Protein Engineering vol.5 no.7 pp.647–657, 1992

Protein secondary structure prediction using logic-based machine learning

Stephen Muggleton[1], Ross D.King[1,3] and Michael J.E.Sternberg[2]

[1]Turing Institute, George House, 36 North Hanover Street, Glasgow G1 2AD and [2]Biomolecular Modelling Laboratory, Imperial Cancer Research Fund, PO Box 123, 44 Lincoln's Inn Fields, London WC2A 3PX, UK

[3]To whom correspondence should be addressed

Many attempts have been made to solve the problem of predicting protein secondary structure from the primary sequence but the best performance results are still disappointing. In this paper, the use of a machine learning algorithm which allows relational descriptions is shown to lead to improved performance. The Inductive Logic Programming computer program, Golem, was applied to learning secondary structure prediction rules for α/α domain type proteins. The input to the program consisted of 12 non-homologous proteins (1612 residues) of known structure, together with a background knowledge describing the chemical and physical properties of the residues. Golem learned a small set of rules that predict which residues are part of the α-helices—based on their positional relationships and chemical and physical properties. The rules were tested on four independent non-homologous proteins (416 residues) giving an accuracy of 81% (±2%). This is an improvement, on identical data, over the previously reported result of 73% by King and Sternberg (1990, *J. Mol. Biol.*, 216, 441–457) using the machine learning program PROMIS, and of 72% using the standard Garnier–Osguthorpe–Robson method. The best previously reported result in the literature for the α/α domain type is 76%, achieved using a neural net approach. Machine learning also has the advantage over neural network and statistical methods in producing more understandable results.

Key words: artificial intelligence/α-helix/machine learning/protein modelling/secondary structure prediction

Introduction

An active research area in the hierarchical approach to the protein folding problem is the prediction of secondary structure from primary structure (Lim, 1974; Gibrat *et al.*, 1987; Bohr *et al.*, 1988, 1990; Qian and Sejnowski, 1988; Seshu *et al.*, 1988; Holley and Karplus, 1989; McGregor *et al.*, 1989, 1990; King and Sternberg, 1990). Most of these approaches involve examining the Brookhaven database (Bernstein *et al.*, 1977) of known protein structures to find general rules relating primary and secondary structure. However, this database is hard for humans to assimilate and understand because it consists of a large amount of abstract symbolic information, although the use of molecular graphics provides some help. Today the best methods of secondary structure prediction achieve an accuracy of 60–65% (Kneller *et al.*, 1990). The generally accepted reason for this poor accuracy is that the predictions are carried out using only local information—long range interactions are not taken into account. Long range interactions are important because when a protein folds up, regions of the sequence which are linearly widely separated become close spatially. Established approaches to the problem of predicting secondary structure have involved hand-crafted rules by experts (Lim, 1974) and Bayesian statistical methods (Gibrat *et al.*, 1987). More recently a variety of machine learning methods have been applied: both neural networks (Bohr *et al.*, 1988, 1990; Qian and Sejnowski, 1988; Holley and Karplus, 1989; McGregor *et al.*, 1989, 1990) and symbolic induction (Seshu *et al.*, 1988; King and Sternberg, 1990). An exact comparison between these methods is very difficult because different workers have used different types of proteins in their data sets.

One approach to achieve a higher accuracy in the prediction of secondary structure is to break the problem down into a number of sub-problems. This is done by splitting the data set of proteins into groups of the same type of domain structure, e.g. proteins with domains only with α-helices (α/α domain type), β-strands (β/β domain type), or alternate α-helices and β-strands (α/β domain type). This allows the learning method to have a more homogeneous data set, resulting in better prediction, and assumes a method of determining the domain type of a protein. The decomposition approach is adopted in this paper where we concentrate solely on proteins of α/α domain type. On these protein types, neural networks have achieved an accuracy of 76% on unseen proteins (Kneller *et al.*, 1990)—using a slightly more homologous database than in this study. These proteins have also been studied using the symbolic induction program PROMIS, which achieved an accuracy of 73% on unseen proteins (King and Sternberg, 1990)—using the same data as this study. Compared with the machine learning method used in this study, PROMIS has a limited representational power. This means that it was not capable of finding some of the important relationships between residues that the new method showed were involved in α-helical formation.

In this paper, Inductive Logic Programming (ILP) is applied to learning the secondary structure of α/α domain type proteins. ILP is a method for automatically discovering logical rules from examples and relevant domain knowledge (Muggleton, 1991). ILP is a new development within the field of symbolic induction and marks an advance in that it is specifically designed to learn structural relationships between objects—a task particularly difficult for most machine learning or statistical methods. The ILP program used in this work is Golem (Muggleton and Feng, 1990). Golem has had considerable previous success in other essentially relational application areas including drug design (King *et al.*, 1992), finite element mesh design (Dolsak and Muggleton, 1991), construction of qualitative models (Bratko *et al.*, 1991) and the construction of diagnostic rules for satellite technology (Feng, 1991).

Materials and methods

Database of proteins

Sixteen proteins were selected for the data set from the Brookhaven data bank (Bernstein *et al.*, 1977). The training proteins used were 155C (cytochrome C550: Timkovich and

Dickerson, 1976), 1CC5 (cytochrome C5 oxidized: Carter *et al.*, 1985), 1CCR (cytochrome C: Ochi *et al.*, 1983), 1CRN (crambin: Hendrickson and Teeter, 1981), 1CTS (citrate synthase: Remington *et al.*, 1982), 1ECD (erythrocruorin reduced deoxy: Steigemann and Weber, 1979), 1HMQ (hemerythrin met: Stekamp *et al.*, 1983), 1MBS (myoglobin met: Scouloudi and Baker, 1978), 2B5C (cytochrome B5), 2C2C (cytochrome C2 oxidized), 2CDV (cytochrome C3) and 3CPV (calcium-binding parvalbumin). The test proteins used were 156B (cytochrome B562 oxidized: Lederer *et al.*, 1981), 1BP2 (phospholipase A2: Dijkstra *et al.*, 1981), 351C (cytochrome C551: Matsuura *et al.*, 1982) and 8PAP (papain: Kamphuis *et al.*, 1984)—in protein 8PAP only the first domain (residues 1 – 108) is used, the other domain is of type β/β. These proteins have high resolution structure and α/α domain type (secondary structure dominated by α-helices, with little if any β-strands)—Sheridan *et al.* (1985). The proteins were also selected to be non-homologous (little structural or sequential similarity). This selection was performed on the basis of a knowledge of protein structure and biology, e.g. there is only one globin structure 1MBS (myoglobin). It was not possible to use a much larger set of proteins because of the limited number of proteins with known α/α domain type structure. The data set of proteins was randomly chosen from all the proteins to give an ~70:30 split (Table I). Secondary structure was assigned using an early implementation of the Kabsch and Sander (1983) algorithim.

Golem

Golem is a program for ILP. The general scheme for ILP methods is shown in Figure 1. This scheme closely resembles that of standard scientific methods. Observations are collected from the outside world (in this study the Brookhaven data bank). These are then combined, by an ILP program, with background knowledge to form inductive hypotheses (rules for deciding secondary structure). These rules are then experimentally tested on additional data. If experimentation leads to high confidence in the hypotheses' validity, they are added to the background knowledge.

Table I. Statistics of the random split of the data into training and test sets

Set		Types of secondary structure						
		α	not α	β	not β	turn	not turn	total
Train	no	848	764	45	1567	719	893	1612
	ratio	0.526	0.474	0.028	0 972	0.446	0 554	
Test	no.	217	199	10	406	189	227	416
	ratio	0.522	0.478	0.024	0.0976	0.454	0.546	
All	no	1065	963	55	1973	908	1120	2028
	ratio	0.525	0.475	0.027	0 973	0.448	0.552	

The top row titles are the types of secondary structure, the left column titles are the splits of the data into training and test sets, no. is the number of residues of that secondary structure type and ratio is the ratio (secondary structure no /total no.)

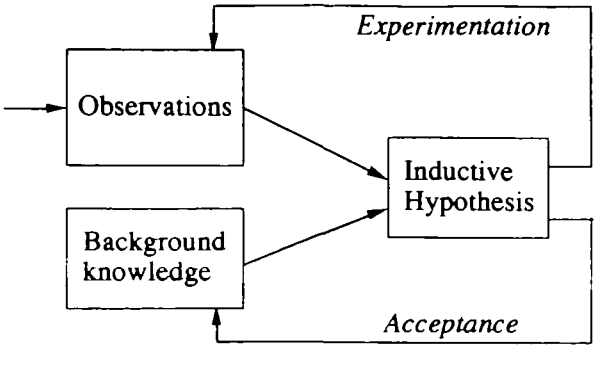

Fig. 1. Inductive Logic Programming scheme

In ILP systems, the descriptive languages used are subsets of first-order predicate calculus. Predicate logic is expressive enough to describe most mathematical concepts. It is also believed to have a strong link with natural language. This combination of expressiveness and ease of comprehension has made first-order predicate calculus a very popular language for artificial intelligence applications. The ability to learn predicate calculus descriptions is a recent advance within machine learning. The computer implementation of predicate logic used in Golem is the language Prolog. Prolog rules can easily express the learned relationships between objects such as molecular structures. Previous machine learning programs have lacked the ability to learn such relationships and neural network and statistical learning techniques also have similar difficulties. This gives ILP learning algorithms such as Golem a potential advantage in learning problems involving chemical structures.

Golem takes as input positive examples, negative examples and background knowledge described as Prolog facts. It produces as output Prolog rules which are generalizations of the examples. The Prolog rules are the least general rules which, given the background knowledge, can produce the input examples and none of the negative examples. The method of generalization is based on the logical idea of Relative Least General Generalization (RLLG). The basic algorithm used in Golem is as follows. First it takes a random sample of pairs of examples. In this application, this will be a set of pairs of residues chosen randomly from the set of all residues in all proteins represented. For each of these pairs Golem computes the set of properties which are common to both residues. These properties are then made into a rule which is true of both the residues in the pair under consideration. For instance, if the only common properties of the residues are that both residues are large and are three residues distant from a more hydrophilic residue then Golem would construct the following explanation for their being part of an α-helix.

```
alpha(Protein,Position):
  residue(Protein,Position,R),
  large (R),
  P3 = Position+3,
  residue(Protein,Position,R3),
  more_hydrophilic(R3,R). (see Representation of the problem)
```

Having built such a rule for all chosen pairs of residues, Golem takes each rule and computes the number of residues which that rule could be used to predict. Clearly these rules might predict some non α-helix residues to be part of an α-helix. Golem therefore chooses the rule which predicts the most α-helix conformation residues while predicting less than a predefined threshold of non α-helix residues. Having found the rule for the best pair, Golem then takes a further sample of as yet unpredicted residues and forms rules which express the common properties of this pair together with each of the individual residues in the sample. Again the rule which produces the best predictions on the training set is chosen. The process of sampling and rule building is continued until no improvement in prediction is produced. The best rule from this process is used to eliminate a set of predicted residues from the training set. The reduced training set is then used to build up further rules. When no further rules can be found the procedure terminates.

Representation of the problem

Three types of file are input into Golem: foreground examples (facts that are true), background facts and negative examples (facts that are false).

Foreground and negative examples. The following is a foreground example: alpha(Protein name,Position), e.g. alpha(155C, 105). This states that the residue at position 105 in protein 155C is an α-helix. The negative examples take the same form but state all residue positions in particular proteins in which the secondary structure is not an α-helix.

Background facts. The background facts contain a large variety of information about protein structure. The most basic is the primary structure information. For instance the fact: position(155C, 119, p) states that the residue at position 119 in protein 155C is proline (the standard 20 character coding for amino acids is used).

Table II. Definition of the more complicated properties of some unary predicates

Unary predicate	Definition
hydro_b_don	hydrogen bond donator
hydro_b_acc	hydrogen bond acceptor
not_aromatic	the complement of the aromatic class
small_or_polar	either small or polar
not_p	everything but proline
not_k	everything but lysine
aromatic_or_very_hydrophobic	either aromatic or very hydrophobic
ar_or_al_or_m	either aromatic or aliphatic or methionine

Because Golem does not have arithmetic information built in, information has to be given about the sequential relationships between the different positions (residues). These arithmetic-type relations allow indexing of the protein sequence relative to the residue being predicted. The first predicate describes nine sequential positions. For instance the fact octf(19, 20, 21, 22, 23, 24, 25, 26, 27), describing the sequence 19−27, can be used to index the four flanking positions on either side of position 23. The second type gives sequences that are considered to be especially important in α-helices (Lim, 1974). Thus, for instance, the background knowledge contains the facts alpha_triplet(5, 6, 9).alpha_pair(5, 8).alpha_pair4(5, 9). The predicate alpha_triplet contains the numbers n, $n + 1$ and $n + 4$. In an α-helix these residues will appear on the same face of the helix. Grouping these numbers together is a heuristic to allow the preferential search for a common relationship between these residues. Similarly, the residues with positions in the alpha_pair predicate (n and $n + 3$), and residues with positions in the predicate alpha_pair4 (n and $n + 4$) are expected to occur on the same face of a helix.

The physical and chemical properties of individual residues are described by the unary predicates hydrophobic, very_hydrophobic, hydrophilic, positive, negative, neutral, large, small, tiny, polar, aliphatic, aromatic, hydro_b_don, hydro_b_acc, not_aromatic, small_or_polar, not_p, ar_or_al_or_m, not_k, aromatic_or_very_hydrophobic (Taylor, 1986). Each of these is expressed in terms of particular facts, such as small(p) meaning that proline is a small residue. The more complicated properties are given in Table II. The rather unusual looking logical combinations, such as aromatic_or_very hydrophobic, have been found useful previously (King

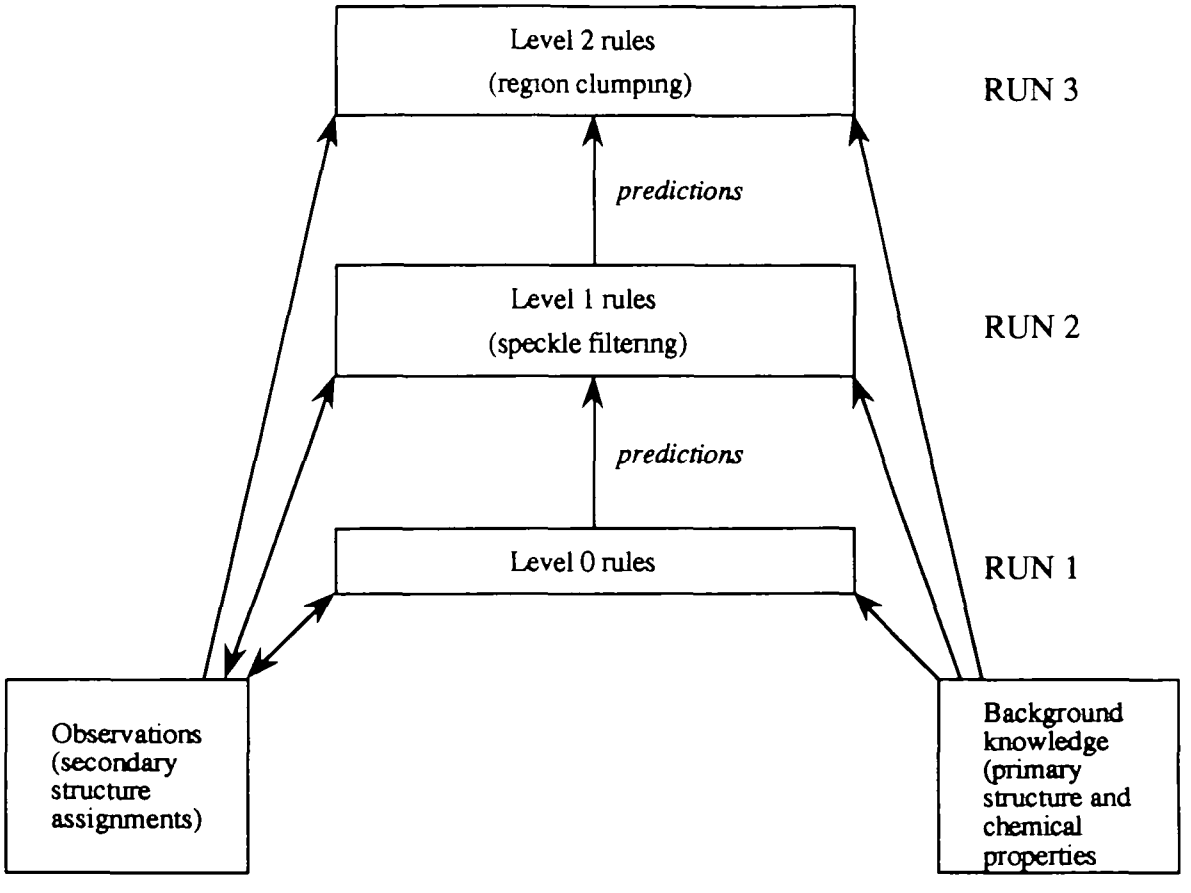

Fig. 2. Process used to generate the three levels of rules showing the flow of information.

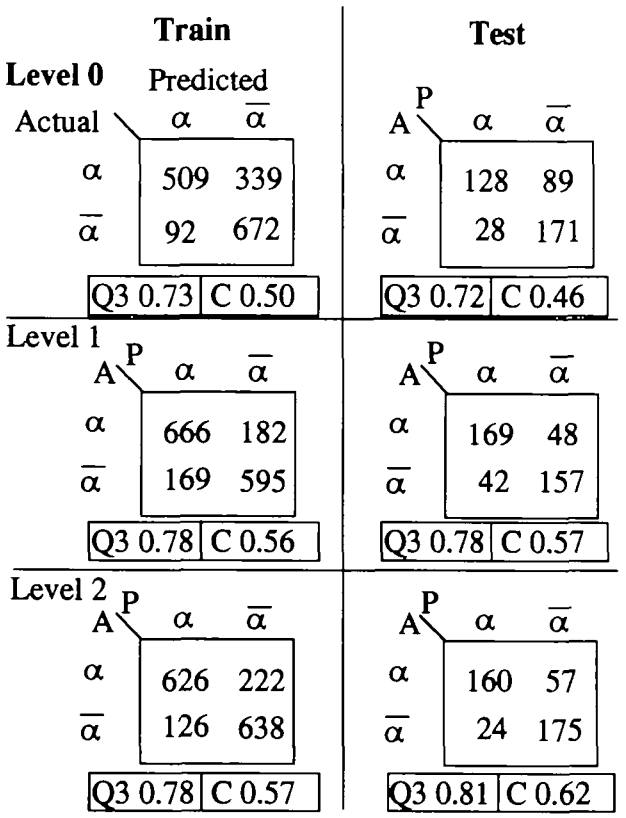

Fig. 3. Confusion matrices and Q_3 percentage accuracies of rules found (P = predicted, A = actual). Each matrix has the form $\begin{pmatrix} A & B \\ C & D \end{pmatrix}$. The Q_3 percentage accuracies below each matrix are calculated as P*100 where $P = (A + D)/(A + B + C + D)$. Each percentage is followed by SE (i.e. ±2). SE is given as S*100 where $S = \sqrt{(P(1 - P)/(A + B + C + D))}$.

and Sternberg, 1990). (For some runs, similar predicates to not_p were created for all 20 residue types.)

Information was also given about the relative sizes and hydrophobicities of the residue. This was described using the binary predicates ltv and lth. Each is expressed in terms of particular facts such as ltv(X, Y), meaning that X is smaller than Y [scale taken from Schulz and Schirmer (1978)], and lth(X, Y), meaning that X is less hydrophobic than Y [scale taken from Eisenberg (1984)].

Experimental procedure

A Golem run takes the form of asking Golem to find good generalization rules. These generalizations can then be either accepted or Golem can be asked to try and find another generalization. A prediction rule is accepted if it has high accuracy and good coverage. If a rule is accepted, then the examples it covers are removed from the background observations (true and false facts), and the rule is added to the background information. Learning stops when no more generalizations can be found within set conditions.

Golem was first run on the training data using the above background information. A certain amount of 'noise' was considered to exist in the data and Golem was set to allow each rule to misclassify up to 10 negative instances. To be accepted, rules had to have >70% accuracy and coverage of at least 3%. If a rule had lower coverage than this it would not be statistically reliable. Learning was stopped when no more rules could be found to meet these conditions. Each determined rule was typically very accurate (often >90% correct classification):

Table III. The Q_3 accuracy of the predictions for each individual protein for the Golem and GOR methods

No.	Protein identification	Set type	Golem Q_3	GOR Q_3
1	155C	train	87.6	64.5
2	1CC5	train	90 4	69.9
3	1CCR	train	66 7	77.5
4	1CRN	train	80.4	65.2
5	1CTS	train	76.2	81 2
6	1ECD	train	76.5	77 9
7	1HMQ	train	68.1	81 4
8	1MBS	train	76.5	69 9
9	2B5C	train	82.4	51 8
10	2C2C	train	88.4	62 5
11	2CDV	train	82.2	68.2
12	3CPV	train	77.8	59 3
13	156B	test	76.7	82 5
14	1BP2	test	81.3	62.6
15	351C	test	78 0	79.3
16	8PAP	test	85 2	71.3

overall ~60% of the instances were classified by the rules as a whole. The accuracy and coverage settings used to find the rules were based largely on subjective judgement and experience. Work is being carried out to replace the need for subjective judgement by objective measures from statistical and algorithmic information theory.

To improve on the coverage found by these first rules, the learning process was iterated. The predicted secondary structure positions found using the first rules (level 0 rules) were added to the background information (Figure 2) and then Golem was re-run to produce new rules (level 1 rules). This forms a kind of bootstrapping learning process, with the output of a lower level of rules providing the input for the next level. This was needed because after the level 0 rules, the predictions made were quite speckled, i.e. only short sequences of α-helix predicted residues interspersed by predictions of coil secondary structure. The level 1 rules have the effect of filtering the speckled prediction and joining together the short sequences of α-helix predictions. The iterative learning process was repeated a second time, with the predicted secondary structure positions from the level 1 rules being added to the background information, and new rules found (level 2 rules). The level 2 rules had the effect of reducing the speckling even more and clumping together sequences of α-helix. Some of the level 1 and 2 rules were finally generalized by hand with the formation of the symmetrical variants of the rules found by Golem.

Results

Applying Golem to the training set produced 21 level 0 rules, five symmetrical level 1 rules and two symmetrical level 2 rules. These rules combined together to produce a Q_3 accuracy of 78% and a Matthews correlation (Schulz and Schirmer, 1978) of 0.57 in the training set, and a Q_3 accuracy of 81% and a Matthews correlation of 0.62 in the test set (Figure 3 and Table III). Q_3 accuracy is defined as $((W + X)/T)$*100, and the Matthews correlation is defined as $((W*X) - (Y*Z))/\sqrt{(X + Y)(X + Z)(W + Y)(W + Z)}$, where W is the number of correct helical predictions, X is the number of correct coil (not helical) predictions, Y is the number of helical residues predicted to be coil, Z is the number of coil residues predicted to be helical and T is the total number of residues. The SE in this Q_3 prediction accuracy was estimated to be ~2%. SE is calculated as

Training Proteins

155C

```
negdaakgekefnkckachmiqapdgtdikggktgpnlygvvgrkiaseegfkygegilevaeknpdltwteanlieyvtdpkplv
----HHHHHHHHHHH----------------------------------HHHHHHHHHH------HHHHHHHHH-----
----HHHHHHHHHHH----------------------------HHHH-----HHHHHHHH--------HHHHHHHH--------
-HH-HHHHHHHHHHHHHH---------H---------------H---HHH--H--HHHHHHH----HHHHHHHHHHHH----HH
0000000100010000000010000000001000000000110001000000000001001000000111000110000010011

kkmtddkgaktkmtfkmgknqadvvaflaqddpda
------------------HHHHHHHHHHHH---
------------------HHHHHHHHHH------
HHHHHHHH-HHHHHHHH---HHHHHHHHHH---HH
0000000010000000100000010110001000
```

1CC5

```
gggarsgddvvakycnachgtgllnapkvgdsaawktradakggldgllaqslsglnamppkgtcadcsddelkaaigkmsgl
------HHHHHHHHHHHHHHHHHHH----HHHHHHHHHHHH----HHHHHHHH------------HHHHHHHHHHHH-
-----HHHHHHHHHHHHHHHHHHHH---HHHHHH-HHHHHHHHHHHHHHHHHH------------HHHHHHHHHHHH-
----------HHHH-------------HHH-HHHHHHHHHH--HHHHHHHH---------------HHHHHHHHHH-
0000001001000010000000000100000000100010000001110010000000000010010001101111011100
```

1CCR

```
asfseappgnpkagekifktkcaqchtvdkgaghkqgpnlnglfgrqsgttpgysystadknmaviweentlydyllnpkkyipgt
---------HHHHHHHHHHHHHHHH-----------------------------HHHHHHHH---HHHHHHHHHHHHHH----
-------------------------------------------------------HHHHH------HHHHH---HHHH---
------------H--HHHHHHHH-----------------------------------HHHHHHHHHHHHHHHHHH---------
0000000000000100010000000001000000001010001000010000000000000000101000011011001000100 1
```

```
kmvfpglkkpqeradlisylkeats
--------HHHHHHHHHHHHHHHH-
-----------HHHHHHHHH-----
HHH--------HHHHHHHHHHHH--
000000100000110110010001 0
```

1CRN

```
ttccpsivarsnfnvcrlpgtpeaicatytgciiipgatcpgdyan
------HHHHHHHHHH-----HHHHHHHH----------------
-----HHHH-----HHHH---HHHHHHHHHH---------------
-------------------------HHH-------------------
00110000100010000000000001000001000000000000
```

1CTS

```
asstnlkdiladlipkeqariktfrqqhgntvvgqitvdmmyggmrgmkglvyetsvldpdegirfrgysipecqkmlpkakggee
------HHHHHHHHHHHHHHHHHHHH----------HHHHH----------------------------HHHHHHH---------
------HHHHHHH----HHHHHHHHHHHHHHHH-HHHHHHHH-----------------------------------------------
--HHHHHHHHHHHHHHHHHHHHHHHH------------H-H---------------H-------HH------HHHHH---------
0000010001000100000000000000000000000000000000000001011000000110100011001000100100001

plpeglfwllvtgqipteeqvswlskewakraalpshvvtmldnfptnlhpmsqlsaaitalnsesnfarayaegihrtkyweliy
--HHHHHHHHH------HHHHHHHHHHHHH----HHHHHHHH-------HHHHHHHHH------HHHHHHHH------HHHHHH
----HHHHHHHHH------HHHHHHH---------HHHHHHHHH-------HHHHHHHHHH----HHHHHHHHH----HHHHHHH
---HHHHHHHH------HHHHHHHHHHHHHHH--HHHHHHH---------HHHHHHHHHHHHHHHHHHHHHHHHHHHH----HH
1111111111110001000010011001000000000100010000000011011101100100010100100000000110011

edcmdliaklpcvaakiyrnlyregssigaidskldwshnftnmlgytdaqftelmrlyltihsdheggnvsahtshlvgsalsdp
HHHHHHHHHHHHHHHHHHHH---------------HHHHHHHHH----HHHHHHHHHHHH--------HHHHHHHH-------H
H-HHHHHHH-----HHHHH--------------------HHHH-----HHHHHHHHHHHHHH-----HHHHHHHHHH------H
HHHHHHHHH--HHHHHHH---HH--------HHH-H-HHHH-HHH-----HHHHHHHHHHHH----------HHHHHHH-------
0111011111111111110000000000000000001101110110100001111111111111100100011001100100011

ylsfaaamnglagplhglanqevlvwltqlqkevgkdvsdeklrdyiwntlnsgrvvpgyghavlrktdprytcqrefalkhlphd
HHHHHHHHHH--------HHHHHHHHHHHH--------HHHHHHHHHH----------------HHHHHHHHHH----
HHH--HHHHHH--------HHHHHHHHHH-------------HHHHHHHHHHHH----HHHHHH--------------------
HHHHHHHHHHH--HHH-HHHHHHHHHHHHHH----HHHHHHHHHH-H-----------HHH-------HHHHHHHHH----
1011101100100000100110110010010001000000001000100100000010010000100111110110010011000 1
```

Fig. 4.

Fig. 4. (Continued)

Fig. 4. (Continued)

```
8PAP
ipeyvdwrqkgavtpvknqgscgscwafsavvtiegiikirtgnlnqyseqelldcdrrsygcnggypwsalqlvaqygihyrnty
----------------------HHHHHHHHHHHHHHHH-------HHHHHHH----------HHHHHHHHH---------
--------------------HHHHHHHHHHHHHHHHHHHH--------HHHHHHH--HHHH-------HHHHHHHHH--------
---HHHHHHH--------------------HHHHHH-HHHH--------HHHHH----------------HHHHHHHHH-------
00001100000101001010010111111111111111000000000111111110100101000001011101100011100001

pyegvqrycrsrekgpyaaktd
----------HHHH--------
-----------------------
-----------------HHHHH
00000000001000000000 10
(N.B. exposure based on 9PAP)
```

Fig. 4. Residue by residue predictions of secondary structure for the training and test data The top line is the primary structure of the proteins The second line is the actual secondary structure (H signifies a residue with α-helix secondary structure, − signifies a residue with coil secondary structure) The third line is the Golem predicted secondary structure The fourth line is the GOR predicted secondary structure The last line is the exposure profile of the protein (1 signifies $<10\%$ of area exposed; 0 signifies $\geq 10\%$ exposed).

$\sqrt{(P(1-P)/T)}$, where $P = Q_3/100$. The prediction rates of the training and test sets agree within the range of the SE. However, the figure of 81% prediction on the test set at level 2 may be a slight overestimate, with a level of 79% being more consistent with the test set prediction. These are the most accurate predictions found by any method to date. The next best reported accuracy is that of 76% (on a slightly more homologous database than used in this study) using a neural network approach (Kneller *et al.*, 1990). A more accurate estimate of prediction accuracy could be made using cross-validation. Unfortunately it takes ~1 h on a workstation to generate each of the 28 rules. This time would have to be multiplied by 16 (once for each protein).

To compare further the accuracy of the Golem predictions, the Standard Garnier−Osguthorpe−Robson (GOR) prediction method was applied to the data (Gibrat *et al.*, 1987). In this comparison the standard residue parameters were used (this biases the results in favour of GOR because the parameters are based on some of the proteins in the test set). In addition, the decision constants were optimized to give the highest accuracy in the training data. GOR produced a Q_3 accuracy of 72% and a Matthews correlation of 0.44 in the training set, and a Q_3 accuracy of 73% and a Matthews correlation of 0.46 in the test set (Figure 3 and Table III). The residue by residue predictions for Golem and GOR are given in Figure 4. Examination of these predictions shows that GOR predictions are more ragged than those of Golem, e.g. producing predictions of helices consisting of single residues. The GOR prediction might have benefited from some filtering process such as that produced by the Golem level 1 and 2 rules. It is interesting to note that occasionally the Golem and GOR predictions both agree on predicting a helix where none exists. e.g. in protein 2B5C. This may be evidence for incipient helix formation prevented by higher level interactions.

In an attempt to characterize the reasons for the successes and failures of the Golem and GOR methods, the exposure profiles of the proteins were produced (Figure 4). These were then examined to see if there was any correlation between particular exposure profiles and prediction success. Unfortunately there appears to be no relationship between the amount of exposure and the success of the prediction methods. Most of the proteins in the data set are quite small and have few residues deeply buried; however, in 1CTS, where there are helices almost completely buried, Golem and GOR seem to have equal success with these buried helices as they do with amphipathic helices

Although Q_3 and the Matthews correlation are the accepted measure for accuracy, it would be better to have a measure more closely related to the information given about conformation, e.g.

```
% level 0 rules
% JOINT(1-21) TRAIN: 509/92 (85%acc,6%cov) TEST. 128/28 (82%acc,59%cov)

% RULE 1    TRAIN: 48/9 (84%acc,6%cov)    TEST: 12/0 (100%acc,5%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
   position(A,D,N), not_aromatic(N), not_k(N),
   position(A,F,L), hydrophobic(L),
   position(A,G,O), not_aromatic(O), not_p(O),
   position(A,B,C), not_aromatic(C), not_p(C),
   position(A,H,Q), not_p(Q), not_k(Q),
   position(A,I,M), hydrophobic(M), lth(N,M), ltv(M,O),
   position(A,K,P), not_aromatic(P), lth(P,L).


% RULE 2    TRAIN: 30/1 (97%acc,4%cov)    TEST  10/0 (100%acc,5%cov)
alpha0(A,B)  - octf(D,E,F,G,B,H,I,J,K),
   position(A,D,Q), not_p(Q), not_k(Q),
   position(A,E,O), not_aromatic(O), small_or_polar(O),
   position(A,F,R),
   position(A,G,P), not_aromatic(P),
   position(A,B,C), very_hydrophobic(C), not_aromatic(C),
   position(A,H,M), large(M), not_aromatic(M),
   position(A,I,L), hydrophobic(L),
   position(A,K,N), large(N), ltv(N,R)

% RULE 3    TRAIN. 34/3 (92%acc,4%cov)    TEST: 7/0 (100%acc,3%cov)
alpha0(A,B) .- octf(D,E,F,G,B,H,I,J,K),
   position(A,D,L), neutral(L), aromatic_or_very_hydrophobic(L),
   position(A,E,R), not_p(R), not_k(R),
   position(A,F,Q), small_or_polar(Q), not_k(Q),
   position(A,B,C), neutral(C),
   position(A,G,O), not_aromatic(O),
   position(A,H,P), not_aromatic(P), not_p(P), not_k(P),
   position(A,I,M), neutral(M), not_aromatic(M), not_p(M),
   position(A,J,N), neutral(N), aromatic_or_very_hydrophobic(N), ltv(N,Q)

% RULE 4    TRAIN: 45/2 (96%,5%cov) TEST· 14/1 (93%acc,6%cov)
alpha0(A,B)  - octf(D,E,F,G,B,H,I,J,K),
   position(A,E,M), neutral(M),
   position(A,F,O), not_aromatic(O), not_p(O), not_k(O),
   position(A,G,P), not_aromatic(P),
   position(A,B,C), large(C), not_aromatic(C), not_k(C),
   position(A,H,N), neutral(N), aromatic_or_very_hydrophobic(N),
   position(A,I,R), not_p(R),
   position(A,J,Q), not_aromatic(Q), not_p(Q), not_k(Q),
   position(A,K,L), hydrophobic(L)

% RULE 5    TRAIN: 49/12 (80%,6%cov)      TEST· 10/1 (91%acc,5%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
   position(A,D,M), not_aromatic(M), not_p(M), not_k(M),
   position(A,E,O), small_or_polar(O), not_p(O), not_k(O),
   position(A,F,N), not_aromatic(N), not_p(N),
   position(A,G,P), not_p(P),
   position(A,B,C), hydrophobic(C), lth(M,C),
   position(A,H,Q), not_p(Q), lth(Q,C),
   position(A,I,L), hydrophobic(L),
   position(A,J,S), ltv(S,M),
   position(A,K,R), not_k(R).

% RULE 6    TRAIN: 31/2 (94%acc,3%cov)    TEST: 7/1 (88%acc,3%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
   position(A,D,N), not_aromatic(N), small_or_polar(N),
   position(A,E,O), not_aromatic(O), not_k(O),
   position(A,F,P), not_aromatic(P),
   position(A,G,M), hydrophilic(M), hydro_b_acc(M),
   position(A,B,C), hydrophobic(C),
   position(A,H,Q), not_aromatic(Q), not_k(Q),
   position(A,I,R), not_aromatic(R), small_or_polar(R),
   position(A,J,L), hydrophobic(L), not_aromatic(L), small_or_polar(L)

% RULE 7    TRAIN  29/1 (97%acc,3%cov)    TEST: 7/1 (88%acc,3%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
   position(A,E,M), polar(M), hydro_b_acc(M),
   position(A,G,L), large(L), not_k(L),
   position(A,B,C), large(C), not_k(C),
   position(A,H,O), not_p(O), not_k(O),
   position(A,J,P),
   position(A,K,N), not_aromatic(N), lth(N,P)

% RULE 8    TRAIN: 40/8 (83%acc,5%cov)    TEST. 6/1 .86%acc,3%cov
alpha0(A,B)  - octf(D,E,F,G,B,H,I,J,K),
   position(A,D,N), not_aromatic(N), small_or_polar(N),
   position(A,E,L), hydrophobic(L), large(L),
```

```
    position(A,F,Q), ltv(L,Q),
    position(A,B,C), not_aromatic(C), not_p(C),
    position(A,H,P), not_p(P), not_k(P),
    position(A,I,M), neutral(M), large(M),
    position(A,K,O), not_aromatic(O), small_or_polar(O), not_p(O)

% RULE 9    TRAIN: 40/2 (95%,5%cov) TEST: 16/3 (84%acc,7%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
    position(A,D,M), not_aromatic(M), not_k(M),
    position(A,E,N), not_aromatic(N), small_or_polar(N), not_k(N),
    position(A,F,R), not_k(R),
    position(A,G,O), not_aromatic(O), not_p(O), not_k(O),
    position(A,B,C), not_aromatic(C),
    position(A,H,P), not_aromatic(P), not_p(P), not_k(P),
    position(A,J,L), hydro_b_don(L), lth(L,C), ltv(L,P),
    position(A,K,Q), not_aromatic(Q), not_k(Q)

% RULE 10   TRAIN: 33/3 (92%acc,4%cov)    TEST  9/2 (82%acc,4%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
    position(A,D,N), not_aromatic(N), not_k(N),
    position(A,E,O), not_aromatic(O),
    position(A,F,P), not_aromatic(P), not_p(P),
    position(A,G,R), not_p(R), not_k(R),
    position(A,B,C), not_p(C), not_k(C),
    position(A,H,Q), not_aromatic(Q),
    position(A,I,L), hydrophobic(L), ltv(C,L), ltv(N,L), ltv(P,L),
    position(A,J,M), hydrophobic(M), not_aromatic(M),
    position(A,K,S), not_p(S), not_k(S).

% RULE 11   TRAIN. 40/3 (93%acc,5%cov)    TEST. 9/2 (82%acc,4%cov)
alpha0(A,B) .- octf(D,E,F,G,B,H,I,J,K),
    position(A,D,O), not_aromatic(O), not_k(O),
    position(A,E,P), not_aromatic(P),
    position(A,F,Q), not_aromatic(Q), not_p(Q), not_k(Q),
    position(A,G,R), not_p(R), ltv(R,P),
    position(A,B,C), not_aromatic(C),
    position(A,H,N), large(N), ltv(N,L),
    position(A,I,L), hydrophobic(L),
    position(A,J,M), hydrophobic(M), not_aromatic(M),
    position(A,K,S), not_p(S).

% RULE 12   TRAIN: 58/3 (95%acc,7%cov)    TEST: 13/3 (81%acc,6%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
    position(A,F,Q), not_p(Q),
    position(A,G,N), not_aromatic(N), not_p(N),
    position(A,B,C), large(C), not_aromatic(C), not_k(C),
    position(A,H,L), hydrophobic(L), not_k(L),
    position(A,I,O), not_aromatic(O), not_p(O),
    position(A,J,P), not_aromatic(P), small_or_polar(N), not_p(P),
    position(A,K,M), hydrophobic(M), not_k(M)

% RULE 13   TRAIN: 29/1 (97%acc,3%cov)    TEST: 4/1 (80%acc,2%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
    position(A,D,N), not_aromatic(N),
    position(A,E,O), not_aromatic(O), small_or_polar(O), not_p(O),
    position(A,F,R), lth(R,N),
    position(A,B,C), not_p(C), not_k(C),
    position(A,H,P), not_aromatic(P), not_p(P), lth(P,Q),
    position(A,I,L), hydrophobic(L), not_aromatic(L), small_or_polar(L),
    position(A,J,Q), not_aromatic(Q), not_p(Q), not_k(Q),
    position(A,K,M), hydrophobic(M).

% RULE 14   TRAIN. 45/4 (92%acc,5%cov)    TEST: 14/3 (82%acc,6%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
    position(A,E,O), not_aromatic(O), not_p(O),
    position(A,F,P), small_or_polar(P), not_aromatic(P), not_p(P),
    position(A,G,Q), not_aromatic(Q), not_k(Q),
    position(A,B,C), hydrophobic(C), neutral(C),
    position(A,H,L), hydrophobic(L), neutral(L),
    position(A,I,M), hydrophobic(M),
    position(A,J,N), neutral(N), not_p(N),
    position(A,K,R), not_aromatic(R), small_or_polar(R).

% RULE 15   TRAIN  28/5 (85%acc,3%cov)    TEST  4/1 (80%acc,2%cov)
alpha0(A,B) :- octf(D,E,F,G,B,H,I,J,K),
    position(A,D,P), not_k(P),
    position(A,E,Q), not_k(Q),
    position(A,F,O), not_aromatic(O), not_p(O),
    position(A,G,L), hydrophobic(L), small_or_polar(L), not_aromatic(L),
    position(A,B,C), polar(C), lth(C,S),
    position(A,H,S),
    position(A,I,R), not_k(R),
    position(A,J,N), neutral(N), not_p(N),
    position(A,K,M), hydrophobic(M), not_aromatic(M)

% RULE 16   TRAIN: 25/1 (96%acc,3%cov)    TEST: 4/1 (80%acc,2%cov)
alpha0(A,B) :- octf(C,D,E,F,B,G,H,I,J), octf(K,L,M,N,C,D,E,F,B),
    position(A,C,S), not_aromatic(S), not_p(S),
    position(A,D,U), small_or_polar(U), not_k(U),
    position(A,E,T), not_aromatic(T), not_p(T), not_k(T),
    position(A,F,V), not_p(V),
    position(A,B,O), not_p(O), not_k(O), ltv(O,R),
    position(A,G,P), very_hydrophobic(P), not_aromatic(P),
    position(A,H,Q), large(Q),
    position(A,I,R), small(R),
    position(A,J,M), not_p(M).

% RULE 17   TRAIN: 34/5 (87%acc,4%cov)    TEST: 4/1 (80%acc,2%cov)
alpha0(A,B) :- octf(C,D,E,F,B,G,H,I,J), octf(K,L,M,N,C,D,E,F,B),
    position(A,C,R), not_aromatic(R), not_p(R), not_k(R),
    position(A,D,S), not_aromatic(S), small_or_polar(S),
    position(A,E,O), very_hydrophobic(O), large(O),
    position(A,F,Q), neutral(Q), not_p(Q),
    position(A,G,P), very_hydrophobic(P), not_aromatic(P),

% RULE 18   TRAIN: 26/3 (90%acc,3%cov)    TEST: 4/0 (100%acc,2%cov)
alpha0(A,B) :- octf(C,D,E,F,B,G,H,I,J), octf(K,L,M,N,C,D,E,F,B),
    position(A,C,T), not_p(T), lth(T,R),
    position(A,D,S), not_aromatic(S),
    position(A,E,R), large(R), ltv(R,T),
    position(A,F,Q), neutral(Q), not_p(Q),
    position(A,B,O), polar(O), lth(O,S),
    position(A,G,P), hydrophobic(P), not_aromatic(P),

% RULE 19   TRAIN: 22/4 (85%acc,3%cov)    TEST: 5/1 (83%acc,2%cov)
alpha0(A,B) .- octf(D,E,F,G,B,H,I,J,K),
    position(A,D,P), not_k(P), not_p(P),
    position(A,E,N), small_or_polar(N),
    position(A,F,R), lth(R,M),
    position(A,G,L), hydrophobic(L), small_or_polar(L),
    position(A,B,C), not_p(C),
    position(A,H,O), small_or_polar(O), not_k(O), not_p(O),
    position(A,I,M), hydrophobic(M), not_aromatic(M),
    position(A,J,S), ltv(S,Q),
    position(A,K,Q), not_k(Q)

RULE 20     TRAIN: 84/20 (81%acc,10%cov)  TEST: 18/4 (82%acc,8%cov)
alpha0(A,B) :- alpha_pair3(B,D), alpha_triplet(D,E,F),alpha_triplet(B,H,E),
    position(A,B,C), not_e(C), not_f(C), not_g(C), not_i(C), not_k(C),
    position(A,F,G),
    position(A,H,I), not_c(I), not_e(I), not_f(I), not_g(I), not_h(I),
    position(A,D,J), not_c(J), not_d(J), not_e(J), not_f(J), not_g(J),
    position(A,E,K), not_d(K), not_e(K), not_f(K), not_h(K), not_r(K),

RULE 21     TRAIN: 72/18 (72%acc,8%cov)   TEST: 18/4 (82%acc,8%cov)
alpha0(A,B) .- alpha_triplet(B,D,E), alpha_pair3(B,G),
    position(A,B,C), not_c(C), not_e(C), not_g(C), not_k(C), not_m(C),
    position(A,D,F), not_c(F), not_d(F), not_f(F), not_g(F), not_h(F),
    position(A,G,H), not_c(H), not_d(H), not_f(H), not_h(H), not_l(H),
    position(A,E,I), not_c(I), not_d(I), not_e(I), not_h(I), not_i(I),

% level 1 rules
% JOINT  TRAIN: 666/169 (80%acc,78%cov)   TEST: 169/42 (80%acc,83%cov)

% RULE 22 TRAIN: 509/92 (85%acc,60%cov)   TEST. 128/28 (82%acc,59%cov)
alpha1(A,B) :- alpha0(A,B).

% RULE 23a  TRAIN: 299/52 (85%acc,35%cov) TEST: 83/10  (89%acc,38%cov)
alpha1(A,B) :- octf(D,E,F,G,B,H,I,J,K), alpha0(A,F), alpha0(A,G).
% RULE 23b  TRAIN: 303/44 (87%acc,36%cov) TEST: 85/7   (92%acc,40%cov)
alpha1(A,B) .- octf(D,E,F,G,B,H,I,J,K), alpha0(A,H), alpha0(A,I).

% RULE 24a  TRAIN: 183/10 (95%acc,22%cov) TEST: 53/2   (96%acc,24%cov)
alpha1(A,B) :- octf(C,D,E,F,B,G,H,I,J), alpha0(A,F), alpha0(A,G),
               alpha0(A,H).
% RULE 24b  TRAIN: 189/5 (97%acc,22%cov)  TEST: 54/2   (96%acc,25%cov)
alpha1(A,B) .- octf(C,D,E,F,B,G,H,I,J), alpha0(A,E), alpha0(A,F),
               alpha0(A,G).

% RULE 25   TRAIN: 102/2 (98%acc,12%cov)  TEST. 35/1   (97%acc,16%cov)
alpha1(A,B) :- octf(C,D,E,F,B,G,H,I,J),
               alpha0(A,E), alpha0(A,F), alpha0(A,H), alpha0(A,I).

% RULE 26a  TRAIN: 102/3 (98%acc,12%cov)  TEST· 36/0   (100%acc,17%cov)
alpha1(A,B) :- octf(C,D,E,F,B,G,H,I,J), alpha0(A,D), alpha0(A,E),
               alpha0(A,G), alpha0(A,H).
% RULE 26b  TRAIN· 86/6 (93%acc,10%cov)   TEST: 33/0   (100%acc,15%cov)
alpha1(A,B) :- octf(C,D,E,F,B,G,H,I,J), alpha0(A,C), alpha0(A,D),
               alpha0(A,G), alpha0(A,H).
% RULE 26c  TRAIN: 88/5 (95%acc,10%cov)   TEST· 32/1   (97%acc,15%cov)
alpha1(A,B) :- octf(C,D,E,F,B,G,H,I,J), alpha0(A,D), alpha0(A,E),
               alpha0(A,H), alpha0(A,I)
% RULE 26d  TRAIN: 87/5 (95%acc,10%cov)   TEST. 32/1   (97%acc,15%cov)
alpha1(A,B) .- octf(C,D,E,F,B,G,H,I,J), alpha0(A,E), alpha0(A,F),
               alpha0(A,I), alpha0(A,J).

% level 2 rules
% JOINT TRAIN  626/126 (83%acc,74%cov)    TEST· 160/24 (87%acc,74%cov)

RULE 27
alpha2(A,B) .- octf(C,D,E,F,B,G,H,I,J), alpha1((A,B), alpha1((A,G),
               alpha1((A,H).
alpha2(A,B) .- octf(C,D,E,F,B,G,H,I,J), alpha1((A,B), alpha1((A,E),
               alpha1(A,F)

% RULE 28
alpha2(A,B) :- octf(C,D,E,F,B,G,H,I,J),alpha1(A,B), alpha1(A,D),
               alpha1(A,E), alpha1(A,F), alpha1(A,G), alpha1(A,H).
alpha2(A,B)  - octf(C,D,E,F,B,G,H,I,J), alpha1(A,B), alpha1(A,G),
               alpha1(A,H), alpha1(A,I).
alpha2(A,B) :- octf(C,D,E,F,B,G,H,I,J), alpha1(A,B), alpha1(A,D),
               alpha1(A,E), alpha1(A,F).
alpha2(A,B) :- octf(C,D,E,F,B,G,H,I,J), alpha1(A,B), alpha1(A,F),
               alpha1(A,G), alpha1(A,H)
alpha2(A,B) .- octf(C,D,E,F,B,G,H,I,J), alpha1(A,B), alpha1(A,E),
               alpha1(A,F), alpha1(A,G).
```

Fig. 5. The list of rules found by Golem at level 0, level 1 and level 2. The performance of each rule on the training and test data is given as the number of correctly predicted residues and wrongly predicted residues (in rule 1 on the training data, 48 residues correct and 8 residues wrong) and the percentage accuracy and coverage. The rules are in Prolog format: Head.-Body. This means that if the conditions in the body are true then the head is true. Taking the example of rule 1: there is an α-helix residue in protein A at position B (the head) if: at position D in protein A (position B − 4) the residue is not aromatic and not lysine; and at position F in protein A (position B − 2) the residue is hydrophobic; and at position D in protein A (position B − 1) the residue is not aromatic and not proline; and at position B in protein A the residue is not aromatic and not proline; and at position H in protein A (position B + 1) the residue is not proline and not lysine; and at position I in protein A (position B + 2) the residue is hydrophobic and has a lower hydrophobicity than the residue at position D and a lower volume than the residue at position G; and at position K in protein A (position B + 4) the residue is not aromatic and has a lower hydrophobicity than the residue at position F. For the level 1 rules, a prediction of a helix by a level 0 rule is signified by alpha0(Protein, Position). The positions of predictions made by the level 1 rules used by the level 2 rules are signified by alpha(Protein,Position).

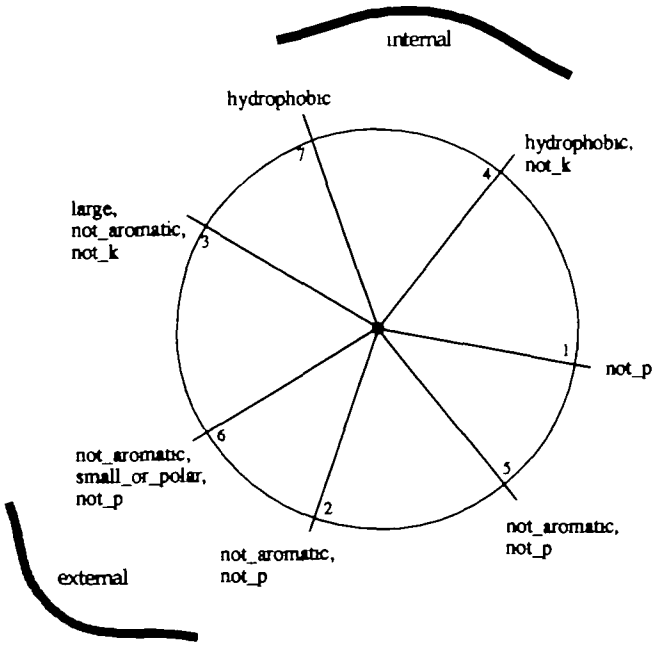

Fig. 6. Helical wheel plan of rule 12

are the correct number of α-helices predicted. It would also be useful to take into account that the boundary between α-helices and coil secondary structure can be ambiguous.

The rules generated by Golem can be considered to be hypotheses about the way α-helices form (Figure 5). They define patterns of relationships which, if they exist in a sequence of residues,will tend to cause a specified residue to form part of an α-helix. For example, considering rule 12, this rule specifies the properties of eight sequential residues which, if held, will cause the middle residue in the sequence (residue B) to form part of a helix. These rules are of particular interest because they were generated automatically and do not reflect any preconceived ideas about possible rule form (except those unavoidably built into the selection of background information). The rules raise many questions about α-helix formation. Examining rule 12, the residue p (proline) is disallowed in a number of positions, but allowed in others—yet proline is normally considered to disrupt protein secondary structure. It is therefore of interest to understand under exactly what circumstances proline can be fitted into an α-helix. One of the most interesting features to be highlighted by the rules was the importance of relative size and hydrophobicity in helix formation, not just absolute values. It is an idea which warrants further investigation (N.B. relative values cannot easily be used in statistical or neural network methods).

One technique of making the level 0 rules more comprehensible is to display them on a helical wheel plan—a projection showing the α-helix from above and the different residue types sticking out from the sides at the correct angle, see rule 12 in Figure 6. Rule 12 shows amphipathicity, the tendency in α-helices of hydrophobic residues and hydrophilic residues to occur on opposite faces of the α-helix. This property is considered central in α-helical structure (Lim, 1974; Schulz and Schirmer, 1978). However, most of the level 0 rules, when displayed on helical wheels, do not display such marked amphipathicity, and a detailed survey of the location of the positive and negative examples of the occurrences of the rules is required. This would involve a database analysis combined with the use of interactive graphics.

One problem raised with the rules, by protein structure experts, is that although they are much more easily understood than an array of numerical parameters or a Hinton diagram (Qian and Sejnowski, 1988) for a neural network, they still appear somewhat complicated. This raises the question about how complicated the rules for forming α-helices are (King and Sternberg, 1990) and it may be that any successful prediction rules are complicated. The failure of Rooman and Wodak (1988) to produce especially high accuracy using patterns with only three residues specified lends support to this idea. One approach to make the rules easier to understand may be to find over-general rules, and then generalize the exception to these rules (Bain, 1991). In such a procedure the over-general rules would tend to be easier to comprehend.

The meaning of the level 1 and 2 rules is much clearer to understand. In proteins, secondary structure elements involve sequences of residues, e.g. an α-helix may occupy 10 sequential residues and then be followed by eight residues in a region of coil. However, the level 0 rules output predictions based only on individual residues. This makes it possible for the level 0 rules to predict a coil in the middle of a sequence of residues predicted to be of α-helix type; this is not possible in terms of structural chemistry. This shortcoming is dealt with by the level 1 and 2 rules which group together isolated residue predictions to make predictions of sequences. For example, rule 23 predicts that a residue will be in an α-helix if both residues on either side have already been predicted to be α-helices; similarly rule 24 predicts that a residue will be in an α-helix if two residues on one side and one residue on the other side have already been predicted to be α-helices.

Discussion

It is intended to extend the application of Golem to the protein folding problem by adding more background knowledge, such as the division of each α-helix into three parts (beginning, middle and end). Analysis has shown that a specific pattern of residues can occur at these positions (Presta and Rose, 1988; Richardson and Richardson, 1988). This is thought to occur because the physical/chemical environment experienced by the three different sections is very different: both the beginning and end have close contact with coil regions, while the middle does not; also a dipole effect causes the end of an α-helix to be more negative than the beginning. Evidence for the usefulness of this division is given by rules 17 and 18. The structure of these rules suggests that they are biased towards the end of α-helices—the residues predicted by these rules occur at the end of the sequence of defined primary structure. Examination of the occurrences of these rules confirms this, showing that their predictions tend to occur at the end of α-helices (and often occur together). Protein secondary structure prediction methods normally only consider local interactions and this is the main reason for their poor success. One way of tackling this problem is that used in this paper of iterative predictions based on previous predictions. This may be extended by using well defined long range interactions such as super-secondary structure and domain structures (Shulze-Kremer and King, 1992) or by using models of constraints (Murzin and Finkelstein, 1988). It is hoped that with the addition of such new types of knowledge, the prediction accuracy of Golem will gradually improve, making it a more useful biological tool.

The advantages Golem enjoys in the protein prediction problem should apply equally well to other problems in chemistry and molecular biology. This is because chemicals are structural

objects and it is most natural to reason and learn about them using relational knowledge. Using the same methodology as described in this paper, we have successfully applied Golem to the problem of forming a Qualitative Structure Activity Relationship in drug design (King *et al.*, 1992). In drug design the foreground data are the activities of the particular drugs, and the background data are the chemical structures of the drugs and the physical/chemical properties of the substituent groups. A further possible application area is in the human genome project which is producing a vast amount of sequential DNA data and has associated with this data a number of important learning problems, e.g. the recognition of promoter sequences, the recognition of translation initiation sequences, etc. Such problems have been investigated using neural network methods (Stromo *et al.*, 1982) and it would be instructive to investigate how well Golem performs in comparison. Golem could also be applied to other problems in chemistry. Some important early machine learning work was done learning the rules for the break-up of molecules in mass spectroscpoy (Meta-DENDRAL: Buchanan and Feigenbaum, 1981); such a problem would be well suited for Golem.

Availability

A Prolog program that implements the Golem prediction rules is available on request. The ILP learning program, Golem, is also available free to academic users.

Acknowledgements

Part of this research was supported by the Esprit Basic Research Action 'ECOLES'. S.M. was supported by a SERC Research Fellowship. R.D.K was supported by the Esprit project 'StatLog'. M.S. is employed by the Imperial Cancer Research Fund This research was carried out at the Turing Institute.

References

Bain,M. (1981) In Birnbaum,L.A and Collins,G. (eds), *Machine Learning: Proceedings of the Eighth International Workshop.* Morgan Kaufmann, San Mateo, pp. 380–384.

Bernstein,F.C., Koetzle,T.F., Williams,G.J B., Meyer,E.F ,Jr, Brice,M.D., Rodgers,J.R., Kennard,O , Shimanouchi,T. and Tasumi,M. (1977) *J. Mol. Biol.*, **112**, 535–542

Bohr,H., Bohr,J., Brunak,S., Cotteril,R.M J., Lautrop,B., Nørkov,L., Olsen,O.H. and Peterson,S.B. (1988) *FEBS Lett.*, **241**, 223–228

Bohr,H., Bohr,J., Brunak,S , Cotteril,R.M.J , Lautrop,B. and Peterson,S.B. (1990) *FEBS Lett.*, **261**, 43–46.

Bratko,I., Muggleton,S. and Varsek,A (1991) In Birnbaum,L A. and Collins,G. (eds), *Machine Learning: Proceedings of the Eighth International Workshop.* Morgan Kaufmann, San Mateo, pp. 385–388.

Buchanan,B.G. and Feigenbaum,E A. (1981) In Webster,B.L. and Nilson,N.J. (eds), *Readings in Artificial Intelligence.* Tioga Publishing Co., Palo Alto, pp 313–322.

Carter,D.C , Melis,K.A., O'Donnel,S.E., Burgess,B.K., Furey,W F., Wang,B.C. and Stout,C.D. (1985) *J. Mol. Biol.*, **184**, 279–295.

Dijkstra,B W., Kalk,K.H., Hol,W.G.J. and Drenth,J. (1981) *J. Mol. Biol.*, **147**, 97–123.

Dolsak,B. and Muggleton,S. (1991) In Muggleton,S. (ed.), *Inductive Logic Programming.* Academic Press, London, pp. 455–473.

Eisenberg,D. (1984) *Annu. Rev. Biochem.*, **53**, 595–623.

Feng,C. (1991) In Birnbaum,L.A. and Collins,G. (eds), *Machine Learning: Proceedings of the Eighth International Workshop.* Morgan Kaufmann, San Mateo, pp. 403–406

Gibrat,J.E., Garnier,J. and Robson,B. (1987) *J. Mol. Biol.*, **189**, 425–443.

Hendrickson,W.A. and Teeter,M.M. (1981) *Nature*, **290**, 103–107.

Holley,L.H. and Karplus,M. (1989) *Proc. Natl Acad. Sci. USA*, **86**, 152–156.

Kabsch,W. and Sander,C (1983) *Biopolymers*, **22**, 2577–2637.

Kamphuis,I.G., Kalk,K.H., Swarte,M.B.A. and Drenth,J. (1984) *J. Mol. Biol.*, **179**, 233–256

King,R.D. and Sternberg,M.J.E. (1990) *J. Mol. Biol.*, **216**, 441–457.

King,R.D., Muggleton,S., Lewis,R.A and Sternberg,M J.E. (1992) *Proc. Natl Acad. Sci. USA*, in press.

Kneller,D.G., Cohen,F.E. and Langridge,R. (1990) *J. Mol. Biol.*, **214**, 171–182.

Lederer,F., Glatigny,A., Bethge,P.H., Bellamy,H.D. and Mathews,F.S. (1981) *J. Mol. Biol.*, **148**, 427–448.

Lim,V.I (1974) *J. Mol. Biol.*, **80**, 857–872.

Matsuura,Y , Takano,T. and Dickerson,R.E. (1982) *J. Mol. Biol.*, **156**, 389–409.

McGregor,M.J., Flores,T.P. and Sternberg,M.J.E. (1989) *Protein Engng*, **2**, 521–526.

McGregor,M.J., Flores,T P. and Sternberg,M.J.E. (1990) *Protein Engng*, **3**, 459–460.

Muggleton,S (1991) *New Gen. Computing*, **8**, 295–318.

Muggleton,S. and Feng,C. (1990) In Arikawa,S., Goto,S , Ohsuga,S. and Yokomori,T (eds), *Proceedings of the First Conference on Algorithmic Learning Theory.* Japanese Society for Artificial Intelligence, Tokyo, pp. 368–381

Murzin,A.G. and Finkelstein,A.V. (1988) *J. Mol. Biol.*, **204**, 749–769.

Ochi,H., Hata,Y., Tanaka,N., Kakudo,M., Sakurai,T., Aihara,S. and Morita,Y. (1983) *J. Mol. Biol.*, **166**, 407–418.

Presta,L.G. and Rose,G.D (1988) *Science*, **240**, 1632–1641.

Qian,N. and Sejnowski,T.J. (1988) *J. Mol. Biol.*, **119**, 537–555.

Remington,S., Wiegand,G. and Huber,R. (1982) *J. Mol. Biol.*, **158**, 111–152.

Richardson,J.S. and Richardson,D C. (1988) *Science*, **240**, 1648–1652.

Rooman,M.J and Wodak,S.J. (1988) *Nature*, **335**, 45–49.

Schulz,G.E. and Schirmer,R.H. (1978) *Principles of Protein Structure.* Springer-Verlag, **7**, 377–390.

Schulze-Kremer,S and King,R.D. (1992) *Protein Engng*, **5**, 377–390

Scouloudi,H and Baker,E.N. (1978) *J. Mol. Biol.*, **126**, 637–660.

Seshu,R , Rendell,L. and Tcheng,D. (1988) In Segne,A.M. (ed.), *Proceedings of the Sixth International Workshop in Charge of Representation and Inductive Bias.* Cornell, NY, pp. 293–305.

Sheridan,R.P., Dixon,S., Venkataghaven,R., Kuntz,I.D. and Scott,K.P. (1985) *Biopolymers*, **24**, 1995–2023.

Steigemann,W. and Weber,E. (1979) *J. Mol. Biol.*, **127**, 309–338.

Stekemp,R.E., Sieker,L.C. and Jensen,L.H. (1983) *Acta Cryst., B*, **39**, 697.

Stromo,G.D., Schneider,L.M., Gold,L.M. and Ehrenfeucht,A. (1982) *Nucleic Acids Res.*, **10**, 2997–3010.

Taylor,W.R (1986) *J. Theor. Biol.*, **119**, 205–221.

Timkovich,R. and Dickerson,R.E (1976) *J. Biol. Chem.*, **251**, 4033–4046.

Received on February 21, 1992; revised on August 3, 1992; accepted on August 20, 1992