Constructive Induction and Protein Tertiary Structure Prediction *

Thomas R. Ioerger[1,4] **Larry Rendell**[1,3,4] **Shankar Subramaniam**[1,2,3,4]
[1]Department of Computer Science
[2]Department of Physiology and Biophysics
[3]National Center for Supercomputing Applications
[4]The Beckman Institute, University of Illinois, 405 N. Mathews Ave., Urbana, IL 61801
ioerger@cs.uiuc.edu, rendell@cs.uiuc.edu, shankar@ncsa.uiuc.edu

Abstract

To date, the only methods that have been used successfully to predict protein structures have been based on identifying homologous proteins whose structures are known. However, such methods are limited by the fact that some proteins have similar structure but no significant sequence homology. We consider two ways of applying machine learning to facilitate protein structure prediction. We argue that a straightforward approach will not be able to improve the accuracy of classification achieved by clustering by alignment scores alone. In contrast, we present a novel constructive induction approach that learns better representations of amino acid sequences in terms of physical and chemical properties. Our learning method combines knowledge and search to shift the representation of sequences so that semantic similarity is more easily recognized by syntactic matching. Our approach promises not only to find new structural relationships among protein sequences, but also expands our understanding of the roles knowledge can play in learning via experience in this challenging domain.

Introduction

Predicting the tertiary structure of a protein is an important but very difficult problem. Previous machine learning approaches to this problem have been limited because of the complex relationship between the low-level descriptions in terms of amino acid sequences and the high-level similarities among three-dimensional folds. [Ragavan and Rendell, 1993] have shown that, in other difficult domains, *constructive induction* can increase the accuracy and comprehensibility of learning over traditional symbolic, connectionist, and statistical methods. Constructive induction generally makes patterns in data more explicit by finding better representations in terms of intermediate concepts. Representation change can be facilitated and learning improved by the use of knowledge [Rendell and Seshu, 1990; Towell *et al.*, 1990]. We are studying how molecular biologist's knowledge of amino acid properties can be incorporated to improve learning in this domain.

One of the ultimate goals of computational biology is to predict the tertiary structure of a protein from its primary amino acid sequence (for a review, see [Schulz and Schirmer, 1979]). Protein structure prediction is important because the rate at which new sequences are being generated far exceeds the rate at which structures are being experimentally determined. It can take years of laboratory work to crystallize a protein for X-ray crystallography [Richards, 1992], current NMR techniques are limited to solving structures of at most 200 residues (amino acids) [Bax, 1989], and methods based on molecular dynamics are so computationally intensive that simulations are highly unlikely to find conformations with globally minimum energy [McCammon and Harvey, 1987].

To date, the only approach that has been used successfully is to identify a similar sequence, based on degree of homology, whose structure is known [Subramaniam *et al.*, 1992]. Currently, structures for about 500 proteins have been deposited in the Protein Data Bank [Bernstein *et al.*, 1977], falling into classes of about 100 distinct folds [Chothia, 1992]. If a new protein is found to have significant sequence similarity to a protein whose structure is known, then the new protein is assumed to have a similar fold [Schulz and Schirmer, 1979]. This approach, called ***homology modeling***, is distinct from methods for predicting secondary structure [King and Sternberg, 1990; Qian and Sejnowski, 1988; Cohen *et al.*, 1986], which have not been successfully extended to predict three-dimensional conformations.

It has been observed that as many as one third of new sequences appear to be similar to known sequences; such statistics have been used to estimate that the number of folds used in biological systems is only on the order of 1000 [Chothia, 1992]. This redundancy suggests that there is a high degree of structural conservation; of all possible protein folds, only a

*This research was supported in part by an NSF Graduate Fellowship (TRI) and the following grants: NSF ASC-89-02829 (SS), NSF IRI-88-22031 (LR), and ONR N00014-88K0124 (David C. Wilkins, for equipment).

small fraction have been selected, and these folds have been opportunistically adapted for a variety of purposes [Neidhart *et al.*, 1990]. Thus the ability to identify a known sequence (with solved structure) similar to a new sequence will, with significant and increasing frequency, provide structural information for analyzing new sequences [Bowie *et al.*, 1991].

To measure sequence similarity, the common method is to align two sequences and then compute their homology (percent of identical residues) [Needleman and Wunsch, 1970]. The significance of a particular homology value can be tested by comparing it to the distribution of homologies from alignments of random sequences with the same amino acid compositions [McLachlan, 1971]. If the homology between two sequences is high enough, they are probably evolutionarily related and hence are adaptations of the same fold. We call this method the ***alignment-homology*** approach.

An interesting limitation of this approach is related to observation that, while proteins with similar sequences have similar structures, proteins with similar structures often do not have similar sequences. For example, mandelate racemase (MR) and muconate lactonizing enzyme (MLE) have an average structural similarity based on the r.m.s. of C_α distances of only 1.3Å, yet their sequences show only 26% similarity [Neidhart *et al.*, 1990]. Thus a new sequence can have the same fold as a known sequence, but the similarity is not detected by the alignment-homology method.

Incorporating Machine Learning

In this section, we examine the potential for machine learning to relieve the aforementioned limitation of the alignment-homology approach to the protein structure prediction problem. One common machine learning technique is to induce a classification scheme (or "classifier") from a set of pre-classified examples that will classify unseen examples with high accuracy (supervised induction; for a review, see [Michalski, 1983]). We will sketch a fairly straightforward feature-based learning method as a thought experiment. Based on an analysis of how the induction algorithm interacts with the alignment algorithm, we will argue that this first learning approach should ***not*** in fact improve protein structure prediction.

In general, it is believed that domain knowledge is needed to improve learning. However, the forms of knowledge and methods for utilizing it differ from domain to domain. We will suggest that molecular biologists have partial knowledge related to the amino-acid-sequence representation of examples. In other domains, knowledge of representation has been exploited by a learning approach called constructive induction. We propose a novel constructive induction technique that uses this knowledge to search for better representations of amino acid sequences that make structural similarities more obvious.

Feature-Based Learning and Sequences

An obvious application of machine learning is to try learning sequence patterns associated with distinct folds. In this scenario, the examples are sequences and the classifications are the fold identities. For example, the structure for MR would be given a distinct class name; the sequences for both MR and MLE would be classified by this class name since they each have such a fold. The learning goal would be to construct a classifier that mapped any new sequence to the correct fold class (or none, if its fold is truly unique) with high accuracy. Success must be measured relative to the predictive accuracy of the alignment-homology method, which already can, by definition, classify all hemoglobins together, all cytochromes together, etc. The real question that emerges is, Will this application of machine learning ***improve*** recognition of structural similarity by sequence homology?

To further explore this proposal, we must consider how a set of sequences could be generalized. While some research has been done on generalizing sequences with rules for pattern completion (e.g. a nondeterministic finite automaton) [Dietterich and Michalski, 1986] or various kinds of grammars [Fu, 1982; Searls and Liebowitz, 1990], most effort in machine learning has focused on techniques for feature-based learning [Rendell, 1986]. The general assumption behind feature-based learning is that training and testing examples can be described by giving a vector (with a fixed number of dimensions) of values. To illustrate, a feature-based description of a protein might be constructed from some of its biochemical properties, such as molecular weight, isoelectric point, solubilities in specific salt solutions, etc. Given this general assumption for feature-based learning, a great number of algorithms and their properties are known for inducing generalizations.

To construct primitive feature-based descriptions of protein sequences, one might suggest treating position 1 of a sequence as feature 1, position 2 as feature 2, etc., with the feature values ranging over the 20 residue names. However, it would seem to be a problem that proteins vary in length, since the number of features must be the same for all examples. In fact, it is clear that insertions and deletions will cause positions that should correspond (based on structural comparison) to shift relative to one another other. These observations suggest that, for generalizing a set of sequences, we should construct a multiple alignment [Needleman and Wunsch, 1970; Dayhoff, 1972], perhaps allowing GAP as a new residue value. After a multiple alignment has been constructed, the features are well-defined by positions in the multiple alignment.

Once features are defined by a multiple alignment, the potential for using feature-based learning to generalize sequences becomes apparent. A fold (or "concept") can be represented by the set of residues observed at each position (conjunctive normal form

[Michalski, 1983]). Furthermore, generalization of residues at a position can be restricted to certain subsets based on our knowledge of likely amino acid replacements, such as 'hydrophobic' or 'bulky' (internal disjunction via tree-structured attributes). Some machine learning techniques can even identify correlations of values at multiple positions (for example by feature-construction [Matheus, 1989; Rendell and Seshu, 1990]). The basis for these generalizations is taken from, and can extend, the molecular biologist's idea of a *consensus sequence* [Dayhoff, 1972].[1]

However, we will now argue that this straightforward application of machine learning will unfortunately not improve the ability to recognize structural similarity. Consider how to use such a concept representation to classify new examples. To see if the feature values of a new sequence match the consensus pattern, we would have to construct the feature-based description of the sequence. But recall that the new sequence probably has a different length than the consensus. We could attempt to assign features values for the new sequence by finding the best possible alignment of it with any of the sequences in the multiple alignment. Cases where there is a good alignment to some known sequence are uninteresting, since by definition the alignment-homology method would have detected this similarity and made the same classification.

In the cases of insignificant homology, however, there can be no confidence in the alignment. If there is no statistical evidence that the best alignment has higher homology than would an alignment of random sequences, then many alternative alignments might have the same homology [McLachlan, 1971]. *So either the new sequence is obviously in the fold, or we cannot construct its feature-based description for analysis by comparison to the consensus.* For similar reasons, it is even difficult to see how to generalize convergent sequences within a fold. For example, it is not clear how to construct a multiple alignment of the various TIM-barrel proteins because there are no reliable pairwise alignments [Chothia, 1988].

Even if we could use machine learning in some variation of the above proposal to learn sequence patterns for folds, this would not facilitate fold recognition in general, but only for certain folds for which convergent sequences are known. What we really need is a new way to apply machine learning to the protein structure prediction problem so that learning improves the ability to recognize structural similarity via homology more generally. Since we do not want to give up the advantages of using the alignment algorithm, which gives us a good method for analyzing global similarity between sequences of different lengths by finding and averaging local similarities [Needleman and Wunsch, 1970], we must look for some way of applying machine learning to the comparison process itself, rather than to the results of comparison.

Learning by Shift of Representation

The previous argument suggests there is an interaction between alignment and learning which we are not exploiting in the right way. The interaction becomes clear when we observe that the alignment-homology method is a classifier itself. The alignment-homology method learns how to classify sequences from preclassified examples by saving the sequences with their fold classifications (determined directly by NMR or X-ray analysis, or indirectly by significant homology with an already classified sequence). Then, given an unclassified example, the alignment-homology method compares it to all the saved examples and returns the classification of the example sequence to which it is most similar (provided the homology is significant). Clearly, the alignment-homology method itself is performing nearest-neighbor learning [Duda and Hart, 1973].

With respect to improving the performance of a nearest-neighbor learning algorithm, it is well known that this algorithm is highly sensitive to the metric used to compare examples [Kibler and Aha, 1987]. For this metric we are using the homology of the alignment, and one of the parameters of the alignment-homology algorithm is the residue distance function [Erickson and Sellers, 1983]. This function returns a real number indicating the degree of mismatch between two aligned residues. In the standard alignment-homology method, the function returns 0 for identical residues and 1 otherwise. However, one variation of the residue distance function that has proved useful for comparing sequences has been the inverse of observed substitution frequencies [McLachlan, 1971; Gribskov *et al.*, 1987].

The rationale behind inverse substitution frequencies as a residue distance function is that it should cause structurally related sequences to appear more similar than truly unrelated sequences [Schulz and Schirmer, 1979]. If two sequences have the same fold, substitutions between them generally must be restricted to chemically or physically similar residues in order to fulfill local roles in determining structure. This biases the observed substitution frequencies because residues that play similar roles exchange more often. By inverting the frequencies, we are counting residues that play similar roles in structure as less distant (because they exchange more frequently), and residues that play different roles as more distant. Sequences from different folds should have a uniform distribution of frequent and infrequent substitutions, canceling the effect of varying mismatch penalties. But

[1] A consensus sequence is constructed from a multiple alignment by indicating the set of most frequent residues occurring at each sequence position. Often the sets are restricted to sets of residues with common properties, such as charge or hydrophobicity. For example, two sequences **...Gly Val Asp Phe...** and **...Gly Ile Glu Glu...** might be represented by the consensus sequence **...Cly hydrophobic negative-charge anything....**

sequences from the same fold should have more of the frequent substitutions, get penalized less, and hence appear more similar overall.

This explanation suggests that we should be looking at sequences in a special way: not as sequences of residue identities, but as sequences of physico-chemical properties. When we see **...Val Tyr Glu...** in a sequence, we actually think "...small hydrophobic residue branched at C_β, aromatic hydroxylated residue, small negatively charged residue that can form H-bonds...." Thus we could achieve the same effect of using a substitution-frequency-based residue distance function with the identity residue distance function by transforming (prior to alignment) the symbols at each sequence position from residue identity into a symbol representing local physico-chemical properties. Furthermore, such transformations could take context into account by including properties of neighboring residues, thus capturing conditional roles. A match would indicate that two residues could play the same role in determining local structure, which is a vast improvement over matches based solely on residue identity.

So we propose that machine learning can be applied to the protein structure prediction problem by learning how to transform amino acid residues to represent local properties involved in determining structure. In the machine learning literature, this approach is generally called *constructive induction* [Michalski, 1983; Rendell and Seshu, 1990]. To improve the performance of a fixed learning algorithm (e.g. the alignment-homology method), constructive induction shifts the representation of examples to provide a more appropriate learning bias [Mitchell, 1980]. Constructive induction is thought to be particularly useful for learning hard concepts in difficult domains by discovering intermediate concepts, which, when added to the representation, make significant patterns in the data more explicit [Rendell and Seshu, 1990]. Constructive induction has been found to significantly improve the accuracy and comprehensibility of learning over standard algorithms, such as decision tree builders (e.g. C4.5), neural nets (e.g. backpropagation), and statistics-based programs (e.g. MARS) [Ragavan and Rendell, 1993].

While constructive induction would seem in principle to be advantageous in the domain of protein structure prediction, current frameworks are not applicable because examples are represented by sequences rather than feature vectors [Matheus, 1989]. In the following sections, we propose a new method of constructive induction that utilizes molecular biologists' knowledge of likely relevant properties of amino acids to search for better representations of sequences, ultimately to make sequence comparisons better reflect structural relationships. Our approach to this important problem is unique in combining traditional statistical analysis with knowledge of amino acid properties, and could potentially discover new structural relationships among protein sequences.

Transformation Functions

As we have proposed above, our learning goal is to find an effective transformation function which will re-represent sequences so that structural similarity is easier to recognize with the alignment-homology method.[2] First we will define transformation functions, and then show how to construct a space of them.

Sequences are constructed in the usual way from finite alphabets and have finite, non-zero lengths: $A \in \Sigma^+$, $A = a_1 a_2 ... a_n$, $a_i \in \Sigma$, where $n \in N$ is the length of A, denoted $|A|$. Σ_{aa} is the usual alphabet for protein sequences, consisting of the 20 amino acid symbols. We will be constructing other alphabets for describing local properties.

In order to transform an entire sequence, we perform a local transformation on each position in the sequence with *local transformation functions*. Then we extend the local transformation function into a *sequence transformation function* by applying it to each position in a sequence.

Definition 1 *A* local transformation function $\mathcal{F}$ *maps a sequence A over one alphabet* Σ_1 *and an index i* $(1 \leq i \leq |A|)$ *to a symbol in another alphabet* Σ_2*:* $\mathcal{F} : {\Sigma_1}^+ \times \mathrm{N} \mapsto \Sigma_2$.

Definition 2 *A* sequence transformation function $\widehat{\mathcal{F}}$ *is an extension of a local transformation function* $\mathcal{F}$ *that maps a sequence A over one alphabet* Σ_1 *to a sequence B of the same length* $(|B| = |A|)$ *over another alphabet* Σ_2 *by applying* $\mathcal{F}$ *to each position in A:* $b_i = \mathcal{F}(A, i)$, $(1 \leq i \leq |A|)$.

The simplest examples of local transformation functions are identities. The function ID_0, when applied to position i of a sequence, simply returns the symbol at that position. Thus $\widehat{ID_0}$ copies one sequence to another: if $B = \widehat{ID_0}(A)$, then $b_i = a_i$ for $1 \leq i \leq |A|$. Other identity functions return neighboring symbols, and their sequence-transformation-function extensions cause shifts. For example, if $B = \widehat{ID_{-1}}(A)$, then $b_i = a_{i-1}$ for $2 \leq i \leq |A|$, and $b_1 = a_1$.[3]

Abstraction

Given this base class of local transformation functions, we can resursively construct more interesting transformations by two processes, the first of which is called

[2] Transformations were also used in [Dietterich and Michalski, 1986] to facilitate sequence learning. Their operator for adding derived attributes subsumes our abstraction operator (section 3.1), and their blocking operator is subsumed by our crossing operator (section 3.2).

[3] There is some freedom in defining boundary conditions; we alternatively might have extended the alphabet of B to contain a symbol for "undefined."

abstraction. Intuitively, to abstract a position in a sequence is to replace the symbol with its class according to some partition of the alphabet.

Definition 3 *An* abstraction function $AB_{\mathcal{P}}$ *maps a sequence A over an alphabet Σ and an index i $(1 \leq i \leq |A|)$ to a symbol in the alphabet $\Sigma/\mathcal{P}$, the class names of the symbols in Σ under the partition $\mathcal{P}$.*

The effect of abstraction is that, when comparing two sequences via the alignment-homology method, some mismatches would be changed to matches because the symbols had been identified together. The most obvious abstraction function is the one that maps the amino acids into their classes heuristically derived from observed substitution frequencies: $AB_{\mathcal{P}_{aa}}$, where $\mathcal{P}_{aa} = \{\{V, I, L, M\}, \{C\}, \{F, Y, W\}, \{K, R, H\}, \{S, T, D, N, G, A, E, Q, P\}\}$ [Dayhoff *et al.*, 1972]. However, abstraction is general and can be used to identify any subset of symbols. For example suppose we partitioned the amino acids into three classes: HYDROPHILIC, HYDROPHOBIC, and AMPHIPATHIC. Then we could single out the property of being HYDROPHOBIC by combining HYDROPHILIC with AMPHIPATHIC via the partition {{HYDROPHILIC,AMPHIPATHIC},{HYDROPHOBIC}}. In terms of constructive induction, the abstraction process disjoins feature values [Michalski, 1983; Rendell and Seshu, 1990].

Crossing

One problem with abstracting residues into classes is that there are multiple dimensions of similarity among amino acids which might get confounded in any single partition. For example, threonine is similar to valine in size and similar to tyrosine because its hydroxyl can participate in hydrogen bonds, but it is not meaningful to identify all three of these residues together. The substitution frequency matrix, although more flexible because of its scalar similarity values, also suffers from such confounding, and must suffice to average out relative similarities based on any and all relevant properties [Dayhoff *et al.*, 1972].

To alleviate this confounding effect, we observe that context often determines the primary roles played by a residue. For example, the important properties of Val in a β-sheet are its hydrophobicity and its branching at C_β (for shielding the polar backbone) but not its smallness [Schulz and Schirmer, 1979]. If we could estimate the local environment, then we could abstract residues conditional on which properties are most likely being exploited. A method for approximating the local environment is to find patterns in neighboring residues and their properties [Schulz and Schirmer, 1979]. Thus we introduce *crossing* as a second process for constructing new transformation functions. Crossing takes symbols from two local transformation functions and forms a new symbol in the product of the two alphabets.

Definition 4 *The* cross $\mathcal{F}_1 \times \mathcal{F}_2$ *of two local transformation functions $\mathcal{F}_1$ (mapping into Σ_1) and $\mathcal{F}_2$ (mapping into Σ_2) maps a sequence (over Σ) and a position index into the cross product of the alphabets of the two functions: $\mathcal{F}_1 \times \mathcal{F}_2 : \Sigma^+ \times \mathrm{N} \mapsto \Sigma_1 \times \Sigma_2$.*

As a *hypothetical* example, suppose that normally hydrophobicity is the most important property, but in turns (say, when glycine is the previous residue) size is most important. We could implement this knowledge in a transformation function like $AB(ID_{-1} \times ID_0)$ that crossed the identity of position $i-1$ with the identity of position i and then abstracted the product symbols in the following way. When the first symbol in the product is glycine, form classes based on the size of the second symbol in the product, and otherwise, form classes based on the hydrophobicity of the second symbol in the product. Thus the symbol $Gly \times Val$ would get mapped into one class, perhaps called NEXT-TO-GLY-AND-SMALL, and $Ser \times Val$ would get mapped into another class, perhaps called NOT-NEXT-TO-GLY-AND-HYDROPHOBIC. In terms of constructive induction, the crossing process conjoins features [Rendell and Seshu, 1990; Michalski, 1983].

Constructing the Space

Through the base cases (identity functions) and recursive cases (abstractions and crossings), a large space of functions can be constructed, similar to extending a set of features by the logical operators $\{\wedge, \vee\}$ [Rendell and Seshu, 1990]. These functions formally capture the ability to compare sequence positions in terms of their local properties. What we hope is that, by finding an appropriate transformation function and applying it to a pair of sequences, the alignment-homology algorithm will be facilitated by a better correspondence between syntactic matching and semantic similarity based on physico-chemical roles played in determining local structure.

Perhaps the ultimate local transformation function would be one that maps sequence positions into secondary structure classes [King and Sternberg, 1990]. If sequences were compared this way, the alignment-homology method would be an extremely good classifier for folds. It is possible that two sequences could have similar secondary sequence patterns and yet fold into distinct global conformations, but this seems highly improbable, especially considering that only on the order of 1000 folds are predicted to be used in biological systems. Since secondary structure is largely determined by properties of residues close in sequence, we expect the space of transformation functions to contain a function that expresses such a representation. Importantly, our approach surpasses secondary structure prediction methods [King and Sternberg, 1990; Qian and Sejnowski, 1988; Cohen *et al.*, 1986] by using such local predictions to recover full three-dimensional conformations.

Searching for Transformations

The constructions mechanize operations known to be important, imparting knowledge of the form of transformations, and relegate the task of search to the computer to instantiate the details of transformations. The space of transformation functions is very large (consider all the possible partitions for abstracting an alphabet of size 20; consider the exponential growth of neighborhood conditions when crossing with more neighbors), so it must be searched intelligently. In order to measure progress we must operationalize our goal into a method for evaluating transformation functions. Since we are looking for a transformation function that improves the ability of the alignment-homology method to recognize structural similarity, the basic test will be to observe the effect that pre-processing sequences with a transformation function has on the predictive accuracy of the alignment-homology method.

The predictive accuracy can be estimated by classifying a training set of sequence *pairs* with insignificant homology, some of which are known to be in the same fold (+ class: SAME-FOLD), and the others known to be in different folds (- class: DIFFERENT-FOLD). Without any transformation, the alignment-homology method would classify all these sequence pairs as DIFFERENT-FOLD, so the goal is to find transformation functions that reduce the false negatives while preserving the true negatives.

If we plot a histogram of homologies from alignments of unrelated sequences (pairs in the training set classified as DIFFERENT-FOLD), we get a distribution with a low mean of roughly 10-20%. If we were to plot a similar histogram for insignificantly homologous sequences classified as SAME-FOLD, we would expect to see a similar distribution since this is the basis for the definition of insignificant homology. The overlap is precisely the reason that syntactic matching (homology using the "null" transformation function $\widehat{ID_0}$) is an inadequate method for recognizing structural similarity: there is poor separation of sequence-pair classifications at low homology. Thus an operational version of our goal is to find a transformation function that separates these two peaks. We can quantitatively evaluate a transformation function (relative to a training set of sequence pairs) by computing the degree of separation $\mathcal{S}$ based on average homologies μ_i, variances σ_i, and sample sizes n_i, where i is the class name: $\mathcal{S} = (\mu_+ - \mu_-)/(\sigma_+{}^2/n_+ + \sigma_-{}^2/n_-)$. This formula captures the notion that the distance between two distibutions depends not only on the distance between the means, but also on how spread out they are.

This evaluation can be used to search the space for effective transformation functions. For example, perhaps a hill-climbing approach would incrementally build increasingly better representations through the crossing and abstracting constructions. Or a genetic algorithm [Booker *et al.*, 1989] that recombines sub-functions of highly rated transformation functions might be effective. Since there are so many possible ways to abstract or cross a particular transformation function, it is clear that some domain knowledge (more than is already built into the transformation operators and alignment-homology algorithm [Matheus, 1989]) will be necessary [Michalski, 1983] [Towell *et al.*, 1990]. Fortunately, the molecular biologist's knowledge of physical and chemical properties that are likely to be involved can be used as suggestions for abstraction functions [Schulz and Schirmer, 1979]. Similarly, the knowledge that the local environment at a seqeunce position is largely determined by up to 5 residues in both directions is useful in restricting the crossing constructs [Schulz and Schirmer, 1979]. By searching the space of transformations near constructions consistent with such knowledge, the evaluation metric can guide us to better transformations, and we might be able to refine our knowledge of the principles determining protein structure by interpreting the search results [Towell *et al.*, 1990]. Our research expands the roles knowledge can play in learning.

In summary, our approach to improving protein structure prediction is essentially to *optimize* the representation of amino acid sequences. In contrast to the nearest-neighbor approach described earlier, this learning method takes pairs of sequences as a training set (rather than single sequences), represents concepts in the form of transformation functions (instead of saved examples), and classifies unseen examples (sequence pairs) as SAME-FOLD or DIFFERENT FOLD (instead of returning actual fold identities of single sequences). The learning is achieved by optimizing a pre-processing transformation of a training set of sequence pairs for predictive accuracy of the alignment-homology classifier. The evaluation is based on the separation of peaks from distributions of alignment scores between sequence pairs of same and of different structure. The shift of representation to fit a fixed learning bias makes this approach a kind of constructive induction, and promises to exploit and augment molecular biologists' knowledge of the physico-chemical roles played by amino acids in determining protein structure.

Preliminary Experimental Results

In this section, we demonstrate the potential for our constructive induction approach to facilitate recognition of structural similarity. Table 1 shows a set of pairs of sequences we used as data. Each pair represents dissimilar sequences with similar folds and will be used as a positive example (SAME-FOLD). Sequences not listed in the same pair are not only different in sequence, but also in structure; such pairs will be used as negative examples (DIFFERENT-FOLD).

To demonstrate that the alignment-homology method would have difficulty classifying this data set, we computed the best possible alignment scores for

Table 1: The data set used for these experiments: pairs of sequences with structural similarity but insignificant sequence homology. The names refer to PDB entries.

name	chain	description
256b 2mhr		cytochrome b562 myohemerythrin
1ald 4xia		aldolase A xylose isomerase
1rhd 1rhd	1-146 152-293	rhodanese rhodanese
1gox 1wsy	 beta	glycolate oxidase tryptophan synthase
1cd4 2fb4	 light	T-cell CD4 immunoglobulin Fab
2hhb 1ecd	alpha	hemoglobin erythrocruorin
2aza 1pcy		azurin plastocyanin
2fbj 1fc1	heavy	IgA Fab IgG1 Fc

each example pair (using the algorithm of [Gotoh, 1982] with a gap start penalty of 3, a gap extension penalty of 0.1, and the identity residue distance function described above). It appeared that the alignment scores were linearly dependent on the minimum length of the sequences being aligned, so this variable was factored out of each score, leaving a number between 0 (no matches) and 1 (identical sequences). The distributions of scores for positive and negative examples are compared in Figure 1 (showing unweighted probability density functions for normal distributions given the means and standard deviations for each set of scores). The distributions are extremely overlapped; the peak separation S is 114.[4]

Interestingly, when amino acid residues are mapped into classes according to substitution frequencies by $AB_{\mathcal{P}_{aa}}$, the ability to distinguish structural similarity does not improve. Figure 2 shows the distributions of alignment scores for SAME-FOLD and DIFFERENT-FOLD sequences pairs. The scores have shifted higher because the abstraction causes more matching. However, the shift was independent of fold similarity; the peaks are still indistinguishable. The separation is −43. Clearly, residue class at a single position is not refined enough to capture secondary structure. The representation is apparently confounding the various roles amino acids can play, hence causing the substitution patterns among sequences with the same structure to appear random.

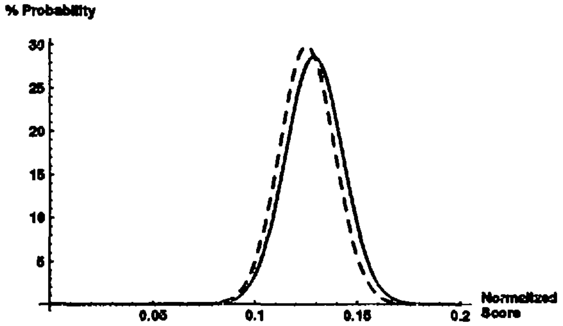

Figure 1: Distributions of alignment scores for sequences pairs (without any transformation) classified as SAME-FOLD (solid line) and DIFFERENT-FOLD (dashed line). The separation of the peaks is 114.

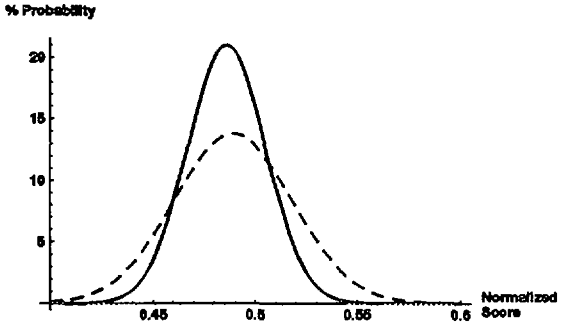

Figure 2: Distributions of alignment scores for sequences pairs when amino acids are transformed into residue classes ($AB_{\mathcal{P}_{aa}}$). The separation of the peaks is -43.

To find a more expressive representation, we crossed the residue class at a position with its neighbors, one each in the N-terminal and C-terminal directions. Simply crossing the residue classes at these three positions produces an alphabet of size 125, since the range of class values is 5 for each position. If such a transformation function were used, most sites would appear dissimilar, including those that should match based on structural comparison. Thus we would like to find a partition of the 125 product symbols that maps together some of the triples which are in fact found in the same secondary structures.

To find such an abstraction function, we used the technique of hill-climbing [Winston, 1984]. First, we randomly partitioned the 125 values into 10 classes. This initial abstraction function, applied on top of the crossing function, did not separate the scores very well (see Figure 3); the initial separation value was -20 (nearly complete overlap, like Figure 2). Then we iteratively perturbed the partition and tested for im-

[4] As defined in the section on searching for transformations, S is an abstract, unitless measure. The separation value of a transformation by itself is meaningless; it is only useful in comparison to the separation of another transformation. Separation can quantitate the observed differences among Figures 1, 2, 3, and 5, and is used for making decisions in our hill-climbing algorithm. Based on our figures, peaks are not noticeably separated until S is on the order of 1000; negative separation indicates that the mean of the positive peak is below the mean of the negative peak.

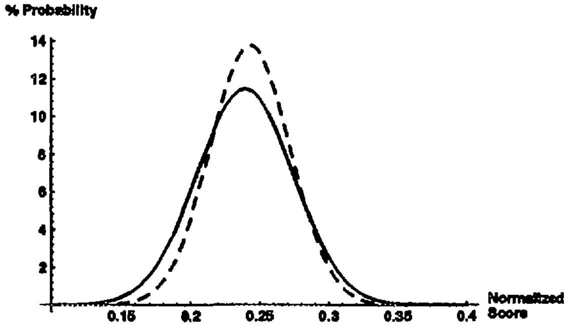

Figure 3: Distributions of alignment scores for sequences pairs when the residue classes at positions $i-1$, i, and $i+1$ are crossed, and then abstracted according the random initial partition in the experiment. The separation of the peaks is -20.

proved separation over the data set. The perturbation was accomplished by randomly choosing one of the 125 values and switching it from its current class to a new one (sometimes naturally emptying a class; 10% of the time creating its own new class). The perturbed partition was evaluated by applying it with the crossing transformation to the sequence pairs and computing the separation of the positive and negative peaks as before. If the separation increased, the perturbed partition was kept for the next iteration.

In our experiment, the initial random partition was perturbed over 500 iterations, and the hill-climbing procedure appeared to be quite successful. Figure 4 shows that the ability to separate the peaks steadily increased. The best partition found had 13 classes with sizes ranging from 2 to 25 product symbols; no interpretable pattern in triplets of residue classes had yet appeared. Figure 5 shows how well this partition separated the peaks; its evaluation was 1338. We suggest that such a transformation function has captured something about local physico-chemical properties that is related to secondary structure. As a consequence, the alignment-homology method has become a better classifier; homologies between sequences that do indeed have similar structure have become incresingly significant and identifiable. Extensions of this experiment should include cross-validation of the results with an independent data set, and might include an analysis of new biophysical principles implied in the learned transformations.

Conclusion

In this paper we considered two ways of applying machine learning to the protein structure prediction problem. We argued that the straightforward approach of applying feature-based learning to generalize sequences in a fold would not be effective since the ability to construct multiple alignments would by itself classify sequences as well. We suggested that machine learning could be more appropriately applied in the form of constructive induction by shifting the representation of amino acids sequences before computing the alignments. We presented a language of transformations that should be able to express local physical and chemical properties that determine protein structure; re-representing sequences with such a function should improve the correspondence between syntactic and semantic similarity. Finding such a function is an immense search task, but offers many possibilities for incorporating and refining molecular biologists' knowledge. The novel learning method we developed for this unique domain will expand current constructive induction frameworks and help us better understand the roles knowledge can play in learning.

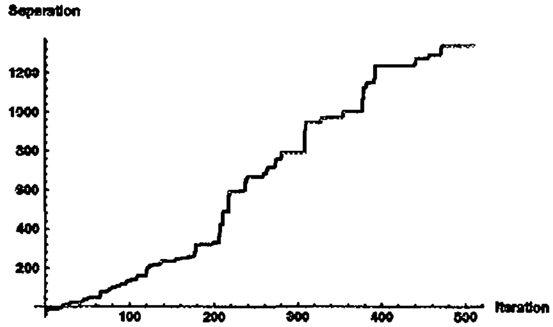

Figure 4: Increase in separation with each perturbation of the partition in the experiment.

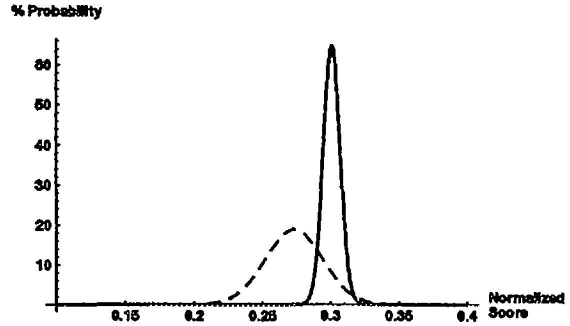

Figure 5: Distributions of alignment scores for sequences pairs when the residue classes at positions $i-1$, i, and $i+1$ are crossed, and then abstracted according the best partition found in the experiment. The separation of the peaks is 1338.

References

Bax, A. 1989. Two-dimensional nmr and protein structure. *Ann. Rev. of Biochemistry.* 58:223–256.

Bernstein, F.; Koetzle, T.; Williams, G.; Meyer, E.; Brice, M.; Rodgers, J.; Kennard, O.; Shimanouchi, T.; and Tasumi, M. 1977. The protein data bank: A computer-based archival file for macromolecular structures. *Journal of Molecular Biology* 112:535–542.

Booker, L.; Goldberg, D.; and Holland, J. 1989. Classifier systems and genetic algorithms. *Artificial Intelligence* 40:235–282.

Bowie, J.; Luthy, R.; and Eisenberg, D. 1991. A method to identify protein sequences that fold into a known 3-dimensional structure. *Science* 253:164–170.

Chothia, C. 1988. The fourteenth barrel rolls out. *Nature* 333:598–599.

Chothia, C. 1992. One thousand families for the molecular biologist. *Nature* 357:543–544.

Cohen, F.; Abarbanel, R.; Kuntz, I.; and Fletterick, R. 1986. Turn prediction in proteins using a complex pattern-matching approach. *Biochemistry* 25:266–275.

Dayhoff, M.; Eck, R.; and Park, C. 1972. A model of evolutionary change in proteins. In Dayhoff, M., editor 1972, *Atlas of Protein Sequence and Structure*, volume 5. Silver Springs, MD: National Biomedical Research Foundation.

Dayhoff, M. 1972. *Atlas of Protein Sequence and Structure*, volume 5. Silver Springs, MD: National Biomedical Research Foundation.

Dietterich, T. and Michalski, R. 1986. Learning to predict sequences. In Michalski, R.; Carbonell, J.; and Mitchell, T., editors 1986, *Machine Learning: An Artificial Intelligence Approach, II.* San Mateo, CA: Morgan Kaufmann Publishers. 63–106.

Duda, R. and Hart, P. 1973. *Pattern Classification and Scene Analysis.* New York: Wiley.

Erickson, B. and Sellers, P. 1983. Recognition of patterns in genetic sequences. In Sankoff, D. and Kruskal, J., editors 1983, *Time Warps, String Edits, and Macromolecules.* Reading, MA: Addison-Wesley. 55–91.

Fu, K. 1982. *Syntactic Pattern Recognition and Applications.* Englewood Cliffs, NJ: Prentice-Hall.

Gotoh, O. 1982. An improved algorithm for matching biological sequences. *Journal of Molecular Biology* 162:705–708.

Gribskov, M.; McLachlan, A.; and Eisenberg, D. 1987. Profile analysis: Detection of distantly related proteins. *Proceedings of the National Academy of Sciences USA* 84:4355–4358.

Kibler, D. and Aha, D. 1987. Learning representative exemplars as concepts: An initial case study. In *Proceedings of the Fourth International Workshop on Machine Learning.* San Mateo, CA: Morgan Kaufmann Publishers. 24–30.

King, R. and Sternberg, M. 1990. Machine learning approach for the prediction of protein secondary structure. *Journal of Molecular Biology* 216:441–457.

Matheus, C. 1989. *Feature Construction: An Analytic Framework and an Application to Decsion Trees.* Ph.D. Dissertation, University of Illinois, Department of Computer Science.

McCammon, J. and Harvey, S. 1987. *Dynamics of Proteins and Nucleic Acids.* New York: Cambridge University Press.

McLachlan, A. 1971. Test for comparing related amino acid sequences. *Journal of Molecular Biology* 61:409–424.

Michalski, R. 1983. A theory and methodology of inductive learning. *Artifical Intelligence* 20:111–161.

Mitchell, T. 1980. *The Need for Biases in Learning Generalizations.* Technical Report CBM-TR-117.

Needleman, S. and Wunsch, C. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48:443–453.

Neidhart, D.; Kenyon, J.; Gerlt, J.; and Petsko, G. 1990. Mandelate racemase and muconate lactonizing enzyme are mechanicaly distinct and structurally homologous. *Nature* 347:692.

Qian, N. and Sejnowski, T. 1988. Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology* 202:865.

Ragavan, H. and Rendell, L. 1993. Lookahead feature construction for learning hard concepts. In *Proceedings of the Tenth International Machine Learning Conference.* to appear.

Rendell, L. and Seshu, R. 1990. Learning hard concepts through constructive induction: Framework and rationale. *Computational Intelligence* 6:247–270.

Rendell, L. 1986. A general framework for induction and a study of selective induction. *Machine Learning* 1:177–226.

Richards, F. 1992. Folded and unfolded proteins: An introduction. In Creighton, T., editor 1992, *Protein Folding.* New York: Freeman. 1–58.

Schulz, G. and Schirmer, R. 1979. *Principles of Protein Structure.* New York: Springer-Verlag.

Searls, D. and Liebowitz, S. 1990. Logic grammars as a vehicle for syntactic pattern recognition. In *Proceedings of the Workshop on Syntactic and Structural Pattern Recognition.* International Association for Pattern Recognition. 402–422.

Subramaniam, S.; Tcheng, D.; Hu, K.; Ragavan, H.; and Rendell, L. 1992. Knowledge engineering for protein structure and motifs: Design of a prototype system. In *Proceedings of the Fourth international Conference on Software Engineering and Knowledge Engineering.* Washington, DC: IEEE Computer Society. 420–433.

Towell, G.; Shavlik, J.; and Noordewier, M. 1990. Refinement of approximate domain theories by knowledge-based neural networks. In *Proc. Eighth Natl. Conf. on Artificial Intelligence.* 861–866.

Winston, P. 1984. *Artifical Intelligence.* Reading, MA: Addison-Wesley.