assign to each of the generated positive patterns $P_1, P_2, \ldots, P_r$ binary (0-1) variables $y_1, y_2, \ldots y_r$ with the convention that $y_k = 1(0)$ means that P_k is (is not) in the selected subset. Let us define $a_{jk} = 1$ if the positive observation point p_j is covered by pattern P_k and $a_{jk} = 0$ otherwise. In order to distinguish pj from the negative points, at least one of the positive patterns covering it must be selected, i.e.

$$\sum_k a_{jk} y_k \geq 1 . (*)$$

In order to distinguish all the positive points from all the negative ones, the vector $(y_1, y_2, \ldots, y_r)$ characterizing the selected subset will have to satisfy the covering constraints (*) for all positive observation points p_j In order to produce a small subset of patterns satisfying these requirements we shall solve the following *set covering problem:*

$$\min \sum_{k=1}^{r} y_k$$

$$\text{s.t.} \sum_{k=1}^{r} a_{jk} y_k \geq 1 \quad \text{for all positive } p_j .$$

$$y_k \in \{0, 1\} \quad k = 1, \ldots r$$

To increase the discriminating power of the selected subset of patterns each positive point should be covered by several patterns. In order to give preferences to patterns possessing some special properties, the objective function

$$\sum_{k=1}^{r} y_k$$

can be replaced by the weighted sum

$$\sum_{k=1}^{r} c_k y_k$$

with appropriately chosen weights c_k.

Weights can be chosen in different ways. The simplest way is to define all $|\omega_k|=1$, assigning thus equal importance to all the patterns. On the other hand, the number of observation points q_k covered by pattern P_k can be viewed as an indication of its relative importance justifying the choice $|\omega_k|= q_k$. It can be emphasized even stronger by chosing $|\omega_k|= q_k^2$, or q_k^3, or 2^{qk}.

A more sophisticated approach to weight selection aiming at the increase the separating power of the discriminant as much as possible, is based on the use of linear programming. The weights are then determined by solving the following linear programming problem:

$$\max g_p + g_n$$

$$s.t. \sum_{k=1}^{t} \omega_k^+ P_k(p) \geq g_p \quad \text{for all positive } p$$

$$\sum_{l=1}^{s} \omega_l^- N_l(p) \leq -g_n \quad \text{for all negative } p$$

$$\omega_k^+ \geq 0 \qquad k = 1, \ldots, r$$

$$\omega_l^- \leq 0 \qquad l = 1, \ldots, s$$

where $P_k(p)(N_l(p))$ is 1 if $P_k(N_l)$ covers p, and is 0 otherwise.

Technical parameters for this stage remain unchanged during the experiment as compared with the original approach [13], 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, each observation belongs to.

An interested reader is referred to [3, 13], for a more detailed description of the Logical Analysis of Data method.

4. Experiments and results

For experiments the data set of protein chains received from the Dictionary of Secondary Structures of Proteins (DSSP) has been used. Properties of amino acids were taken from

the ProtScale (as described in Table 4). First four characters denote PDB entry identifier, the fifth character describes a part of the protein chain. For experiments all protein chains from a given PDB entry have been used.

Table 4. PDB identifiers of chains used for experiments

107*l*	1*crn*	1*rhd*	2*pcy*	4*rhv*1
114*l*	1*cse*1	1*s*01	2*phh*	4*rhv*2
121*p*	1*cse*2	1*sh*1	2*sns*	4*rhv*3
127*l*	1*eca*	1*ubq*	2*stv*	4*rhv*4
12*ca*	1*etu*1	2*aat*	3*ait*	4*rxn*
132*l*	1*etu*2	2*alp*	3*b*5*c*	5*ldh*
146*l*	1*fdx*	2*cab*	3*blm*	5*lyz*
155*l*	1*fkf*	2*cyp*	3*cd*4	6*acn*
159*l*	1*hip*	2*fxb*	3*cla*	6*cpa*
165*l*	1*l*58	2*gbp*	3*cln*	6*cpp*
170*l*	1*lap*	2*gcr*	3*ebx*	6*cts*1
173*l*	1*mrt*	2*gn*5	3*icb*	6*cts*2
187*l*	1*paz*	2*i*1b	3*pgm*	6*cts*3
1*acx*	1*ppt*	2*lh*7	3*rnt*	6*dfr*
1*azu*	1*pyp*	2*lhb*	4*cms*1	6*hir*
1*bds*	1*r*091	2*mev*1	4*cms*2	7*icd*
1*bmv*1	1*r*092	2*mev*2	4*cpv*	7*rsa*
1*bmv*2	1*r*093	2*mev*3	4*fxn*	8*abp*
1*cbh*	1*r*094	2*mev*4	4*gr*1	8*adh*
1*cc*5	1*rbp*	2*mhu*	4*pfk*	9*pap*

During experiments about 20 000 observations have been created. Unfortunately, it has not been possible to generate rules based at the same time on the whole set of observations. One had to create some numbers of smaller sets of observations and rules have been generated for these subsets separately. Output results shown below, present an average accuracy for the secondary structure prediction for all observations. Building observation subsets one wanted to find the strongest rules, thus, one didn't care whether or not observations collected in one subset were derived from one protein chain. Results of the experiments for all three secondary structures and for the sets simultaneously consisting of two classes are shown in Fig. 2 through 4. Numbers on horizontal axis correspond to the property number as given in Table 3.

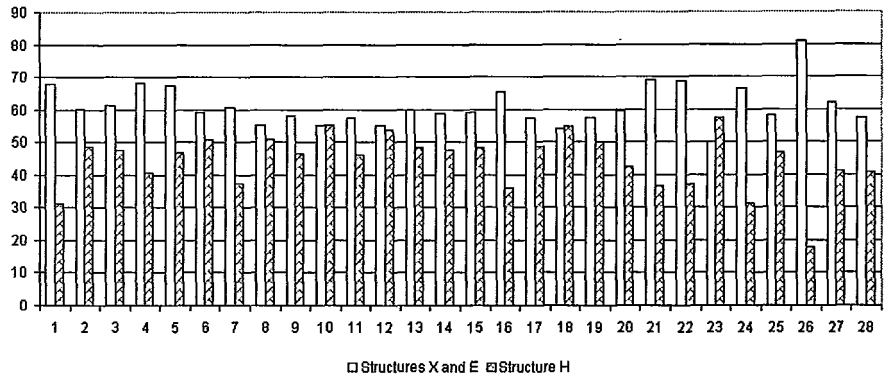

Fig. 2. Prediction accuracy for structure H

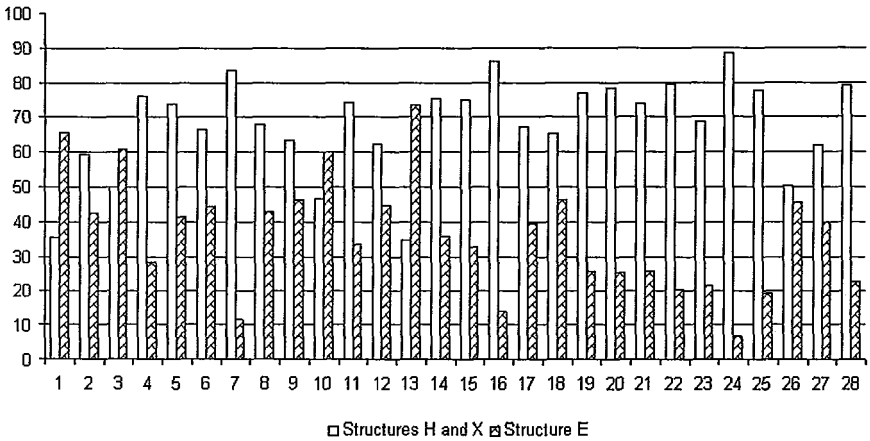

Fig. 3. Prediction accuracy for structure E

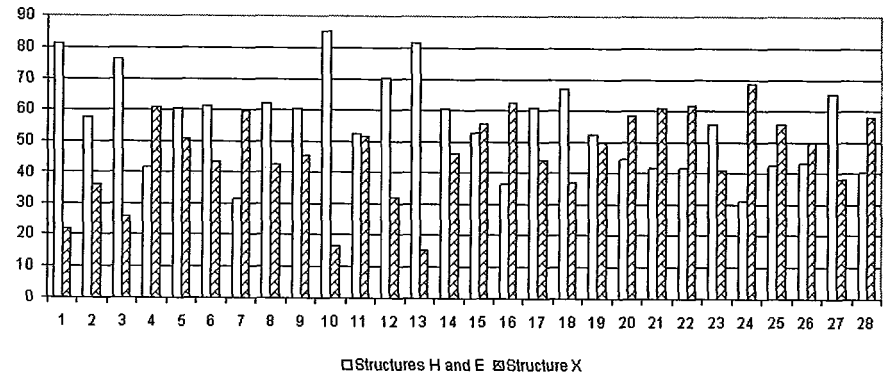

Fig. 4. Prediction accuracy for structure X

Prediction accuracy for structure H is between 18 and 57%, for structure E between 7 and 74% and for structure X between 15 and 69%. Average accuracy for the best property for all three structures is about 55%. Better results have been obtained when patterns have been generated for a class consisting of two secondary structures. In this case the accuracy of prediction exceeded 80%.

5. CONCLUSIONS

The obtained results are average as compared with other methods for the protein prediction. A comparision has been made with the algorithm based on the Rough Set theory. Results obtained using this method were similar to the results obtained using logical analysis of data and none of the two methods could prove its superiority. The difficulty in getting better results can be situated in a construction of training and testing data sets. Observations used for experiments in one data set should belong to one organism or be responsible for the same function. On the other hand, a positive aspect of the experiment has been an extraction of the set of the most promising amino acids properties. From the set of 54 properties, 5 of them have been extracted which had the most important influence on the created secondary structures. This is a good standpoint for a continuation of the research in this field.

References

[1] M. Anthony, *Accuracy of techniques for the logical analysis of data,* Rutcor Research Raport, 23-96 (1996).

[2] F. C. Berenstein, T. F. Koeyzle, G. J. B. Williams, J. E. F. Meyer, M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi and M. Tatsumi, *The protein Data Bank: A computer based archival file for macromolecidar structures,* Journal of Molecular Biology, **112**, 525-542 (1977).

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

[4] E. Boros, P. L. Hammer, T. Ibaraki and A. Kogan, *Logical Analysis of Numerical Data,* Rutcor Research Raport, 04-97 (1997).

[5] E. Boros, P. L. Hammer, A. Kogan, E. Mayoraz and I. Muchnik, *Logical Analysis of Data - Overview,* Rutcor Research Raport, 1-94 (1994).

[6] J. U. Bowie, R. Luthy and D. Eisenberg, *A method to identify protein sequences that fold into a known three-dimensional structure,* Science, **253**, 164-170 (1991).

[7] Chou and G. Fasman, *Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins,* Biochemistry, **13**, 211-222 (1974).

[8] Y. Crama, P. L. Hammer and T. Ibaraki, *Cause-effect relationships andpartally defined Boolean functions,* Annals of Operations Research, **16**, 299-326 (1998).

[9] O. Ekin, P. L. Hammer and A. Kogan, *Convexity and logical analysis of data,* Rutcor Research Raport, 5-98 (1998).

[10] G. F. Fasman, *Protein conformation prediction. In Prediction of Protein Structure and the Principles of Protein Conformation* (G. D. Fasman, ed.), 193-316, Plenum, New York and London (1989).

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

[12] C. Geourjon and G. Deleage, *SOPM: a self optimized method for protein secondary structure prediction,* Protein Eng. **7**,. 157-164 (1994).

[13] P. L. Hammer, *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.

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

[15] R. D. King and M. J. E. Sternberg, *Machine learning approach for the prediction of protein secondary structure,* J. Mol. Biol. **216**, 441-457 (1990).

[16] A. Kolinski, L. Jaroszewski, P. Rotkiewicz and J. Skolnick, *An efficient Monte Carlo model of protein chains. Modeling the short-range correlations between side group centers of mass,* J. Phys. Chem. **02**, 4628-4637 (1998).

[17] J. M. Levin, B. Robson and J. Gamier, *An algorithm for secondary structure determination in proteins based on sequence similarity*, FEBS Letters **205**, 303-308 (1986).

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

[19] E. Mayoraz, I. Dubchak and I. Muchnik, *Relation between protein structure, sequence homology and composition of amino acids,* Rutcor Research Raport, 6-95 (1995).

[20] S. Muggleton, R. D. King and M. J. E. Sternberg, *Protein secondary structure prediction using logic based machine learning,* Protein Eng. **5**, 647-657 (1992).

[21] B. Rost and C. Sander, *Prediction of protein secondary structure at better than 70% accuracy,* JMB **232**, 584-599 (1993).

[22] B. Rost and C. Sander, *Combining evolutionary information and neural networks to predict protein secondary structure,* Proteins **19**, 55-72 (1994).

[23] J. Skolnick and A. Kolinski, *Monte Carlo approaches to the protein folding problem, Monte Carlo Methods in Chemical Physics,* D. Ferguson, J. I. Siepmann and D. G. Truhlar, Eds., Advances in Chemical Physics, John Wiley & Sons, **105**, 203-242 (1999).

[24] T-M. Yi and E. S. Lander, *Protein secondary structure prediction using nearest-neighbor methods,* Journal of Molecular Biology, **232**, 117-1129 (1993).