CHAPTER

5

Developing Hierarchical Representations for Protein Structures: An Incremental Approach

Xiru Zhang & David Waltz

1 Introduction

The protein folding problem has been attacked from many directions. One set of approaches tries to find out correlations between short subsequences of proteins and the structures they form, using empirical information from crystallographic databases. AI research has repeatedly demonstrated the importance of representation in making these kinds of inferences. In this chapter, we describe an attempt to find a good representation for protein substructure. Our goal is to represent protein structures in such a way that they can, on the hand, reflect the enormous complexity and variety of different protein structures, and yet on the other hand facilitate the identification of similar substructures across different proteins. Our method for identifying a good representation for protein structure is embodied in a program called GENEREP[1], which automatically generates hi-

erarchical structural representations for a protein of known structure.

Our approach proceeded in three stages. First, we selected a set of objectively definable primitives that captured all local information in as compact a form as possible. This step was accomplished by an unusual variation on principal component analysis. Using these primitives to represent proteins of known structure, we then looked for commonly co-occurring collections of primitives, which we used to define substructure families by an analog of k-means classification. Finally, we looked at how these families of structures are combined in sequence along a protein chain by heuristically inferring finite state automata that use the structure families to recognize proteins of known structure.

We hope this paper can serve both the AI and molecular biology communities. We believe the techniques described here are generally useful in designing representations for complex, ordered data in general, such as speech processing or economic predictions. We also present the derived representation as a useful tool for analysis of protein structures in biological domains. Our representation captures much of the important information about a protein conformation in a very compact form, which is more amenable to analysis than many of the alternatives.

2 Why Worry About Representation of Protein Structures?

2.1 The Issue of Representation in AI

The importance of representation in problem solving has long been emphasized in AI; see, for example, [Brachman & Levesque, 1985]. Researchers in the recent resurgence of connectionism have also started to realize its importance [e.g. Tesauro & Sejnowski, 1989]. A general lesson from AI is that good representations should make the right things explicit and expose natural constraints. In most of the traditional AI work, representations were designed by users and hand-coded; see [Charniak & McDermott, 1985] and [Winston, 1984] for summary of such work. Recently, with the development of connectionism, it has been shown that interesting representations can also be computed "on the fly'' from the input data. For example, [Hinton, 1988] developed internal representations for family relationships by training an auto-association networks with a set of examples; [Ellman, 1989] trained a recurrent network on a corpus of sentences, and the network was able to abstract noun/verb agreement. Researchers in computer vision have also been concerned with computing concise representations of large amounts of visual input data [Sanger, 1989; Saund, 1988]. Here, we attempt to bring some of this experience to bear in developing representations of protein structure.

2.2 Existing Representations of Protein Structures

A common format of known protein structure data (such as in

Brookhaven Protein Databank) gives lists of 3D coordinates for the atoms of all of the amino acids in a protein. This is not a good representation for the purpose of structure prediction because it is difficult to identify similar substructures across different proteins and, consequently, difficult to carry out generalization and abstraction.

Another way to represent the three-dimensional structures is to use a "distance matrix.'' For a protein sequence of N residues, the corresponding distance matrix contains NxN entries, each representing the distance between the C_α atoms of a pair of residues. Similar to the 3D-coordinate representation, a distance matrix carries almost all the information about the protein structure (except the handedness of a helix), but still it is not obvious how to build an abstraction hierarchy on top of it.

Another common way to represent the protein structure is to assign each residue in the protein to one of several secondary structure classes. Secondary structure is a relatively coarse characterization of protein conformation, indicating basically whether the amino acid residues are packed closely (α helix) or stretched into an extended strand (β sheet). Parts of proteins that don't match either category are generally labeled random coil.

Research so far on protein structure prediction has mainly focused on predicting secondary structures, e.g. [Qian & Sejnowski, 1988; Levin, Robson & Garnier, 1986; Chou & Fasman, 1974; Rooman & Wodak, 1988]. However, given the 3D coordinates of all the residues in a protein, researchers differ on how to assign secondary structures. There is broad agreement on what a typical α helix or β sheet is, but a real protein structure is complicated, and non-typical cases are common.[2] Coil is not really one class of local structures, but rather it includes many very different structures. Also, though it is known that groups of α helices and/or β sheets often form higher level structures (often called super-secondary structures) [Richardson, 1981]—and some researchers have even tried to predict particular super-secondary structures, such as $\beta\alpha\beta$ [Taylor & Thornton, 1984]—there has not been a rigorous, generally agreed way to identify different super-secondary structures in the known proteins.

The Ramachandran plot [Schulz & Schirmer, 1979], plots ϕ vs. ψ angles for all the residues in the set of protein structures used in this work. The definition of these angles is shown in Figure 1, and a Ramachandran plot is shown in Figure 2. We can see that the angles are not evenly distributed. There are two dense regions; these correspond to α helices and β sheets, respectively. We can also see clearly that this categorization does not capture the richness and variety in protein structure.

Thus, a good representation for protein structures is in demand for the purpose of structure prediction. It should produce a coherent description of protein structures at the residue level, secondary structure level and super-secondary structure level.

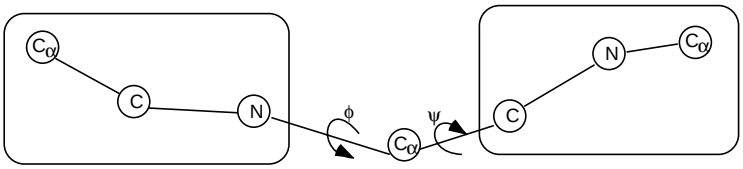

Figure 1: The definition of phi (ϕ) and psi (ψ) angles in a peptide chain. Most of the bond angles and bond lengths in an amino acid are quite rigidly fixed, as is the peptide bond that holds the amino acids together. There are two principal degrees of freedom in the backbone of a peptide chain: These angles are defined around α carbons by the indicated planes.

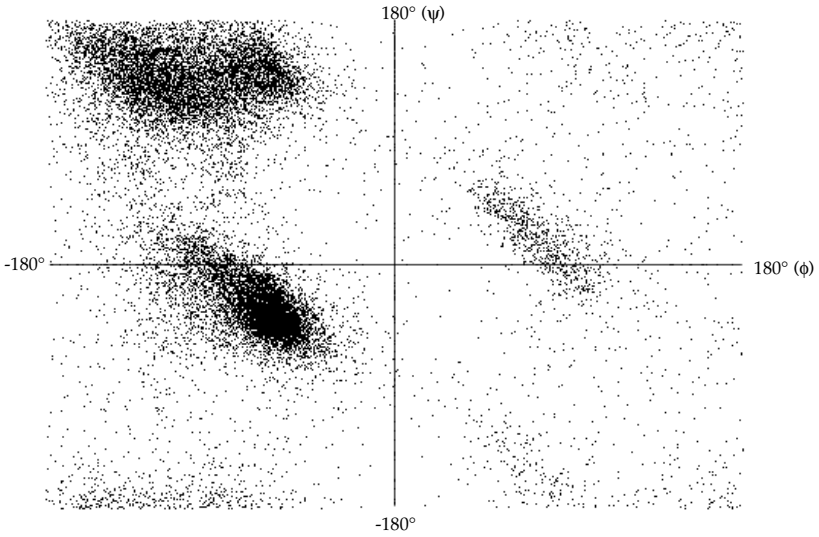

Figure 2. A Ramachandran plot of the ϕ versus ψ angles found in our dataset. Notice that although there are two main groups, there is significant variance within the groups, and many points outside of them.

3 Method

Our goal is: given the 3-D coordinates of all the residues in a protein, generate a representation that can both reflect the enormously complexity of protein structures and facilitate the identification of similar substructures across different proteins. This is exactly the kind of representation problem AI has been concerned with. Ideally, it should be possible to describe a protein structure at several levels of abstraction, so we desire a hierarchical representation.

We have taken an incremental, bottom-up approach in developing representations for protein structures. The first step is to find a set of lowest level

primitives with which we will build higher level structures. These primitives must capture as much relevant information about the structure as possible, and do so in a maximally compact way. In order to define the base level, we apply a neural network-based techniques related to principal component analysis to a dataset that contains the structural state of each residue in a large number of proteins.

The second step is to group continuous stretches of structural states to form local structures, roughly corresponding to what have been called secondary structures. We take the primitives we developed in step one, and use them to find groups of similar residues, using a method related to k-means classification; this provides a level of description roughly commensurate with secondary structure. Finally, we assemble our groups of related local structures to form higher level structures, which corresponds to super-secondary structures.

The advantages of this approach are (a) given a protein's residue coordinates, it can generate representations at all three levels discussed above automatically—higher level representations are built upon lower level ones; (b) these representations are grounded on a few objective, observable structural parameters whose accuracy depends only on the accuracy of the crystal data, rather than some subjective judgment; (c) it is straightforward to compute a distance (similarity/difference) between any two structures in this representation; thus when categories are formed, it is possible to compute how different one category is from another, or how far a particular instance is from the mean for each category.

All of the inference was done on a subset of protein structures taken from the Brookhaven Protein Databank. The subset consisted of 105 non-homologous protein structures; we call our subset the Protein Structure DataBase (PSDB) in the following discussion.

3.1 Defining Primitives

3.1.1 Source data. Abstraction and generalization must be solidly grounded. In this work, each residue in a protein in PSDB is associated with a number of structural parameters. The three parameters used here are the dihedral angles (ϕ, ψ) and water accessibility (ω)[3]. Dihedral angles represent a residue's spatial orientation relative to its two immediate neighbors in the protein sequence, and the water accessibility reflects whether the residue is on the surface of or inside the protein. ω is included here because it is generally believed that hydrophobic interaction plays an important role in protein folding. Also, whether a residue is on the surface of or inside a protein is an important source of structural information.[4] The residue state vector for residue *i* is defined as a 9-tuple:

$$SV_i = <\omega_{i-1}, \Phi_{i-1}, \Psi_{i-1}, \omega_i, \Phi_i, \Psi_i, \omega_{i+1}, \Phi_{i+1}, \Psi_{i+1}>$$

That is, each SV_i depends on residue *i*'s ϕ , ψ and ω parameters and on those of its two nearest neighbors in the sequence. In this work, ω takes a binary value: either 0 (inside) or 1 (on surface). ϕ and ψ are are rounded to the nearest multiple of 20 degrees. Any pair of residues that have at least 3 identical angles and no angles that differ by more than 20 degrees are forced to have identical state vectors. Residue state vectors include all aspects of protein structure of concern here; it is on this basis that the abstraction hierarchy is built.

All the state vectors for all of the residues in the entire PSDB were computed, and 7159 distinct residue state vectors were found. This is a highly nonrandom distribution; in theory, there are about $3.8*10^7$ possible residue state vectors. In addition, the histogram of occurrence of vectors is highly skewed. The mean number of times a state vector occurs in the database is 3; the most frequent one occurs 2027 times.

3.1.2 Computing Canonical Representations by an Auto-association Network. Computing the state vector for each amino acid residue in a protein structure provides a great deal of information about the structure, but in a less than ideal form. The elements of the state vectors are highly dependent upon each other, and it is unclear how to measure the distance between a pair of vectors. The different dimensions of the vector have different value ranges and value distributions; it is not clear how to weight their relative importance. A canonical representation is needed to make useful information explicit and strip away obscuring clutter. Our approach was to use an auto-associative back-propagation network [McClelland & Rummelhart, 1986] to automatically identify the intrinsic features implied in the state vectors.

It has been shown that, when properly designed, the hidden unit values of an auto-association back-propagation network will identify the intrinsic features of its inputs [Bourlard & Kamp, 1988; Saund, 1986]. A network trained to reproduce values at its input units (within a given accuracy) using a smaller number of hidden units has found a more compact encoding of the information in the input unit values. This process is related to principal component analysis (see section 5). In addition, something else is available for free: if the hidden unit values for each residue state vector are used as its new representation, the lower half of the trained network (input→hidden layers) can be used as an encoder, and the upper half (hidden→output layers) can be used as a decoder.

At this point, we needed a mapping of the state vectors to binary vectors as required by the autoassociative network encoding process. Since the accuracy of ϕ and ψ angles is around 20° in PSDB, and these angles range over [-180°, 180°], 18 units are used to encode one angle. The unit activity then is smeared by a Gaussian function to the unit's neighbors, thus similar angle values will have encodings near each other in Hamming space. This encoding of real values is similar to that in [Saund, 1986]. Four units are used to encode each ω value. This is required so that the backpropagation error signal for ω will not be overwhelmed by that from

the angles. The network and encoding are shown in Figure 3.

After the network is trained on all of the state vectors, it can be used as an encoder and decoder, to translate from state vectors to the newly defined primitives. Each residue state vector can be mapped to a 20-element vector on [0,1] obtained from the 20 hidden units of the backpropagation network. These are called residue feature vectors. But treating the production of these vectors solely as a blackbox encoding is somewhat unsatisfying; what do the values of the hidden units mean?

Each hidden unit captures certain features of the input vectors. One example is a hidden unit which is sensitive primarily to the first, fourth and seventh position of the input vectors, that is, to the ω values. For example, when the input vectors have the form <0,?,?,0,?,?,0,?,?>[5], the output of the 6th hidden unit is always close to 0. Another, more complex example demonstrates a distributed representation: when one hidden node is low ($V_0 \leq 0.3$) and another hidden node is high ($V_2 \geq 0.8$) the input vectors are always of the form <?,?,?,?,?,?,?,-120,160>, indicating the beginning of a β sheet.

The hidden unit value distributions were plotted for all 7159 distinct residue state vectors. The values of each of the hidden nodes over the range of training examples took on one of three distinctive distributions: bimodal, multimodal and normally distributed. Figure 4 shows one example from each kind.

We now have a method for translating objective information about the amino acid residues in a protein structure into a set of independent, compatible features. The next step is to assemble these examples of protein structure into general classes, based on the feature vectors we have just defined. These features provide the basis for an objective, general classification.

3.2 Finding Common Structures Using the Primitives

We claim that the hidden unit values represent intrinsic features of the network inputs. The residue feature vector representation not only provides a good representation for clustering, but also a way to measure the "distance'' between different clusters (how similar two classes are) and the "distance'' between a particular instance and the center of the cluster it belongs to (how typical it is to the class). This distance measure allows us to apply a standard clustering algorithm to find groups of similar structures from the examples that we have.

3.2.1 The Clustering Algorithm. The clustering on the 7159 20-element residue feature vectors was carried by a clustering procedure implemented on the Connection Machine CM-2 which is similar to K-means clustering.[6] Briefly, it does the following:

1. Get arguments: n — the number of clusters required; m — the number of iterations desired;
2. Randomly select n vectors from the 7159 residue feature vectors as "seeds'' of the n clusters;
3. For each of the rest of the feature vectors, find the closest seed and put

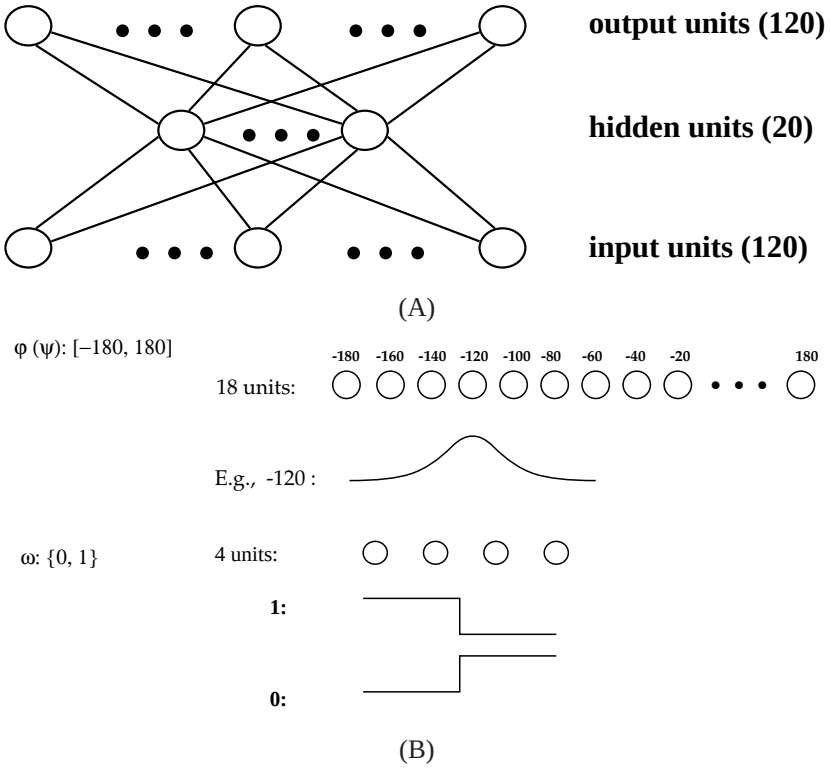

Figure 3. The design of an auto-association backpropagation network for transforming the residue state vectors to a canonical form. (A) The net contains 120 input units, 20 hidden units, and 120 output units. After training, the hidden unit values are taken as the canonical form. (B) The parameters ϕ and ψ are encoded as the activities of the input/output units in the backpropagation network by quantizing the angle to the nearest multiple of 20° and smearing the value over several neighbors. ω, which is binary, is encoded with four bits.

the vector into that cluster;

4. Compute the deviation in each cluster, then compute the average deviation of all clusters;
5. Repeat m times from Step 2 to Step 4 above, and select as the result the clustering that has the smallest average deviation.

3.2.2 Clustering Results. Therefore, a classification of residue state vectors based on the feature vectors should put similar structural states into the same class. Using a method related to K-means clustering, the residue feature vectors were classified into clusters with small average deviations. We looked for something around 20 classes at the beginning, and we found that using 23 clusters produced the grouping with the smallest overall deviations.

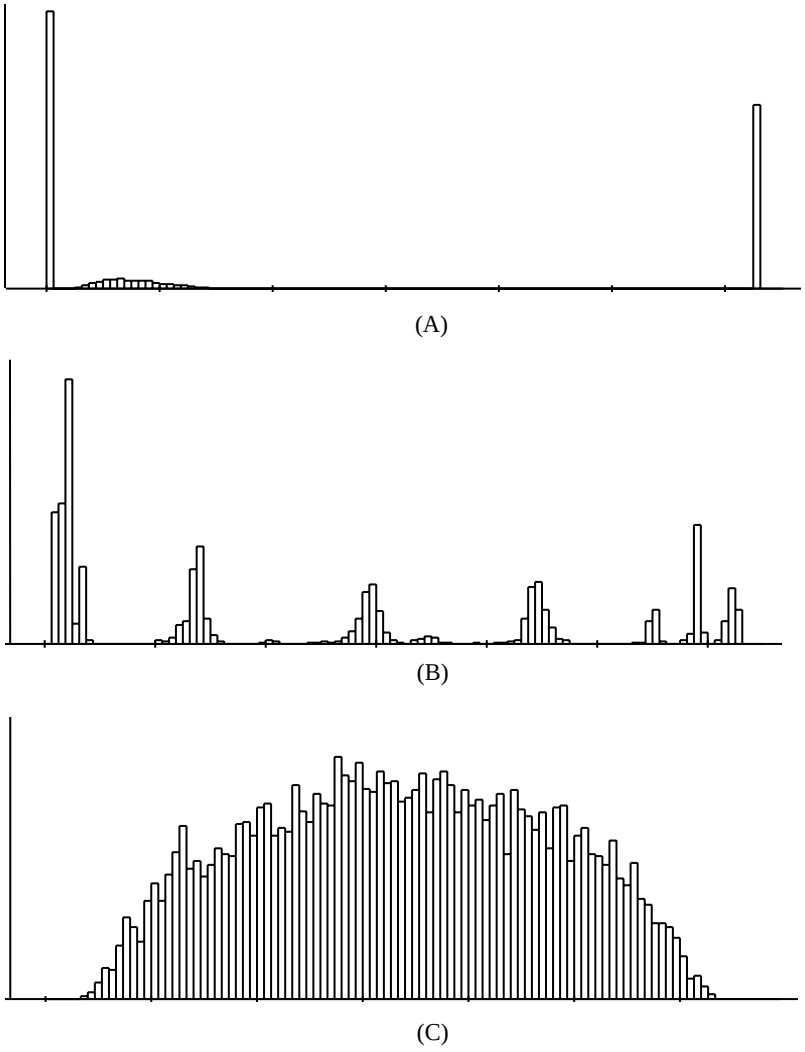

Figure 4. Three kinds of distributions of the hidden unit values. (a) V_6 (The 6th hidden unit). This kind of hidden units divides all the inputs into two classes. There are six hidden units with this kind of distribution. (b) V_1. This kind of units classifies all the inputs into a few categories. There are six such units. (c) V_0. Eight units have this kind of normal distribution.

Figure 5 shows an example of a cluster. 50 residue state vectors from this cluster are plotted by ϕ/ψ angle. It is clear that they share strong family resemblance. The 23 residue structural classes found by the clustering procedure are denoted by C_1, C_2, ..., C_{23}.

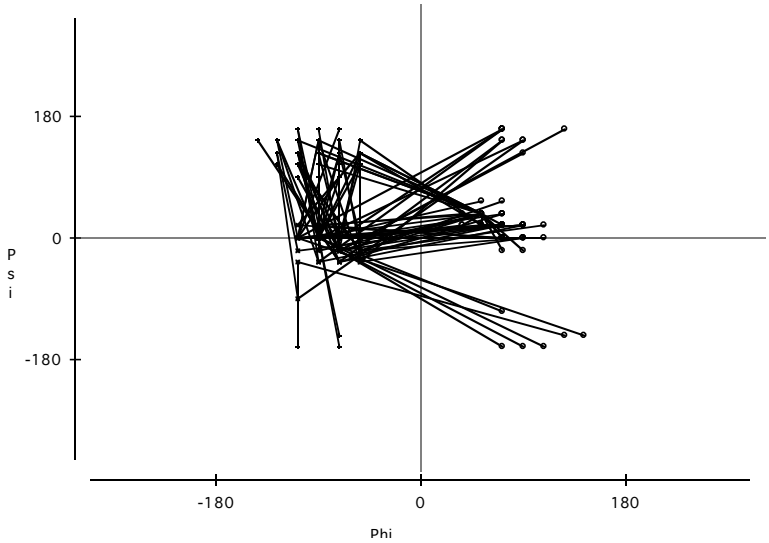

Figure 5. A cluster of residue state vectors made by a K-mean clustering procedure. Three consecutive $<\phi,\psi>$ pairs are three points on the $\phi-\psi$ plane joined by two straight lines, each line starts with a ∘, goes through an x *and ends with a +. ω's are not displayed here.*

3.3 Correlation Between Residue State Classes And Amino Acids

It was found that there are strong amino acid preferences for each of the 23 classes computed above. That is, some amino acids appear very frequently (or rarely) in particular classes. Figure 6 shows the results of a χ^2 correlation test between the 20 amino acids and the 23 classes in PSDB.

3.4. Identification of Common Substructures

In PSDB, strong correlations exist among structural classes C_1, C_2, ... C_{23} themselves, also. That is, in a protein, when C_i occurs at one place, some C_j tends to occur at another place. A number of class patterns were identified based on this kind of correlation.

3.4.1 Labeling the Residues with Structural Classes. Given 3D protein structure, we can compute a 20-element feature vector for each residue by the trained lower half of the auto-association network in Figure 8. Then from the feature vector, we can determine which of the 23 structural classes the residue belongs to. Thus all the residues in the sequence can be labeled for structural class membership. That is, the structure of the protein can be represented as (assuming there are n residues):

$$C^2C^3C^4C^5C^6C^7 \ldots C^{n-1}$$

where $C^i \in \{C_1, C_2, \ldots C_{23}\}$, i = 2, 3, ... (n-1). The first and the last residues each have only one neighbor residue, and thus their structural class-

	A	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	Y
C1	•			•		•			•	•	•	•	•		•		•	•		•
C2						•							•	•						
C3			•			•		•				•	•					•		•
C4	•		•	•	•	•		•		•			•		•	•	•	•	•	•
C5					•				•	•		•	•							
C6													•							
C7	•	•			•		•	•					•	•				•		•
C8					•			•		•			•			•				
C9					•	•			•	•			•			•		•		•
C10	•		•	•	•				•		•	•	•	•	•	•	•	•		•
C11	•	•	•	•			•	•	•	•		•	•		•	•		•		
C12						•							•		•			•		
C13	•		•	•		•		•	•			•	•	•		•		•		•
C14	•	•	•	•	•	•		•	•	•	•	•		•		•	•	•	•	•
C15			•			•		•	•	•	•	•	•	•		•		•		
C16	•	•		•																
C17			•	•																
C18			•																	
C19	•	•	•																	
C20		•																		
C21	•			•																
C22		•																		
C23	•			•	•	•			•	•			•	•	•	•				

Figure 6 The contingency table for χ^2 tests on the correlations between amino acids and classes. Each column corresponds to an amino acid, and each row corresponds to a class. A mark means the corresponding amino acid and class are correlated with a confidence level >95%

es cannot be computed.

3.4.2 Repetitive Patterns. It is known that there are some repetitive local structures in proteins, mainly α helices and β sheets. Pattern matching techniques using fixed pattern size do not work here because α helices and β sheets occur with different lengths. Finite state automata (FSA's) can easily recognize sequences of symbols of variable length. A set of heuristics was used to inductively generate FSA's from instances in PSDB, and these automata were then used to identify all the similar structures. The heuristics used were:

(1) $C_i\ C_i\ ...\ C_i\ (N, M) \rightarrow C_i^+$

(2) $C_i\ ...\ C_i\ C_j\ ...C_j\ C_i\ C_i ...\ C_j ...\ (N, M) \rightarrow (Ci^*Cj^*)^+$

Heuristic (1) says that if in a protein sequence, a structural class C_i occurs continuously along the sequence for at least N residues, and this occurs in M different protein sequences, then generate regular expression C_i^+ as a general representation for such continuous, repetitive local structures.[7] Heuristic (2) is similar to (1), except that it deals with two interleaving structural classes C_i and C_j.

Four regular expressions (representing the FSA's) were generated from protein sequences labeled with 23 classes $\{C_1, C_2, ... C_{23}\}$:

(1) $RE_1 = C_{21}^+$

(2) $RE_2 = C_{14}^+$

(3) $RE_3 = (C_1{*}C_{21}{*})^+$

(4) $RE_4 = (C_{17}{*}C_{23}{*})^+$

The average residue state vector values for these classes are:

	ω_1	ϕ_1	ψ_1	ω_2	ϕ_2	ψ_2	ω_3	ϕ_3	ψ_3
C_{21}:	<0.3	-54	-32	0.3	-67	-34	0.0	-75	-36>
C_1:	<0.3	-61	-26	0.4	-46	-37	1.0	-65	-26>
C_{14}:	<0.2	-92	104	0.0	-104	110	0.0	-88	93>
C_{17}:	<0.4	-93	107	0.3	-81	108	1.0	-88	116>
C_{23}:	<0.3	-73	70	1.0	-89	113	0.0	-82	96>

These repetitive patterns correspond to α helices (RE_1 and RE_3) and β sheets (RE_2 and RE_4). The main difference between RE_1 and RE_3, and between RE_2 and RE_4 is whether the local structure is on the surface of or inside proteins.

3.4.3 Non-repetitive Structural Class Patterns. After the repetitive local structures were identified, structural classes that occur often at the beginning or the end of RE_1, RE_2, RE_3 and RE_4, and non-repetitive class patterns that happen frequently in PSDB were found. Some interesting phenomena were observed. For example, class pattern $C_{10}C_7$ occurs often at the beginning of RE_3 (61 times in PSDB), but never occurs at the end of RE_3 while class pattern C_3C_4 occurs 79 times at the end of RE_3, but never at the beginning. This suggests that classes in these two sets are not just variations of the classes inside RE_3, but rather they have specific preference for places they occur. Also about 100 non-repetitive class patterns (with length ≥ 4) were found that occur 20 times or more in PSDB.

Table 3 shows how many α helices and β sheets (identified by RE_1, RE_2, RE_3, RE_4) have common sequences proceeding them (heads) or sequences that follow them (tails) in PSDB. More than 75% of the helices and sheets have both head and tail ((413-89-11)/413 = 75.9%, (695-137-21)/695 = 77.3%). Only 3% of the helices and sheets have neither head and tail (11/413 = 2.7%, 21/695 = 3%). Thus, the occurrences of the heads and the tails suggests strongly the existence of the corresponding secondary structures.

Finally, groups of RE_i's are found that are close to each other in space and less than 15 residues apart along the sequence. For example, $RE_2...RE_3...RE_2$ (two sheets with a helix in between) occurs 18 times in PSDB. This is an example of what has been called super-secondary structure.

4 Summary and Discussion

4.1 Protein Structures

The success of protein structure prediction research depends on whether "rules" can be found that govern the correspondence between amino acid se-

Head and Tail of Helices and Sheets

RE's	Total	Lack One	Lack Two	Lack Head	Lack Tail
RE_1 & RE_3	413	89	11	54	57
RE_2 & RE_4	695	137	21	86	93

Table 3: The number of occurrences of the ``heads" and the ``tails" of the secondary structures identified by RE_1, RE_2, RE_3, RE_4.

quences and structures they form. We argue that representation plays an important role here—good representation exposes natural constraints.

In this paper, starting with a few primitive structural features about residues and some generalization techniques, we have developed representations for protein structures at several levels. As shown elsewhere [Zhang, 1990], we obtained a much higher secondary structure prediction accuracy with this representation than other representations; these representations greatly facilitate the prediction of protein structures. The correlations among structures at different levels revealed by these representations impose constraints about which amino acids will form what structures and where (in relation to other structures). This suggests that instead of predicting the state of each residue as an isolated individual (which is the case in most secondary structure prediction work today), the structural states of all the residues in a protein should be treated as a mutually related whole. The structure hierarchy described in this chapter is one way that these relations can be found and represented.

The representation here also has its limitations. Right now it only covers certain super-secondary structures—those that are close to one another in space and not very far away from each other along the sequence. It cannot account for all the global interactions.

4.2 General

In this paper, several computational tools have been successfully applied to the problem:

Feature Extraction. This was done by an auto-association network and proved to be a very useful tool for the purpose. Auto-association networks have been shown to be similar to principal component analysis [Bourlard, et al., 1988]. However, with non-linear input/output units, their dynamics are not yet fully understood. A principal component analysis method was applied to the same problem but did not produce as good a result in terms of forming meaningful clusters of protein local structures. One explanation for this is that the original dimensions need to be properly weighted in the principal component analysis to be successful.

Primitive Identification In this chapter, clustering of the original data based on their canonical features gave rise to meaningful categories. This gives an interesting example about the relationship between symbolic and non-sym-

bolic reasoning: the original data — ϕ, ψ angles and ω's — are clearly non-symbolic, and yet the labels for the final structural classes are symbolic. These symbols (which correspond to the structural classes) emerged from the computation on non-symbolic data. They facilitate reasoning by identifying similar things and omitting details (abstraction!). They differ from the symbols in classic AI in that they are "backed up'' by non-symbolic (numeric) information, so they can be compared, combined or broken into smaller pieces.

Correlation Among Primitives. Finite state automata and pattern matching techniques have been used to determine correlations among representation primitives. The sequential nature of the input data was explored to make such techniques applicable.

The above techniques could be applied to other representations of protein structures such as the distance matrix, or to problems in other domains (maybe in slightly different form), such as speech recognition and text processing. It is hoped that the lessons learned in this work will shed light on research in these domains as well.

Acknowledgment

We would like to thank Jill L. Mesirov and Robert Jones of Thinking Machines Corporation for many comments and suggestions on this work, and Chris Sander for providing the DSSP program and for helpful discussions. This work was supported in part by the Defense Advanced Research Projects Agency, administered by the U.S. Air Force Office of Scientific Research under contract number F49620-88-C-0058, while the first author was a graduate student at Brandeis University, and in part by Thinking Machines Corp. Cambridge, MA, both author's current affilitation.

Notes

1. It stands for "GENErate REPresentations."
2. When comparing the assignment of secondary structures by crystallographers and the one by Kabsch and Sander [Kabsch, 1983] (which is commonly used by structure predictors) for some protein sequences in Brookhaven Protein Databank, we found that they classify as many as 20% of the residues differently.
3. These parameters were computed from PSBD atomic coordinates by Kabsch and Sander's program DSSP.[Kabsch, 1983]
4. w is based on the number of water molecules bound to each residue. The w value computed by DSSP is normalized to be in [0, 1].
5. Where ? means any value.
6. Anand Bodapati at Thinking Machines Corp. kindly provided the initial

code. We modified it to suit our need.

7. The regular expression also specifies a FSA that can recognize all the sequences that can be represented by this expression.

References

Bourlard, H. & Kamp, Y. (1988). Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. *Biological Cybernetics*, 59: 291-294.

Brachman, R. & Levesque, H. (1985). *Readings in Knowledge Representation*. Morgan Kaufmann,

Charniak, E. & McDermott, D. (1985). *Introduction to Artificial Intelligence* . Reading, MA: Addision-Wesley.

Chou, P. & Fasman, G. (1974). Prediction of Protein Conformation. *Biochemistry, 13*(2),

Ellman, J. (1989). *Representation and Structure in Connectionist Models* (CRL TR 8903). Center for Research in Language, UCSD.

Hinton, G. (1988). Representing Part-Whole Hierarchies in Connectionist Networks. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society* (pp. 48-54).

Kabsch, W. & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers, 22*(12), 2577-2637.

Levin, J. M., Robson, B. & Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *205*(2), 303-308.

McClelland, J. & Rummelhart, D. (1986). *Parallel Distributed Processing*. Cambridge, MA: MIT Press,

Qian, N. & Sejnowski, T. (1988). Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *Journal of Molecular Biology, 202*, 865-884.

Richardson, J. (1981). The Anatomy and Taxonomy of Protein Structure. *Advances in Protein Chemistry, 34*, 167-339.

Rooman, M. & Wodak, S. (1988). Identification of Predictive Sequence Motifs Limited by Protein Structure Data Base Size. *Nature, 335*(1), 45-49.

Sanger, T. (1989). *Optimal Unsupervised Learning in Feedforward Neural Networks* (1086). MIT Artificial Intelligence Laboratory.

Saund, E. (1986). Abstraction and Representation of Continuous Variables in Connectionist Networks. In *Proceedings of Fifth National Conference On Artificial Intelligence* (pp. 638 - 644). Morgan Kaufmann.

Saund, E. (1988). *The Role of Knowledge in Visual Shape Representation* (1092). MIT Artificial Intelligence Laboratory.

Schulz, G. E. & Schirmer, R. H. (1979). *Principles of Protein Structure* . New York: Springer-Verlag.

Taylor, W. & Thornton, J. (1984). Recognition of Super-secondary Structure in Proteins. *Journal of Molecular Biology, 173,*

Tesauro, G. & Sejnowski, T. J. (1989). A Parallel Network that Learns to Play Backgammon. *Artificial Intelligence, 39*, 357-390.

Winston, P. (1984). *Artificial Intelligence* (2nd ed.). Reading, MA: Addison-Wesley.

Zhang, X. (1990). *Exploration on Protein Structures: Representation and Prediction*. Ph.D., Brandeis University,