19

PROMIS: Experiments in Machine Learning and Protein Folding

R. D. King†
The Turing Institute
Glasgow, UK

Abstract

The aim of these experiments is to test the use of machine learning as a tool for forming theories from data. A machine-learning program (PROMIS) was developed to form rules for predicting protein secondary structure from primary structure—an important unsolved problem in molecular biology. PROMIS uses a top-down controlled hill-climbing beam search with the rules for predicting secondary structure being search states. Structured background knowledge is used to transform the search space and control generalization. Six rules were found that are humanly comprehensible and provide a chemically meaningful description of the important factors in formation of secondary structure. These rules predicted protein secondary structure with a Q_3 accuracy of 57 per cent, which is comparable with the most commonly used prediction methods. Variations of the rules were found with different accuracies. These were found to help highlight the important features of the rule. Rules were also found which used threshold logic to match sequences of primary structure. These rules were found to be suitable for predicting turn secondary structure. PROMIS is an example of the application of machine learning to molecular biological databases where there is an increasing demand for some form of automated discovery.

1. INTRODUCTION

Perhaps the most promising and yet most difficult application of machine learning is in the area of scientific discovery: 'the most technically gripping challenge, ... will be how to spread the computer wave from the front end of the scientific process, the telescopes, microscopes, ... spark chambers, and the like, back to recognition and reasoning processes by which the chaos of data is finally consolidated into orderly discovery' (Michie 1982). For scientific discovery, machine learning is viewed as a tool to aid working scientists in forming theories from data. Such tools

†Present address: Brainware, Gustav-Meyer-Allee 25, 1000 Berlin 65, FRG

are needed because it often proves difficult for a scientist to perceive patterns in data, even though strong patterns exist. Difficulty in perceiving patterns may occur for a number of reasons: for example patterns may be obscured because there is a very large amount of data, or because the data may be in a difficult form. This paper describes PROMIS (protein machine induction system), a program designed to aid in the formation and creation of theories about the formation of protein secondary structure from primary structure (King 1987).

1.1. The problem

Proteins are the most complicated chemicals that exist. They are responsible for almost all the important tasks within living systems. In a human it is estimated that there are around 100,000 different types of protein. Typical types of proteins are: enzymes (e.g. DNA polymerase), transport proteins (e.g. haemoglobin), protection proteins (e.g. HIV antibody), and toxins (e.g. cobra venom). The conformation of a protein, its three-dimensional shape, determines its function. Anfinsen *et al.* (1961), in an elegant series of experiments, showed that for a given environment, the conformation of a protein is uniquely coded for in the one-dimensional structure of a gene. It is now possible in molecular biology to create a gene and to have this gene translated into the three-dimensional shape of a protein: yet it is not possible to create new useful proteins by doing this, as the rules governing the conversion of the one-dimensional information into three-dimensional information and the rules relating conformation to function are not understood. PROMIS is concerned with finding rules relating one-dimensional and three-dimensional information—'the protein folding problem'.

A gene forms the conformation of a protein in the following way: the DNA sequence is first translated by use of the genetic code into a sequence of amino acids which is the primary structure of a protein (via a matching RNA sequence), the one-dimensional primary structure then spontaneously folds itself into the final conformation of the protein. Between the primary structure and the conformation there is a level of structure known as the secondary structure (Schulz and Schirmer 1978). In finding rules relating the primary structure and the conformation, it is simplest to split the problem into rules relating primary to secondary structure and rules relating secondary structure to conformation (Cohen *et al.* 1982; Lathrop *et al.* 1987). PROMIS is designed to discover rules that convert primary structure into secondary structure. The difficulty of predicting secondary structure comes from the fact that it is the sequence of primary structure as a whole which determines the secondary structure of any particular position, and so any individual amino acids's contribution cannot be said to be context free. In PROMIS, this

Godian knot is cut by considering long-range interactions to be 'noise' and local regions of secondary structure to be caused by the corresponding local region of primary structure, an assumption of locality.

It is thought probable that much of the knowledge necessary for predicting protein secondary structure already implicitly exists in data bases, hidden by the bulk and difficult nature of the information. There are around 70 proteins of known primary and secondary structure, which together give around 10,000 positions of primary structure where the corresponding secondary is known. The secondary structure information has been acquired by protein crystallography and is very difficult to obtain. The primary structure information is much more easily obtained by genetic sequencing methods. In recent years there has been an explosive growth in genetic sequence information and there are now around 10,000,000 primary structure positions known (Smith 1987). The great imbalance of information is set to get even worse with the prospective sequencing of the human genome (Roberts 1987). A solution to the protein folding problem would allow us to exploit this increase in information fully by removing the bottleneck of crystallography.

1.2. The suitability of the problem

The problem of predicting a protein's secondary structure from its primary structure is increasingly becoming a test bed for applications of machine learning. There are several reasons for this:

1. It is of the highest scientific importance.
2. It is of potentially great practical importance.
3. It is a well-known hard and intractable subject and as such presents a great challenge to machine learning technology.
4. Human and statistical methods have fared poorly in attempting to find regularities in the data and solve the problem.
5. There exists a large and growing amount of symbolic data of relevance to the problem (consisting of example proteins of known primary and secondary structure).
6. There exists relevant background knowledge in a form that can readily be applied in a machine-learning program.
7. The data is available in a machine-readable form.
8. There is a reasonably well accepted measure of success, allowing comparison between different machine learning techniques and also more conventional methods.

1.3. Previous work

There have been three types of traditional approach to the problem of secondary structure prediction: methods based on statistics, methods based on chemical theory, and most recently methods based on homology (exemplars), (Sternberg 1983). The most successful achieve an accuracy of around 60 per cent. Statistical methods examine the data base of known primary and secondary structures to find statistical trends, little domain knowledge is used and the rules produced are not in a form comprehensible by people or related to chemical theory, e.g. Gibrat *et al.* (1987). Chemical theory methods use knowledge of molecular structure to produce prediction rules. These rules are comprehensible but mainly ignore the empirical evidence of the data base of known protein structures, e.g. Lim (1974a, b) and Cohen *et al.* (1983). The homology-based methods use domain knowledge to match unknown sequences with known sequences to make their predictions. These predictions suffer by having no explanation in chemical theory and by not producing any new knowledge; the methods closely resemble exemplar-based learning algorithms, e.g. Levin *et al.* (1986).

Apart from PROMIS, two other machine learning approaches have been applied to protein secondary structure prediction. Qian and Sejnowski (1988) made an extensive study of the application of neural networks to the problem. This work achieved an impressive accuracy of prediction and raised several important points about protein secondary structure prediction, but it had the disadvantages of involving a large number of numerical parameters and treating protein folding as a black box with no explanation in human comprehensible terms (a Hinton diagram means nothing to a molecular biologist). Seshu *et al.* (1988) applied the learning program PLS1 to the problem (this program is similar to ID3). PLS1 failed to achieve results significantly better than the default accuracy. They also applied their program NTC (New Term Constructor) to the problem. NTC consists of a complex suite of programs designed to carry out constructive induction in 'hard' domains (Rendell 1988). NTC achieved reasonable results but did not produce concepts comprehensible to molecular biologists.

2. METHODS

The learning problem is: given the proteins of known primary and secondary structure, find generalized relationships between the existing primary and secondary structure which can be used to predict an unknown secondary structure from a known primary structure. Inductive learning is taken to be a heuristic search for a goal through a space of symbolic descriptions generated by application of various rules of inference to the initial observational statements (Mitchell 1982). The

search method used was designed specifically for induction of strings in the presence of noise over a large search space.

2.1. Inputs

The primary and secondary structure of a protein can be considered to be two related strings of characters, where there is a one-to-one mapping between the primary structure and the secondary structure.

[$p1$, $p2$, $p3$, $p4$, $p5$, ..., pn] primary structure [] brackets indicate a sequence.
[$s1$, $s2$, $s3$, $s4$, $s5$, ..., sn] secondary structure

In the alphabet of life there are 20 letters in the primary structure and three letters in the secondary structure. The primary letters are represented as follows (p, g, c, a, s, n, v, t, d, i, l, m, f, y, w, h, k, e, r, q); these are the 20 genetically coded amino acid residues; () brackets indicate a set. The secondary lettes are (A, B, T). These are the three types of secondary structure, alpha-helix, beta-sheet, and turns, respectively.

2.2. Outputs

The concepts induced by the learning program should be good descriptions of the data and useful in prediction. The concepts should also be comprehensible to the domain scientists using the program, that is, they should fit in with current scientific ideas about the domain and perhaps even the scientist's own biases. The concepts should also be simple enough to be represented in a machine learning program.

The output of PROMIS is rules that predict secondary structure from primary structure. The general form of the rules is:

if the string of classes [CC, Dc, Ec] occurs
(i.e. a string of residues occurs [w, x, y]
where the residue w belongs to the class Cc
where the residue x belongs to the class Dc
where the residue y belongs to the class Ec)
then the residues are all in the secondary stucture type S.

For example, using the rule [positive, negative, positive] → A

with the positive class = (h, k, r)
with the negative class = (d, e)

then the primary sequence

[h, d, r] is predicted to have
[A, A, A,] as a corresponding secondary structure by use of the rule.

These rules are similar in form to that used by domain experts in encoding knowledge about proteins.

Learning is carried out in a representation that is different from the input data: that is, not at the residue level (Barr and Feigenbaum 1983, Dietterich and Michalski 1983). Background knowledge is used to group the residues into classes sharing a particular chemical property, or conjunction or disjunction of several properties. For example the residues (*r, k, h*) form the class of 'positive' residues and the residues (*d, e*) form the class of 'negative' residues. There is also the class 'charged' which consists of the residues (*r, k, h, d, e*) and is the conjunction of the classes 'positive' and 'negative'. The reason for grouping residues into classes is to be able to produce rules that can specify more than one primary sequence (that is, they are more general). There are 71 classes used and each residue is a member of 30 classes on average, which means that for a primary sequence of length n there are around 30^n possible class sequences. The classifications used in this work are those of Taylor (1986).

2.3. Background knowledge used as search operators

The class representation of the residues is equivalent to a generalization hierarchy. The graph is a directed acyclic graph and not a tree because any particular node, with the exception of the root node and its children, can have more than one parent; this is because a particular class can be a subclass of several different complex classes; it is a tangled hierarchy. The root node in this example is the class of all residues ('all'). Generalization structures are described by Michalski and Stepp (1983).

The transformation of the set representation into a generalization lattice immediately suggests the method of induction known as climbing a generalization tree (Michalski 1983). This is based on the fact that the ancestor nodes of a set consist of inductive generalizations of that set. Thus, to carry out induction a rule containing a set can be generalized to a rule containing a set that is an ancestor of the original set.

This can be more formally represented thus:

if the rule exists $[Bs] \rightarrow A$,

where *Bs* and *Cs* are sets, *Cs* is an ancestor of *Bs* and *A* is a secondary conformation type, then the following inductive inference can be made:

$[Cs] \rightarrow A$.

A possible example of this is:

from [positive] $\rightarrow A$, infer the generalization [charged] $\rightarrow A$.

Because of the nature of the complex classes some of the climbing tree inductions can be represented as dropping conditions or adding alternatives. An example of dropping a condition is the generalization of [large_and_polar] to [large]. In this example the condition of polarity

has been dropped as the set [large] is an ancestor of [large_and_polar]. An example of adding an alternative is the generalization of [small] to [small_or_polar]. In this example the alternative of polarity has been added as the set [small_or_polar] is an ancestor of [small].

Specialization can easily be carried out with the use of the generalization tree by simply reversing specialization and moving down the tree. A possible example of this is:

from [charged] → A, infer the generalization [positive] → A.

A method of increasing the length of the string is also needed. This can be achieved by adding a new class to either end of the string of classes, a form of specialization specific to string induction. A possible example of this is:

[positive] → A, becomes [positive, negative] → A

The operators used in PROMIS were restricted to: lengthening one end of a rule at a time by adding a new class to the rule's condition and using the generalization tree operators to generalize and specialize on one class of the rule's condition at any one time.

2.4. Rule evaluation

The goal of the search is to find general powerful rules for converting primary structure into secondary structure. To do this a rule evaluation function and a method of assessing statistical significance are needed. The existence of a very large amount of noise in the data (associated with the restrictions inherent in our data representation) means that the best rules should not be expected to be necessarily 100 per cent accurate. It is also expected that no single rule will have 100 per cent coverage.

To find the evaluation of a rule, the sections of primary and secondary sequence are collected where the rule applies in the data base. The sequences of actual secondary structure are then compared with the predicted secondary structure to count how many positions were correctly predicted and how many positions were incorrectly predicted. The evaluation function used is:

$$(P - N)/(P + N + M);$$

where P = the number of correctly predicted positions, N = the number of incorrectly predicted positions and M = the number of positions not predicted. For example: if the primary sequence

[f, g, h, h, g, h]

is found to have the following secondary structure in the data base

[A, A, A, B, B, B]

and it is predicted to have the following secondary structure by a rule

$[A, A, A, A, X, X]$ (X = no prediction made)

then the number of correctly predicted positions is three, the number of incorrectly predicted positions is one, the number of positions not predicted is two and the evaluation of the rule is 0.333. The justification of this evaluation function is that it increases with correct predictions, decreases with incorrect predictions, and is normalized for a given example set; a similar evaluation functions is used in NEWGEM (Mozetic 1986).

It is important that any relationship between primary and secondary structure that is found in the data base should be statistically significant. This is because the rule is to be used to predict unknown secondary structure in the future, and thus must represent a real relationship, not just one that arises through the chance existence of particular primary and secondary structures within the data base. As a heuristic for finding significant rules, a threshold test is used in PROMIS. This involves introducing a threshold number of positions which a rule must cover before it is considered to be significant. For example, if the threshold is set at 100, then the number of correctly predicted and incorrectly predicted positions must be > 100. A similar method is used in RULEGEN from Meta-DENDRAL (Buchanan and Feigenbaum 1981), in SEQUOIA (Haiech *et al.* 1986—SEQUOIA also includes a threshold for incorrect coverage) and in the work of Rooman and Wodak (1988).

2.5. Control of search

Complete search of the rule space is not possible. It is therefore necessary to use some form of heuristic search. The method adopted uses top-down 'generate and test' control, because additional heuristics can be easily applied and because it has good noise immunity (Mitchell 1982). This method has the disadvantage that it involves many passes through the data.

This algorithm (see below) is a form of hill-climbing beam search. It was chosen as it avoids the large memory requirements of best-first search while still avoiding premature commitment to a particular branch of the search tree, see Bisiani (1987). Beam search is used in many induction programs, as, for example, AQ15(Michalski *et al.* 1986) and CN2 (Clark and Niblett 1987).

Algorithm

```
begin
  add an initial beam set of rules
  repeat
    new rules are generated from the beam set by use of the operators,
```

the new rules are evaluated,
store any rule from the beam set that does not produce a better rule,
the beam set becomes the best evaluated new rules,
until
the beam set is empty or a set number of iterations have passed without a rule entering the store with the highest known evaluation,
the best rule found is that with the highest evaluation in the store.
end

The examples of the best rule may then be covered and the process repeated.

In the search PROMIS faces the difficult problem of knowing when to stop and accept a local maximum as the best that can be found, given the limited resources of time and space allowed for the search. Many, if not most practical search problems suffer from lack of a simple test to tell when the goal state has been found. This limits the value of the traditional search formalism and algorithms that exploit it. The method of allowing several iterations to occur between finding the best rule and stopping, is a compromise between best-first and hill-climbing search and allows some local maxima to be avoided.

3. RESULTS

3.1. Data

The example set of proteins used came from the standard Brookhaven data base, via the molecular biology laboratory of Birkbeck College in the University of London. The secondary structure is objectively designated using a modified algorithm from Kabsch and Sander (1983a), which assigns secondary structure on the basis of a known tertiary structure. The proteins used were selected from the data base by M Sternberg, an expert in the subject of protein structure. The Brookhaven data base contains ~ 100 proteins. This was trimmed down to 61 proteins by removing polypeptides and homologous proteins. In addition, only one polypeptide chain was selected from any protein; this was done to make the data as unbiased as possible. The test and training set were split randomly to give a 7:3 division. There are 8024 positions in the training set; 2161 are alpha-helices, 1466 are beta-sheets, and 4397 are turns. There are 3283 positions in the test set, 917 are alpha-helices, 668 are beta sheets and 1698 are turns.

3.2. Experiment 1

PROMIS was used to find general rules for predicting secondary structure from primary structure and six rules were found (King 1988) (see Table 1). All the rules were found starting with the rule,

Table 1. Individual evaluation of rules found. Rules 1a, 2a and 3a are for predicting alpha-helices, rules 1b and 2b are for predicting beta-sheets, rule 1t is for predicting turns. 'Evaluation' is the evaluation function described above in section 2 *Methods*. '% covered' is the amount of secondary structure of the type predicted covered. '% correct' is the accuracy of the prediction for the positions covered. '% correct based on frequency of secondary structure' gives the frequency of the predicted type of secondary structure in the test set. The decrease in coverage and accuracy from the training set to the test set probably represents some overfitting of the rules on the training set.

	On training data			On test data			% correct based on frequency of secondary structure
Rule	Evaluation	% covered	% correct	Evaluation	% covered	% correct	
1a	0.0339	17	79	0.0079	13	56	28
2a	0.0147	12	64	−0.0006	8	49	28
3a	0.0242	12	78	0.0155	7	83	28
1b	0.0067	16	57	0.0049	17	54	20
2b	0.0014	10	52	−0.0006	3	48	20
1t	0.1639	78	62	0.1167	81	58	52

[all] → required secondary structure;

where 'all' matches every type of primary structure and the required secondary structure is either alpha-helix, beta-sheet, or turn.

The beam size was 10, made up as follows: first the highest-scoring rule according to the evaluation function; then three rules selected to be different in at least two places from this rule; finally the next highest-scoring six rules, making 10 in all.

These rules were found to perform comparably with the best published claims for rules produced by domain experts. Exact comparison is difficult because of the imprecise reports of the hand-produced rules, see, for example Cohen *et al.* (1983).

All the rules found show agreement with accepted knowledge about the chemistry and structure of proteins. Yet the only knowledge about proteins that was coded into PROMIS was about residue classes. The higher-level features found in the rules such as the amphilicity in alpha-helix rules and hydrophobic cores in beta-sheet rules were found empirically by PROMIS. As little chemical knowledge was built into PROMIS, discovery of these concepts gives credence to the rules. The lack of knowledge also allows PROMIS not to be bound to existing theory and lets it form new and potentially useful concepts about proteins. Some such concepts have been found and are being further investigated in collaboration with domain experts.

The typical form of the rules is illustrated by the first rule to be found for predicting alpha-helix secondary structure (rule 1a) (see Figure 1 and Table 2).

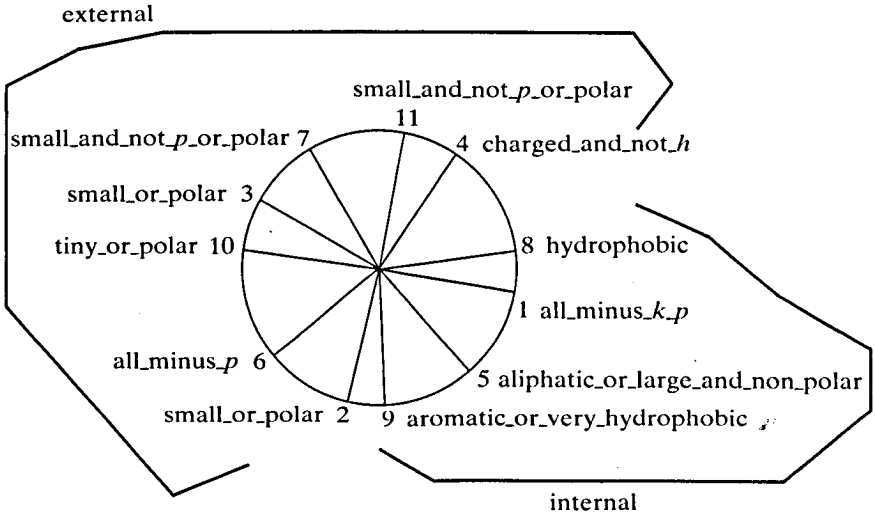

Figure 1. Helical wheel plan of rule 1a. This shows the actual position of the classes in an alpha helix.

Table 2. The occurrences of rule 1a in the test set. The 'protein id' is the short standard identification name of the protein; 'pos' is the sequence position of the first amino-acid residue covered by the rule; 'secondary structure' is the sequence of secondary structure occurring (the rule predicts the sequence to be all 'a' alpha-helix type), 'primary structure' is the sequence of secondary structure from which the rule predicts the primary structure.

protein id	pos	secondary structure	primary_structure
1ABP	19	[a a a a a a a a a a a a]	[t e w k f a d k a g k]
1SBT	38	[t t t t t t t t t b b b]	[s h p d l k v a g g a]
1TIM	1	[t t t t t b b b b b t]	[a p r k f f v g g n w]
1TIM	50	[a a a a t t t t t b b b]	[a r q k l d a k i g v]
1TIM	54	[t t t t b b b b b b t]	[l d a k i g v a a q n]
1TIM	108	[a a a a a a a a a a a t]	[i g q k v a h a l a e]
1TIM	179	[a a a a a a a a a a a a]	[q a q e v h e k l r g]
1TIM	183	[a a a a a a a a a a a a]	[v h e k l r g w l k t]
1TIM	201	[a a t b b b b t t t t]	[v q s r i i y g g s v]
1TIM	235	[t t t a a a a a a t t]	[l k p e f v d i i n a]
2ADK	24	[a a a a a a a a a t t t]	[q c e k i v q k y g y]
2ADK	101	[a a a a a a a a a t t t]	[q g e e f e r k i g q]
2ADK	146	[a a a a a a a a a a a a]	[i k k r l e t y y k a]
2CNA	57	[t t t b b b b b b b t]	[v d k r l s a v v s y]
2MDH	299	[a a a a a a a a a a a a]	[s r e k m n e t a k e]
351C	26	[a a a a a a a a a a a t]	[a y k d v a a k f a g]
3DFR	34	[t t t t b b b b b a a]	[t v g k i m v v g r r]
3DFR	75	[b b t t a a a a a a a]	[v v h d v a a v f a y]
4FXN	11	[a a a a a a a a a a a a]	[n t e k m a e l i a k]
8PAP	153	[t t t t t t b b b b b]	[c g n k v d h a v a a]

[all_minus_*k_p*, small_or_polar, small_or_polar,
charge_and _not_*h*, aliphatic_or_large_and_non_polar,
all_minus_*p*, small_and_not_*p*_or_polar, hydrophobic,
aromatic_or_very_hydrophobic, tiny_or_polar,
small_and_not_*p*_or_polar] → Alpha-helix

In Figure 1 the most important thing to note is the amphiphilic nature of the rule, that is, the hydrophilic and hydrophobic residues are positioned on separate sides of the helix, corresponding to the faces of the helix which face externally (hydrophobic) and internally (hydrophilic); this separation is thought to be a major explanation for helix formation. The hydrophobic residues are arranged in the classic sequence $n(5)$, $n+3$ (8), $n+4$ (9); the same is true for the hydrophilic

residues $n(7)$, $n+3$ (10), $n+4$ (11) and $n-4$ (3), $n-3$ (4), $n(7)$, (Schiffer and Edmundson 1967).

The six prediction rules found are put together to produce a complete method for predicting protein secondary structure from primary structure. This method takes as an input a sequence of primary structure and produces as an output the corresponding secondary structure. Such a prediction method is directly comparable with any other secondary structure prediction method. The six rules combined to produce a Q_3 value of 63 per cent in the training data and 57 per cent in the test data. This accuracy is comparable with the most commonly used prediction methods such as that of Chou-Fasman, Robson, and Lim (Kabsch and Sander 1983b). When combined with protein domain-type specific rules obtained in the same machine learning study (King 1988), they produced a Q_3 value of 67 per cent for secondary structure prediction in the training data, and 60 per cent in the test set. This accuracy is comparable with the best available other methods for protein prediction.

3.3. Experiment 2 (variations on a theme)

In experiments with an alternative rule evaluation method, a threshold percentage accuracy *Ta* is set. If a rule meets this accuracy the number of positive examples is then maximized while still maintaining the heuristic for rule significance. The starting place of the search in these experiments has been the rules generated in Experiment 1.

The aim of these experiments is to find variations of successful rules which have different accuracies. This, it is hoped will highlight the important constant features of a rule across the range of accuracies. For example: if a successful rule is found with an accuracy of 65 per cent, then variations of the rule might be sought with an accuracy of >80 per cent; the resulting rule is likely to have fewer examples, but it will show what features are important in making the rule more accurate; conversely if a lower accuracy is set then the most general features of the rule will tend to show through and possibly make the rule more comprehensible to a domain expert.

The results of searching for rules of varied accuracy are illustrated by the case of rule 1a, the first rule found for predicting alpha helices (Figure 2). The highest accuracy rule was found by the specialization of only three classes. The rule selected for with an accuracy of >80 per cent was found by extending the length of the rule (a form of specialization). The class added was 'hydrophobic_or_small' which fits in with the amphiphilicity of the rule. Lower accuracy rules were found by generalizing the hydrophobic classes and extending the rule with the class 'all'. Interestingly the position of the new classes 'all' is such that they lie between the external and internal faces of the helix. There is an

Selection	Highest	>80%	Best	>70%	>60%	>50%	>40%
Coverage	10%	16%	17%	22%	26%	33%	69%
1	←	←	all_minus_k_p	→	→	→	*all*
2	←	*small_or_polar_ and_not_ aromatic*	small_or_polar	→	→	→	*all*
3	←	←	small_or_polar	→	→	→	→
4	←	←	charged_and_ not_h	→	→	*charged*	*all_minus_p*
5	←	←	aliphatic_or_ large_and_non_ polar	→	*aromatic_or_ very_ hydrophobic*	*hydrophobic*	*aromatic_or_ very_ hydrophobic*
6	←	←	all_minus_p	→	→	→	→
7	*small_and_not_ p_or_ hydrophilic*	←	small_and_not_ p_or_polar	→	→	→	→
8	←	←	hydrophobic	*hydrophobic_ or_small_and_ not_p*	→	*hydrophobic_ or_small_and_ not_p*	→
9	*aromatic_or_ aliphatic_or_m*	←	aromatic_or_ very_ hydrophobic	→	→	→	→
10	*tiny_or_polar_ and_not_ aromatic*	←	tiny_or_polar	→	→	→	→
11	←	←	small_and_not_ p_or_polar	→	→	→	→
+1		*hyrophobic_or_ small*		*all*	*all_minus_p*		
+2				*all_minus_p*			

Figure 2. Variations of rule 1a selected for at different 'Selection' accuracies. The 'Coverage' is the amount of coverage obtained at this accuracy. The 'highest' accuracy is the highest possible accuracy that still allowed the threshold number of examples to be found. The 'best' accuracy is rule 1a. The order of classes goes from top to bottom. An arrow through a box means that it contains the same class as 'best' for that position, a cross means that no class exists.

inversely proportional relationship between correctness of prediction and coverage; the higher the accuracy the lower the coverage (Table 3). Both coverage and correctness decrease in the test set although coverage decreases less. The only rule that does not vary much between training and test is the lowest accuracy rule. This rule manages to cover 60 per cent of all alpha-helix residues with an accuracy that is too low for direct use in prediction but still much higher than the percentage of alpha-helix in the data base (28 per cent).

Table 3. The accuracy, coverage, and evaluation of variations of rule 1a selected for at different accuracies with a constant threshold; 'high' is the highest possible accuracy found for the threshold number of examples, 'best' is the original rule 1a.

	On training data			On test data		
Rule	Evaluation	% covered	% correct	Evaluation	% covered	% correct
high	0.0248	10	89	0.0057	9	70
>80	0.0332	16	81	0.0027	12	56
best	0.0339	17	79	0.0032	13	56
>70	0.0338	22	70	0.0021	14	47
>60	0.0237	26	60	0.0027	18	47
>50	0.0006	33	50	0.0142	23	39
>40	−0.0884	69	40	−0.0376	60	39

3.4. Experiment 3

This experiment is based on using a form of threshold logic in the representation of rules. The idea is that a rule can be said to match a primary structure even if all the class positions of the rule do not match the primary structure exactly. An example makes this clearer:
The rule

[positive, negative, positive] → A
positive = (h, k, r)
negative = (d, e)

makes a mistake in matching the primary sequence

[h, t, r]

(the mistake being that t is not a member of the set 'negative'), therefore the primary sequence [h, t, r] cannot be said to match the rule head [positive, negative, positive]. However, if one mistake in matching was allowed the sequence would be considered an example of the rule.

Allowing one error in matching a rule is equivalent to several rules, each rule having one position as a 'wild card'.

For example, allowing one error in matching the rule

[positive, negative, positive] → A,

is equivalent to the three rules:

[all, negative, positive] → A
[positive, all, positive] → A
[positive, negative, all] → A,

It is simpler to write down one rule and allow one error in matching, than to write down the variations of the rule with the class 'all'; this is especially so if the rule is of greater length.

It is hoped that, by changing the representation in this way, and allowing mistakes in matching, more powerful rules will be found. These should cover more examples of a particular secondary structure while still retaining high accuracy and most importantly human comprehensibility. Mistake matching implies a model of secondary structure where every position in a primary sequence is not vital in forming the secondary structure, and where any one position can be substituted without certain loss of the corresponding secondary structure.

The use of various degrees of matching is a common idea in pattern matching (Slagle and Gini 1987). Mistake matching is also related to the technique for dealing with noise known as 'flexible interpretation' of rules (Michalski *et al.* 1986, Michalski 1987). The idea also receives support from the fact that it is one of the types of representation that has already been used in protein structure prediction (Cohen *et al.* 1983, Cohen *et al.* 1986); however, no attempt has been made to evaluate the representation's suitability for the problem.

The usefulness of normal rules (with complete matching) was compared with rules that make one mistake in matching. For 13 different rule types and appropriate splits of the data, rules were sought for both strict matching and mistake matching and the results compared (King 1988). Mistake matching rules were found to be less successful than normal rules in describing and predicting alpha-helices and beta-sheets, but more successful in describing and predicting turns; the difference between the two types of rule is more marked in the test set than in the training set. The different success rate of mistake matching rules in describing and predicting secondary structure types is probably mainly due to the structural form of the different secondary types. In alpha-helices and beta-sheets, every position is important and the inclusion of an incompatible residue at a position may disrupt the whole protein structure, e.g. a hydrophilic residue within the internal face of an alpha-helix. In a turn, every position is not so vital and residues can be added without disrupting the whole structure of the protein, e.g. it is known that mutations in proteins tend to occur in turns (Thornton 1986). This difference in structural nature between turns and other types of secondary structure means that mistake matching rules are badly suited for describing alpha-helices and beta-sheets but well suited for describing turns.

The two types of rule often resembled each other, suggesting that they were just different ways of describing the same regularity in the data. For example:

The first rule to be found for predicting turns was:

[all, tiny_or_small_and_polar, all, tiny_or_polar_and_not_aromatic_or_*p*] → *T*

with an evaluation of 0.1167 in the test data. The corresponding mistake matching rule was:

[glycine, small_and_polar_or_*p*, all, tiny_or_polar_and_not_aromatic_or_*p*] → *T*

with an evaluation of 0.1648 in the test data.

4. DISCUSSION AND CONCLUSION

Molecular biologists are finding themselves submerged in information about macromolecular structure (von Heijne 1988). The amount of such information is already so great that it is beyond human ability to digest it all; and with the development of faster sequencing machines and the proposed sequencing of the human genome the amount of information is set to increase by several orders of magnitude.

The data are held in very large data bases, the most important of which are GenBank which contains DNA sequence data (Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA), PIR which contains protein primary structure data (National Biomedical; Research Foundation, Georgetown University Medical Center, 3900 Reservoir Rd., NW, Washington DC 20007, USA) and Brookhaven which contains tertiary structure (Brookhaven National Laboratory, Upton, New York 11973, USA). These data bases hold many important secrets, some of which should be capable of being discovered by machine learning. Possible problems in molecular biology to which PROMIS or other machine-learning programs could be applied are recognition of patterns for: antigen binding sites, prediction of RNA secondary/tertiary structure, protein initiation sequences in MRNA, protein coding sites in DNA, gene intron/extron juncture sites, DNA transcription promoters, etc. (Haiech *et al.* 1986).

PROMIS has learned rules that predict secondary structure with an accuracy comparable with other existing prediction methods. In contrast to most other competing methods, the rules are in a form that is humanly comprehensible and they represent theories about the formation of protein secondary structure. The true importance of these rules will only be discovered by a more general airing of the rules to workers in the subject.

It was found that variations in the generality of a rule could highlight its important features to molecular biologists and provide aids to their

understanding. A form of rule representation allowing mistakes in matching the conditional part of the rule was found to be unsuitable for predicting alpha-helices and beta-sheets but suitable for predicting turns.

Protein secondary prediction is a well-known difficult problem. It is therefore unreasonable to expect the problem to be solved easily by the application of any single new method. However, the inductive learning approach seems to be capable of making a useful contribution to solving the problem.

Acknowledgements

I would like to thank my supervisors Peter Mowforth and Profesor McGregor, along with my domain expert Mike Sternberg. I would also like to thank Professor Michie and Pete Clark for their advice. This work was supported by a grant from the Department of Computer Science at Strathclyde University and a grant from the SERC.

REFERENCES

Anfinsen, C. B., Harber, E., Sela, M., and White, F. H. (1961). The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences (U.S.)* **47**, 1309–14.

Barr, A. and Feigenbaum, E. A. (eds) (1983). *The Handbook of Artificial Intelligence.* Pitman, London.

Bisiani, R. (1987). Beam Search. In *Encyclopaedia of artificial intelligence* (eds S. C. Shapiro, and D. Eckroth) pp. 56–57. Wiley Interscience.

Buchanan, B. G. and Feigenbaum, W. A. (1981). Dendral and Meta-Dendral: their application dimension. In *Readings in Artificial Intelligence* (eds B. L. Webster and N. J. Nilsson) pp. 313–22. Tioga, Palo Alto, Ca.

Clark, P. and Niblett, T. (1987). Induction in noisy domains. In *Progress in machine learning* (eds I. Bratko and N. L. Lavrac) pp. 11–30. Sigma Press, Wimslow, England.

Cohen, F. E., Sternberg, M. S. E., and Taylor, W. R. (1982). Analysis and prediction of the packing of alpha-helices against a beta-sheet in the tertiary structure of globular proteins. *J. Mol. Biol.*, **156**, pp. 821–62.

Cohen, F. E., Abarbanel, R. M., Kuntz, I. D., and Fletterick, R. J. (1983). Secondary structure assignment for alpha/beta proteins by a combinatorial approach. *Biochemistry*, **22**, pp. 4894–905.

Cohen, F. E., Abarbanel, R. M., Kuntz, I. D., and Fletterick, R. J. (1986). Turn prediction in proteins using a pattern-matching approach. *Biochemistry*, **25**, pp. 266–75.

Dietterich, T. G., and Michalski, R. S. (1983). A comparative review of selected methods for learning from examples. In *Machine learning: an artificial intelligence approach* (eds R. S. Michalski, J. Carbonell, J. G. and T. Mitchell) pp. 41–81. Tioga, Palo Alto, Ca.

Gibrat, J. F., Garnier, J., and Robson, B. (1987). Further development of protein secondary structure prediction using information theory: new parameters and consideration of residue pairs. *J. Mol. Biol.* **198**, pp. 425–43.

Haiech, J., Quinqueton, J., and Sallantin, J. (1986). SEQUOIA: Concept formation from sequential data. *Proc. EWSL-86*, Paris.

Kabsch, W. and Sander, C. (1983a). Dictionary of protein structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, pp. 2577–637.

Kabsch, W. and Sander, C. (1983b). How good are predictions of protein secondary structure? *F.E.B.S. Letters*, **155**, pp. 179–82.

King, R. D. (1987). An inductive learning approach to the problem of predicting a protein's secondary structure from its amino acid sequence. In *Progress in machine learning* (eds I. Bratko and N. L. Lavrac) pp. 230–50. Sigma Press, Wimslow, England.

King, R. D. (1988). A machine learning approach to the problem of predicting a protein's secondary structure from its primary structure. Ph.D. Thesis, University of Strathclyde, U.K.

Lathrop, R. H., Webster, T. A., and Smith, T. F. (1987). ARIADNE: Pattern-directed inference and hierarchical abstraction in protein structure recognition. *Communications of the ACM* **30**, pp. 909–21.

Levin, S. M., Robson, B., and Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *F.E.B.S.* **205**, pp. 303–8.

Lim, V. I. (1974a). Structural principles of the globular organization of protein chains: a stereochemical theory of globular protein secondary structure. *J. Mol. Biol.* **80**, pp. 857–72.

Lim, V. I. (1974b). Algorithm for prediction of alpha-helical and beta-structural regions in globular proteins. *J. Mol. Biol.* **80**, pp. 873–94.

Michalski, R. S. (1983). A theory and methodology of inductive learning. In *Machine learning: an artificial intelligence approach* (eds R. S. Michalski, J. Carbonell, J. G. and T. Mitchell) pp. 83–134. Tioga, Palo Alto, Ca.

Michalski, R. S. (1987). How to learn imprecise concepts: A method for employing a two tiered knowledge representation in learning. *Proc. Fourth International Workshop on Machine Learning*, pp. 50–8. Morgan Kaufmann, Los Altos, Ca.

Michalski, R. S., Mozetic, I., Hong, J., and Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proc. A.A.A.I.–5*, pp. 1041–5. Morgan Kaufmann, Los Altos, Ca.

Michalski, R. S., and Stepp, R. E. (1983). Learning from observation: conceptual clustering. In *Machine learning: an artificial intelligence approach* (eds R. S. Michalski, J. Carbonell, J. G. and T. Mitchell) pp. 331–64. Tioga, Palo Alto, Ca.

Michie, D. (1982). *Machine intelligence and related topics.* Gordon and Breach Science Publishers.

Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, **18**, pp. 203–26.

Mozetic, I. (1986). Knowledge extraction through learning from examples. In *Machine learning: a guide to current research* (eds T. Mitchell, J. Carbonell, and R. S. Michalski) pp. 227–31. Kluwer Academic Publishers.

Qian, H. and Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins through neural network models. *J. Mol. Biol.* **202**, pp. 865–84.

Rendell, L. (1988). Learning hard concepts. *Proc. International Workshop in Change of Representation and Inductive Bias–I.* pp. 70–100.

Roberts, L. (1987). New sequencers to take on the genome. *Science*, **238**, pp. 271–3.

Rooman, M. J. and Wodak, S. J. (1988). Identification of predictive sequence motifs limited by protein structure data base size. *Nature*, **335**, pp. 45–9.

Schiffer, M., and Edmundson, A. E. (1967). Use of helical wheels to represent the structures of proteins and to identify segments with helical potential. *Biophysical Journal*, **7**, pp. 121–35.

Schulz, G. E. and Schirmer, R. H. (1978). *Principles of protein structure.* Springer-Verlag.

Seshu, R., Rendell, L., and Tcheng, D. (1988). Managing constructive induction using subcomponent assessment and multiple-objective optimization. *Proc. International Workshop in Change of Representation and Inductive Bias–I.* pp. 293–305.

Slagle, J. and Gini, M. (1987). Pattern Matching. In *Encyclopaedia of artificial intelligence* (eds S. C. Shapiro and D. Eckroth) pp. 716–20. Wiley Interscience.
Smith, L. M. (1987). Automated DNA sequence analysis: guide to biotechnology products and instruments science. *Nature*, **235**, No. 11, G89.
Sternberg, M. S. E. (1983). The analysis and prediction of protein structure. In *Computing in biological science* (eds Geisow and Barrett), Elsevier Biomedical Press.
Taylor, W. R. (1986). The classification of amino acid conservation. *J. Theor. Biol.*, **119**, pp. 205–21.
Thornton, J. (1986). Loop regions in proteins: their structure, prediction, and antigenicity. *Proc. SERC Collaborative Computational Project in Protein Crystallography (CCP4)*, Daresbury.
von Heijne, G. (1988). Getting sense out of sequence data. *Nature* **333**, pp. 605–7.