INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

A MULTI-LEVEL NEAREST-NEIGHBOUR ALGORITHM FOR PREDICTING PROTEIN SECONDARY STRUCTURE

Iustin Lazar

A Thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science
Concordia University
Montréal, Québec, Canada

March 1998

Abstract

A Multi-Level Nearest-Neighbour Algorithm for Predicting Protein Secondary Structure

Iustin Lazar

A thesis on machine learning and prediction of protein secondary structure.

We develop a variation of the nearest-neighbour algorithm that adopts a multi-level strategy together with a variable window size. The algorithm is applied to the problem of predicting the secondary structure of a protein given its primary structure: that is, given a sequence of amino-acids, output a sequence of secondary structures (helix, sheet, or coil).

A new training set is developed that is orthogonal, and covers the known classes of proteins.

Overall accuracy is 65.0%, with 68.7% accuracy for helices, 66.3% accuracy for sheets, and 61.4% for coils. This compares well with existing methods, in that the best results for a single nearest-neighbour classifier is 65.1% by Salzberg and Cost in 1992. Our accuracy rate for sheets is better than known methods, but our accuracy rate for coils is much lower than existing methods.

Acknowledgements

I would like to thank my supervisor, Dr. Gregory Butler, for his patience and valuable guidance.

I would also like to thank to Peter Montgomery, who graduated in Biochemistry, for his contribution to the construction of the training sets.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, and *Fonds pour la Formation de Chercheurs et l'Aide a la Recherche.*

Table of Contents

List of Figures

List of Tables

Chapter 1. INTRODUCTION

1.1. Problem Statement

The rapid development of powerful biochemical concepts and techniques in recent years has enabled investigators to tackle some of the most challenging and fundamental problems in biology and medicine. For exemple, it is now known that common molecular patterns and principles underlie the diverse expressions of life. These patterns of chemical transformations in biological systems are determined by proteins. That is why the study of structure and function of proteins are so important.

Proteins are a unique class of macromolecules being able to specifically recognize and interact with highly diverse molecules; they are built of amino acids, playing various and crucial roles in virtually all biological processes ([Str88]).

Amino acids are the basic structural units of proteins. Each protein is built of a unique, precisely defined amino acid sequence; these sequences are genetically determined. An α-amino acid consists of an amino group, a hydrogen atom, and a distinctive R group bonded to a carbon atom, which is

called the α-carbon becuse it is adjacent to the carboxyl (acidic) group. The R group is referred to as a side chain.

There are twenty amino acids (or twenty two, considering the variations in two of them) varying in size, shape, charge, hydrogen-bonding capacity and chemical reactivity. All proteins in all species, from bacteria to humans, are constructed from the same set of twenty amino acids ([Str88]). The remarkable range of functions performed by proteins is the result of the diversity and versatility of this twenty kinds of building blocks.

Proteins are made of a sequence of amino acids, which interact due to their positions and chemical and physical properties. This sequence uniquely determines the three-dimensional structure of the proteins which is, in its turn, in a one-to-one relationship with the protein functionality.

Secondary structure refers to the spatial arrangement of amino acid residues that are near one another in the linear sequence. Some of these relations are of a regular kind, giving rise to a periodic structure. In such cases, the chain of amino acids folds into regular repeating structures called α-helix, β-sheet, β-turn and coil.

Tertiary structure refers to the spatial arrangement of amino acid residues that are far apart in the linear sequence (see Fig.1.1). Proteins have well defined three dimensional structures, but their computation is extremely expensive in terms of time and resources. Function arises from conformation, which is the three-dimensional arrangement of atoms.

The problem is to determine the functionality of a protein, starting from its sequence of amino acids. In order to determine its functionality, we must know the three-dimensional structure of the protein. Hence, the problem can be re-formulated: knowing the sequence of amino acids of a protein, determine its three-dimensional structure (also called tertiary structure) - see Fig.1.1.

The above statement constitutes one of the fundamental problems that are still incompletely solved: classical methods, as spectroscopy and crystallography are very expensive and provide a low productivity, in the context of an exploding number of known protein sequences (produced by large scale sequencing projects), in contrast to the much slower increase in the number of known protein structures.

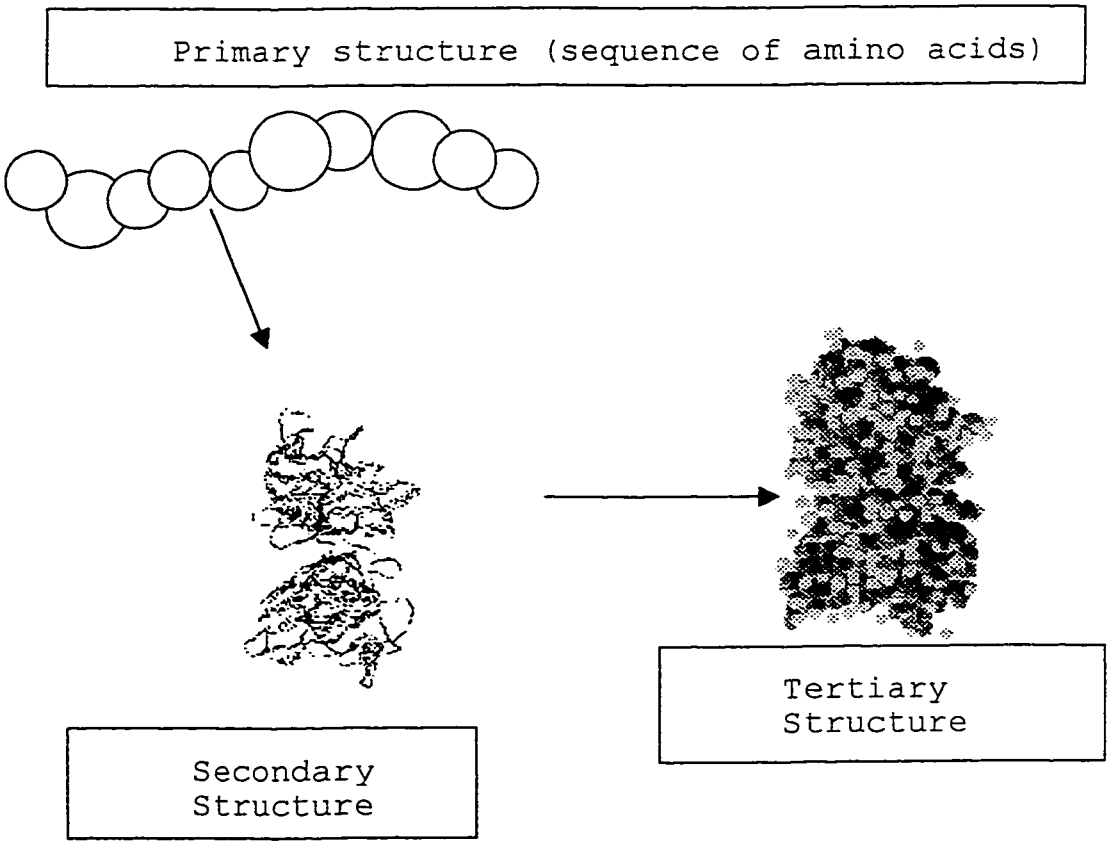

Fig.1.1. Problem definition

A key in finding the "hidden" relation between the primary and tertiary structure of a protein is the prediction of its secondary structure. The latter can be defined as being the spatial arrangement of amino acid residues that are near one another in the linear sequence.

Predicting the secondary structure of proteins is an important and necessary stage in predicting and understanding the tertiary structure of proteins. Secondary structure information can be incorporated into simulations that attempt to fold proteins. This information can also be

used to enhance the accuracy of programs designed to identify proteins that are homologous to a query sequence.

This is the subject of our thesis: an attempt to use an artificial intelligence method in order to predict the secondary structure of a protein. The method we chose is the nearest neighbor algorithm.

The nearest neighbor method was already used by several authors, either by itself or in combination with other methods (in hybrid systems). Each time a peculiar implementation was adopted, and some improvements /specialization were considered. We adopted our own approach both in implementation details and at a strategic level.

1.2. Overview of the Work

The protein secondary structure prediction is a typical classification problem: based on the sequence of amino acids of a protein, one tries to predict the secondary structure (its class). A given data set of proteins with known secondary structures is 'learned' by an algorithm. Then, an unknown protein is analysed by the same algorithm and, comparing it with the learned structures, similarities are

searched. The structure of the new protein is predicted, based on the similarities found with the learned patterns.

We chose the nearest-neighbor algorithm. The nearest-neighbor classifiers can be used to predict the secondary structure of proteins, with good results, compared to the other methods. The nearest-neighbor rule states that a test instance is classified according to the classifications of "nearby" training examples from a database of known secondary structures. An instance-based algorithm stores a series of training instances in its memory and uses a distance metric to compare new instances to those stored. New instances are classified according to the closest exemplar from memory.

There are several differences between the various nearest-neighbor algorithms used in different works and our approach. We started using a pure nearest-neighbor algorithm, cleaned from experimentally determined coefficients (as we encountered in some articles ([Cos93])).

Instead of a fixed window size, we adopted an extensible one, which we considered to be a natural way of allowing the algorithm to extend by itself the width of a found secondary structure. In our approach, more learned

secondary structures may simultaneously contribute to determine a new secondary structure in a tested protein.

We also attempt to predict the group of classes of the protein, using statistical information (the percentage of amino acids having preferences for certain secondary structures).

These are the two main differences between our version of the method and the others used by different authors.

We used two main groups of classes of proteins for our tests: α proteins and β proteins. Within each group, almost each class was represented by 2 or 3 proteins. The latter have different functionality, such that the data orthogonality requirement was guaranteed: there are no similarities between the proteins. One protein at a time was used as test protein and all the others were used as training set. The fact that there are no similarities between proteins ensures that the quality of the result is not influenced by high similarities of the test and training set.

1.3. Summary of the State of the Art

Best results were obtained with hybrid systems, especially by those using multiple sequence alignments - [Sal95], [Ros93] - see Tab.1.3. Different accuracy rates were reached for each secondary structure type, α-helix being predicted better than β-sheet. The β-sheets are more difficult to discover, and improving this factor is one of the goals of any approach. Not all authors specify the β-sheet accuracy.

Salamov & Solovyev [Sal95] reached 72.2% of accuracy with a system combining multiple sequence alignment and a nearest-neighbor algorithm, using a majority vote. This is the best known result. Rost & Sander ([Ros93]) used a system combining multiple sequence alignments and artificial neural networks (ANN), obtaining an accuracy of 70.8%; they reached 65.4% for β-sheet.

Cost & Salzberg [Cos92], [Cos93] achieved 71.0% accuracy with a nearest-neighbor algorithm (N-N) using a weighting scheme. This is the best single classifier. But this result was obtained on a particular data set, while on different data sets, the overall accuracy was 65.1% [Ros93].

Zhang ([Zha92]) reached 66.4% by combining neural networks with nearest-neighbor (called memory-based

reasoning) and statistical methods. Yi & Lander [Yi93] obtained 68% of accuracy with a hybrid system made of six nearest-neighbor modules and an artificial neural network as combiner. Maclin [Mac93] reached on overall accuracy of 63.6% using multi-layered network approach, while the Combine program ([Bio88]) reached 65.8 .

Tab.1.3. Summary of the State of the Art

Author	N-N	ANN	Statistics	Sequence Alignment	Induction	%	%α	%β
[Sal95]	Yes			Yes		72.2		
[Cos92]	Yes					71.0 65.1		
[Ros93]		Yes		Yes		70.8		65
[Yi93]	Yes	Yes				68.0		
[Zha92]	Yes	Yes	Yes			66.4		
[Bio88]			Yes			65.8		45
[Mac93]		Yes				63.6		
[Kin90]					Yes	60.0		

1.4. Summary of Results

The average result accuracy of the tests on α-proteins was 68.7 for α-helix and 65.4% for coil, using a set of 14 proteins from 5 classes.

The result accuracy of the tests on β-proteins was 66.3% for β-sheet and 59.0% for coil.

The overall per-residue accuracy is 65%. These results are an improvement for prediction of β-sheets and are competitive with overall accuracy results obtained by more sophisticated systems, it being known that the best results were obtained by authors using hybrid systems.

1.5. Overview of Chapters to Come

Chapter 2 on background has 4 parts. First, we describe our domain of application: biochemistry. We present basic notions of proteins and their components (amino acids) and structures: primary, secondary, tertiary, quaternary structures. We briefly detail our exact field of application: secondary structures, α-helices, β-sheets, coils, turns.

In the second part of this chapter we make a general presentation of Machine Learning. We define the learning process and the relation between Knowledge Acquisition and Machine Learning. We present the Machine Learning approaches and systems which are classified according to:

- the underlying learning strategy,
- the type of the knowledge acquired, and
- the domain of application.

The third part presents training and testing techniques used in Machine Learning, especially in our domain, underlying the criteria which guide the construction of learn and test sets. Different approaches are described, as well as measures of performance in classification.

The fourth part of chapter 2 contains the problem description and Machine Learning methods presently used in different systems for the prediction of secondary structure of proteins.

Chapter 3 describes basic principles of the Nearest Neighbor Method and significant details of our implementation.

Chapter 4 starts presenting the data format (Protein DataBase - Brookhaven University) of the protein files. Then we describe the structure of the data set, the test strategy and the results we obtained. At the end of this chapter, we interpret our results.

Chapter 5 presents the conclusions of the present thesis: after a summary of the work, we add a summary of results and results interpretation. At the end, we suggest future improvements.

Chapter 2. BACKGROUND

2.1. Domain Description

2.1.1. Biochemistry

Biochemistry is the study of the molecular basis of life. The amazing diversity of life relies on common molecular patterns and principles: organisms as different as a bacteria and human beings use the same building blocks to construct macromolecules. The flow of genetic information from DNA (deoxyribonucleic acid) to RNA (ribonucleic acid) is the same in all organisms [Str88].

The chemical basis of many central processes being understood, secrets of the medical domain could be elucidated: molecular mechanisms of many diseases, inborn errors of metabolism, and clinical diagnosis.

Biochemistry is also a basis for the rational design of new drugs and agriculture benefits from recombinant DNA technology. Powerful biochemical concepts and techniques allow investigators to tackle some of the most challenging problems in biology and medicine: the control of growth of cells, the causes of cancer, the mechanism of memory, the

mechanism by which cells find each other when forming a complex organ, and many more.

2.1.2. Proteins

Proteins are a unique class of macromolecules in being able to specifically recognize and interact with highly diverse molecules. Proteins (a word coined by J.J.Berzelius in 1838 to emphasize the importance of this class of molecules, derived from the Greek word 'proteios' which means 'of the first rank') are molecules built of amino acids, playing various and crucial roles in virtually all biological processes ([Str88]): enzymatic catalysis, transport and storage, coordinated motion, mechanical support, immune protection, generation and transmission of nerve impulses, and control of growth and differentiation.

2.1.3. Amino Acids

Amino acids are the basic structural units of proteins. Each protein is built of a unique, precisely defined amino acid sequence. A series of studies in the late 1950s and early 1960s revealed that the amino acid sequences of proteins are genetically determined. The sequence of

nucleotides in DNA, specifies a complementary sequence of nucleotides in RNA, which in turn specifies the amino acid sequence of a protein ([Str88]). An α-amino acid consists of an amino group, a hydrogen atom, and a distinctive R group bonded to a carbon atom, which is called the α-carbon becuse it is adjacent to the carboxyl (acidic) group. The R group is referred to as a side chain.

There are twenty amino acids varying in size, shape, charge, hydrogen-bonding capacity and chemical reactivity. All proteins in all species, from bacteria to humans, are constructed from the same set of twenty amino acids([Str88]).

There are two amino acids - Aspartic acid (Asp) and Glutamic acid (Glu) - presenting light variations - Aspargine (Asx) and Glutamine (Glx). Aspargine and Glutamine are uncharged derivatives, containing a terminal amide group in place of a carboxylate. Including the latter amino acids, the total number reaches 22. Even if Aspargine and Glutamine are extremely rare in proteins, we consider them in the basic set of amino acids, in order to respect the exact composition of proteins, as it is presented in the PDB protein files (see chapter 4.1).

Amino acids are often designated by either a three-letter abbreviation or a one-letter symbol to facilitate

concise communication - see Tab.2.1.3. ([Str88]). The abbreviations for amino acids are the first three letters of their names, except for tryptophan (Trp), aspargine (Asn), glutamine (Gln) and isoleucine (Ile). The symbols for the small amino acids are the first letters of their names (e.g. G for glycine and L for leucine). The other symbols have been agreed upon by convention. We use the three-letter abbreviation because it is clearer for all users (biochemists or programmers) and because it is also used in the PDB files (our data).

Tab.2.1.3. Abbreviations for amino acids

Amino acid	Three letter abbreviation	One-letter symbol
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Aspargine or aspartic acid	Asx	B
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamic acid	Glu	E
Glutamine or glutamic acid	Glx	Z
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

2.1.4. Protein Structures

2.1.4.1. Primary Structure.

Each protein has a unique, precisely defined amino acid sequence. This is called the primary structure of the protein (see Fig.2.1.4.1.). The primary structure is thus a complete description of the covalent connections of the protein.

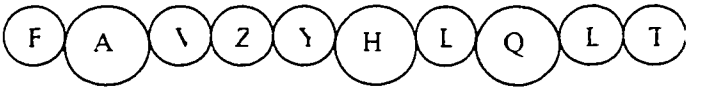

Fig. 2.1.4.1. The primary structure of a protein

The amino acid sequence is the link between the genetic message in DNA and the three-dimensional structure that guides a protein's biological function.

2.1.4.2. Secondary Structure.

A segment of the primary structure of a protein has a local spatial arrangement, due to local amino acid interactions. These local arrangements have a regular form and are known as secondary structure types, being classified as α-helix, β-sheet, β-turn or coil.

The secondary structure of a protein is the linear sequence of such local conformation classifications (e.g. for a primary sequence Ala-Val-Leu-Glu-Cys-His-Val-Ile-Ala-Pro-His-Ile-Ala, the secondary structure might be ααααααββββTCCCC, where α stands for α-helix, β for β-sheet, T for β-turn and C for coil).

Secondary structure types have a spatial arrangement and their interconnection generates the three dimensional form of a protein (see Fig.2.1.4.2.a.).

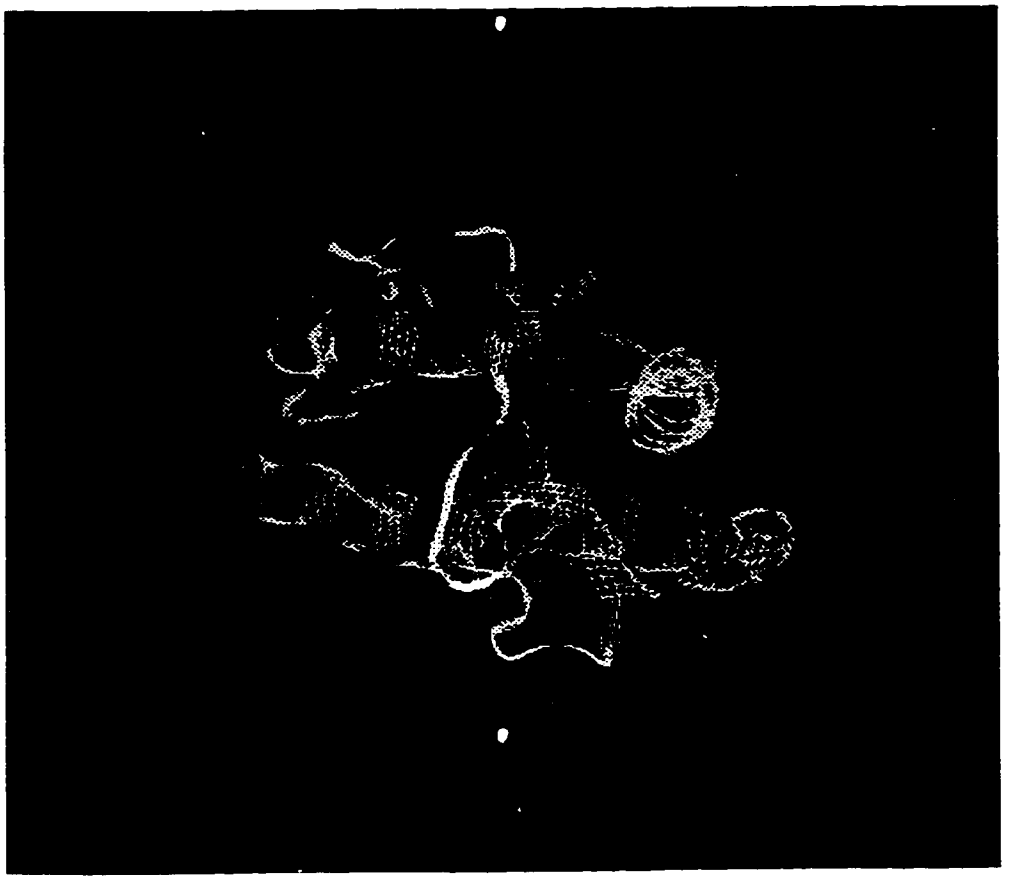

Fig.2.1.4.2.a. The secondary structures of a protein (2fx2: Electron transport)

Some of these relations are of a regular kind, giving rise to a periodic structure. In such case, the chain of amino acids folds into regularly repeating structures. Pauling and Corey (1951) called these structures α-helix and β-sheet - see Fig.2.1.4.2.b.

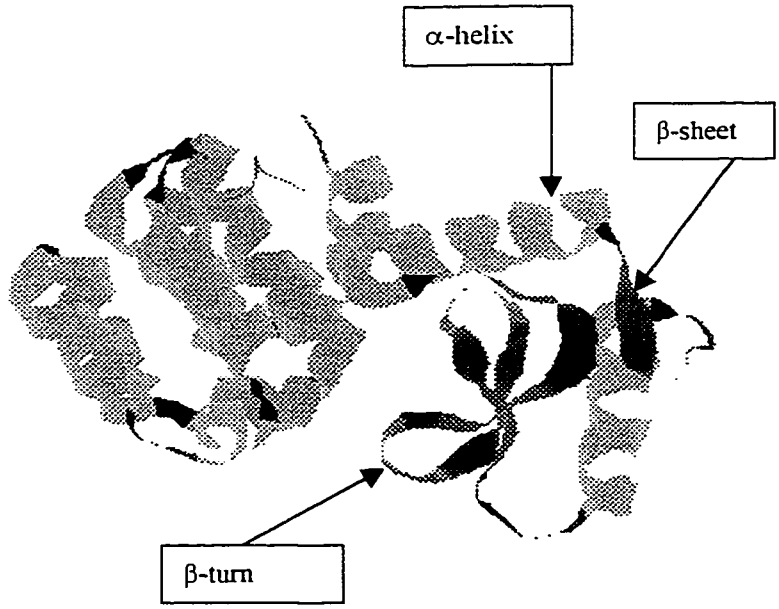

Fig.2.1.4.2.b. α-helix and β-sheet secondary structures of the 3lzm PDB-protein (HYDROLASE (O-GLYCOSYL))

Other structures are called β-turn and coil. The atom bonds of the amino acids determine the type of these structures.

2.1.4.2.1. α-helix

The α-helix is a rodlike structure (see Fig.2.1.4.2.1). It is stabilized by hydrogen bonds between the NH and CO groups of the main chain. The CO group of each amino acid is hydrogen bonded to the NH group of the amino acid that is situated four residues ahead in the linear sequence. The α-helices found in proteins are right-handed, that is, the screw sense of the helix is clockwise.

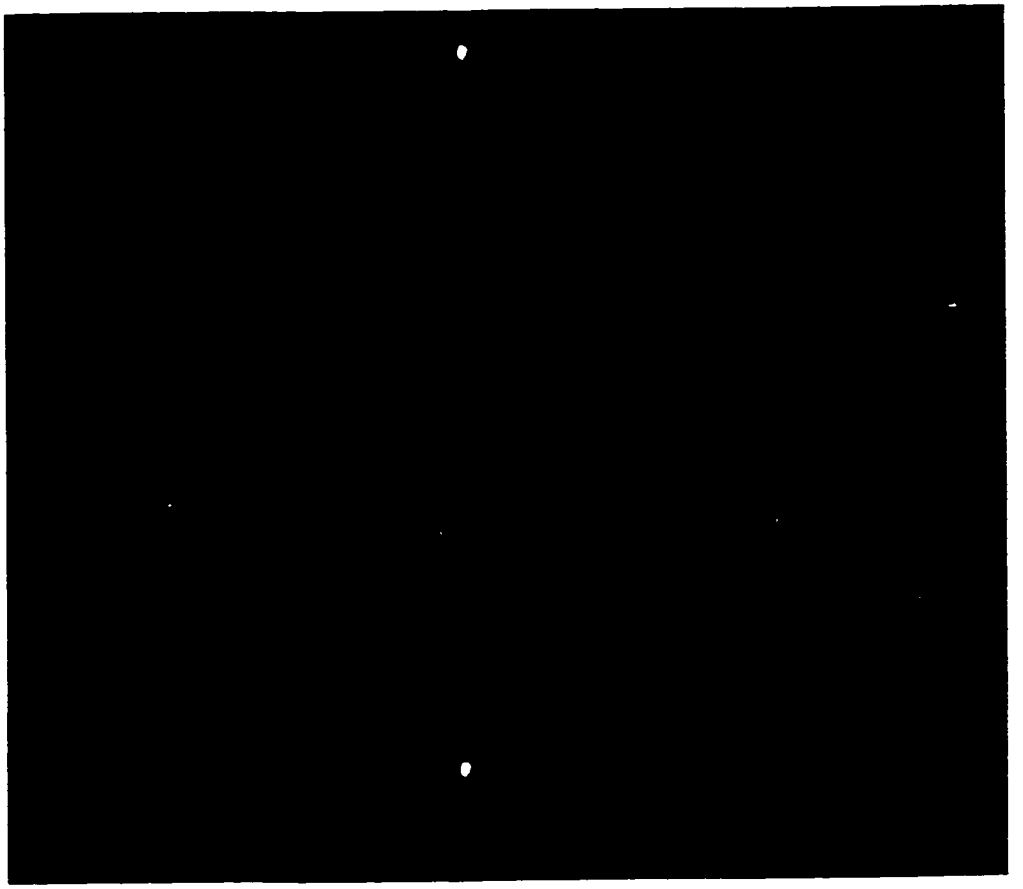

Fig.2.1.4.2.1. Two α-helix secondary structures

The α-helix content of proteins of known three-dimensional structure is highly variable. In some, such as myoglobin and hemoglobin, the α-helix is the major structural motif. Other proteins are virtually devoid of α-helix. In most proteins, the single-stranded α-helix discussed above is usually a rather short rod, typically less than 27 residues. In some proteins, the α helical theme is extended to much longer rods, as long as 700 amino acids or more (see Fig.2.1.4.2.2.). Two or more such α helices can entwine to form a cable. The helical cables in these proteins serve a mechanical role.

The elucidation of the structure of α-helix is a landmark in molecular biology because it demonstrated that the conformation of a polypeptide chain can be predicted if the properties of its components are rigorously and precisely known.

2.1.4.2.2. β-sheet

In 1951, Pauling and Corey discovered another periodic structural motif, the β-pleated-sheet (see Fig. 2.1.4.2.2.).

Fig.2.1.4.2.2. The β-sheet secondary structure

A polypeptide chain in the β-sheet is almost fully extended (flat). The axial distance between adjacent amino

acids is 3.5 A, in contrast with 1.5 A for the α-helix. Another difference is that the β-pleated-sheet is stabilized by hydrogen bonds between NH and CO groups in different polypeptide chains. Adjacent chains in a β-pleated-sheet can run in the same direction, called a parallel β-sheet or in opposite directions, called an antiparallel β-sheet. β-sheet regions are a recurring structural motif in many proteins.

2.1.4.2.3. β-turns

Most proteins have compact, globular shapes due to numerous reversals of the direction of their polypeptide chains. Many of these chain reversals are accomplished by a common structural element called a β-turn (see Fig.2.1.4.2.3.). The essence is that the CO group of residue n in a polypeptide is hydrogen bonded to the NH group of residue n + 3. A polypeptide chain can thus abruptly reverse its direction. β-turns often connect antiparallel β strands; hence their name.

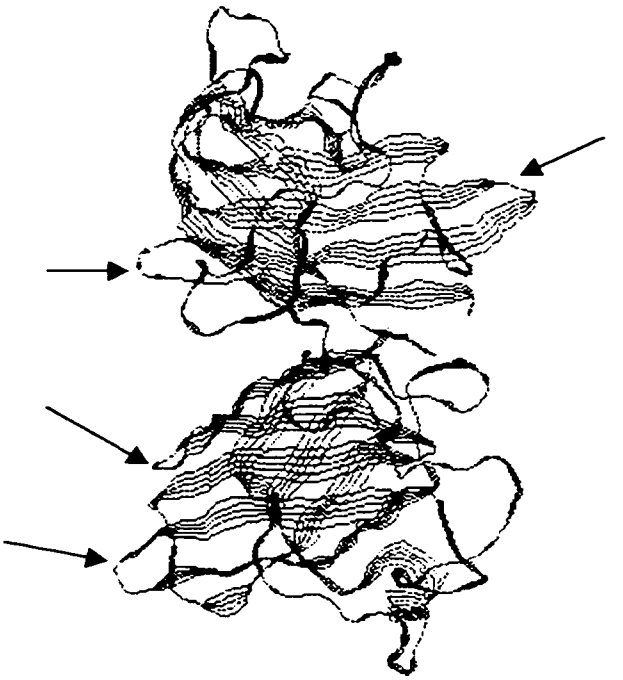

Fig. 2.1.4.2.3. β-turns reversing the direction of β-sheets into a β-Greek-key protein (1cob: Oxidoreductase)

2.1.4.3. Super-secondary Structure

Super-secondary structure refers to clusters of secondary structure. For example, a β-strand separated from another β-strand by an α-helix is found in many proteins. This motif is called a β-α-β unit. Super-secondary structures are intermediates between secondary and tertiary structure.

2.1.4.4. Tertiary Structure.

Tertiary structure, or 3D conformation, refers to the spatial arrangement of all amino acid residues of a protein (see Fig.2.1.4.4.). The dividing line between secondary and tertiary structure is a matter of taste, because secondary structures also have a spatial arrangement. From our point of view - the prediction of protein secondary structures - only the sequences of secondary structure classes are significant and we neglect their spatial arrangement. Proteins have well-defined three-dimensional structures. Function arises from conformation, which is the three-dimensional arrangement of atoms in a structure. Amino acid sequences are important because they codify the conformation of proteins ([Str88]). Weak, noncovalent bonds play key roles in the faithful replication of DNA, the folding of proteins into intincate three-dimensional forms, the specific recognition of substrates by enzymes, and the detection of signal molecules.

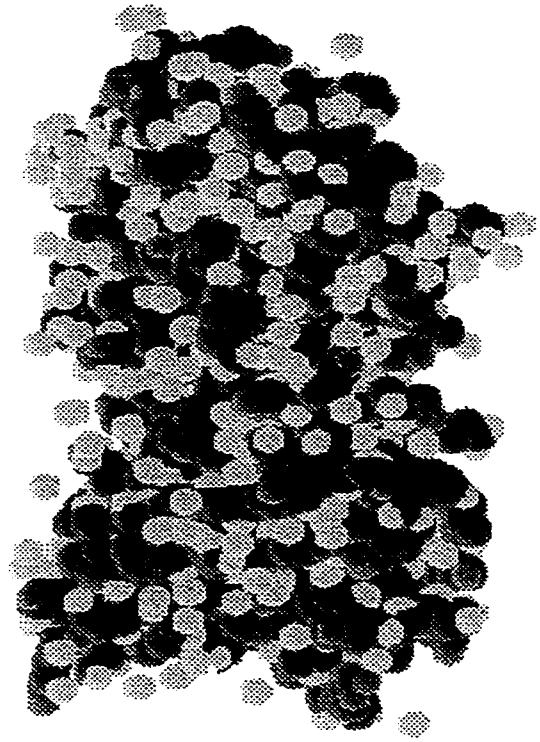

Fig.2.1.4.4. The tertiary structure of a protein (1cob - Oxidoreductase)

2.1.4.5. Quaternary Structure

Proteins containing more than one polypeptide chain exhibit an additional level of structural organization. Each polypeptide chain in such a protein is called a subunit. Quaternary structure refers to the spatial arrangement of subunits and the nature of their contact.

2.1.5. Protein Classes

Levitt & Chothia [Lev76] classified proteins in four major classes. Their method uses topology/packing diagrams which are two-dimensional representations of the three-dimensional structure of a protein. They identified

- all-α proteins, which have only α-helix secondary structures;
- all-β proteins, which have mainly β-sheet secondary structures;
- α+β proteins, having α-helix and β-strand secondary structure segments that do not mix but tend to segregate along the polypeptide chain. These proteins consist of a mixture of all-α and all-β regions; and
- α/β proteins, having mixed or approximately alternating segments of α-helical and β-strand secondary structure. α-helices and β-strands occur one after the other so that most α-helices are separated by β-strands along the sequence and vice versa.

2.2. Machine Learning (General Presentation)

The ability to learn is one of the most important attributes of intelligent behaviour. This is also one of the most striking differences between how people and computers work: humans, while performing any kind of activity, usually simultaneously expend efforts to improve the way they perform it. In other words, human performance of any task is inseparably intertwined with a learning process, while computers are typically only executors of procedures supplied to them. They may execute very efficiently, but they do not improve themselves with experience.

Computer programs able to construct new knowledge or improve existing stored knowledge are under research. But typically, the input information (examples, facts, etc.) is typed in by the human instructor.

Machine learning offers an immense diversity of research tasks and testing grounds. This diversity is due to the fact that learning can accompany any kind of problem solving and hence it may be studied in many different contexts (e.g. decision making, planning, control, task execution, signal recognition, classification) - see [Kod90].

Fields concerned with understanding intelligence include cognitive science, artificial intelligence, pattern recognition, information science, psychology, education, epistemology, and related disciplines. Progress in the theory and computer modeling of learning processes is of great interest to all these fields - see [Mic83]. Machine learning progress is central to the development of artificial intelligence, affecting most of its areas: expert systems, problem solving, computer vision, speech understanding, autonomous robotics, intelligent tutoring systems, conceptual analysis of databases, etc.

Learning processes include the acquisition of new declarative knowledge, the development of motor and cognitive skills through instruction or practice, the organization of new knowledge into general, effective representations, and the discovery of new facts and theories through observation and experimentation.

The study and computer modeling of learning processes in their multiple manifestations constitutes the subject matter of machine learning [Mic83].

Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.

Knowledge Acquisition and Machine Learning represent two complementary approaches to the acquisition, improving and organization of knowledge for knowledge-based systems. Knowledge Acquisition has focused on improving and partially automating the acquisition of knowledge from human experts by knowledge engineers. In contrast, Machine Learning has focused on developing autonomous algorithms for acquiring knowledge from data, and for knowledge compilation and organization. Currently, both fields are moving toward an integrated approach, using machine learning techniques to automate knowledge acquisition from experts, and knowledge acquisition techniques to guide and assist the learning process. This is one of the central research directions in AI, and also one of the most rapidly growing, due to its applicability to a wide range of practical problems - [Tec95].

Verification of the learned knowledge is through experimental testing on other data sets, the goal being the improvement of the Knowledge Base system's competence and/or efficiency in problem solving - [Tec95].

Machine Learning assumes that there already exists a representation language and background knowledge before any learning will take place.

2.2.1. Machine Learning Approaches

One may classify machine learning systems along many different dimensions (see [Mic83]), on the basis of:

- the underlying learning strategies used,
- the representation of knowledge, and
- the application domain.

Each point in the space defined by the above dimensions corresponds to a particular learning strategy, employing a particular knowledge representation, applied to a particular domain. Since existing learning systems employ multiple representations and processes, and many have been applied to more than one domain, such systems are characterized by several points in the space.

In every learning situation, the learner transforms information provided by a teacher (or environment) into some new form in which it is stored for future use. The nature of this transformation determines the type of learning strategy used - [Mic86].

The classification based on the underlying learning strategy includes: rote learning and direct implementation of new knowledge; learning by instruction; learning by deduction; learning by analogy; learning by induction

(learning from examples, learning from observation and discovery); learning by abduction; neural net learning; genetic algorithms and evolutionary computation; multistrategy learning (see [Tec95], [Mic86]).

The classification according to the type of knowledge representation contains: parameters in algebraic expressions (e.g. perceptrons); decision trees; formal grammars; production rules; logic-based expressions and related formalisms; graphs and networks; frames and schemas; computer programs and other procedural encodings; taxonomies (hierarchies); multiple representations.

The domain of application may be, for instance, one of the following:

agriculture; biology; chemistry; cognitive modeling (simulating human learning processes); computer programming; education; expert systems (high-performance, domain-specific AI programs); game playing; general methods (no specific domain); image recognition; mathematics; medical diagnosis; music; natural language processing; physics; planning and problem-solving; robotics; sequence prediction; speech recognition.

Kodratoff & Michalski [Kod90] proposed a classification of learning based on a few interrelated criteria, giving a general view of the whole field (see Fig.2.2.1). It illustrates important distinctions among various categories, which should not be viewed as having precise, sharp boundaries, but rather as labels for central tendencies that can be transformed from one to another. The criteria for classification include the primary purpose of the learning method, the type of input information, the type of primary inference employed, and finally, the role of the learner's prior knowledge in the learning process.

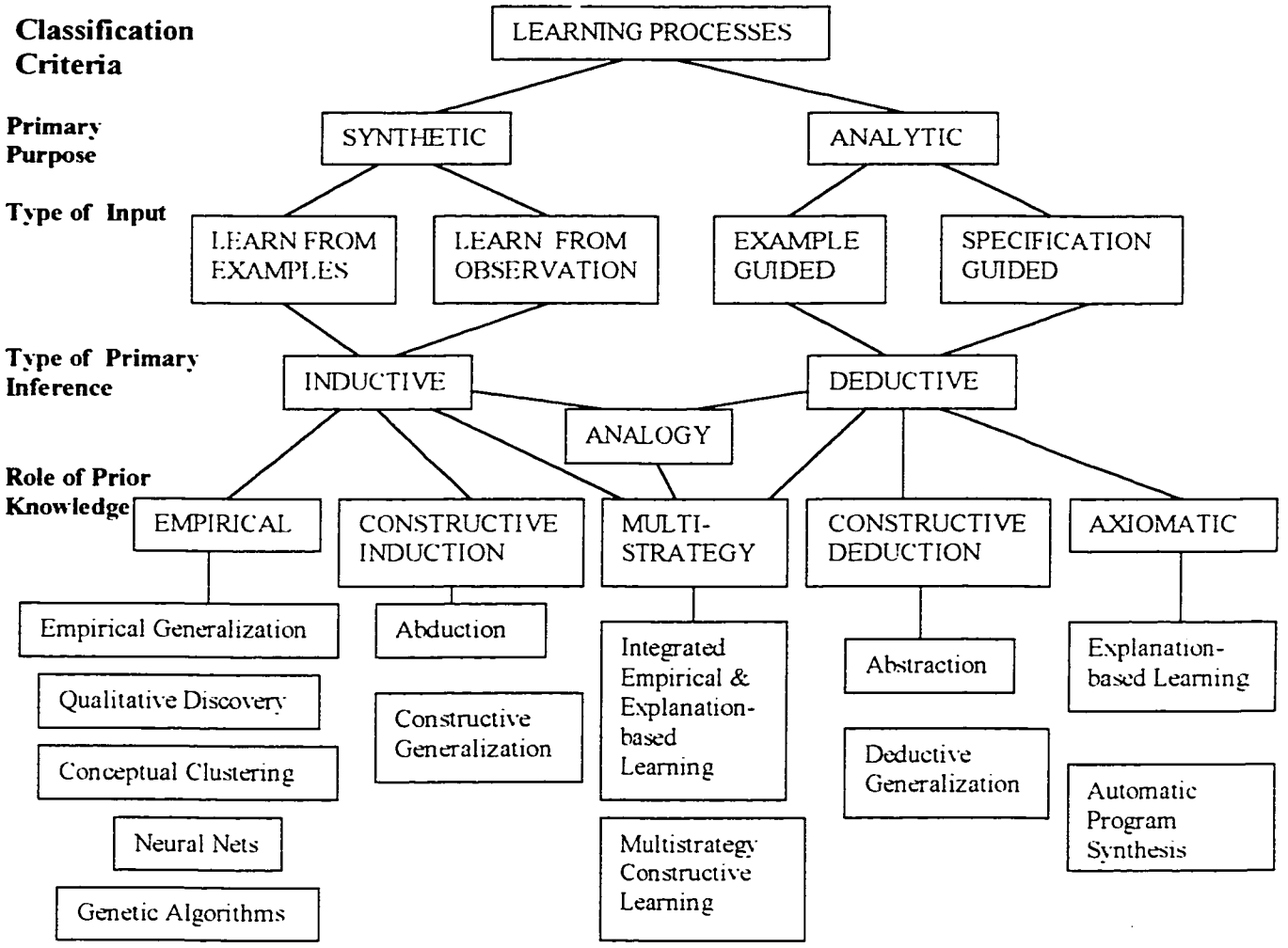

Fig.2.2.1. A multicriteria classification of machine learning methods [Kod90]

Learning processes can be classified into synthetic and analytic on the basis of their main goal [Kod90]. Synthetic learning aims primarily at creating new or better knowledge, and analytic learning aims at reformulating given knowledge

into a better form. Synthetic learning employs induction as the primary inference, while analytic learning employs deduction.

Induction is a process of hypothesizing premises that entail given consequents, while deduction is a derivation of consequents from given premises. The following relationship states that P and BK entails C:

$$P \,\&\, BK \succ C \qquad (1)$$

where P is a premise, BK is the reasoner's background knowledge, and C is a consequent. If the reasoner observes C, then it may hypothesize P, by entailing (explain or generalize) the observation. This is an inductive inference. Its type depends on the nature and the role of background knowledge [Kod90].

From the point of view of machine learning (artificial intelligence), secondary structure prediction is an instance of inductive learning, generalizing from known examples to solve new cases. Different algorithms may work according to different principles and can generalize in different ways: a concept description can be matched with examples in a simple and direct way, or can employ a substantial amount of background knowledge and inference [Kod90]. The matching

procedures are quite sophisticated in order to allow the system to recognize new examples.

2.3. Training and testing

We will approach this chapter only from the point of view of exemplar-based learning, since this is the method we are developing and because different learning processes have completely different particulars. We will illustrate with examples from our domain, protein secondary structure prediction.

The training phase consists of learning a set of classified examples (i.e. of proteins with known secondary structures, in our case) which form the training set.

The testing phase consists of finding the class to which a new example belongs, using the learned examples, memorized during the training phase. In protein secondary structure prediction terms, testing means finding the type of the secondary structures of a new protein (see Fig.2.4.b.).

2.3.1. Criteria for Building the Training and Testing Sets

The training exemplars must not have very close similarities with the data in the testing set. This guarantees that the quality of results is not due to such similarities. We called this independence between learning and testing sets, **orthogonality.**

At the same time, the testing set must contain samples from the whole spectrum of the domain. We called this quality **coverage** and it guarantees that the result accuracy is very likely to be the same for most randomly chosen examples to be tested.

2.3.2. Approaches

Several approaches are possible. Each of them can not be judged without considering the orthogonality principle, which was introduced above and which is also developed in the chapter dedicated to the Data Set.

In the **standard** approach, ideally, the training and testing sets are disjoint. If the training set is $L = \{l_1, l_2 \ldots l_m\}$ and the testing set is $T = \{t_1, t_2 \ldots t_n\}$, then the

number of tests to be done will be the cardinality of T (i.e. n). There is a single training set.

The **jack-knife** approach builds more training sets: if the data set is $L = \{l_1, l_2 \ldots l_n\}$, one protein at a time is selected as test set: $T_i = \{l_i\}$ while the remaining proteins are used as training set: $L_i = L - T_i$. There are m training sets and m tests.

The **bulk** version of **jack-knife** (also called k-way cross validation) randomly partitions the data set $L = \{l_1, l_2 \ldots l_n\}$ into k subsets $L_1, L_2, \ldots L_k$ of equal size; the training set is $L - L_i$ and the test set is L_i. There are k training sets and m tests.

In a **blind test**, test data is previously unclassified. The real secondary structures of a test protein being unknown, the results can only be compared to those obtained by using other methods, on the same test protein. One can compare the results of our method with the results obtained by Combine [Bio88], GORIII [Gib87] or PEBLS [Cos93], for example.

2.3.3. Measures of Performance

Different measures of performance are possible. For exemplar-based reasoning, the most obvious metric is the percentage of correctly classified instances. In our case, since secondary structures - which must be predicted - are not points, but rather segmerts, a more precise performance metric is needed: we must also determine to which degree a secondary structure is correctly predicted, that is, which percentage of its entire length is discovered.

Hence, our metric is concerned with *efficiency* rather than correcteness, because we deal with the percentage of correctly classified instances. In a classification domain, the predictive accuracy of a learning algorithm depends on predicting properties of *unknown* instances and not on summarizing aspects of already processed instances.

2.3.4. Performance in Classification

Fig.2.3.4 presents the learning curve of a classification process, NTGrowth, a variant of the nearest neighbor algorithm which selectively retains only some instances in memory. This classification method was evaluated by Aha and Kibler [Aha91]. NTGrowth bases its decision on whether to save an instance on that instance's predictive accuracy and its contribution to overall accuracy. The test domain consists of 300 cardiology cases containing 139 positive instances of heart disease. Each case has an associated class and is described by 13 numeric attributes. The data set was roughly divided in two, one half being used for training and the other for testing.

Kibler and Langely [Kib90] conclude from the experiment that most learning occurs early, after the algorithm has processed a moderate number of training instances, say 25. A slight increase in the classification accuracy occurs with additional instances added to the training set. The system reaches its asymptotic performance just before 50 instances in the training set. Thus, the learning curve shows the benefit of inspecting new training instances and estimates wether the improved predictive accuracy is worth the cost.

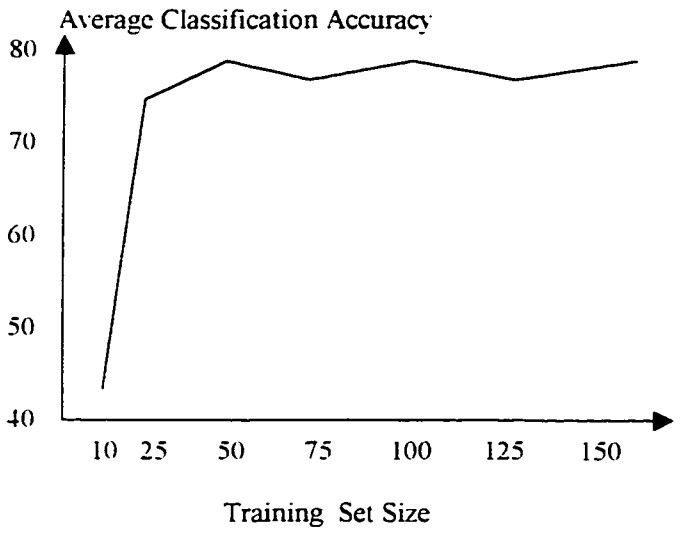

Fig. 2.3.4. The learning curve

A training set which is enhanced with new instances can increase the performance of a method, within certain limits. Gibrat & Garnier showed that improvements of the order of 7% over the original method ([Gar87]) could be obtained by increasing the number of proteins stored in the database [Gib87]. This is natural, since a rich training set is able to offer more useful information than a less complete set. Most authors used data sets around 50 proteins or even above 100 (107-[Zha91]; 128-[Cos93]; 130-[Ros93]; 110-[Yi93]). Yi and Lander obtained best results for a data set of 50 to 80 proteins, while the predictive accuracy for 100 exemplars was as low as for 20.

2.4. Machine Learning and Protein Structures

In the field of artificial intelligence (AI), secondary structure prediction is a typical classification problem: based on the sequence of amino acids of a protein (its features), one tries to predict the secondary structure of the protein (its class) - see Fig. 2.4.a.

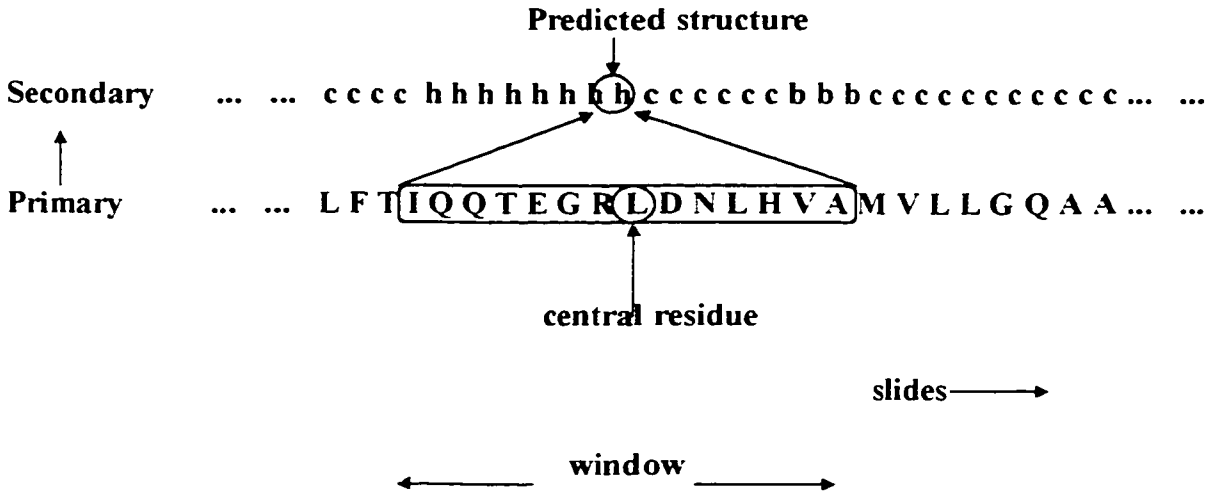

Fig.2.4.a. Secondary structure prediction uses the primary structure of the protein

A window is moved along an amino acid sequence to extract correlations between the residues and the secondary structure state of the central residue.

A machine learning approach uses a training set to learn its known structures and then classifies an unknown protein based on the similarities it finds with the training set (see Fig.2.4.b).

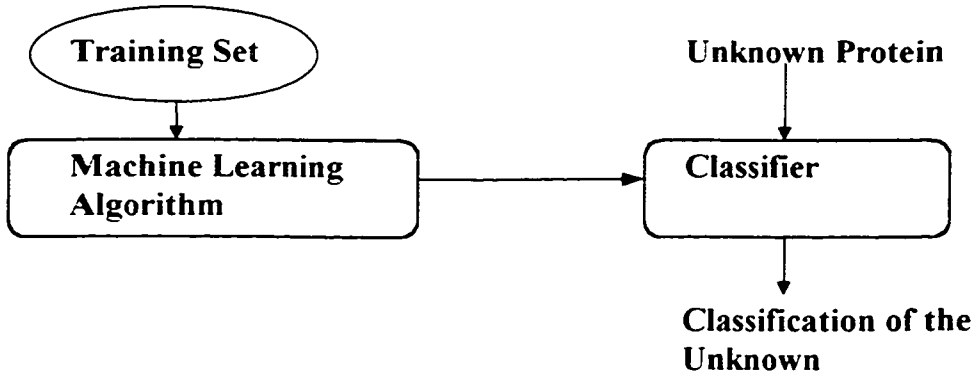

Fig.2.4.b. The Machine Learning Approach

Traditional classification methods have been used to predict protein secondary structure:

- statistical methods - [Nag73]; [Cho74]; [Nag75]; [Gar78]; [Sch79]; [Lev86]; [Gib87]; [Bio88]; [Kan88]; [Lev88]; [Fas89]; [Gar91]; [Sto92]; [Mug92]; [Cos93]; [Yi93].
- neural networks - [Qia88]; [Boh88]; [Boh90]; [Hol89]; [Bos90]; [Kne90]; [Hir92]; [Mac93]; [Sto92]; [Zha92], [Ros93].
- pattern-matching - [Coh83], [Coh86]; [Tay83]; [Roo91]; [Roo89]; [Room91]; [Ste90]; [Pre92].

- evolutionary conservation (sequence alignment) - [Max79]; [Zve87]; [Fra89]; [Ben90]; [Bar91]; [Nie91]; [Ouz91]; [Mus92]; [Rus92]; [Gib93].
- induction - [Kin90].

All these methods achieve aproximately the same accuracy level. This is due to the fact that these methods are obliged, because of the complexity of the problem, to neglect the multitude of properties of amino acids and their combinations. The formation of secondary structures is only to a certain degree due to local interaction of amino acids ([Nag75]; [Tay88]; [Zho92]). Physical and chemical properties of amino acids and bonds can be described by the ALOHA system [Pin90] which is mainly conceived for knowledge acquisition by learning. King & Sternberg [Kin90] classify amino acids based on their properties and use induction to predict the secondary structures.

During 1983-1993 these methods reached from 60 to 64% in overall three-state accuracy [Ros93]. Some methods had poor results in predicting certain secondary structures, such as 45% for β-sheet ([Bio88]), which is not significantly (just 12%) above the chance value (33%).

The real power of these methods should be checked by having no significant similarities between test and training sets and by using severe validation techniques.

Some of the most enthusiastic results reached an overall acuracy of 66.4 % ([Zha92]) or between 63 and 65 % (64.3%: [Qia88]; 63.2%: [Hol89]; [Kne90]; [Sto92]). It is even claimed that predictions can not be better than 65 +/- 2 % ([Gar93]). Rost & Sander ([Ros93]) observed that 70.8% of the three states are predicted accurately.

The best results were obtained by authors using hybrid systems (see Fig.2.4.c.):

- multiple sequence alignments + neural networks: 70.8% [Ros93]
- multiple sequence alignments + nearest neighbor: 72.2% [Sal95]
- neural network + memory based reasoning + statistical: 66.4% [Zha92]

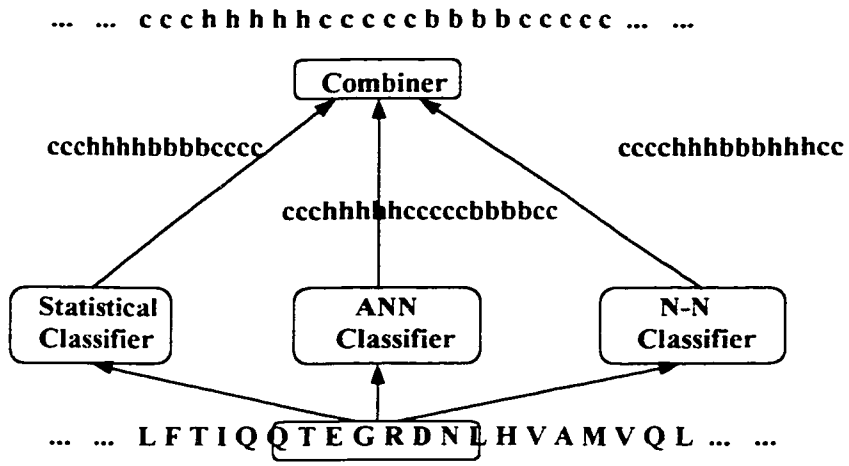

Fig.2.4.c. Hybrid System

Rigorous cross-validation improved the confidence in the accuracy of the results: Maclin [Mac93] reached on overall accuracy of 63.6% using multi-layered network approach, and Cost & Salzberg [Cos92], [Cos93] obtained 65.1%, while Zhang ([Zha92]) combining networks with other methods, reached 66.4%. Their test and training sets present sequential homologies. The Combine program ([Bio88]) reached 65.8%, SIMPA ([Lev88]) obtained 63.2%, GORIII ([Gib87]) achieved 62.4%, all three being reported to have had no significant similarities between test and training sets (no sequential homology). ALB ([Pti83]) reached 63.8%, also on sets without sequential homology. Rost ([Ros93]) who obtained the best results, using a two-layered feed-forward

neural network, also used a non-redundant database (of 130 proteins).

The accuracy of prediction of each secondary structure type is also an important factor: GORIII ([Gib87]) poorly predicted β strands (46%), while Rost & Sander ([Ros93]) reached 65.4% for the same secondary structure type.

AI methods that use a database of known examples to classify the test instance, known as nearest-neighbor systems, achieved good results in different domains ([Aha91]).

2.4.1. Statistical Methods

These methods rely on probabilities derived from the training set. For example, for each secondary state S_i, given a window $(a_1,...a_n)$ of n residues, we might consider the conditional probability $p(S_i \mid a_1,...a_n)$ of S_i, given $(a_1,...a_n)$. The prediction will be the secondary state having the probability with the maximum value in the set:

$$prediction = \left\{ s_j \mid \max_j p(s_j \mid a_1 a_2 \ldots a_n) \right\}, \qquad s_j \in \{\alpha - helix, \beta - sheet, coil\}$$

Bayes Theorem could be used to compute the above probabilities:

$$p(s \mid a_1 a_2 \ldots a_n) = \frac{p(s)\, p(a_1 a_2 \ldots a_n \mid s)}{p(a_1 a_2 \ldots a_n)}$$

Zhang et al. [Zha92] eliminated third and higher order correlations, thus postulating:

$$p(a_1 a_2 \ldots a_n \mid s) \approx \prod_i p(a_i \mid s) \left[1 + C_f \sum_{i<k} f_{ik} \left(\frac{p(a_i a_k \mid s)}{p(a_i \mid s) p(a_k \mid s)} - 1 \right) \right]$$

where C_f (≈ 1.5) is a compensation factor and f_{ik} is the reliability.

$$f_{ik} = \sqrt{\frac{p(a_i \mid s) p(a_k \mid s)}{(1 - p(a_i \mid s))(1 - p(a_k \mid s))}}$$

But there is not currently enough protein structure data available to compute the frequencies of ($a_1, \ldots a_n$) in each state S_j. We need a simpler estimation, e.g. by using the Bahadur-Lazarsfeld expansion (see [Zha92]). Due to the limited sample size, third and higher order correlations are ignored.

Another approach that also uses statistical information is known as "exemplar-based reasoning" or "nearest neighbour

method" or "memory-based reasoning". The nearest-neighbor rule states that a test instance is classified according to the classifications of "the nearest" training examples from a database of known secondary structures. In order to compute the distance between the test instance and the training examples, the relative frequency of occurrence of amino acids in secondary structures is used. More details are provided in Chapter 3.

2.4.2. Neural Networks

Artificial neural networks have been used in many applications, including protein seondary structure prediction ([Qia88]; [Kne90]). An artificial neural network usually consists of a large number of simple processing units, called basic perceptrons - see Fig.2.4.2.a, connected by weighted links. The output is computed by each unit by applying an activation function to each input.

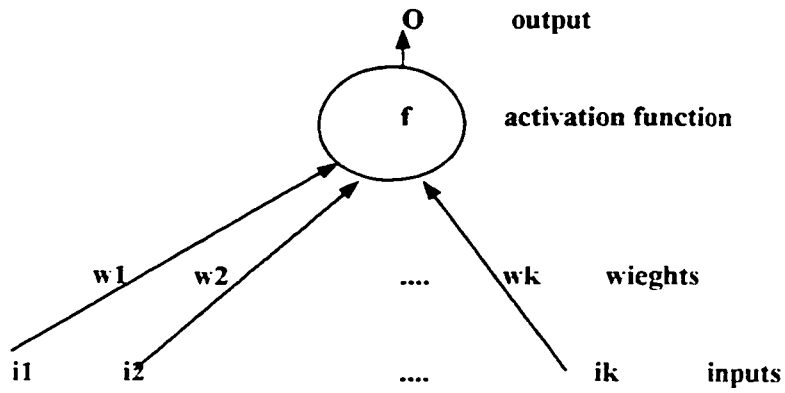

Fig.2.4.2.a. Basic perceptron

A layered neural network contains an input layer, an output layer and one or more "hidden layers" placed between the input and the output layers - see Fig.2.4.2.b.

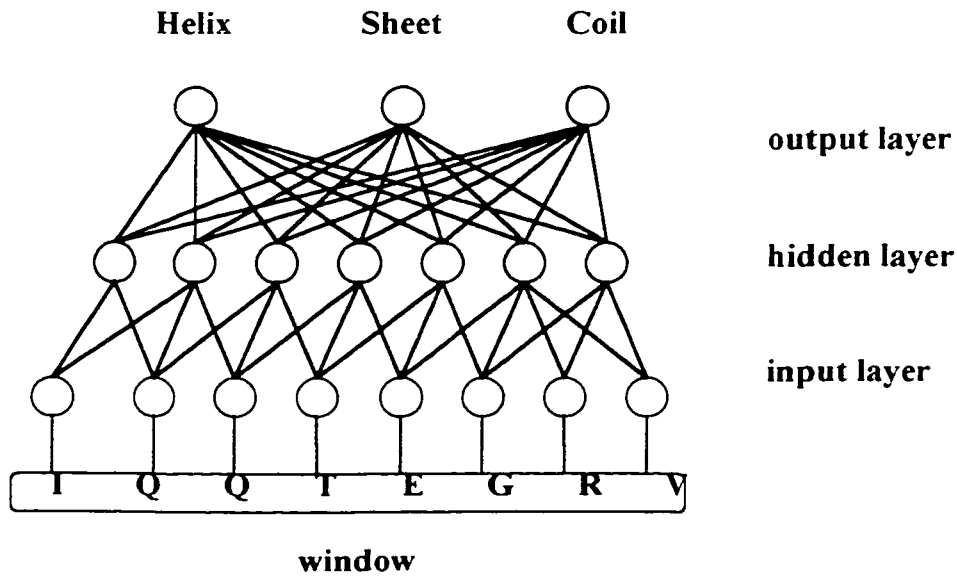

Fig.2.4.2.b. A layered neural network

A feed-forward network computes its output in the following fashion: the input pattern sets the input layer; the outputs of this layer, computed by the activation function will be applied to the input of the next hidden layer; thus, one layer at a time, from the input to the hidden to the output layer, the units compute their outputs by applying the activation function to the weighted sum of the outputs from the units of the lower layer. The weights are associated to the links between units. Often, the sigmoid function is used as activation function:

$$O(i, j) = \frac{1}{1 + e^{-x}}$$

O being the output of unit j at layer i; x is the weighted sum of outputs from nodes (units) at one layer below:

$$x = \sum_{l=1}^{k} w_l i_l$$

A complex system for protein structure classification may contain several neural networks, the results of which are used by a combiner to compute the output, i.e. the secondary structure (see Fig.2.4.2.c).

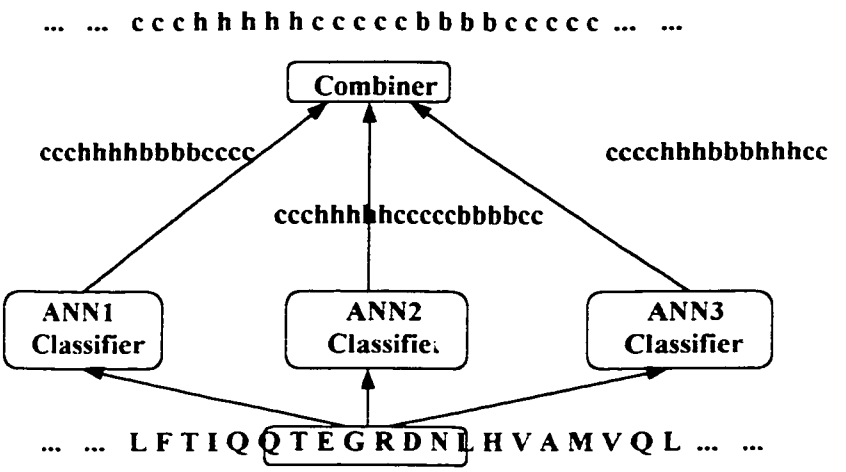

Fig.2.4.2.c. A multiple neural network.

The back-propagation algorithm trains a layered network by adjusting the link weights of the net, using a set of training examples.

The major limitation of neural networks (as well as of genetic algorithms) is the difficulty of introducing large amounts of domain specific knowledge to them and explicitly exploiting that knowledge or any feedback information in the learning process. To illustrate, let us suppose that a neural network incorrectly classifies some examples. To correct the mistake, the system modifies its knowledge representation by stepwise corrections, rather than by an explicit analysis of the reasons for the mistake. This might explain why such systems tend to exhibit relatively slow rates of learning. Another weakness is the lack of

transparency of the results of learning. The knowledge acquired by neural networks is not in the form that people can easily understand. The comprehensibility principle has not been viewed as a major issue in implementing such systems. For that reason, they are called *subsymbolic learning systems* - see [Kod90].

2.4.3. Pattern Matching and Induction

A systematic and comprehensive analysis of the relations between amino acid sequence and secondary structure has shown ([Roo88]) that short amino acid sequence patterns that predict secondary structure with high level of accuracy can be found. The ability of finding such patterns is limited by the size of the available databases, since the frequency of sequence patterns is generally too low to produce enough non-random sequence-structure relations.

The databases are scanned for simple amino acid patterns of the type Gly-X-Ala-X-X-Val, e.g., where X denotes any residue. An association between structural families and patterns in amino acid sequence is also searched.

King & Sternberg [Kin90] overcame the limitation of only being able to specify 3 residues in a pattern (due to

the low frequency of occurence of larger patterns) by grouping residues into classes. These classes are then used to specify patterns. Amino acids can be grouped into the following classes, according to their physico-chemical properties: hydrophobic, hydrophilic, polar, small, positive, charged, negative, small or polar, charged or hydrophilic, charged minus h, large minus aliphatic, etc .

This scheme allows more complex patterns to be described as each class specifies several residues; for example:

hydrophobic = C = {h, w, y, f, m, l, i, v, c, a, g, t, k}

positive = D = {h, k, r}

negative = E = {d, e}

charged = F = {d, e, r, k, h}.

The patterns can be expressed in terms of such classes and in terms of their position relative to the central residue:

[C D E F] is a sequence of classes as defined above. King & Sternberg [Kin90] use symbolic induction to produce rules that are meaningful in terms of chemical properties of the residues. The rules are in the form:

'if the sequence of classes [C D E] ocurs, then they are all in the secondary structure type S'

That is, a sequence of residues [w x y] occurs where the 1st position residue w belongs to the class C, where the 2nd position residue x belongs to the class D, where the 3rd position residue y belongs to the class E.

For example, the rule *'[positive negative charged]-> Helix'* means that the primary sequence [h d r] is predicted to have [Helix Helix Helix] as a corresponding secondary structure. This type of rule used by [Kin90] resembles that used by [Coh83]. An implicit assumption of this form of rule, is that what is important in forming secondary structure from primary structure is not the particular residues at each position, but some specific chemical property or combination of properties.

The advantage of the pattern representation is that it is a hint to the rules based on chemical properties that govern protein folding.

2.4.4. Evolutionary Conservation

Other authors developed algorithms based on the secondary structure propensities for aligned residues and on the observation that insertions and high sequence variability tend to occur in loop regions between secondary structures ([Zve87], [Gib93], etc.)- see Fig.2.4.4.a.

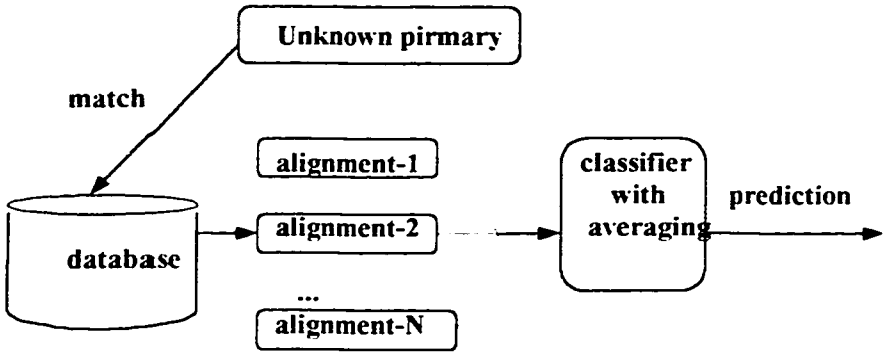

Fig.2.4.4.a. Combining with alignment information

The first step is to obtain the alignments. A standard method is the dynamic programming approach of Needleman & Wunsch (1970).

There is, however, more information about secondary structure available from aligned sequences than that obtained by averaging the residue propensities. The crystal structures of protein families show that sequence insertion and sections of high sequence variability occur in loop regions between the secondary structures. In addition, residues involved in secondary structure packing tend to be hydrophobic.

A sequence relationship is searched between the test sequence and the learned sequences, by computing a matrix of similarity for the compared sequences: amino acids with

similar physico-chemical properties in the corresponding positions of the two sequences will have greater weights, while amino acids with different properties will have lower weights. Based on a Venn diagram representation of the chemical properties of the amino acids (see Fig.2.4.4.b.), one can quantify the extent of sequence conservation at any position i along the chain by a conservation number. The aim is to have a high conservation number when similar chemical types of amino acids occur at a position, and low value when there is a high variability of corresponding residues or an insertion. The prediction of a secondary structure is penalized where there is a low conservation number.

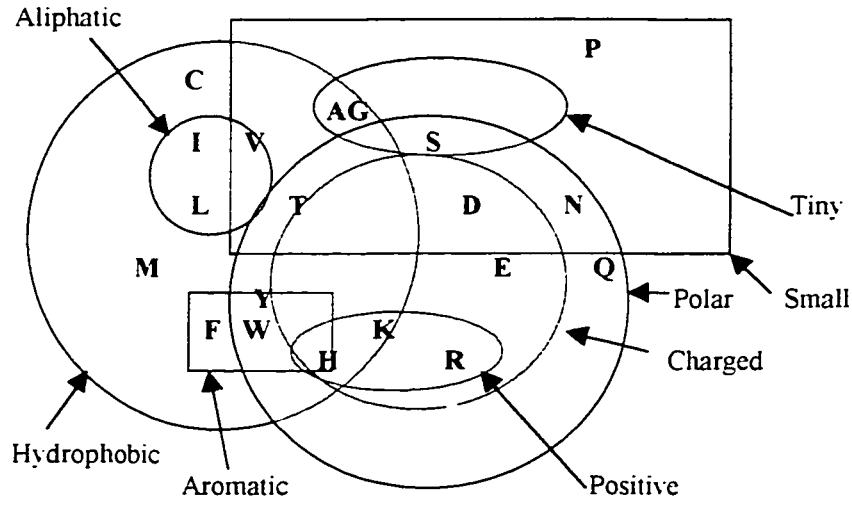

Fig.2.4.4.b. Venn diagram representation of the chemical properties of amino acids [Pin92]

Chapter 3. THE NEAREST-NEIGHBOR ALGORITHM

3.1. Problem Description

Given the amino acid sequence (the primary structure) of a protein, we wish to predict its secondary structure. The experimental analysis of the amino acid sequence of a protein is a much simpler task than the experimental determination of its tertiary structure. An intermediate step between the primary and the tertiary structure is the prediction of the secondary structure. This intermediate step is valuable because the prediction of tertiary structure is still a computationally intensive task.

3.2. Method Description - Nearest-Neighbor

Nearest-neighbor classifiers can be used to predict the secondary structure of proteins with good results. The nearest-neighbor rule states that a test instance is classified according to the classifications of "nearby" training examples. The training examples are taken from a

database of proteins with known secondary structures. This method uses a distance metric to compute how far the test instance is from the learned structures.

3.2.1. Basic Principles

An instance-based algorithm stores a series of training instances in its memory and uses a distance metric to compare new instances to those stored. New instances are classified according to the closest exemplar from memory.

In 1986, Stanfill and Waltz presented a powerful method for measuring the distance between values of features in domains with symbolic feature values. They applied their technique to the English pronunciation problem with impressive initial results ([Sta86]). Their Value Difference Metric (**VDM**) takes into account the overall similarity of classification of all instances for each possible value of each feature. Being symbolic features, their posssible values are in a finite domain. Their metric is represented by a matrix that is derived statistically by defining the distance between all values of a feature.

In our case, the features are the secondary structure types at a certain sequence position and the possible values

are the amino acid symbols. The distance δ between two amino acids V_1, V_2 for a specific feature is defined as:

$$\delta(V_1, V_2) = \sum_{i=1}^{n} \left| \frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right|^k \qquad (1)$$

The distance between the values is a sum over all n classes; for the protein data, there might be 3 classes (Helix, Sheet, Turn) or 4 classes (Helix, Sheet, Turn, Coil). The matrix entry C_{1i} is the number of times V_1 was classified into category i, and C_1 is the total number of times value V_1 occurred. The constant k is set to 1/2, 1, or 2, depending on the kind of metric desired: for k=1, it is a Manhattan distance; for k=2, it is an Euclidean distance.

The idea of this metric is to compute a matrix of value differences for each feature in the input data in the following way: values are similar if they occur with the same relative frequency for all classifications. The term C_{1i}/C_1 represents the likelihood that the central residue will be classified as class i given that the feature in question has the value V_1. Two values are similar if they give similar likelihoods for all classifications. Equation (1) finds overall similarity between two values by computing

the sum of differences of these likelihoods over all classifications.

We constuct a separate value difference matrix for each feature, by counting the number of occurrences of each value for each class.

Equation (1) defines a geometric distance on a fixed, finite set of values. It is a metric. That is, it has the properties that a value has a distance zero to itself; it has a positive distance to all the other values; the distances are symmetric; and the distances obey the laws of triangle inequality:

i. $\delta(a, b) > 0, \quad a \neq b$

ii. $\delta(a, b) = \delta(b, a)$

iii. $\delta(a, a) = 0$

iv. $\delta(a, b) + \delta(b, c) \geq \delta(a, c)$

Stanfill and Waltz's VDM also used a weighted term w_i which makes their total distance metric Δ non-symmetric: $\Delta(X, Y) \neq \Delta(Y, X)$. The tctal distance Δ between two instances is given by:

$$\Delta(X,Y) = w_X w_Y \sum_{i=1}^{N} \delta(x_i, y_i)^r \qquad (2)$$

where X and Y represent two windows, $X=(x_1..x_n)$ and $Y=(y_1..y_n)$ for the protein folding domain, and where Y is a learned exemplar and X is a new example. The variables x_i and y_i are the values for the ith feature for X and Y, each example having N features. Weights w_X and w_Y are assigned to exemplars. In our method, these weights are always 1.

In the classical approach, a fixed sized window is used to identify the class (secondary structure) of each residue in the test sequence (see Fig.3.2.1.a.). At the end of this process, a post processing-algorithm based on the minimal sequence length restrictions is used to decide if the residues of the test sequence belong to the found class.

Cost and Salzberg [Cos93] used the minimal sequence length restrictions of Holley and Karplus (1989): the β-sheet must consist of a contiguous sequence of no fewer than two such residues, and an α-helix must consist of a contiguous sequence of no fewer than four residues. If the residues do not conform to these restrictions, they are re-classified as coil. Qian and Sejnowski [Qia88] used another type of post-processing, called "cascaded neural net": the

output of a lower-level network was fed into the input of an upper-level network which re-classifies some residues.

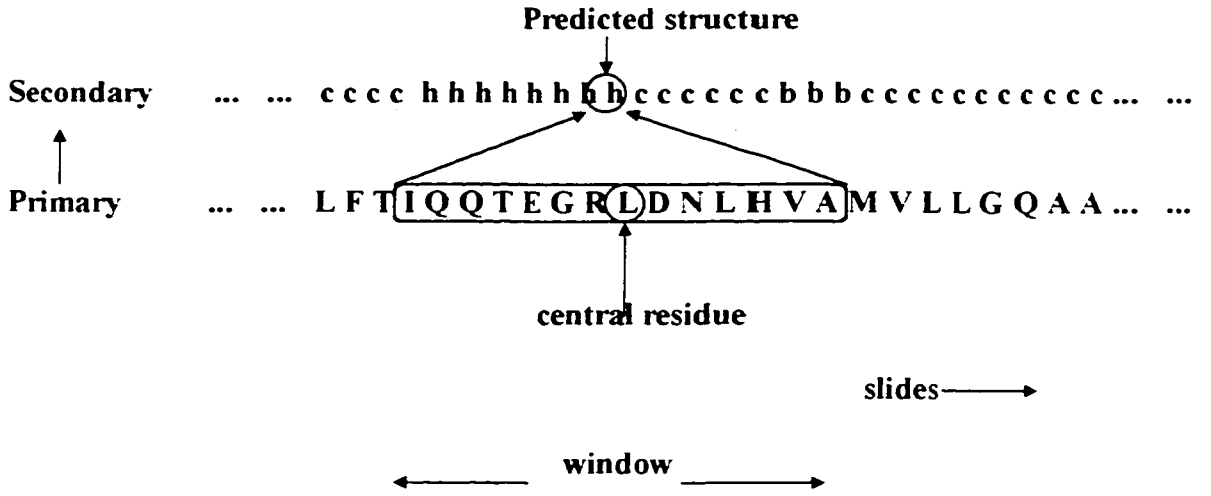

Fig.3.2.1.a.Classical approach

No nearest-neighbour method can predict β-turns, because these structures are very short (3..4 amino acids) and the patterns which build them are found with equal frequency in α-helices and β-sheets.

3.2.2. Modifying Parameters

Different parameters must be adjusted in order to maximize the prediction accuracy. This task involves a lot of preliminary tests. Some of the parameters are pre-established before the start of the required test (e.g the training set, the window size). Some others are adjusted as the test proceeds (e.g. the weights). Not all of these parameters must be used: some methods may not use certain parameters (e.g. our method does not use weights, nor pre-fixed window size).

3.2.2.1. Training set

Both the number and the content of the training examples strongly influence the quality of the prediction, because the amino acid distribution statistics rely on these factors. Ideally, the training set should reflect the distribution of amino acids in secondary structures as they occur in real life. If the training set has an amino acid distribution in secondary structures different from that of the real life, then, statistically, the prediction will be deformed according to the particular distribution of the training set. For example, let us say some of the amino

acids clearly favoring the α-helix in real life - ALA, LEU, GLU - have relative frequencies of occurence in protein secondary structures that favor β-sheet in our training set (due to more numerous β-sheet secondary structures in the training set, containing the three amino acids: ALA, LEU, GLU). In this case, a window in a test protein, containing (ALA, LEU, GLU, XXX) - where XXX is any amino acid - may be estimated as a β-sheet instead of an α-helix.

Even adding proteins containing only turns and coils to our training set will influence the prediction of α-helix and β-sheet, because the relative frequency of occurence of amino acids in α-helix and β-sheet will change.

3.2.2.2. Window size

The window is a fixed length sequence of residues from a protein chain for which we must classify the central residue in the window as α-helix, β-sheet or coil (see Fig. 3.2.2.2.).

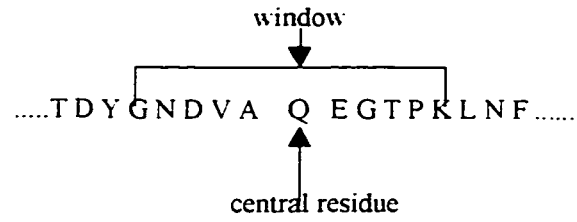

Fig.3.2.2.2. The central residue and the window

The window size influences the accuracy of the results. Qian and Sejnowski [Qia88] found the optimal window size to be approximately 17 residues, while Cost and Salzberg [Cos93] found that best results were obtained with window sizes of 17 and 19. Yi and Lander [Yi93] determined that their nearest-neighbor method operated best with a window size of 19, but good results were also achieved with window sizes as large as n = 25 or n = 41. Zhang [Zha92] used a window length of 13 for a hybrid system including a nearest-neighbor subsystem. Salamov and Solovyev [Sal95] used a window length of 19. Rooman [Roo90] used patterns of most 7 residues for their rule-based systems.

Our window is variable, that is it dynamically expends itself at run-time (starting at a size of 4), trying to maximize the length of the predicted secondary structure. This expansion occurs as long as the distance between the two windows (the learned one and the tested one) is below a threshold value.

3.2.2.3. Exponents

Stanfill and Waltz [Sta86] used the value k = 2 in their version of equation (1), while Cost and Salzberg [Cos93] equally obtained good performance for k = 1 and r = 1. Salamov and Solovyev [Sal95] used k = 2 and r = 2, while Zhang [Zha92] used k = 1 and r = 1, but with a metric considering the second degree conditional probabilities too (the conditional probability of a secondary structure given both the current amino acid and its neighbor occur). In general, exponents of value 2 improve the precision. If the improvement is not significant, exponents equal to 1 are prefered for simplicity.

3.2.2.4. Weights

The exemplars which are used more frequently to classify new protein sequences are called 'reliable', while those exemplars used less frequently are called 'unreliable'.

In order to increase the influence of 'reliable' exemplars relative to 'unreliable' exemplars, each instance in the training set receives a weight: reliable exemplars are given smaller weights ($w_x \approx 1$), making them appear closer to a test example. A weight w_x is the ratio of the

number of uses of an exemplar to the number of correct uses of the exemplar. If the exemplar is accurate, it will have a weight $w_i \approx 1$, while unreliable exemplars will get a weight $w_i > 1$.

Exceptions are thus seen quite far, while noise may even be eliminated. Reliable exemplars remain the 'rule' for examining the test instances. Without this capability, more instances would be required for learning, according to Cost and Salzberg [Cos93]. Yi and Lander [Yi93] also used this principle, known in the artificial intelligence literature as the Alien Identification Rule. We did not use weights in our approach.

3.3. Method Description - Multi-Level Nearest-Neighbor

3.3.1. Differences from other Similar Approaches

There are three main differences between our method and those we encountered in our research, which we will describe starting with low-level equation details and ending with high-level sterategy approaches

- First, our equations present some differences compared with the Stanfill-Waltz or other approaches: there are no weights and no distance coefficients (k = 1, r = 1); the distance between two amino-acids only considers the searched secondary structure type.
- Second, our window is variable, that is it dynamically expands itself at run-time (starting at a size of 4), trying to maximize the length of the predicted secondary structure. This expansion occurs as long as the distance between the two windows (the learned one and the tested one) is below a threshold value. At the same time, a *whole window* in the test sequence must be similar (that is quite close) to a subwindow of a learned secondary structure, in order to be considered as a found secondary structure. More

such *whole windows* will form a longer found secondary structure if they overlap or are adjacent. From this point of view, the classical approach might be seen as an extreme case, where the *preliminary found window* may have a length of one (the central residue) and more such *preliminary found windows* may form a *found window*. In our approach, the *preliminary found window* may not be shorter than four (the value of HELIX_WIN_SIZE, SHEET_WIN_SIZE) residues.

- Third, we predict α-helix first and then β-sheet (this order can be reversed, since the two predictions are independent), solving the eventual conflicts (amino acids found in both classes) based on the computed distances associated to the found secondary structures.
- Finally, there is an attempt to predict the group of classes to which the protein belongs, by using statistical information on the distribution of the α-helix, β-sheet and β-turn preferring amino acids. This is also useful in building the training set.

The distance is computed between the window of the new example *Y* and a window of a learned exemplar *X;* if the distance is below a threshold value, the windows are

increased and the distance is computed again. As long as the distance is below the threshold value, we decide that a secondary structure is found in the test sequence (having the same type as the matching window of the learned secondary structure and the same length) and we continue the expansion process of the window. All windows of the learned exemplars X_i are used in the scanning process of finding the nearest neighbours. The final secondary structure might be close to more than one subwindows of the exemplars X_i.

3.3.2. Implementation Details

We will describe the implementation details of our approach, comparing them with the classical nearest-neighbor method.

3.3.2.1. Distance Between Amino Acids

Our approach is based on a modified nearest-neighbour method. Our equations are based on those of Stanfill and Waltz, but having several simplifications. The distance δ between two amino acids V_1, V_2 (i.e. between two values) for a specific feature is defined as:

$$\delta(V_1, V_2) = \left| \frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2} \right| \qquad (3)$$

That is, we use a symmetric Manhattan metric, where i is the secondary structure type for which we are computing the distance. Since our approach is centered more on searching a certain class type (e.g.α-helix), we considered that it was not significant if the two amino acids have different distributions for the other classes (β-sheet, β-turn).

3.3.2.2. Threshold Value

The nearest-neighbor rule states that a test instance is classified according to the classifications of "nearby" training examples. In our approach, there is a supplementary condition: "nearby" means that the distance between the test instance and the training example is (equal or) smaller than a **threshold value.**

By increasing the threshold value, we admit farther training examples to be taken into consideration in the classification process. If the threshold value is too high, too many examples are considered as "nearby" and we have an overprediction.

By diminishing the threshold value, we consider only the nearest examples in the classification process. If the threshold value is too low (but there are still enough training examples in our training set), an underprediction will result.

3.3.2.3. Distance Between Two Windows

The total basic distance Δ between two instances $X = (x_1..x_n)$ and $Y = (y_1..y_n)$ is given by:

$$\Delta(X,Y) = \sum_{i=1}^{N} \delta(x_i, y_i) \qquad (4)$$

Our window is variable: it dynamically expands itself at run-time (starting at a size of 4), trying to maximize the length of the predicted secondary structure. This expansion occurs as long as the distance between the two windows (the learned one and the tested one) is below a threshold value.

In our approach, an initial window win$_i$ (see Fig.3.3.2.3.a.) of HELIX_WIN_SIZE or SHEET_WIN_SIZE residues (established to a value of 4) of the test sequence is compared with a window of the same size of a learned secondary structure. If the distance between the two windows is below the threshold value, we consider that a secondary structure is found in the test sequence, in the position of the window win$_i$. This window is then expanded (both for the learned secondary structure and for the test sequence) and the comparison is made on the increased windows (win$_{i+1}$). If the distance is still below the threshold value, the window expansion process continues. Otherwise, it ends. All our distances are normalized: the distance between two windows is divided by the window length.

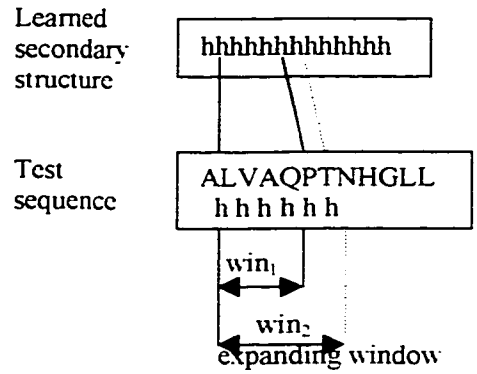

Fig.3.3.2.3.a. Our approach

The expansion process can also occur due to more subwindows win_i from different learned structures (see Fig.3.3.2.3.b.). More subwindows of the learned protein secondary structures of the same type (e.g. α-helix) each contribute to the decision that the window of the test protein belongs to their type. For each subwindow we use the closest distances (computed with equation 4) to the training set, which are together concluding the result: in Fig.3.3.2.3.b., three windows of different learned secondary structures (α-$helix_i$, marked $h_ih_ih_ih_i$..., where i=1..3) are the closest to one of the subwindows of the discovered secondary structure.

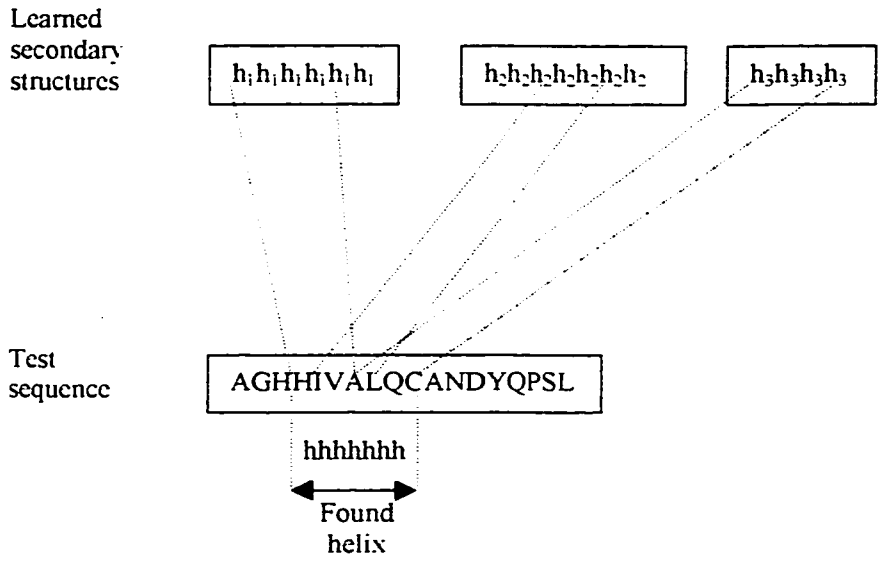

Fig.3.3.2.3.b. Finding secondary structures

The distance between the found secondary structure (of the test sequence) and the training set is:

$$\Delta(X,\Lambda) = \sum_{j=1}^{n} \sum_{i=1}^{lengthWindow\ j} \delta(x_i, y_{ijl}) \qquad (5)$$

where X is the new example, Λ is the training set with a cardinality of L: $\Lambda = \{Y_1, \ldots, Y_L\}$; x_i and y_{ijl} are the values for the ith feature (amino acid) for X and Y; j is the subwindow index; l is the training set index; n is the number of subwindows; '*lengthWindow j*' is the length of the j-th window; and N is the number of features being considered (the window size):

$$N = \sum_{j=1}^{n} lengthWindow_j \qquad (6)$$

Both n and $lengthWindow_j$ are variables, not constants; they are dynamically computed during the program execution. The program is trying to minimize the distance and to maximize the window length of the found secondary structure.

All learned secondary structures are used during this process and the scanning continues inside each such secondary structure (Fig. 3.3.2.3.c.):

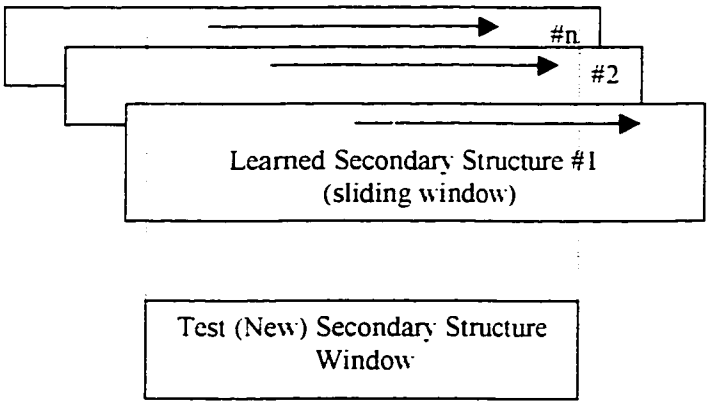

Fig.3.3.2.3.c. The scanning process

In terms of pseudo-code, the function which detects secondary structures is as follows:

```
for each position of the whole sequence of the Test Protein
    for all learned secondary structures of all Learned
                                                Proteins
        distance = ComputeDistance;
        if (distance <= smallestDistance)
            smallestDistance = distance;
    if smallestDistance > thresholdValue // not a
                         // secondary structure (any more)
        if it was already a secondary structure
            end it;
    else                 // it is a secondary structure
        if it is a new secondary structure
            initialize it;
        else             // it is an old one
            expand it; // increase its size
```

There are some consequences of our approach:

- a whole secondary structure (window) is found from the first stage without post-processing;
- smaller granularity smilarities can be found (resulting in smaller distances between the test protein window and the training set); and
- overprediction may occur due to the smaller found distances, which may lead to false similarities. This effect can be counter-balanced by a higher threshold value, especially in the case of β-sheet.

3.3.2.4. Degree of Confidence. Conflict Resolution

The distance between a learned secondary structure and a test window having the same class type represents our degree of confidence in the learning process. If an amino acid in the test sequence is predicted as belonging to an α-helix in the first phase and also predicted as belonging to a β-sheet during the second phase of our program, then this conflict of contradictory prediction must be solved.

If two examples of different types - e.g. an α-helix and a β-sheet - are close to the test instance, but the α-helix is closer ($d_\alpha < d_\beta$), then we will conclude that the test instance is an α-helix, with a degree of confidence of

$$\frac{d_\beta}{d_\alpha + d_\beta}$$

3.3.2.5. Groups of Classes

At the highest, strategic level, we attempt to predict the group of classes the protein belongs to by using statistical information: the percentage of amino acids in the protein sequence having certain affinities. This decision should be improved by using specific biochemical

information (i.e. amino acids properties). In our opinion, this is an important direction to be considered in a multi-level approach. At the first stage, the general group of classes of the protein has to be discovered and at a second stage, the secondary structure of the protein is predicted. We concentrated our efforts on two large groups of protein classes: α and β.

As Fig.3.3.2.6. shows, 80% of the α proteins have α-preferring amino acids in a percentage greater than 44%, which clearly distinguishes them from the other groups of classes. The α group of protein classes also has a low percentage of β preferring amino acids (below 23%). The β group of protein classes overlaps with α/β and α+β groups: only 75% of the β proteins have β preferring amino acids in a percentage greater than 24%, while their α-preferring amino acids do not represent more than 38%. These statistics also suggest why α structures are easier to predict than β structures.

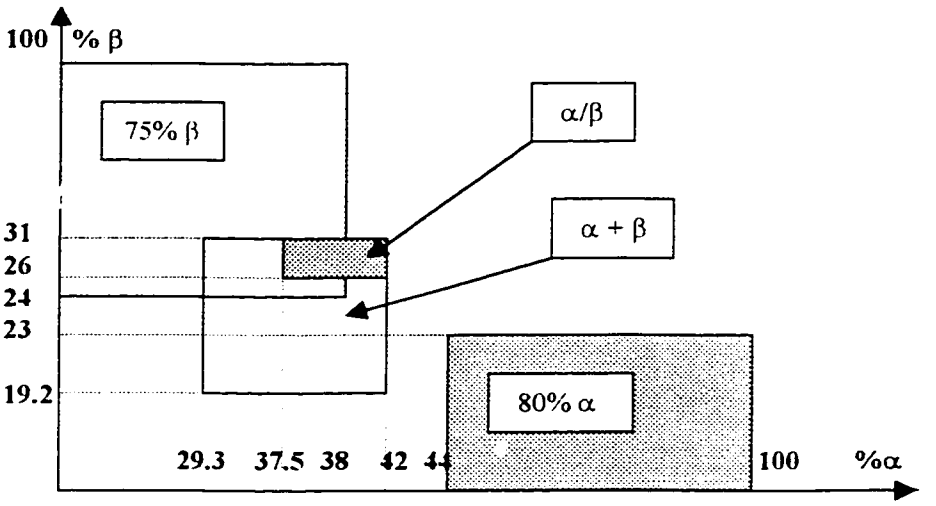

Fig.3.3.2.6. The distribution of α and β preferring amino acids inside the protein groups of classes (pure α, pure β, α+β, α/β)

These statistics are essential in constructing the training set, because they show the distribution of α and β preferring amino acids inside the protein groups of classes.

3.3.2.6. Parameter tuning

For each class (α-helix, β-sheet) we adopted a different threshold value: $ThresholdValue_{\alpha\text{-helix}}$, $ThresholdValue_{\beta\text{-sheet}}$. The threshold values were established during the program development and were not modified later on.

The training set was built using the statistics on the distribution of α and β preferring amino acids inside the protein groups of classes, as explained above. More details are provided in the next chapter.

The initial window size (win_i) was chosen to be the same for α-helix and β-sheet: HELIX_WIN_SIZE=4, SHEET_WIN_SIZE=4. It could be set to independent values for each class.

A finer tuning of the above parameters, based on more tests, is still possible.

Chapter 4. TRAINING, TESTING AND TEST RESULTS

Proteins in solution form globular structures, the net result of which is that residues which are sequentially very far from each other may be physically quite close, and have significant effects on each other. For this reason, secondary structures cannot be completely determined from primary structure, with the present methods. Qian and Sejnowski [Qia88] say that no method incorporating only local information can perform much better than current results in the 60%..70% range (for non-homologous proteins). There are also membrane proteins which have a different physical environment from water-soluble globular proteins and hence, different rules have to be learned to predict their structures.

Before performing training and testing, one must construct the learning and testing sets, according to the following principles:

- the training set must be well balanced, that is it should reflect the distribution of amino acids in secondary structures in the same ratio as in real life; and

- the training set and the test set must have a low degree of similarity.

The first principle will guarantee good predictions (within the limits of performance of the method) for an "average" protein. Rost and Sander [Ros93] also mention in their article the importance of "balanced prediction by balanced training", which is an elegant way to improve poor predicting performance.

The second principle will guarantee that good results are not due to similarities between learned proteins and test proteins. It is known that functional similarity and structural similarity are equivalent, that is one implies the other. Hence, we wish that learned and test proteins have different functionalities, in order to respect this principle.

4.1. Data Description (PDB files)

The Protein Data Bank (PDB) of the Brookhaven University is used by most of the biochemistry specialists involved in protein secondary structure prediction. PDB contains several thousand proteins with identified structures, which are updated as techniques evolve.

A PDB file is an ASCII labeled text file. The labels accompany each line and they identify the file sections:

- the HEADER line contains the protein name and the date
- the COMPND line
- the SOURCE line contains the biological medium
- the AUTHOR, REVDAT, JRNL lines contain the author name, the journal date and name where the protein structure was presented
- the REMARK lines contain different supplementary information
- the SEQRES lines contain the protein amino acid sequence
- the HELIX, SHEET, TURN lines contain the discovered secondary structures (using spectroscopy and crystallography):
  - the secondary structure type (the name of the label: e.g. TURN), an ordinal number of the secondary structure (H1, H2, H3,... for helices; S1, S2,... for sheets, T1, T2,... for turns; or TURN 1 for the first turn in the example below), the name and position of the first amino acid in the structure (e.g. VAL 13), the name and position of the last amino acid in the sequence (e.g. LYS 16).

- other lines (SSBOND, ORIGXn, SCALEn, etc) contain additional information on amino acid positions, etc.

The PDB code (4 letters and/or digits) is written on each line of a PDB file, as well as the line number. We only use the information contained in the HEADER, SEQRES, HELIX, SHEET, TURN lines; some other labels are used as delimiters. We present below the 1cse PDB file:

```
HEADER  COMPLEX(SERINE  PROTEINASE-INHIBITOR)        03-JUN-88       1CSE
REMARK 9  CORRECT E.C. CODE ON COMPND RECORD.  15-JAN-95.             1CSE
SEQRES 1 I 71  ACE THR GLU PHE GLY SER GLU LEU LYS SER PHE PRO GLU  1CSE
SEQRES 2 I 71  VAL VAL GLY LYS THR VAL ASP GLN ALA ARG GLU TYR PHE  1CSE
SEQRES 3 I 71  THR LEU HIS TYR PRO GLN TYR ASN VAL TYR PHE LEU PRO  1CSE
SEQRES 4 I 71 GLU GLY SER PRO VAL THR LEU ASP LEU ARG TYR ASN ARG  1CSEB
SEQRES 5 I 71  VAL ARG VAL PHE TYR ASN PRO GLY THR ASN VAL VAL ASN  1CSE
SEQRES 6 I 71  HIS VAL PRO HIS VAL GLY                              1CSE
FTNOTE                                                              1CSE
FORMUL3 CA    2(CA1 ++)                                             1CSE
FORMU4  HOH   *432(H2 O1)                                           1CSE
HELIX IA PHE I   10  VAL I   14  5                                  1CSE
HELIX IB THR I   17  TYR I   29  1                                  1CSE
SHEET S1I 4 LYS I   8  PHE I  10  0                                 1CSE
SHEET S1I 4 HIS I  65  GLY I  70 -1                                 1CSE
SHEET S1I 4 ARG I  51  TYR I  56 -1                                 1CSE
SHEET S1I 4 ASN I  33  LEU I  37  1                                 1CSE
TURN 1I VAL I  13  LYS I  16    TYPE II                             1CSE
TURN 2I TYR I  29  TYR I  32     TYPE I                             1CSE
TURN 3I PRO I  38  SER I  41     TYPE II                            1CSE
TURN 4I ARG I  48  ARG I  51     TYPE I                             1CSE
TURN 5I ASN I  57  THR I  60     TYPE II                            1CSE
SITE 1 RSB  2 LEU I  45  ASP I  46                                  1CSE
CRYST1   38.300   41.500   57.000 111.80  85.80 104.70 P 1       1  1CSE
```

4.2. Data Set

The data set is built upon three principles: **representativeness, orthogonality** and **coverage.** Representativeness is a property of the training set, while orthogonality is a property of the relation of similarity between the training and the test sets. Coverage should characterize the test set.

The training set must be representative, that is it should reflect the distribution of amino acids in secondary structures in the same ratio as in real life. We call such a set "well balanced". This will guarantee good predictions (within the limits of performance of the method) for an "average" protein.

By orthogonality we mean that proteins in the training set and those in the test set are functionally independent, hence structurally independent. That is, we do not have two proteins with similar structures. Hence, the quality of the results is not influenced by the similarity of the proteins in the data set.

By coverage we mean that (almost) all protein classes of a group (α,β) are represented into the test set. This guarantees the diversity of the test set by covering a large spectrum of proteins to be tested. Where missing, the proteins specified by Orengo et al. [Ore93] - our reference - where either not available in PDB, or they contained severe errors. For example, for the α group of classes, we have the following representation (see [Ore93]):

- α: Globin: 1mbc, 1thb, 2lh3
- α: Orthogonal: 1utg, 1fia,3sdp
- α: EFHand: 4cpv, 2scp, 4icb
- α: Up/Down: 256b, 2hmz, 2tmv
- α: Complex Up/Down: (none)
- α: Metal Rich: 1ycc, 451c

Tab.4.2.a. The α data set

Group	Class	PDB file	Length	# HELIX	# SHEET	# TURN	%H	%S	%T	%C
α	globin	1mbc	153	8	-	-	78	0	0	22
α	globin	1thb	141	8	-	-	78	0	0	22
α	globin	2lh3	153	7	-	-	82	0	0	18
α	orthogonal	1utg	70	4	-	1	61	0	6	33
α	orthogonal	1fia	98	4	-	6	55	0	25	20
α	orthogonal	3sdp	195	6	3	-	40	13	0	47
α	EFhand	4cpv	109	6	-	7	64	0	24	12
α	EFhand	2scp	174	8	4	1	64	7	2	27
α	EFhand	4icb	76	5	-	-	74	0	0	26
α	Up/Down	256b	106	5	-	4	80	0	15	5
α	Up/Down	2hmz	113	4	-	-	62	0	0	38
α	Up/Down	2tmv	158	6	-	2	53	0	5	42
α	Metal Rich	1ycc	107	5	2	6	59	7	23	11
α	Metal Rich	451c	82	5	-	5	51	0	24	25

Tab.4.2.b. The β Test Set

Group	Class	PDB file	Length	#H	#S	#T	%H	%S	%T	%C
β	Orthogonal Barrel	1ifc	132	2	11	8	12	58	23	7
β	Orthogonal Barrel	1rbp	182	2	9	8	9	44	17	30
β	Orthogonal Barrel	1bbp	173	4	11	8	11	42	20	27
β	Greek Key	2sga	181	-	15	-	0	50	50	0
β	Greek Key	1cob	151	1	8	13	4	36	34	26
β	Greek Key	2rhe	114	1	7	7	7	45	24	24
β	Greek Key	7pcy	98	1	8	7	7	60	28	5
β	Jelly Rolls	1tnf	157	1	10	-	3	53	0	44
β	Jelly Rolls	2cna	237	1	17	29	2	50	48	0
β	Complex Sandwich	5hvp	99	1	11	4	6	54	16	24
β	Complex Sandwich	2rsp	124	1	7	-	7	48	0	45
β	Trefoil	3fgf	146	-	12	7	0	49	19	32
β	Trefoil	8i1b	152	1	12	9	4	48	26	22
β	Disulfide Rich	1pi2	63	-	4	-	0	49	0	51
β	Disulfide Rich	3ebx	62	-	5	5	0	53	32	15

Zhang & al. [Zha92] used a database of 107 proteins having 19,861 residues. Their sequence homology is less than 50%, as is that of Rooman [Roo90]. Rooman defines sequence identity in terms of common patterns, e.g. A-A-X-X-

K. Rost & Sander [Ros93] underline that for chains of more than 80 residues, the mutual similarity should be less than 25%. They used 130 chains. King & Sternberg [Kin90] used 43 proteins for the training set and 18 for the test set; the proteins they used were "selected to remove homologous proteins". We built our data set based on the latter idea: selecting proteins without similarities. Our data set is smaller. In spite of the reduced and non-homologous data set, we obtained encouraging results.

Using the statistics of relative frequencies of occurrence of amino acid residues in secondary structures of proteins - from [Str88] (see Tab.4.2.c) - we considered three groups of amino acids:

- **α-helix** favoring amino acids: ALA, CYS, LEU, MET, GLU, GLN,HIS, LYS.
- **β-sheet** favoring amino acids: VAL, ILE, PHE, TYR, TRP, THR.
- **β-turn** favoring amino acids: GLY, SER, ASP, ASN, PRO.

ARG does not have any preference for any secondary structure, hence it was not included in our three groups.

Tab.4.2.c. Amino acids preferences

Amino acid	α-helix	β-sheet	β-turn
ALA	1.29	0.90	0.78
CYS	1.11	0.74	0.80
LEU	1.30	1.02	0.59
MET	1.47	0.97	0.39
GLU	1.44	0.75	1.00
GLN	1.27	0.80	0.97
HIS	1.22	1.08	0.69
LYS	1.23	0.77	0.96
VAL	0.91	1.49	0.47
ILE	0.97	1.45	0.51
PHE	1.07	1.32	0.58
TYR	0.72	1.25	1.05
TRP	0.99	1.14	0.75
THR	0.82	1.21	1.03
GLY	0.56	0.92	1.64
SER	0.82	0.95	1.33
ASP	1.04	0.72	1.41
ASN	0.90	0.76	1.28
PRO	0.52	0.64	1.91
ARG	0.96	0.99	0.88

Based on the three groups of amino acids, we computed the statistics (shown in Tab.4.2.d., Tab.4.2.e.) on our data set. The difference from 100% is due to the non-occurence of the Arg amino acid.

Tab.4.2.d. Amino acids preferences of the α data set

Group	Protein	**α-helix** favoring amino acids	**β-sheet** favoring amino acids	**β-turn** favoring amino acids
α	1mbc	56%	21%	18%
α	1thb	48%	23%	26%
α	2lh3	47%	30%	21%
α	1utg	47%	22%	27%
α	1fia	44%	22%	23%
α	3sdp	44%	23%	25%
α	4cpv	48%	23%	27%
α	2scp	35%	28%	33%
α	4icb	55%	17%	27%
α	256b	55%	15%	25%
α	2hmz	39%	30%	26%
α	2tmv	28%	34%	30%
α	1ycc	45%	23%	28%
α	451c	48%	20%	29%

The majority (77%) of the α proteins are made of **α-helix** favoring amino acids in a proportion greater than 44%.

Tab.4.2.e. Amino acids preferences of the β Test Set

Group	Protein	**α-helix** favoring amino acids	**β-sheet** favoring amino acids	**β-turn** favoring amino acids
β	1ifc	38%	31%	25%
β	1rbp	36%	26%	29%
β	1bbp	33.5%	34.5%	31%
β	2sga	24.8%	31%	40%
β	1cob	33%	27%	33%
β	2rhe	31%	24%	42%
β	7pcy	30%	31%	36%
β	1tnf	39.5%	25.5%	29.3%
β	2cna	29%	30%	38%
β	5hvp	36.3%	32.3%	27.3%
β	2rsp	36.3%	24.2%	31.5%
β	3fgf	39%	22%	31.5%
β	8i1b	41.5%	25%	31%
β	1pi2	46%	11%	35%
β	3ebx	37%	24.2%	33.9%

Most of our β proteins (86.5%) are made of **β-sheet** favoring amino acids in a propotion greater than 24%. Equally, most of them (also 86.5%) present **α-helix** favoring amino acids in a percentage lower than 40%.

We had to use a β-training set different from our initial β-data set (which remained only as β-test set) in

the case of the β proteins, because too many proteins in the latter had too strong α-preferring presence (greater than 35.5%) - 9 out of 15 β proteins - while our training set has a better amino acid preference distribution (only 2 proteins out of 10 have such a strong α-preferring presence)- see Tab.4.2.f. and Fig.3.3.2.6.

Tab.4.2.f. Amino acids preferences of the β Training set

Group	Protein	**α-helix** favoring amino acids	**β-sheet** favoring amino acids	**β-turn** favoring amino acids
β	1acx	32.4%	25.9%	40.7%
β	1atx	32.6%	21.7%	41.3%
β	1hoe	33.8%	33.8%	28.4%
β	1paz	43.9%	28.4%	26.8%
β	2aza	46.5%	25.6%	27.1%
β	2er7	23%	38.2%	38.5%
β	2pab	35.4%	32.3%	29.1%
β	2por	34.5%	27.9%	35.2%
β	2stv	32.8%	30.2%	30.2%
β	3hla	35.3%	30.3%	29.3%

4.3. Error sources

The errors in prediction are either intrinsic to the method, or general to all methods, due to the simplification done to such a complex problem, in order to manage it. There are also input errors (of PDB files), which we tried to eliminate.

4.3.1. Errors of PDB files

The following types of errors exist in the PDB files.

- measurement errors: crystallography and other methods present an inherent imprecision, which in turn influences the learning process and hence the results.

- overlapping structures: some secondary structures of PDB files are overlapping - normally, this should not occur. The consequence is that the statistics are not accurately reflecting the real distribution of amino acids into secondary structures and hence, the prediction is affected.

- rough errors: secondary structures of some PDB files are beyond the limits of the protein sequence. We eliminated such files from the learn and test sets.

4.3.2. Errors of our method

Our method has the following kinds of errors.

- errors of this type of algorithms: it does not take into account other properties of the amino acids such as polarity, charge, dimension, hydrophobicity, etc. The consequence is that the precision of the prediction cannot be better than the experimental threshold of 70%.

- learn/test errors: the limited number of learned proteins and the distribution of their amino acids into secondary structures generally leads to imperfect statistics, from the point of view of the test proteins.

4.3.3. General errors (common to most methods)

Most machine learning methods suffer the following common kinds of errors.

- completely new proteins (without any equivalent into the training set), having peculiar structures (e.g. protein inhibitors) will lead to less precise predictions.

- the distant secondary structures slightly influence the positions of amino acids around a central residue.

- the protein spatial (3D) configuration also influences the secondary arrangement. Being difficult to take it into account, it is negleted.

4.4. Results

The 14 pure α proteins we included in our data set (see Tab.4.2.a.) were used as both Training set and Test Set: one protein at a time was excluded from the α Set and used as test protein, while the rest of 13 proteins were used as Training set for the former protein. That is, we used a jack-knife testing for the α proteins.

The results of the prediction are shown in Tab.4.4.a. We present the helix prediction accuracy and coil prediction accuracy. Based on these results, we built the average

prediction accuracy on the α set for all results for α-helix (68.7%) and for coil (65.4%). All are computed on a per-residue basis (see Tab.4.4.b.).

Tab.4.4.a.Results of the prediction on the α data set

Group	Class	PDB file	Length	%H prediction	%C prediction
α	globin	1mbc	153	63.6%	65.2%
α	globin	1thb	141	61.6%	68.6%
α	globin	2lh3	153	74.5%	64.3%
α	orthogonal	1utg	70	77.2%	75.6%
α	orthogonal	1fia	98	79.5%	45.4%
α	orthogonal	3sdp	195	64.5%	58.9%
α	EFhand	4cpv	109	62.8%	62.5%
α	EFhand	2scp	174	56.7%	72.9%
α	EFhand	4icb	76	65.4%	77.5%
α	Up/Down	256b	106	85.9%	75.7%
α	Up/Down	2hmz	113	85.7%	73.6%
α	Up/Down	2tmv	158	78.3%	59.1%
α	Metal Rich	1ycc	107	42.3%	77.9%
α	Metal Rich	451c	82	71.4%	67.3%

Tab. 4.4.b. Average prediction accuracy on the α set

Average predction accuracy for the top n results	%HELIX prediction	%COIL prediction
n = 14 (all results)	68.7%	65.4%

Tab. 4.4.c Results of the prediction on the β test set

Group	Class	PDB file	Length	%S prediction	%C prediction
β	Orthogonal Barrel	1ifc	132	85.7%	31.9%
β	Orthogonal Barrel	1rbp	182	55.5%	62.7%
β	Orthogonal Barrel	1bbp	173	65.2%	51.6%
β	Greek Key	2sga	181	81.5%	68.6%
β	Greek Key	1cob	151	62.5%	60.3%
β	Greek Key	2rhe	114	80.4%	63%
β	Greek Key	7pcy	98	51.6%	73.1%
β	Jelly Rolls	1tnf	157	54.2%	50%
β	Jelly Rolls	2cna	237	56.2%	69.8%
β	Complex Sandwich	5hvp	99	62.2%	42.4%
β	Complex Sandwich	2rsp	124	74.6%	63.7%
β	Trefoil	3fgf	146	73.2%	50%
β	Trefoil	8i1b	152	69.8%	54.4%
β	Disulfide Rich	1pi2	63	53.8%	61.2%
β	Disulfide Rich	3ebx	62	69.7%	64.1%

The results of the prediction on the β set are shown in Tab.4.4.c. The average results accuracy is printed in Tab.4.4.d. - it is 66.3% for β-sheet. Three results are better than 80% and seven results are better than 69.7%.

The overall prediction accuracy is 65% (including coil prediction). These results are competitive with those obtained by more sophisticated systems like the hybrid system of Rost & Sander ([Ros93]) who combined multiple sequence alignments and neural networks, obtaining an accuracy of 70.8%; they reached 65.4% for β-sheet.

Tab.4.4.d. Average prediction accuracy for the β set.

Average predction accuracy for the n results	**%SHEET prediction**	**%COIL prediction**
n = 15 (all)	66.3%	59.0%

The computation time is 10 minutes for each secondary structure type (α-helix, β-sheet) for a protein of length 140, using a training set of 15 proteins. We used an IBM PC 386DX (40MHz, 16M RAM).

4.5. Results Interpretation

Like all methods, we obtained better results in predicting α-helix than β-sheet. That is because of the biochemical particularities of the β-sheet, which make it more difficult to predict - the diversity of the combinations of amino acids in β-sheets is even greater than that for α-helices and remote amino acids interactions are also important [Str88].

The prediction accuracy of the coil is slightly smaller than that of α-helix; in the case of β proteins, the coil prediction is 6.6% below that of β-sheet. Both can be explained by the over-prediction which we already mentioned when describing our method.

We underline that these results were obtained on data sets with a good orthogonality and in spite of the relatively small number of proteins we used as training set.

Chapter 5. CONCLUSIONS

The nearest-neighbor algorithm is a method that behaves well in the field of protein secondary structure prediction. Our nearest-neighbor classifier achieves 65% accuracy without using any weights, on a set of 29 proteins having a good orthogonality (by construction) and in spite of the relatively small number of proteins involved in the training set.

The nearest-neighbor algorithm allows one to embed domain knowledge in its structure (thus transforming the method into a hybrid one). The performance of our approach could be enhanced by including environment class information into the algorithm, such as physical and chemical properties of amino acids and patterns having known behaviour, such as:

- [X, tiny or (small and polar) or p, X, tiny or (polar minus aromatic) or p] -> COIL [Kin90]
- A-A-X-X-K -> HELIX [Roo90]

where X is any amino acid.

The present approach can also be used in a hybrid system, together with multiple sequence alignments. The best

predictors to date use hybrid systems, with knowledge of environment classes, and multiple alignment.

More work must be done in both creating larger training sets and testing more proteins, including those of α/β and $\alpha+\beta$ groups of classes. The training sets must be well balanced, in order to be near the natural distribution of amino acids in secondary structure types, as we explained in the previous chapter. The construction of the training set is facilitated by statistics provided by our method, but domain knowledge is still crucial. A larger training set might improve the performance of the system, as Kibler and Langely [Kib90] conclude from their experiment that even if most learning occurs early, after the algorithm has processed a moderate number of training instances, a slight increase of the classification accuracy occurs with additional instances added to the training set. The system reaches its asymptotic performance using about 50 instances in the training set.

Chapter 6. BIBLIOGRAPHY

[Aha91] Aha, D.W., Kibler, D. & Albert, M.A. (1991). Instance-based learning algorithms. *Machine Learning, 6,* 37-66.

[Bar91] Barton, G.J., Newman, R.H., Freemont P.S. & Crumpton, M.J.(1991). Amino acid sequence analysis of the annexin super-gene family of proteins. *European Journal of Biochemistry* 198, 749-760.

[Ben90] Benner, S.A. & Gerloff, D. (1990). Patterns of divergence in homologous proteins as indicators of secondary and tertiary structure: a prediction of the structure of the catalytic domain of protein kinases. *Advances in Enzyme Regulation* 31, 121-181.

[Bio88] Biou, V., Gibrat, J.F. Levin, J.M., Robson, B. & Garnier, J.(1988). Secondary structure prediction: combination of three different methods. *Protein Engineering 2,* 185-191.

[Boh88] Bohr, H., Bohr, J., Brunak, S., Cotterill, R.M.J., Lautrup, B., Norskov, L., Olsen, O. H. & Petersen, S.B. (1988). Protein secondary structure and homology by neural networks. *FEBS Letters*, 241, 223-228.

[Boh90] Bohr, H., Bohr, J., Brunak, S., Fredhom, H., Lautrup, B. & Petersen, S.B. (1990). A novel approach to prediction of the 3-dimensional structures of protein backbones by neural networks. *FEBS Letters*, 261, 43-46.

[Bor92] Bork, P., Ouzonis, C., Sander, C., Scharf, M., Schneider, R. & Sonnhammer, E. (1992). What's in a genome? *Nature, 358*, 287.

[Bos90] Bossa, F. & Pascarella, S. (1990). PRONET: a microcomputer program for predicting the secondary structure of proteins with a neural network. *CABIOS*, 5, 319, 320.

[Cho74] Chou, P.Y. & Fasman, U.D. (1974). Prediction of protein conformation. *Biochemistry*, 13, 211-215.

[Coh83] Cohen, F.E., Abarbanel, R.M., Kunz, I.D. & Fletterick, R.J. (1983). Secondary structure assignment for alpha/beta proteins by a combinatorial approach. *Biochemistry*, 22, 4894-4904.

[Coh86] Cohen, F.E., Abarbanel, R.M., Kunz, I.D. & Fletterik, R.J. (1986). Turn prediction in proteins using a pattern-matching approach. *Biochemistry*, 25, 266-275.

[Cos92] Cost, S. & Salzberg, S. (1992). Predicting Protein Secondary Structure with a Nearest-neighbor Algorithm. *Journal of Molecular Biology*, 227, 371-374.

[Cos93] Cost, S. & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning 10:1*, 57-78.

[Fas89] Fasman, G.F. (1989). Protein conformation prediction. *In Prediction of Protein Structure and the Principles of Protein Conformation (Fasman, G.D., ed.), pp. 193-316, Plenum, New York and London.*

[Fra89] Frampton, J., Leutz, A., Gibson, T.J. & Graf, T. (1989). DNA-binding domain ancestry. *Nature*, 342, 134.

[Gar91] Garett, R.C., Thornton, J.M. & Taylor, W.R. (1991). An extension of secondary structure prediction towards the prediction of the tertiary structure. *FEBS Letters*, 280, 141-146.

[Gar78] Garnier, J., Osguthorpe, D.J. & Robson, B. (1978). Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *Journal of Molecular Biology*, 120, 97-120.

[Gar93] Garnier, J. (1993). Prediction of protein structure. *In Biological Sequences: Finding Structure and Function by Neural Networks (Brunak, S., ed.), Institute for Scientific Interchange Foundation, Torino, Italy.*

[Gib87] Gibrat, J.-F., Garnier, J. & Robson, B. (1987). Further developments of protein secondary structure prediction using information theory. New Parameters and consideration of residue pairs. *Journal of Molecular Biology 198*, 425-443.

[Gib93] Gibson, T.J., Thompson, J.D. & Abagyan, R.A. (1993). Proposed structure for the DNA-binding domain of the helix-loop-helix family of eukaryotic gene regulatory proteins. *Protein Engineering* 6, 41-50.

[Hir92] Hirst, J.D. & Sternberg, M.J.E. (1992). Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry*, 31, 615-623.

[Hol89] Holley, H.L. & Karplus, M. (1989). Protein secondary structure prediction with a neural network. *Proceedings of the National Academy of Sciences, U.S.A. 86*, 152-156.

[Kan88] Kanehisa, M. (1988). A multivariate analysis method for discriminating protein secondary structural segments. *Protein Engineering 2*, 87-92.

[Kib90] Kibler, D. & Langley, P. (1990). Machine Learning as Experimental Science. *Readings in Machine Learning/edited*

by J.Shavlik & Th.G.Diettrich. Morgan Kaufman Publishers, 38-44.

[Kin90] King, R.D. & Sternberg, M.J.E. (1990). Machine learning approach for the prediction of protein secondary structure. *Journal of Molecular Biology, 216,* 441-457.

[Kne90] Kneller, D.G., Cohen, F.E. & Langridge, R. (1990). Improvements in protein secondary structure prediction by an enhanced neural network. *Journal of Molecular Biology* 214, 171-182.

[Kod90] Kodratoff, Y. & Michalski, R. (1990). *Machine Learning. An Artificial Intelligence Approach. Vol.III. Morgan Kaufmann Publishers.*

[Lev76] Levitt, M. & Chothia, C. (1976). Structural patterns in globular proteins. *Nature,* 261, 552-558.

[Lev88] Levin, J.M. & Garnier, J. (1988). Improvements in a secondary structure prediction method based on a search for local sequence homologies and its use as a model building tool. *Biochimica et Biophysica Acta, 955,* 283-295.

[Lev86] Levin, J.M., Robson, B. & Garnier, J. (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS Letters, 205*, 303-308.

[Mac93] Maclin, R. & Shavlik, J. W. (1993). Using knowledge-based neural networks to improve algorithms: refining the Chou-Fasman algorithm for protein folding. *Machine Learning*.

[Max79] Maxfield, F.R. & Scheraga, H.A. (1979). Improvements in the prediction of protein topography by reduction of statistical errors. *Biochemistry*, 18, 697-704.

[Mic83] Michalski, R.S., Carbonell J.G. & Mitchell T.M. Editors (1983). *Machine Learning. An Artificial Intelligence Approach. Vol.I. Tioga Publishing Company, Palo Alto, California.*

[Mic86] Michalski, R.S., Carbonell J.G. & Mitchell T.M. Editors (1986). *Machine Learning. An Artificial Intelligence Approach. Vol.II. Morgan Kaufmann Publishers.*

[Mug92] Muggleton, S., King, R. D. & Sternberg, M.J.E. (1992). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering* 5, 647-657.

[Mus92] Musacchio, A., Gibson, T., Lehto, V.-P. & Saraste, M. (1992). SH3 - an abundant protein domain in search of a function. *FEBS Letters*, 307, 55-61.

[Nag73] Nagano, K. (1973). Logical analysis of the mechanism of protein folding. *Journal of Molecular Biology* 75, 401-420.

[Nag75] Nagano, K. & Hasegawa, K. (1975). Logical analysis of the mechanism of protein folding. *Journal of Molecular Biology* 94, 257-281.

[Nie91] Niermann, T. & Kirschner, K. (1991). Improving the prediction of secondary structure of "TIM-barrel" enzymes (Corrigendum). *Protein Engineering* 4, 359-370.

[Ore93] Orengo, C.A., Flores, T.P., Taylor, W.R. & Thornton, J.M. (1993). Identification and classification of protein fold families. *Protein Engineering, 6:5*, 485-500.

[Ouz91] Ouzounis, C.A. & Melvin, W.T. (1991). Primary and secondary structural patterns in eukaryotic cytochrome P-450 families correspond to structures of the helix-rich domain of Pseudomonas putyida cytochrome P-450cam. *European Journal of Biochemistry,* 198, 307-315.

[Pin90] Pingand P. (1990). Etude d'un environnement permettant l'acquisition de connaissance par apprentissage. Aplication à l'analyse structurelle des protéines. *Thèse de doctorat. Académie de Montpellier. Université de Montpellier - Sciences et Techniques du Languedoc.*

[Pre92] Presnell, S.R., Cohen, B.I. & Cohen, F.E. (1992). A segment-based approach to protein secondary structure prediction. *Biochemistry,* 31, 983-993.

[Pti83] Ptitsyn, O.B. & Finkelstein, A.V. (1983). Theory of protein secondary structure and algorithm of its prediction. *Biopolymers, 22,* 15-25.

[Qia88] Qian, N. & Sejnowski, T.J. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology,* 202, 865-884.

[Roo91] Rooman, M.J. & Wodak, S. (1991). Weak correlation between predictive power of individual sequence patterns and overall prediction accuracy in proteins. *Proteins: Struct. Func. Genet.*, 9, 69-78.

[Roo89] Rooman, M.J., Wodak, S. & Thornton, J.M. (1989). Amino acid sequence templates derived from recurrent turn motifs in proteins: critical evaluation of their predictive power. *Protein Engineering* 3, 23-27.

[Room91] Rooman, M.J., Kocher, J.P. & Wodak, S.J. (1991). Prediction of protein backbone conformation based on seven structure assignments: influence of local interactions. *Journal of Molecular Biology,* 221, 961-979.

[Ros93] Rost, B. & Sander, C. (1993). Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology, 232,* 584-599.

[Rus92] Russell, R.B., Breed, J. & Barton, G.J. (1992). Conservation analysis and structure prediction of the SH2 family of phosphotyrosine binding domains. *FEBS Letters,* 304, 15-20.

[Sal95] Salamov, A.A. & Solovyev, V.V. (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology, 247,* 11-15.

[Sch79] Schulz, G.E. & Schirmer, R.H. (1979). *Principles of Protein Structure, Springer, New York.*

[Sta86] Stanfill, C. & Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM, 29:12,* 1213-1228.

[Ste90] Sternberg, M.J.E. & King, R.D.(1990). Machine learning approach for the prediction of protein secondary structure. *Journal of Molecular Biology,* 216, 441-457.

[Sto92] Stolorz, P., Lapedes, A. & Xia, Y. (1992). Predicting protein secondary structure using neural net and statistical methods. *Journal of Molecular Biology,* 225, 363-377.

[Str88] Stryer, L. (1988). *Biochemistry, 3rd Ed. W.H.Freeman & Comp - New York.*

[Tay83] Taylor, W.R. & Thornton, J.M. (1983). Prediction of super-secondary structure in proteins. *Nature*, 301, 540-542.

[Tay88] Taylor, W.R. (1988). Pattern matching methods in protein sequence comparison and structure prediction. *Protein Engineering 2*, 77-86.

[Tec95] Tecuci, G. & Kodratoff, Y. (1995). *Machine Learning and Knowledge Acquisition: Integrated Approaches. Academic Press, London.*

[Yi93] Yi, T.M. & Lander, E.S. (1993). Protein secondary structure prediction using nearest-neighbor methods. *Journal of Molecular Biology. 232*, 1117-1129.

[Zha92] Zhang, X., Mesirov, J.P. & Waltz, D.L. (1992). Hybrid system for protein secondary structure prediction. *Journal of Molecular Biology* 225, 1049-1063.

[Zho92] Zhong, L., Johnson, W. & Curtis, J. (1992) Environment affects amino acid preference for secondary

structure. *Proceedings of the National Academy of Sciences U.S.A. 89*, 4462-4465.

[Zve87] Zvelebil, M.J., Barton, G.J., Taylor, W.R. & Sternberg, M.J.E.(1987). Prediction of protein secondary structure and active sites using alignment of homologous sequences. *Journal of Molecular Biology* 195, 957- 961.