Markku Siermala

Local Prediction of Secondary Structures of Proteins from Viewpoints of Rare Structure

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Information Sciences of the University of Tampere, for public discussion in the Paavo Koli Auditorium on May 3, 2002, at 12 noon.

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF TAMPERE

A-2002-3

TAMPERE 2002

Supervisors: Professor Martti Juhola
Department of Computer and Information Sciences,
University of Tampere

Professor Mauno Vihinen
Institute of Medical Technology,
University of Tampere

Opponent: Professor Tapio Grönfors
Department of Computer Science and Applied Mathematics
University of Kuopio

Reviewers: Professor Timo Järvi
Department of Computer Science
University of Turku

Professor Olli Yli-Harja
Institute of Signal Processing
Tampere University of Technology

Department of Computer and Information Sciences
FIN-33014 UNIVERSITY OF TAMPERE
Finland

Electronic dissertation
Acta Electronica Universitatis Tamperensis 180
ISBN 951-44-5358-1
ISSN 1456-954X
http://acta.uta.fi

ISBN 951-44-5342-5
ISSN 1457-2060

Tampereen yliopistopaino Oy
Tampere 2002

Abstract

This dissertation deals with the local prediction of protein secondary structure from the viewpoint of rare secondary structures. Protein three-dimensional structures are needed in the biomedical field because structures indicate something about the functions of proteins, and functions are almost everything that happens in a living cell. Unfortunately, it is difficult to ascertain the structure of a protein, because the details of the structure are located at the level of atoms. However, an amino acid sequence is fairly easy to solve and can also be produced from a DNA sequence. This could be a shortcut to the structure and function of proteins. We searched for ways to better understand the prediction challenge of secondary structures. Our research started with polyproline type II secondary structure prediction. The results showed that a neural network behaved well when the learning and test sets had a uniform class distribution. However, the identification of amino acid sequences that represent a rare class was difficult with class distribution of the real world. In this context, prediction was hampered by imbalanced class distribution. We developed spectrum and response analysis for the neural network which reveal the reasons for a certain decision. The frequencies of prolines affected a major part of decisions and this was almost all that a neural network could learn from the data. Apparently input sequences can take the evolutionary pre-information to the learning process. With the polyproline II structure this was a promosing idea and aroused interest in using the method with other structures and other pre-information types. With hyperspheres we developed a learning algorithm that achieved excellent prediction accuracy with all known secondary structure types. Unfortunately, the method leaves cases unclassified - if uncertain generalization is reduced, hyperspheres can achieve better prediction accuracies. Finally, for all secondary structure types we analyzed the space used and found explanations for how the structure types behave in the sequence space. The results showed that polyproline II is an exception among other types because of its sensitivity to the amino acid proline. We were able to show that for half of sequences the nearest case seek its one's way to the distance as cases were randomly generated. Therefore, in the sequence space there are no large clusters. Rather, around the individual case (sequence) there is a sphere with high probability of achieving the same secondary structure type.

Keywords: secondary structure prediction, neural network, machine learning

Acknowledgements

This work was carried out in the Department of Computer and Information Sciences, University of Tampere, during the years 1998 - 2002.

I wish to warmly thank my supervisor, Professor Martti Juhola, Ph.D. of the University of Tampere, who helped me through this doctoral research. I also wish thank sincerely Professor Mauno Vihinen, Ph.D. of the University of Tampere, who was my supervisor in the bioinformatics domain.

I am grateful to the Heads of the Department of Computer and Information Sciences, Professor Kari-Jouko Räihä, Ph.D., Professor Pertti Järvinen, Ph.D., Professor Seppo Visala, Ph.D., and Professor Jyrki Nummenmaa, Ph.D., for the opportunity to work and complete my dissertation at the University of Tampere.

I want to thank Jorma Laurikkala, Ph.D., Kati Viikki M.Sc., Suvi Karvonen, M.Sc., and Charles Cathal, TEFL., for their help and persevering attitude regarding my limited ability to produce readable English. I wish to thank Jorma Laurikkala, Ph.D., for assistance in dealing with logistic regression and Jyrki Nummenmaa, Ph.D., for assistance in dealing with scattering ratio. I also want to thank all the members of the research group of Professor Juhola and the personnel in the Department of Computer and Information Sciences for inspiring conversations and a pleasant working atmosphere.

I want to thank my wife Anita, my son Joni, my father Jorma and my mother Helena for their encouragement during this work. I also wish to express my greatest gratitude to all my friends for their interest in my work.

This work has been supported financially by the University of Tampere, Tampere Graduate School in Information Science and Engineering, the Academy of Finland, the Ella & Georg

Ehrnrooth Foundation and the Finnish Cultural Foundation, the Regional Fund of Central Ostrobotnia.

List of Abbreviations

Abbreviation	Description
CASP	The Critical Assessment of Techniques for Protein Structure Prediction
DNA	Deoxyribonucleic acid
DSSP	Dictionary of Protein Secondary Structure
MLP	Multilayer perceptron neural network
NMR	Nuclear magnetic resonance
PAM & PAM250	Amino acids substitution tables
PDB	The Protein Data Bank
PPII	Polyproline type II secondary structure
PSIPRED and PHD	Protein secondary structure prediction methods

List of original publications

This dissertation is based on the following articles referred to in the text of their Roman numerals.

I. Siermala M, Juhola M, Vihinen M: On preprosessing of protein sequences for neural network prediction of polyproline type II secondary structures, Computers in Biology and Medicine 31 (2001) 385-398.

II. Siermala M, Juhola M, Vihinen M: Neural network prediction of polyproline type II secondary structure. In: Hasman A, Blobel B, Dudeck J, Engelbrecht R, Gell G, Prokosch H-U (eds.): Medical Infobahn for Europe: Proceedings of MIE2000 and GMDS2000, Studies in Health Technology and Informatics, vol. 77, IOS Press, Amsterdam, 2000, pp. 475-479.

III. Siermala M, Juhola M, Vihinen M: On postprosessing of neural network prediction of polyproline type II secondary structures: Network spectrum, response analysis, and scattering, submitted to Neural Computing and Applications.

IV. Siermala M, Juhola M, Vihinen M: Binary vector or real value coding for secondary structure prediction? A case study of polyproline type II prediction. In: Crespo J, Maojo V, and Martin F (eds.), Medical Data Analysis, Proceedings of Second International Symposium, ISMDA 2001, Madrid, Spain, October 2001.

V. Siermala M: Prediction of protein secondary structures of all types using new hypersphere machine learning method. In: Quaglini S, Barahona P, Andreassen S (eds.): Artificial Intelligence In Medicine, 8th European Conference on Artificial Intelligence in Medicine in Europe, Lecture Notes in Artificial Intelligence, vol. 2101, Springer, Berlin, 2001, pp. 117-120.

VI. Siermala M, Juhola M: Behaviour of protein secondary structure types in the sequence space of certain length, accepted to Intelligent Data Analysis.

Contents

1. Introduction

1.1. Machine learning

Machine learning algorithms have a great practical value. They are very useful in the area of data mining, domains where humans may not have the knowledge or understanding to develop effective algorithms and domains where a computer program must dynamically adapt to changing conditions [Mit97].

Machine learning concerns the question how to construct computer programs that automatically improve with experience [Mit97]. On the other hand, artificial intelligence is based on the questions what is intelligence and how we can model it [CHK93]. The basic task of machine learning is more practical and the role of artificial intelligence can be seen as a support to the machine learning field. Machine learning research also utilizes knowledge from statistics, philosophy, information theory and biology etc. [Mit97].

The basic idea behind machine learning is that an object can be described by the values of its attributes [Gar01]. The aim of the methods is to find an unknown function that can correctly classify the given examples by using only attribute information. Learning involves searching, at the worst, through a space of all possible hypotheses to find the hypothesis that best fits the available training examples [Mit97].

We can make several lists that include central machine learning methods. Mitchell presents methods that seem to be important. His book includes methods on concept learning, decision trees learning, neural networks learning, Bayesian learning, instant based learning, genetic algorithm, learning sets of rules, and reinforcement learning [Mit97]. In addition to this list there could be Markov models that are useful with temporal learning tasks [BB98], and support vector machines that draw an optimal hyperplane in a high dimensional feature space [DD01].

The first four papers in this dissertation describe how a neural network accomplished a difficult learning task. Therefore neural networks are next reviewed. They are popular tools that provide useful information in the area of artificial intelligence and machine learning. Researchers in the field of artificial intelligence assume that artificial neural networks allow us to understand how biological neural networks work. In the machine learning field, researchers are interested in neural networks because they make it possible to build more intelligent machines and models [And97].

The history of artificial neural networks started in 1943 when McCulloch and Pitts [MP43] built models based on ideas of actual neurons. The first models were simple networks that used binary decision units. Two decades later Rosenblatt with some colleagues [Ros62] constructed a weighted neural network model with perceptron units. The efficiency of neural models was increased when it was realized that networks can be built with hidden layers. With hidden layers and a Werbos backpropagation learning algorithm, the networks achieved more and more complicated models [Tay97]. One very important property of multilayer perceptrons is universality, which means their capacity to approximate any function at any desired accuracy [Alm97].

Modern models of neural networks are composed of simple parallel elements (see Figure 1). Models for elements (neurons) are based on the biological nervous system [DB98]. In general, a simple neuron may have several details, but most of them have many numerical input lines (i.e. input attributes) and one output line (i.e. value for performed function) [Mic97]. A transformation function that is inside an individual neuron may be one of the several functions. A function takes every input signal into account and performs a transformation operation. The topology of neural networks represents how individual neurons are connected to each other and the external world. Topology plays a crucial role in the functionality and performance of the network [Fie97].

The function of neural networks is largely made up of connections between elements. Learning can be achieved by modifying the weight of connections [DB98]. There are three major classes of artificial learning types: supervised, unsupervised and reinforcement

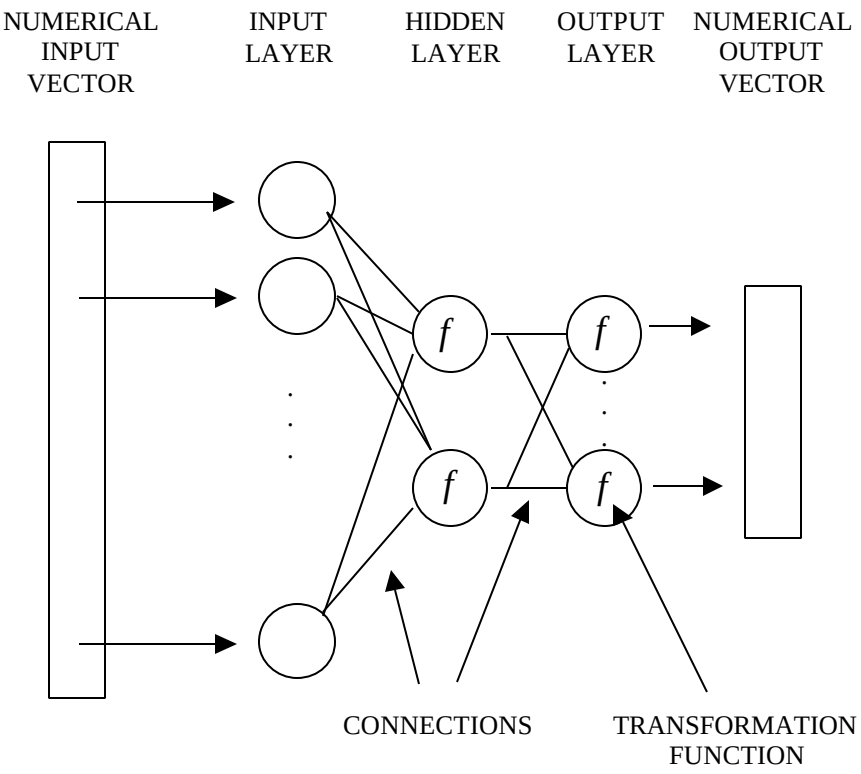

Figure 1. Feed forward multilayer perceptron neural network and its basic elements.

learning. Supervised models (as in Figure 1) assume that a training set is available which contains both input patterns and the corresponding desired output patterns [Alm97]. Unsupervised learning resembles feature extraction models and there is no external teacher to oversee the learning process [Hay94]. Reinforcement learning is between supervised and unsupervised models [Wer97], where a process maximizes a performance index called reinforcement signal [Hay94].

1.2. On protein structures

The genetic information (i.e. instructions for construct) of a living organism is stored in a 1-dimensional code in DNA. The code is dissolved in the synthesis of proteins that takes place at the surface of ribosomes in any cell of a living organism [UTY98+]. In each type of protein the polypeptide chain is folded into a specific three-dimensional structure [Leh79] and the function of a protein is determined by its three-dimensional structure [CB00].

Proteins cause almost all the events in the cells of a living organism. Therefore, there are urgent requirements to determine the three-dimensional structures of proteins.

The structure of a protein has different levels and it has an energially and structurally optimized form [Tur97]. The *primary structure* is the amino acid sequence of the protein and can be presented by a sequence with 20 letters, where each letter indicates an individual amino acid. The *secondary structure* describes the areas in the primary structure where secondary structure elements occur in the backbone of the protein. In the backbone there are also locations where there are no regular secondary structures [SO97]. The *tertiary structure* is the three-dimensional structure of a single protein chain. The *quaternary structure* is the three-dimensional native structure of complex of several chains [CB00, MKV89]. Figure 2 presents several viewpoints to the three-dimensional structure of a Calcium/Phospholipid-Binding Protein (the Protein Data Bank (PDB) code is 1AOW).

X-ray diffraction (crystallography) methods for obtaining protein structure information may be accurate, but some steps are uncertain [QS88]. Moreover, X-ray crystallography methods require that a protein can be crystallized, however, this is not always the case. Proteins cannot be brought into a sufficiently concentrated solution for liquid-state nuclear magnetic resonance (NMR) spectroscopy [Eth02] and the result of an NMR study is not as detailed and accurate as that obtained crystallographically [PDB]. However, the NMR method provides useful information on the dynamic properties of molecules. Electronic microscopy can also provide three-dimensional information from biochemical molecules. Unfortunately, all these methods are time-consuming and they need much pure protein data, which may be very difficult to procure.

The fact is that the primary structure information is much easier to get than information of higher level structures. These are the practical reasons that compel us to predict three-dimensional structural properties using primary structure information.

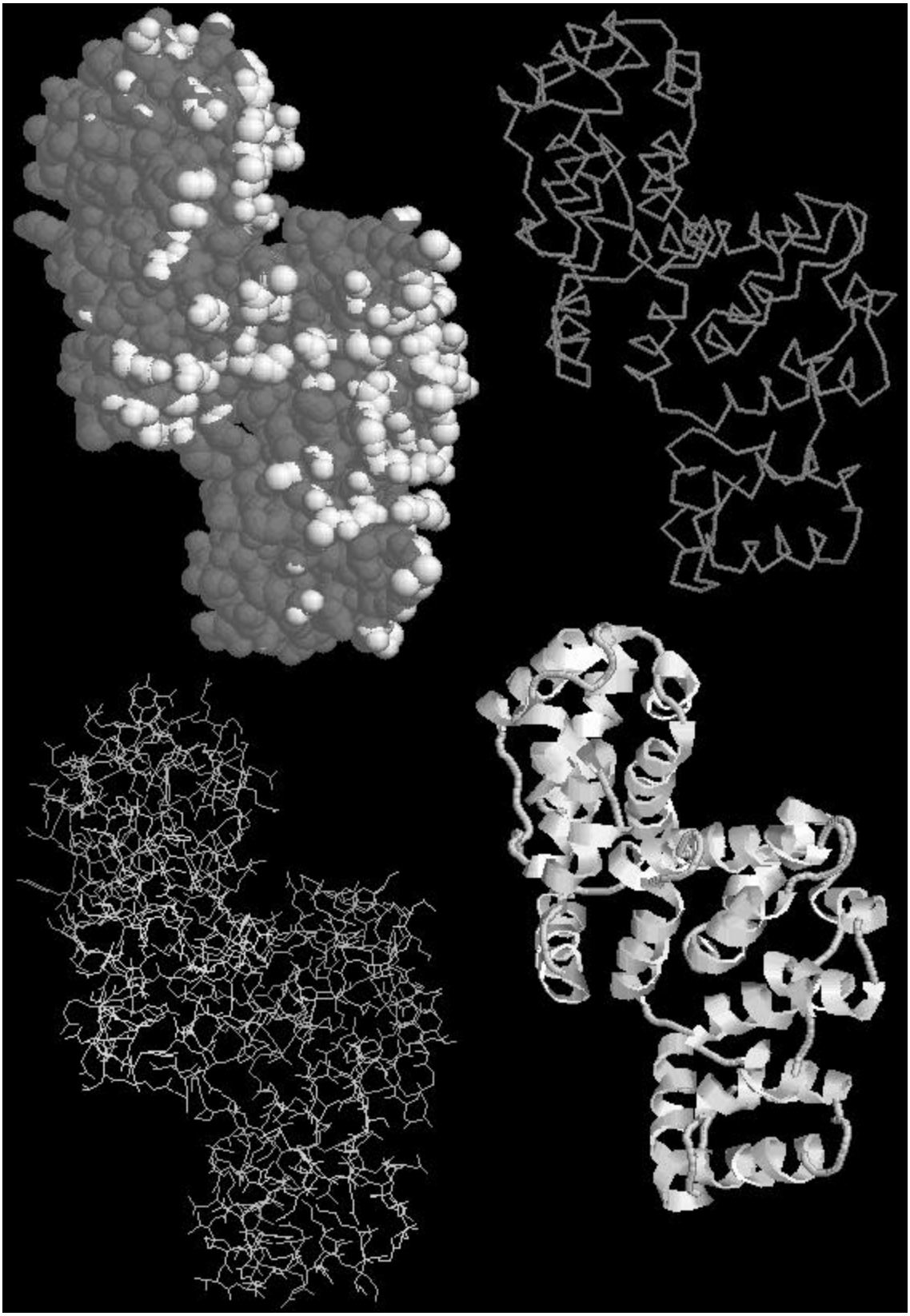

Figure 2. Four views of three-dimensional structures of protein. In the upper left corner, the atoms are modelled with spheres, in the upper right corner there is a protein backbone, in the bottom left corner each bond is presented, and finally in the bottom right corner secondary structures are highlighted.

The main dogma motivating protein structure prediction is that the three-dimensional structure of a protein is determined by its sequence [HSS92+] and its environment, without any great effect from external factors (i.e. chaperones and enzymes). Secondary structure prediction is a major part of obtaining some structural information from any newly-determined sequence [Ste96].

Why do we need to develop more efficient structure prediction methods? It is important because the sequence-structure gap is constantly increasing and current methods are not accurasies enough. Large scale genome sequencing projects produce a huge amount of sequence data, but structure determination techniques fail to keep up with this sequence production [Rost97].

1.3. Secondary structures

In the protein chain, two amino acids are connected via a peptide bond, where a carboxyl group of the previous amino acid reacts with the amino group of the next amino acid and thus forms the backbone in proteins. Using the peptide bonds, long chains of amino acids can be generated (see Figure 3). The peptide bond is inflexible, but flexibility for rotation is placed around the α-carbon (called the ϕ-angle and the ψ-angle, which together form the ϕψ-space). Combinations of angles ϕ and ψ are restricted to small regions in natural proteins. A protein can fold into a specific three-dimensional structure by using this freedom of rotation [CB00]. Regular behaviour in the combination of ϕ and ψ angles is a requirement for a regular structure, i.e. a regular secondary structure element in the protein backbone. However, not all secondary structures are regular. The appearance of a certain secondary structure type in the place of certain amino acid depends slightly on amino acid itself and the amino acid context.

What is the meaning of secondary structures in a protein in a biological environment? At least it is known to be a major factor determining a three-dimensional fold [MBJ01], and three-dimensional information can then provide information for the functions of proteins [BG01].

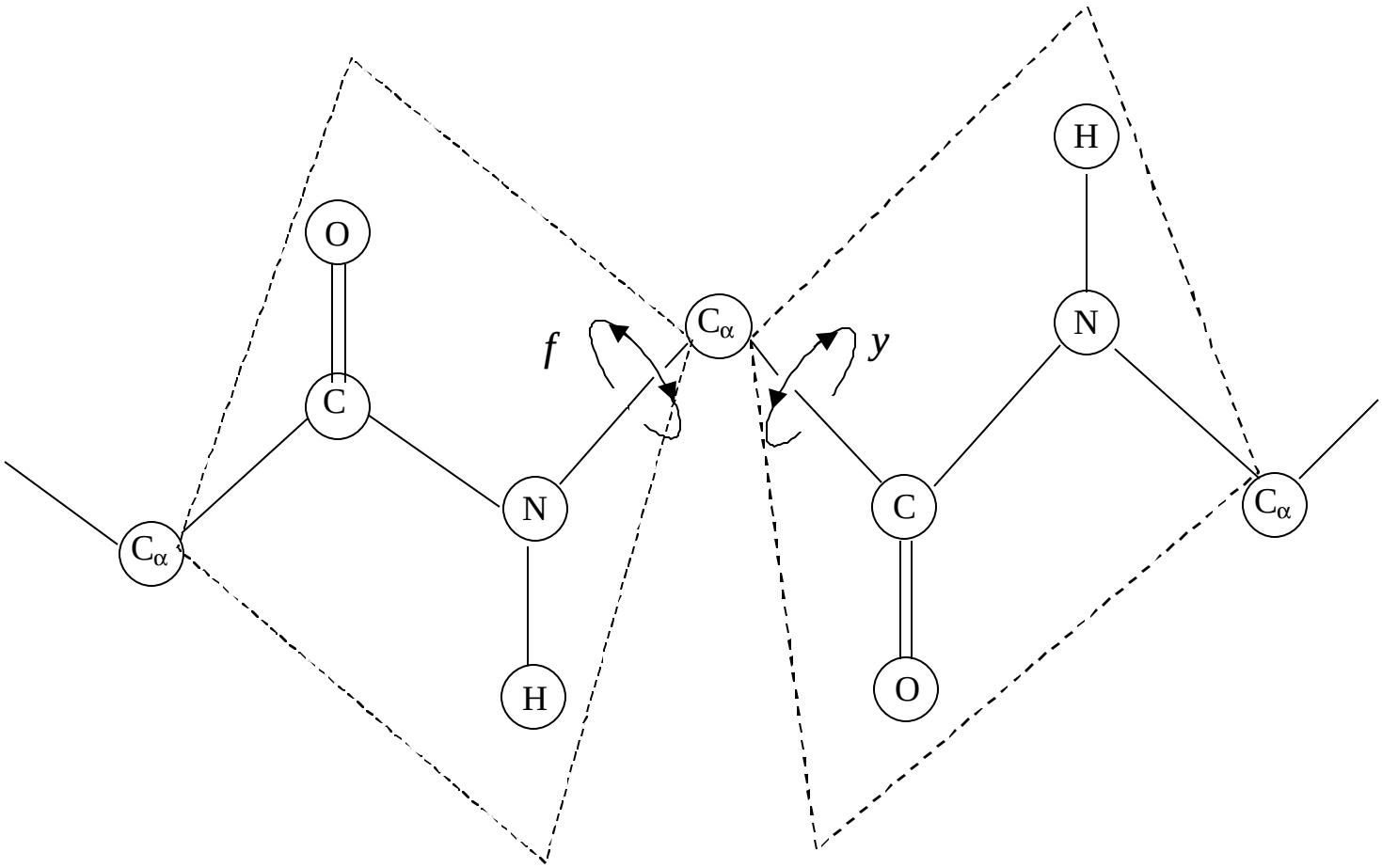

Figure 3. Peptide bonds and torsion angles. Only the bonds around of alpha carbon (C_α) can rotate; the C-N single bonds of the planar peptide groups (broken line) are rigid [Leh79].

For example, the secondary structure α-helix has 3.6 amino acids per turn and appears around point ϕ= -57° and ψ = -47° in the $\phi\psi$ space (see bottom right corner of Figure 2, where there are many α-helix structures in the backbone). In the same way the other structure types appear in the protein backbone by forming structure elements. Every secondary structure has some role for the whole three-dimensional structure of the protein. See Figure 4 and Table 1, listing all secondary structure types in our data set.

Table 1. Names, symbols and definitions for seven secondary structure types.

SYMBOL	T	E	H	G	S	B	PPII
NAME	H-bonded turn	extended strand participates in beta-ladder	alpha-helix	3-10-helix	bend	residue in isolated beta-bridge	polyproline type II
TYPE	regular	regular	regular	regular	irregular	irregular	regular
phi psi	many	-139 +135	-57 -47	-49 -26		many	-78 +149
phi psi		-119 +113					

In this dissertation the polyproline type II secondary structure (PPII) plays a major role and therefore needs detailed examination. The PPII structure forms left-handed triangular helices and forms a cluster with ϕ and ψ at points -75° and 145°, respectively [AS93]. The length of PPII elements is typically 4-8 residues [AS94]. The PPII structure is rare (frequency 1.26%), but it has special biochemical properties; for example, it has an important role in several signalling pathways [Sho95, SK98, WWS98, Bud99, and McP99]. In theory, a right-handed polyproline I is also possible, but was never detected in nature [Sza97].

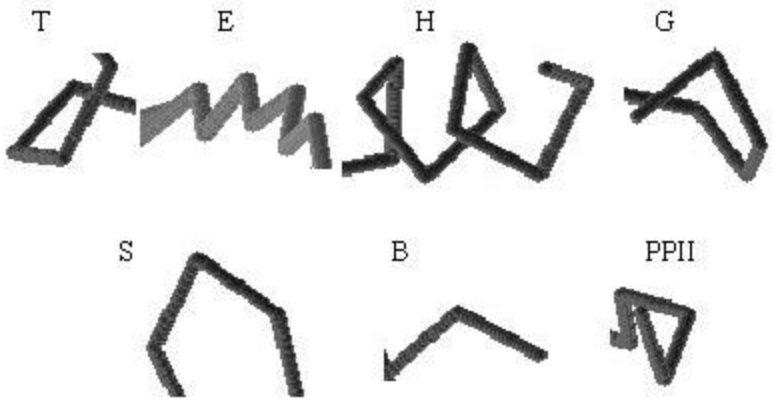

Figure 4. Behaviour of protein backbone with different secondary structure types.

1.4. Machine learning in bioinformatics and secondary structure prediction of proteins

The computational analysis of biological sequences has completely altered the nature of biochemistry since the late 1980s. The new term bioinformatics has been used instead of computational molecular biology for some time now. Basically, bioinformatics is the

analysis of biomedical data. More precisely, bioinformatics conceptualizes biology in terms of molecules and applies informatics techniques to understand and organize the information associated with these molecules on a large scale [LGG01].

Computational tools have become essential components of the research process in the area of bioinformatics. The majority of biological data are inherently noisy. Models must therefore be probabilistic. The methods classify sequences, detect weak similarities, separate protein coding regions from DNA, predict the structure and function of molecules, and reconstruct the evolutionary history etc. [BB98]. All these tasks need efficient computational tools designed to solve specific problems.

The essential theoretical tools in bioinformatics are probabilistic models and information theory. Frequently used algorithms include dynamic programming, gradient descent, expectation maximization, Markov chains, simulated annealing, and genetic algorithms etc. Machine-learning techniques are excellent for the task of discarding and combining redundant sequence information. Currently, widely used machine learning methods in bioinformatics are neural networks, hidden Markov models, several hybrid systems, stochastic grammars and trees. We cannot forget the major role of the Internet, which can provide a huge amount of data and modern tools for analyzing this data over the World Wide Web [BB98].

Neural networks have many uses in bioinformatics. Neural networks are used to predict properties or structures of molecules [e.g. QS88, BBF99+, FC96, RS94, Ros97b, RSR93, SGT99] and are used for biological sequence analysis [e.g. ASB99, FA97, HR96, JS00, PBB90+, SS97, CEB00, OAX97+, CDK00+] and many other purposes [e.g. CC95, KSM92].

The whole three-dimensional structure of a protein can be predicted with the threading method (i.e. fold recognition methods). The technique is based on searching for similar sequences (two similar sequences having the same or relative sequences or subsequences when similarity is high) from a database where there are already structurally known proteins

[RSS97, RE97]. Another method to predict the whole three-dimensional structure is homology modeling. The method is based on searching for a shared evolution history (homology). Similar and homology sequences also provide useful information for secondary structure prediction. Unfortunately, new sequences do not always have similar or homology sequences in the set of structurally known proteins.

Given the crucial role of secondary structures, it is important to review their prediction methods in detail. Early secondary structure prediction methods were based on either simple stereochemical principles (chemical properties of the amino acids and principles of protein architecture [KI02]) or statistics [Jon99]. One of the most widely used, the Chou-Fasman method [CF78] uses an individual statistically defined parameter for every amino acid and for every predicted structure type [PF90].

Nowadays neural networks are very popular tools for predicting the secondary structure of a protein but there are also other approaches. One can make prediction, for example, with a library of sequence-structure motifs [ByBa98], local alignments [SaSo97], support vector machine [HS01], hidden Markov models [LGT98+], linear regression [GGG99+], and decision trees [SML99]. It is also popular to combine modern methods like position-specific scoring matrices with neural networks [Jon99], neural networks and Markov chains [BB98], information theory and pattern recognition [GLG90+], and neural networks with multiple sequence alignment [Ros97c]. The tradition of using external information to improve prediction accuracy is fairly long [DR90]. Research is diversified but complete function between sequence and structure is still missing.

The Critical Assessment of Techniques for Protein Structure Prediction (CASP) is way to evaluate prediction methods. The center has been set up to provide the means of objective testing of these methods via the process of blind prediction. In addition to supporting the CASP meetings their goal is to promote an objective evaluation of prediction methods on a continuing basis [CASP02]. Unfortunately CASP concentrates only on three-state predictions (helix, strand and coil) in the secondary structure prediction category. Our questions consist different class composition.

Many secondary structure prediction methods build models by using a relatively short input window of amino acids, centred at the prediction site. It does not use long-range information and it is therefore a *local method* [BBF99+].

Neural networks need data in numerical form. Therefore categorical variables need modification. The direct sequence encoding method preserves positional information, converts each individual molecular residue into a vector and can only deal with fixed-length sequence windows. It is also possible to use an indirect sequence encoding method that can be used with a varying length sequence and can utilize the overall information measure of a complete sequence string, but at the same time it disregards the ordering information [WM00]. The first work with neural networks in the secondary structure prediction is that of Qian and Sejnowski, who used the same encoding (direct and orthogonal) method as in the NETtalk system. In their network the input layer was arranged in 13 groups. Each group had 21 units, where there were one 1 and twenty 0s. There were 20 units for amino acids and one for spacers. Spacers replaced sequence position where there was no amino acid (i.e. gap in a sequence) [QS88].

Since the first prediction methods, the accuracy of the methods has been important. Prediction accuracy is the number of all cases that get a correct classification divided by the number of all predicted cases. The correct classification means that the method correctly predict secondary structure type that lie in the backbone in the place of a certain amino acid.

Today the best single predictors (PSIPRED and PHD [Jon99, PHD02 and Ros96]) are based on neural network architectures. The secondary structure prediction method is rated at clearly over 70% average accuracy for (water-soluble globular) proteins, in the three states helix, strand, and loop. However, the best early statistical methods (Ptitsyn & Finkelstein) achieved an accuracy of 63%. A method based on sequence similarity (Levin & Garnier) achieved approximately the same accuracy. Early pattern recognition methods achieved an accuracy of 64%.

What actually is a good or bad prediction result? Unfortunately, it is difficult to compare several prediction results to each other. Sternberg wrote aptly: "How good are secondary structure prediction methods? Which of the many secondary structure prediction methods is best? At present there can be no unequivocal answers to these questions, as a number of methodological problems exist that preclude definitive answer. Because of these problems the reader is hereby warned that it is dangerous to read a paper about a secondary structure prediction method, notice its headline accuracy in the abstract, and assume that the method will produce the same accuracy on the protein you are interested in, and that the method is necessarily better than another prediction method with a lower headline accuracy" [Ste96].

In the preceding book Sternberg enumerates reasons for his point of view. Researchers may select widely different subsets of proteins. Test proteins may have an evolutionary relationship to the learning proteins. Secondary structure definitions may vary. Prediction studies may consider different types of secondary structure types [Ste96].

2. Results

2.1 Goals of the research

Throughout this research there have been two main goals: How can better methods be established to predict secondary structures of proteins? How can difficult machine learning problems be detected and encountered?

At practical level, the work began with a question as to whether we can predict rare polyproline type II secondary structures. This was very interesting, because in the literature there was no report describing the prediction of rare secondary structures.

The second question was, how to obtain information from a taught multilayer peceptron neural network. This was interesting, because supervised neural networks are understood as black boxes that cannot give causes for a certain decision. We also consider how we can detect difficult learning tasks. We encountered this question, because a published method behaved inappropriately for our results.

Encoding problems are very important in the area of machine learning. We tried to develop a good encoding method for amino acid sequences that takes chemical properties of amino acids into account and, at the same time, saves used memory.

Neural networks have problems with rare secondary structures. Therefore, we tried to find another way to look at the secondary structure prediction problem. The final questions were, how secondary structure types behave in the space of local sequence information, whether behaviour explains the prediction problems, and how we can use this information to build better prediction methods.

2.2. General issues of the research

The papers in this dissertation describe our work in the area of the protein secondary structure prediction. The papers look at the area from several viewpoints. We produce general methods for the machine learning field, we use our methods and already known methods to produce solutions for bioinformatics, and uncover properties which affect the context of the protein secondary structure prediction.

Figure 5 presents the structure of the research process. Papers I-IV deal with polyproline type II (PPII) secondary structure prediction problems. Paper I presents a hard preprocessing project on the database selection to the prediction task. Paper II presents details on the neural network prediction work and describes the results. Paper III considers problems as to whether we can get some information from a taught neural network and how we can contemplate complicated sequence space and learnability in this way. Paper IV describes our solutions to the sequence encoding problems. Papers V and VI deal with all known secondary structure types in the Dictionary of Protein Secondary Structure (DSSP). Paper V presents a new general algorithm that can make accurate predictions for all secondary structure types. Paper VI is a description of the work in which we tried to uncover the behaviour of the secondary structure types in the sequence space.

Every paper in this dissertation can be presented under the claim that it supports the observations detected in the current research. The claims are given in the titles of the following subsections.

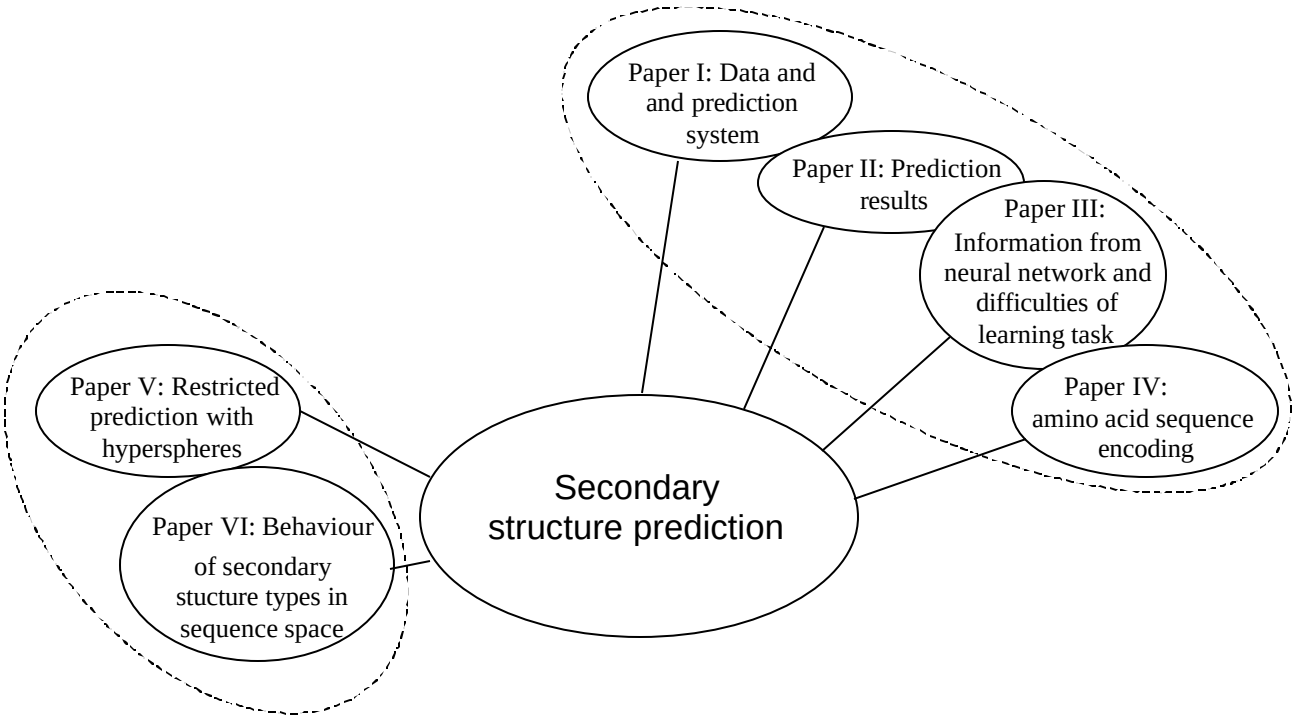

Figure 5. Structure of the dissertation. The papers deal with different problems in the field of secondary structure prediction. The first four papers describe how neural networks behave with PPII secondary structure data, while the last two papers concentrate on the data with all known secondary structure types.

2.3. A huge amount of data but not enough (Paper I)

Our work with the PPII secondary structure predictions started with a data preprocessing project. A great amount of protein structure data were available in digital form. In the protein data bank [PDB, BWF00+] there was structure information for about 8000 macromolecules in autumn 1998.

The database had information on molecules that are not proteins. There was also redundant information and some proteins with low resolution. Proteins with these properties were not suitable for our purpose.

Inside protein families (sets of proteins having the same evolutionary background) there was still too high sequence similarity (perhaps also high sequence identity), which means that there was also too high structural similarity. If in the data set there is too high similarity, prediction accuracy may be distorted. We therefore used the PAM250 substitution table [DSO78] and the pairwise sequence comparison of Needleman and Wunch [NW70] to reduce similarity from protein families. Inside the families we compared all sequences against others and if an identity value was higher than the limit 0.65, we discarded a protein with lower resolution (i.e. the accuracy of biochemical measurement was lower). For large families especially, the method was slow, but fast enough for our calculations.

Three-dimensional information of proteins was a suitable format in the DSSP files [KS83]. Information in DSSP files based on amino acids streochemical properties. There was sequence information, atom coordinates, torsion angles, and several secondary structure information for known secondary structure types. We used torsion angles as did Azhubei and Sternberg [AS93, AS94] to locate PPII secondary structures from an amino acid sequence. By using these files we set the structural conditions for torsion angles that differentiated PPII and non-PPII cases.

After that, no further use was made of torsion angles and no other structural information; machine learning techniques must find deviating characters of PPII and non-PPII sequences. *This is absolutely fundamental to the secondary structure prediction.*

By using a common windowing technique (see for example [PBB90+]) the global information disappeared, but there was no other choice because we used neural networks (multilayer perceptron neural network (MLP)) for the learning task and this approach needed a certain number of input attributes. We used Matlab neural network package to practical prediction work [DB92, DB98].

Although we found about 4000 rare PPII occurrences for an MLP, the situation was problematic. Naturally, the remaining windowed sequences (more than 300 000) belonged to the non-PPII class and the distribution was markedly non-uniform. The MLP needs the

same number of learning cases for all classes [Swi96]. Therefore, we had few choices to equalize the sizes of classes [KM97, KHM98]. In this situation it was better to prune either non-PPII classes or multiply PPII classes. It was reasonable to multiply PPII cases and this way to take better account of the characters of non-PPII classes (i.e. sample stratification [CEB00]). Multiplying was impossible, because orthogonal encoding methods [BB98] were used. Encoding needed 20 input nodes for one amino acid and one sequence needed a vector whose dimension was 260 (13 times 20). Matrices for both classes would include about 150 - 160 million elements. Thus, the only solution was to prune non-PPII cases. Despite the pruning work, the matrices that included the whole learning material had over two million elements, which was enough for Matlab environment. However, the sequence space and, consequently, the 260-dimensional binary vector space were almost empty. These problems are the consequence of infrequent PPII occurrences, used neural network implementation and the huge size of the space.

We measured internal and external distances for PPII structures. The internal distance means the Hamming measure, where for all PPII cases the distances to the other PPII cases were measured. In the same way external distances mean distances from the PPII cases to the non-PPII cases. It showed that inside the PPII class there were only slightly more relative cases than outside the class.

Amino acid proline greatly affects classifications, as frequencies show. Prolines occur over 4 times more frequently in the middle of PPII sequences than in the middle of non-PPII sequences. Proline also accumulates around the middle part of PPII sequences.

Swingler provided a method to determine learnability values for the machine learning data set [Swi96]. The method uses Shannon's information theory. Learnability value can be understood so that if both classes have identical cases, conditional entropy between classes is high. Our results showed that data was easy to learn in this sense. Nevertheless, it was not convincing, since an MLP could not learn perfectly with our learning set. The method did not take advantage of the situation of the cases in the object space. Therefore, data with high Swingler's learnability value may be easily learnable or not. Moreover, when a taught MLP

is tested with a separate test set, the situation is more complicated and data with high learnability do not tell us how difficult the prediction problem is. On the other hand, poor learnability is probably a strong message of difficult data. This is not the only criticism of this method (see for example [VJP01+]). Our solution to learnability value is presented in paper III.

2.4. Badly distorted distribution causes problems for prediction (Paper II)

Theoretical considerations showed that the distribution between classes in a learning set should be almost uniform [Swi96, DB91]. This is quite easy to carry out, but unfortunately, this operation ensures that an MLP learns a uniform distribution. This action does not prove that the method works with test cases where a naturally rare phenomenon has the natural distribution. This action only proves that an MLP learns some characters from the learning set and supposes that it learns some common features that also affect in a test set. Therefore we used a uniformly distributed test set and achieved quite good prediction accuracy.

However, we also tested MLPs with naturally distributed test sets. MLPs learned from uniformly distributed learning sets that the PPII class uses about a half of the object space (the reasons for this become clear in Paper VI). With a test set the method tried to classify cases with almost the same ratio; the number of false positives increased dramatically and the prediction accuracy of the small class decreased.

Swingler also considered this problem [Swi96] and gave a solution. His method calculates *a posteriori* probabilities for an MLP that were learned from a uniformly distributed learning set. The method calculates a value that takes network output values, frequencies of the classes and actual class probabilities into account. The method outputs *a posteriori* probabilities for input cases that respect the natural distribution. Unfortunately, the method did not give improvements. Actually, the sensitivity of the method prevents all PPII predictions. Therefore all test cases were classified into the non-PPII classes - the method did not work in this situation.

After publishing the Paper II we tried our simple *frequency compelling* solution on PPII secondary structure prediction. It predicts a test set in a normal way and then arranges outputs of MLP (and input cases) in ascending order. Suppose that N is the number of cases in the test set and f is the frequency of the class of interest in the learning set. Then select Nf cases, thereby obtaining the best prediction values to get classification for the class of interest. The method worked slightly better than Swingler's method, but the best prediction accuracy was only 8.5% and the method found 11% of PPII cases. We can summarize that there are many cases in the non-PPII class that resemble PPII cases too closely. However, with uniformly distributed learning and test sets prediction accuracy was 73.7%.

The rule of thumb for the MLPs stated that the number of learning cases should be about ten times the number of connections inside a network [Bis95]. In the network used with one hidden layer the size of the input layer was 260 and the number of hidden nodes range between 2 and 25 (the number 4 gave the best result). With 4 hidden nodes there were $260 \cdot 4 + 4 \cdot 2 = 1048$ connections in the network. Our data set was slightly smaller than that requirement.

We tested several numbers of hidden nodes and found that an MLP with a relatively small number of hidden nodes gave the best results. This was surprising, because the data seemed to be so complicated. Moreover, in the literature there were results that an MLP without hidden nodes predicted typical protein secondary structures almost as well as a network with hidden nodes [QS88, RSR93]. We can conclude that there are no strong non-linearities in data sets for protein secondary structure prediction.

It is clear that PPII structures were difficult to predict, but we found some positive aspects from this problem. These problems led us to develop a hypersphere machine learning method (Paper V) that was not disturbed by uniform distributions. We can also point out that the prediction accuracy relative to the density is higher (2.6 times with the basic MLP and 6.7 times with frequency compelling method) than, for example, for α-helix in conventional prediction context.

2.5. A neural network may explain the causes for decision (Paper III)

Our work with MLPs inspired us to search for explanations for MLP decisions. How can we find those characters in the input vector that cause a marked effect in the network and advocate some classification? These are not new questions in the history of MLPs (see [Swi96]) but we tried to find better and simpler answers.

A spectrum of an MLP shows how strongly each value of a categorical variable affects an output node of an MLP. The method requires that variables are encoded to a binary vector as in the secondary structure prediction of protein, i.e. all values of the variables have their own input nodes. Moreover, all classes need their own output nodes. The algorithm is simple; the method inputs vectors when there is one 1 and others are 0 to the MLP. Input 1 is delivered to each input node one by one and after every such input the method considers network output nodes. The spectrum is a set of output values and results are easy to visualize.

As expected, amino acid proline had a strong influence on the PPII classification (as in [AS93, AS94]). It is somehow surprising that there are several amino acids in an exact place that resists the appearance of a PPII structure decision. The method does not give much more information on the data than a simple frequency consideration. However, deviations between the frequency and spectrum could be an indication of a nonlinear relation between input variables. See the half mirror ("puolipeili" in Finnish) example in [Sie99].

The response analysis of the MLP is more advanced than the spectrum analysis. The MLP with one or more hidden layers can form non-linear classification surfaces in the variable space. There may be situations where certain values of certain variables jointly affect classification greatly, but not alone. The response analysis tries to find these variables and certain values of variables by means of the MLPs learned. These variables and values are possibly the generators of the phenomenon that we are interested in.

For the MLP we used a binary vector input that can be transformed into the natural amino acid sequence of a certain length (called permitted input). The method with a genetic algorithm generates permitted inputs to the MLP and, for every input, checks the values of the output nodes. If the output nodes showed strong classification, the input was accepted for refinement (*switch off* state in the original algorithm). The refinement tries to elucidate the phenomenon by eliminating unimportant inputs.

The method showed that a strong PPII decision comes out of the MLP almost exclusively due to proline. Therefore, we can conclude that the PPII structure can originate in the backbone, when there are prolines in and around the middle part of the sequence. This was already known, but a new observation was that remote (four residue) amino acid S slightly strengthens the appearance of PPII.

Swingler describes Pilkington's method of capturing information from an MLP [Swi96]. He suggests that non-linearities can be expressed by calculating the correlations between each input value and the sensitivity of each output unit at each point in the variable space. If calculations produce non-zero correlations, there are non-linearities in the data.

Swingler criticized Pilkington's method, because for a non-linear relationship it is possible to lead to a zero correlation. After that he presented his own solution to solve the problem why a certain classification was done on an individual input vector. The method was based on the variance of the partial derivatives of each output unit with respect to each input unit as random input patterns were presented to the network. The method can tell why certain output value was given, can help in altering the inputs to achieve a desired output, can tell about decision boundaries and provide extra confidence in a network output by providing an explanation of how that output was produced [Swi96].

To summarize, Swingler's method is based on using a difference in certain output units relative to a difference in certain input units and helps to find areas of the input space where there is high or low non-linearity. A difference between Swingler's and our methods is that without the derivative and other such considerations, our methods concentrate on finding if

there are values of certain variables that strongly affect the classification (i.e. generators of a phenomenon).

2.6. Scattering is a learnability indicator that takes the positions of cases into account (Paper III)

How can we recognize difficult learning tasks? Swingler's information theoretical method considers whether the learning cases are identical and belong to different classes [Swi96]. On the other hand, computational learning theory concentrates on different questions. How many training examples does a learning system need to converge to a successful hypothesis, how much computational effort does a learning system need to converge to a succesful hypothesis and how many training examples will a learning system misclassify before converging to a successful hypothesis [Mit97]?

The preceding methods do not take into account positions on the cases in the variable space. We believe that positions of cases are especially important for prediction accuracy when learning methods generalize decision rules throughout the whole variable space and then try to predict unseen query cases. For these problems we developed a new method to measure the learnability of the dataset; we call it a scattering value. This simple method takes into account information theoretically problematic learning tasks and also positions of cases in the variable space. We can consider the results via the parameter and linechart that uncover, for example, clusters of a single class.

More specifically, our technique considers whether the classes are easily separable in variable space. If so, the generalization is more reliable. In other words, our methods recognize situations which require a complicated function from variable space to the set of classes. Swingler's method considers a number of positions in spaces where a certain point is mapped to several classes but does not pay attention to the complexity of the function.

First, the method randomly selects a case from the dataset, removes it, and starts to build a queue in which to put the class label of the case. Second, it selects the nearest case (used

distance concept in PPII secondary structure predictions is the number of amino acids that differ at the same positions) from the previous one, removes it and still continues to build the queue until there are no cases in the set. If there are several nearest cases, it randomly selects one from among them. Third, the technique calculates a learnability value from the queue. The algorithm reads the queue from begining to end and compares a current class label to the previous one and sums all situations where labels deviate. The learnability value is the number of changes between classes in queue divided by the number of the theoretical maximum of changes. The algorithm could travel through the variable space several times, but is almost independent of the starting point (i.e. reliable indicator).

We computed a scattering value of the data that includes PPII and non-PPII cases and used the Hamming distance metric. The results showed that the classes are badly mixed with each other, because in the learning set the scattering value was 0.38, which for uniformly distributed classes is 0.5. We also tested cases which the MLP classified as PPII cases (true positive and false positive). For these cases the scattering value was 0.71 - they were totally jumbled together.

We also performed the scattering process for every known secondary structure type from the data set (same data set as described in Papers V and VI). The result, as expected, was quite bad. The scattering ratio was 0.53, which means that the nearest case came more probably from a class other than its own - the situation concerning our data set was very problematic.

The scattering method is common in the machine learning field and gives valuable pre-information on a data set.

2.7. A space-saving method that may include external information (Paper IV)

It is quite surprising that the original orthogonal coding method for MLPs [QS88] is so popular. The orthogonal encoding method uses 21 or 20 input nodes for one amino acid. However, this grows markedly along with the number of connections in the network.

Wu and McLarty presented several sequence encoding methods in [WM00]. There are methods available that do not need so many input nodes for one amino acid as does orthogonal coding. The methods simplify amino acid information by using lower dimensional vectors to present, for example, hydrophobity categories. They even use evolutionary information via the PAM substitution table [DSO78]. The use of the substitution table is fairly reasonable, because it includes information on how common replacement of amino acid *i* by amino acid *j* is in a natural amino acid sequence [BB98].

We designed new methods to include several kinds of information for the sequence encoding task. The method was tested with evolutionary information from the PAM250 substitution table [DSO78]. It could be reasonable to use several types of pre information, because it helps to generalize, i.e. it may bring remote sequences closer in the huge, muddled and almost empty space.

Our real value encoding method requires amino acid distance values in some sense (for example, evolutionary distances). We tested the real value coding method with evolutionary information for the PPII secondary structure prediction task. In a way, the word “real” refers more to the property of the natural presentation than to real values (from R^1, R^2, R^3, …).

The substitution table PAM250 [DSO78] includes relational evolutionary information instead of distances. Therefore, we transformed the relations to the distances using a genetic algorithm. The fitness function of the genetic algorithm [Gol89] maximizes negative ”correlation” between the PAM250 matrix [DSO78] and (randomly initialized) distance matrix. Another condition for the fitness function is that amino acid distance to itself must be zero.

When the distance information between amino acids is known, the real value coding method can be used. First a dimension for vectors to present one amino acid is selected. We used 2, 3 and 5 dimensional presentations (cf. the orthogonal method uses dimension 20 for one amino acid). The method randomly selects 20 floating point numbers with a previously

selected dimension to present each amino acid. Floating point presentations are optimized with the genetic algorithm to present positions that maintain target distances in the selected space. The evolutionary information from the PAM250 [DSO78] substitution table is conflicting. Therefore the target distances generated were conflicting. Thus in the Euclidean space the method could not achieve positions as the target distances required.

We tested 3 and 5 dimensional encodings to the PPII secondary structure prediction. The same sets were used as with the orthogonal encoding methods (in Papers I, II and III). First, sequences were encoded into a new form. Second, an MLP was taught to separate PPII from non-PPII sequences. Unfortunately the results were not quite so good as with the orthogonal coding method. Probably the PAM250 [DSO78] does not have enough information for the rare PPII structure. In the future it will be possible also to include other information such as electrostatic, hydrophobic, binding etc. (see for example [AF92], [MF98]). Therefore, our method can be seen as a framework to adjoin different pre-information sources to the sequence presentation (see for example [NHH00]).

2.8. Increasing of prediction accuracy has its price (Paper V)

It was already known that high identity (or similarity) between two natural protein sequences indicates that there is also some similarity between their secondary structures [Ros97b]. Let a distance between two windowed sequences be a number of amino acids that differ at the same positions. By using this distance concept we computed the probability that the same secondary structure would be located in the middle of the sequences. This analysis showed that there was an area in the sequence space around the point (a natural windowed amino acid sequence), where the probability of detecting points from the same class is high and decreases as the distance increases.

This analysis suggests that we can use the sphere around the point to make predictions. Actually, this leads to a nearest neighbours-like method. The method forms somewhat similar decision boundaries as a restricted Colomb energy neural network [Jut97]. The method requires some learning and it can leave query cases without classification. That is a

consequence of the learning algorithm that determines a radius around each case in the learning set that extends only halfway from the nearest enemy (i.e. a case that comes from another class). This procedure may leave unclassified areas in the used space where query cases cannot have any classifications. The method was called hypersphere, because it builds spheres over the learning cases according to the radius. Despite being an individual solution, the hypersphere method is common to all machine learning fields.

Figure 6 illustrates the functioning of the hypersphere algorithm in a two-class case. The circles represent class 1 and crosses class 2 respectively. At the bottom of Figure 6(*a)* the algorithm has detected case *c*, which is the nearest enemy to the case *j*. Next, distance *e* is measured and radius *r* is calculated to form a sphere over case *j*. Radius *r* must be a half of the distance *e*, because the likelihood of predicting an unknown case correctly is greater when it is closer than *e/2*. At the top of Figures 6(a), (b) and (c) there are two cases which are at the same location, but come from different classes. For both the cases the radius is zero, because the distance to the nearest enemy is zero.

In Figure 6(b) the algorithm has calculated two hyperspheres which together form a complex volume in the object space. Finally, algorithm 1 has completed with the class in Figure 6(c) and a very complex volume has been built over the cases that came from this class. Every query case that belongs to this volume is classified into class 1. The broken lines in 2(c) approximate a decision boundary of an MLP (or other machine learning method). When an MLP method meets two cases at the same location (at the top of Figure 6(c)), with the winner takes all method it has to make a compromise and both cases are classified into the same class. The hypersphere method avoids decision-making in this area by restricting generalization.

The data set was different from that of the MLP predictions for the PPII structure. We did not separate learning and test sets at the level of a protein, but separated test cases from data after the windowing task. This simplifies a crossvalidation task. Unfortunately it may increase the number of identical sequences between the test and learning sets if several parts

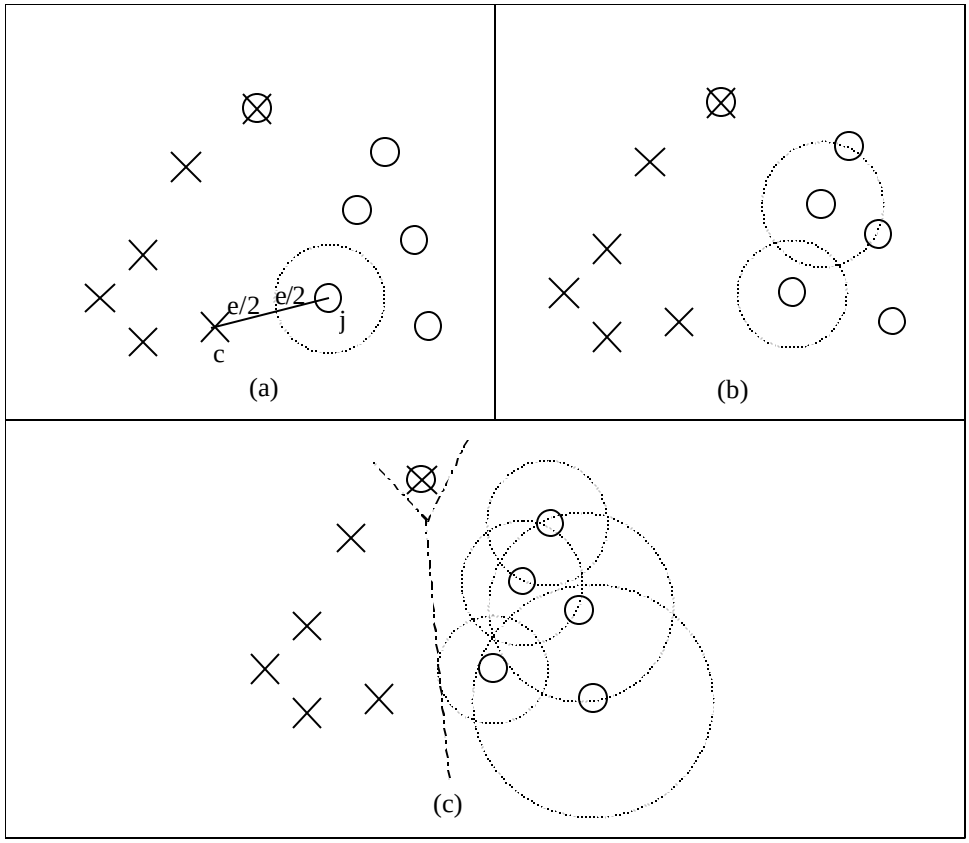

Figure 6. Example of how the hypersphere method builds a complex decision volume in 2-dimensional Euclidean space. Length of radiuses are half of distances to the nearest case that come from another class.

of a protein are in a data set. The fact is that in the PDB there are many proteins that are made up of several identical parts of sequences and therefore there may be several similar chains in a protein. We used only the first part of a protein in the data.

The hypersphere prediction results had special properties and cannot be directly compared to the prediction results of ordinary methods. The method can leave query cases without a class label. With protein sequence data, the method leaves about 70% of cases without the classification. If classes are easily separable in the variable space, the hypersphere method probably does not leave any cases without classification. However, only one noisy case in the condensation of classes can disturb the method and affect the “enemy area” in the wrong place. All in all, this property is very useful with protein sequence data, because excessive

generalization affects the decreasing of the prediction accuracy. Therefore, the prediction accuracy with this method was excellent if we are interested in the certainty of the classification. Haykin wrote that MLPs construct global approximations and can therefore generalize in regions of the input space where little or no training data are available [Hay94]. It is clear that the prediction accuracy of MLPs decreases in such regions. Therefore, it is advisable to avoid generalization just as hyperspheres do, but the penalty is that cases could be left without classifications.

There is some correlation between the sizes of classes and the prediction accuracy (see results from Paper V and Figure 7). Only PPII does not belong to the regression line. Our hypothesis for correlation is that noisy enemy points decrease the prediction accuracy and this occurs to a more considerable extent in small classes. The hypothesis for the behaviour of the PPII structure is that the structure is sensitive to amino acid proline and therefore it is more scattered over the sequence space than others are.

Despite the fact that the prediction results are not quite comparable, we have a method that is accurate and can be used with all secondary structure types. In other words, the method is not disturbed by skew distributions. Average prediction accuracy approached 90% (weighted prediction accuracy Q_8 was 93.5%), which is high. Missing rate results also show that in the test set there are only few cases that are situated in the “enemy area” - only PPII sequences caused some exceptions. To summarize, the results showed that there can be some way to raise prediction accuracies above the pointed upper limit (see upper limit consideration for α-helix from [HC92]).

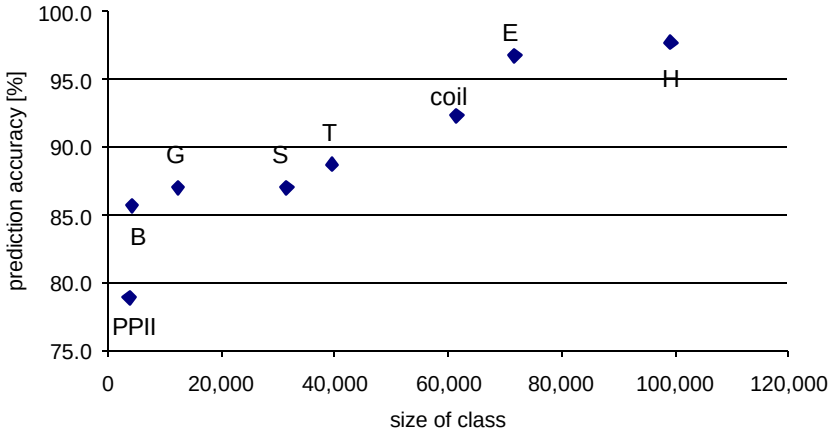

Figure 7. Scatter plot for size of secondary structure sets versus prediction accuracy. See Table 1 for full names of secondary structures.

2.9. Sequences of a certain length build a huge space that is almost empty and very disordered with secondary structure types (Paper VI)

Baldi and Brunak wrote: “The set of all amino acid segments of length 13, where the central residue is in a helical conformation, is scattered over a very large part of the sequence space. The same holds true for other types of protein secondary structures like sheets and turns.” They continue: “The different structural categories are typically not found in nicely separated regions of sequence space; rather, islands of sheets are found in sequence regions where segments preferably adopt a helical conformation, and vice versa.”[BB98] Thus, incoherence between secondary structure types was realized in the prediction attempts.

Paper VI introduced computational and theoretical methods to reveal how the protein secondary structure types behave in a sequence space (i.e. amount of incoherence). The size of volume can be seen as an indicator of incoherence of a class. The simplest method to consider the size of volume is to compute the sum of the internal distances inside the

individual classes. For every class the algorithm sums distances from every case to the others. Two other methods for the size of volume are Gaussian kernels [Jut97] and hyperspheres. The methods capture a part of the sequence space around cases and compute the size of this part for every secondary structure type by using a simple numerical integration method. The theoretical methods described in Paper VI also show how our data set behaves in the sequence space. The methods are: expectation values of the distance inside of class and the theoretical distances to the nearest case. The methods utilize a technique where we study whether some interesting phenomenon is randomly formed. This is investigated by generating an artificial random process (formula).

The theoretical expectation value of distances inside classes gave information on what the average distance is inside a class if cases are uniformly distributed over the sequence space. The results showed that in nature average distances deviate only slightly from the situation in which cases are uniformly distributed in the space.

Results with the nearest case analysis showed something interesting: in the natural data set many of the nearest cases seek their way to the distances that can be achieved when the cases are scattered uniformly over the sequence space. Only roughly half of cases come closer than the theoretical value - many cases behave as if they were generated by chance.

Measured volumes gave more information about the structures of the classes in the space and also gave some explanation as to why the prediction work is so difficult. The clear evidence of the problems in the prediction work is that the sequence space of length 13 is almost empty and this is very problematic if the cases are almost uniformly distributed. In the database there were about 324,000 cases. Neverthless, there was approximately 75% of unused space.

The computational methods look at the space from different viewpoints. Therefore, the results deviate slightly from each other. The greatest difference is that the internal distance of PPII is the smallest, but it used the most space when a volume was measured with Gaussian kernels and hyperspheres. This discrepancy needs an explanation. We speculate

that the difference is a consequence of our distance concept and the structural sensitivity of amino acid proline. For PPII it is not so important where the sequence is in the space. It is important that the prolines are in the middle of a sequence or around the middle part. The high frequency of the proline also affects weak closeness between PPII sequences and therefore internal distances are the smallest. On the other hand, PPII sequences are so far from each other that the methods with numerical integration cannot detect this weak closeness.

Other secondary structure types behaved more normally than PPII. The results of structure B (residue in isolated β-bridge) were good; probably it would be the best structure to predict if test and learning sets are uniformly distributed. Unfortunately, structure type B is rare and therefore the major classes inhibit its prediction. The secondary structures H (α-helix) and E (β-strand) are almost equally compact. Therefore, we can conclude that well known difficulties in predicting structure E (in conventional three-state prediction) are partly caused by its frequencies (see next chapter).

For the windowed protein data and secondary structure classes it seems clear that there is no strong organisation of the classes related to the whole space (i.e. higher level organization). Rather, we demonstrated in Paper VI that the organisation of the classes in the neighbourhoods of individual natural sequences (lower lever organisation) is strong. This situation is problematic for conventional prediction methods that construct a global approximation over the input space.

In conventional secondary structure predictions α, β, and coil structures are included in the learning set and the frequencies of this classes are roughly 30%, 20%, and 50% respectively. Conventional prediction results can be understood as we presented in Paper VI. Approximately 50% of the cases are within the theoretical distance (as shown in the nearest case analysis) and, therefore, they follow lower level organisation (we can suppose the same ratio between test and learning sets). These 50% of the cases can be predicted correctly. The remaining 50% do not follow either higher or lower level organization and the correct classification of these cases depends only on chance (described in the next

section). The conventional method correctly guesses 25% of coil cases from among cases that do not lie inside the theoretical distance (by guessing the accuracy is 0.5^2) and in the same way 9% (0.3^2) for type α and 4% (0.2^2) for type β. Therefore, the method correctly predicts 38% of cases that do not follow the lower level organization. The prediction accuracy is then 50% + 0.38 · 50% = 69% for the whole test set. This explanation is appropriate for the accuracy of the conventional prediction methods if we suppose that they use local sequence information, form a global decision surface over the input space, and do not use external information (for example, alignment information).

3. Basic perspective on the prediction

Upper limit considerations of the prediction accuracy of some secondary structure types are considered in the literature (see for example [HC92]). The secondary structure prediction accuracy depends on data, definitions and structure types. In addition we contend that the lowest limit is also important. With class frequencies it is possible to calculate the lowest limit for prediction accuracy. The term lowest limit refers here to what is obtained by guessing classifications for the query cases. When we know the lowest limit, it is easier to consider how good the prediction method actually is. This is useful not only with secondary structure prediction, but also with other classification and prediction tasks.

Consider the situation where we have a data set of cases from classes C_1, C_2, …, C_n and the frequencies of the classes are f_1, f_2,…, f_n, respectively. Symbol $|C_i|$ refers to the number of cases that belong to the class C_i, where i goes from 1 to n. What is the prediction accuracy if we guess classification for all cases in the data set? Select arbitrary $|C_i|$ cases to belong to class i and other cases belong to the other classes. The number of correct guesses for class C_i is $f_i|C_i|$ (we can suppose it to be independent of other classes and guessing order). Therefore, the total accuracy and theoretical lowest limit lw in percents is

$$
\begin{aligned}
lw &= 100\ \% \cdot \frac{f_1 |C_1| + f_2 |C_2| + \ldots + f_n |C_n|}{|C_1| + |C_2| + \ldots + |C_n|} \\
&= 100\ \% \cdot \frac{f_1 |C_1|}{|C_1| + |C_2| + \ldots + |C_n|} + \ldots + 100\ \% \cdot \frac{f_n |C_n|}{|C_1| + |C_2| + \ldots + |C_n|} \\
&= 100\ \% \cdot \sum_{i=1}^{n} f_i^2 .
\end{aligned}
$$

This consideration offers some useful results for this dissertation. For example, the lowest limit for two classes whose sizes are the same is 50%. For the natural distribution in the PPII structure prediction the lowest limit is about 97.52%, because the majority class is quite easy to predict with high accuracy, but for the PPII class guessing decreases accuracy down to 0.016%. For the original three-state prediction, α-structure (30% of data), β-structure (20% of data), and coil (50% of data) the lowest limit is 38%. Our hypersphere

prediction with eight classes gave a lowest limit of 20%. In the prediction task where the method tried to separate α-helix from the others [HC92], the lowest limit was 58%.

It is not easy to find an upper limit of the prediction, because with several "tricks" we can increase accuracy (for example leaving cases without any classification). However, there always exists the theoretical lowest limit that could very easily be achieved and should be taken into account as an individual learning task into account, when we consider the efficiency of the prediction methods. It should be noted that if we try to get below the theoretical lowest limit, we must use the information that can also be used to exceed the lowest limit.

We can “put the methods on the same line” and calculate the prediction efficiency value. Let *a* be an achieved accuracy. Furthermore, *lw* is the lowest limit and 1 is a correct prediction. Then the prediction efficiency value *e* is

$$e = \frac{a - lw}{1 - lw}.$$

The ratio falls within the interval [0,1]. Values that are near 1 describe good efficiency and numbers that are near value zero mean low efficiency. Hyperspheres yielded an efficiency value of 0.87. The original three state predictions (used prediction accuracy was 75%) gave an efficiency value of 0.60. Our best MLP prediction result for PPII gave a value of 0.26.

4. Discussion and current understanding

During the process of this dissertation the opinions of the author have matured regarding the interpretations, meanings of our methods, results achieved and also the whole secondary structure prediction challenge. This section concerns the present state of things.

The extent of the difficulty in secondary structure prediction depends on the relation, how much torsion angles reflect on the amino acid chain, or how much a single amino acid or amino acid context determines what the torsion angles in the backbone are. We already know and it is a well known fact, that proteins are structurally similar if the sequence similarity is high, but they may have structurally similar parts even when the sequence similarity is low [BB98]. This situation simply tells us that if we find similar sequences, the structures are similar. Conversely, it tells us nothing if the sequences are not similar. With small data set this compels us to generalize somehow, but at the same time, generalization is a step into the unknown.

We can also consider the effect and meaning of regularization conditions for torsion angles. In PPII prediction work we check the torsion angles for false positive test cases. There are many cases where torsion angles are within to the permitted area for a PPII structure, but the regularization conditions (conditions for torsion angles to get the similar values in single secondary structure element) are not fulfilled. Obviously, some amino acids or certain amino acid contexts were determined for torsion angles to be the permitted area of PPII, but somehow the angles did not behave in a regular way. Thus, the effect of regularization conditions for torsion angles decreases prediction accuracy. Another and more important question is whether the secondary structures still work in a biological way even if regularization conditions are not fulfilled, but torsion angles are within PPII area - this remained open.

The first prediction results in the natural distribution of the PPII structure revealed the “character of Nature”. Without any balancing method a large class was easy to predict

accurately, but a small one was “hidden” behind the majority class. The posterior probability of Swingler was too sensitive for the rare class and did not give any PPII classifications because there are no large PPII clusters in the sequence space. The frequency compelling method forced an MLP to classify as many PPII cases as the frequency imposes. Accuracy was over 8%, but not as much as we expected. The results showed that the classes are extremely difficult to separate in input space. Therefore, we can conclude that an MLP does not work well with rare secondary structure types.

One can conceive of MLPs as black boxes. We showed that it is possible to “ask” them what the input level reasons are for making some classification. The orthogonal sequence encoding method allows us to input every sequence that can be formed in the sequence space. With the genetic algorithm we found that prolines in the middle and around the middle part of sequences greatly affect PPII classification. There can also be seen some indicators that amino acid S, from three or four residues away, could favour an occurence of the PPII structure.

The scattering value pays attention to the situations of cases in the variable space. Markedly separate classes obtain low scattering values in range [0,1] and vice versa. Our data set for the PPII prediction task had a high scattering value. Thus, a great part of cases of the secondary structure classes was scattered over the sequence space. We also detected this fact in the prediction work.

The next attempt to improve the prediction results was to concentrate on sequence encoding problems: how can a categorical variable be changed to the numerical forms that take properties from nature into account? We produced the real value coding method. The prediction results for the PPII structure with a uniformly distributed test set achieved almost as good prediction as with orthogonal coding method. This was a positive surprise and led us to assume that the method works even better in another sequence presentation context.

Tests showed that if we draw nearer to some natural sequence of a certain length, the probability of representing the same secondary structure type increases. It is easy to form a

sphere around the case, and this way capture some part of the space as do our hyperspheres. It should be pointed out that in Paper V there are also prediction results for the 1-nearest neighbour algorithm in a naturally distributed test set with eight classes. This arrangement produced much better prediction results for PPII than did our best MLP. Our explanation is that the 1-nearest neighbour algorithm is more independent of frequencies than is an MLP. It was a little surprising that the old and simple methods work better than the newest and most advanced one. This situation also allows us to hypothesize that if we iterate the *n*-nearest neighbour method and increase *n* at every iteration, we find fewer and fewer cases of rare classes. Hypersphere learning exploits exactly this property to achieve a good prediction accuracy for rare classes, too.

Many of our results (scattering, nearest case, volumes) let us conclude for the windowed protein data and secondary structure classes that the higher level organization (i.e. organization of classes related to the whole space) in the sequence space is weak, but the lower lever organization (i.e. organization of classes in neighbourhoods of individual natural sequences) is strong. This situation is very problematic for methods that construct a global approximation over the input space, but can be in control with local methods that avoid excessive generalization.

The results of conventional secondary structure predictions (i.e. α, β, and coil, frequencies 30%, 20%, and 50% respectively) can be understood by means of the nearest case analysis (in Paper VI), lowest limit equation (Chapter 3) and previous organization consideration. The nearest case analysis showed that approximately 50% of the cases are within the theoretical distance. We can suppose the same ratio between the test and learning sets. Thus, these 50% of the cases (type A) follow the lower level organization with the other cases and can be predicted correctly. The other 50% (type B) do not follow either higher or lower level organization and correct classification for these cases depends only on chance. Therefore, the method can correctly predict approximately 69% of the whole test set. This consideration leads to the same level of accuracy as the average accuracy of the prediction methods that use local sequence information, form a global decision surface over the input space, and do not use external information.

The sequence space of a certain length has special properties that do not appear in the Euclidean space. For example, if we move from the position (sequence) S1 to the position (sequence) S2 in the space and the distance between S1 and S2 is greater than 1, there are two or more different shortest paths to walk this route (i.e. a way to change S1 to S2). Therefore, the direction is a more complex concept in this context than in the Euclidean space.

The methods developed can be used in a great part of the machine learning field and in the field of bioinformatics. For example, we planned to build a server that provides hypersphere predictions via the Internet. We expect that the method will be useful in the protein structure prediction context and also help biochemists in experimental structure solving work.

It is time to ponder over the more profound questions that deal with Nature itself. Local sequence information does not provide final answers for protein function prediction, structure prediction or even secondary structure prediction. Future systems should understand more about the biochemical conditions pertaining in the protein folding process. They must understand more about forces at the atomic level. Then we would be in a better position to predict three-dimensional structures of macromolecules that would allow us to make better predictions for the function of proteins. This may enable the development of better drugs, and we may even face an organism reconstruction problem (see [Kan98]).

To summarize, this dissertation presented several viewpoints on the secondary structure prediction problem. The original arrangement (rare PPII secondary structure and neural network) led to a more profound question of dealing with a neural network, learnability, sequence encoding, machine learning, and behaviour of all secondary structures in the sequence space. Neural networks seem to obey too slavishly the laws of frequencies. They worked with PPII secondary structure prediction in the case where the learning and test sets were artificially balanced. Neural networks with the genetic algorithm can reveal nonlinear interaction between input variables. Sequence encoding methods called real value coding can include much pre-information. We tested it for PPII with a neural network as secondary

structure prediction with evolutionary knowledge. We developed a new machine learning algorithm that also accurately predicts rare secondary structure types. This simple method was accurate because in the sequence space, secondary structure types do not form large clusters. Rather, around an individual case (sequence) there is a sphere with a high probability area for the same secondary structure type. The sequence space of a length of 13 seems to be almost empty and the organization of classes related to the whole space is weak. Therefore, machine learning methods have problems especially in predicting rare structure types.

References

[AF92] Avbelj F, Fele L: Role of main-chain electrostatics, hydrophobic effect and side-chai conformational entropy in determining the secondary structure of proteins, Journal of Molecular Biology 279 (1998) 665-684.

[Alm97] Almeida L: Multilayer peceptrons, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. C1.2.1-30.

[And97] Andersson J: Foreword, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. G4.1:1-9.

[Arb97] Arbib M: Neurons and neural networks: the most abstract view, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. B1.1:1.

[AS93] Adzhubei A, Sternberg M: Left-handed polyproline II helices commonly occur in globular proteins, Journal of Molecular Biology 229 (1993) 472-493.

[AS94] Adzhubei A, Sternberg M: Conservation of polyproline II helices in homologous proteins: implications for structure prediction by model building, Protein Science 3 (1994) 2395-2410.

[ASB99] Allex C, Shavlik J, Blattner F: Neural network input representation that produce accurate consensus sequences from DNA fragments assemblies, Bioinformatics 15 (1999) 723-728.

[BB98] Baldi P, Brunak S: Bioinformatics, The MIT Press, Cambridge 1998.

[BBF99+] Baldi P, Brunak S, Frasconi P, Soda G, Pollastri G: Exploiting the past and the future in protein secondary structure prediction, Bioinformatics 15 (1999) 937-946.

[BG01] Bock J, Gough D: Predicting protein-protein interactions from primary structure, Bioinformatics 17 (2001) 455-460.

[Bis97] Bishop C, Neural network for pattern recognition, Clarendon Press, Oxford 1997.

[Bud99] Buday L: Membrane-targeting of signaling molecules by SH2/SH3 domain-containing adaptor proteins, Biochimica et Biophysica Acta 1422 (1999) 187-204.

[BWF00+] Berman H, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, Shindyalov I, Bourne P: The protein data bank, Nucleic Acids Research 28 (2000) 235-242.

[ByBa] Bystroff C, Baker D: Prediction of local structure in proteins using a library of sequence-structure motifs, Journal of Molecular Biology 281 (1998) 565-577.

[CASP02] Critical Assessment of techniques for protein Structure Prediction, CASP, http://predictioncenter.llnl.gov/ (loaded Feb 2002) 2001.

[CB00] Clote P, Backofen R: Computational molecular biology, John Wiley & Sons, Ltd 2000.

[CC95] Cai Y, Chen C: Artificial neural network method for discriminating coding regions of eukaryotic genes, CABIOS 11 (1995) 497-501.

[CDK00+] Cai D, Delcher A, Kao B, Kasif S: Modeling splice sites with Bayes network, Bioinformatics 16 (2000) 152-158.

[CEB00] Choe W, Ersoy O, Bina M: Neural network schemes for detecting rare events in human genomic DNA, Bioinformatics 16 (2000) 1062-1072.

[CF78] Chou P, Fasman G: Prediction of the secondary structure of proteins from their amino acid sequence, Advances in Enzymology and Related Areas of Molecular Biology, 47 (1978) 145 - 148

[CHK93] Carlson L, Hyvönen E, Karanta I, Syrjänen M: Tekoälyn ensyklopedia, Gaudeamus, Helsinki 1993 (in Finnish).

[DB91] DeRouin E, Brown J: Neural network training on unequally represented classes, Martin Marietta Corporation, 1991.

[DB92] Demuth H, Beale M: Neural network toolbox for use with Matlab, The Math Works, 1992.

[DB98] Demuth H, Beale M: Neural network toolbox for use with Matlab, The Math Works, 1998.

[DD01] Ding C, Dubchak I: Multi-class protein fold recognition using support vector machines and neural networks, Bioinformatics 17 (2001) 349-358.

[DR90] Deleage G, Roux B: Use of class prediction to improve protein secondary structure prediction, in Fasman G (ed.), Prediction of protein structure and the principles of protein conformation, Plenum Press, New York and London 1990, pp. 587-596.

[DSO78] Dayhoff M, Schwartz R, Orcutt B: A model of evolutionary chance in proteins, matrices for detecting distant relationships, in Dayhof (ed.), Atlas of protein sequence and structure, Vol. 5, National Biomedical Research Foundation, Washington DC, 1978, pp. 345-358.

[Eth02] ETH: Eidgenössische Technische Hochschule Zürich: Solid-state nuclear magnetic resonance, http://www.nmr.ethz.ch/research_projs/structure.html (loaded Feb. 2002), 2000.

[FA97] Frichman D, Argos P: A neural network for recognizing distantly related protein sequences, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. G4.1:1-9.

[FC96] Faricelli P, Casadio R: HTTP: a neural network-based method fo predicting the topology of helical transmembrane domains in proteins, CABIOS 12 (1996) 41-48.

[Fie97] Fiesler E: Topology, in Fiesler E, Beale R (eds), Handbook of neural computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. B2.2:1.

[Gar01] Garian R: Prediction of quaternary structure from primary structure, Bioinformatics 17 (2001) 551-556.

[GGG99+] Guermeur Y, Geourjon C, Gallinari P, Deléage G: Improved performance in protein secondary structure prediction by inhomogeneous score combination, Bioinformatics 15 (1999) 413-421.

[GLG90+] Garnier J, Levin J, Gibrat J, Biou V: Secondary structure prediction and protein design, in Kay J, Lunt G, Osguthorbe D (eds.) Protein structure, prediction and design, The Biochemical Society, 1990, London.

[Gol89] Goldberg D: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading 1989.

[Hay94] Haykin S: Neural networks: A comprehensive foundation, Prentice Hall, London, 1994.

[HC92] Hayward S, Collins J: Limits on α-helix prediction with neural network models, Proteins 14 (1992) 372-381.

[HR96] Hanke J, Reich J: Kohonen map as a visualization tool for the analysis of protein sequences: multiple alignments, domains and segments of secondary structures, CABIOS 12 (1996) 447-454.

[HS01] Hua S, Sun Z: A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach, Journal of Molecular Biology 308 (2001) 397-407.

[HSS92+] Haug E, Sand O, Sjaastad Ö, Toverund K: Ihmisen fysiologia, WSOY, 1992.

[Jon99] Jones D: Protein secondary structure prediction based on position-specific scoring matrices, Journal of Molecular Biology 292 (1999) 195-202.

[JS00] Jagla B, Schuchhardt, Adaptive encoding neural networks for the recognition of human signal peptide cleavage sites, Bioinformatics 16 (2000) 245-250.

[Jut97] Jutten C: Supervised composite networks, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. G4.1:1-9.

[Kan98] Kanehisa M: Grand challenges in bioinformatics, Bioinformatics 14 (1998) 309.

[KHM98] Kubat M, Holte R, Matwin S: Machine learning for the detection of oil spills in satellite radar images, Machine Learning 30 (1998) 195-215.

[KI02] Karolinska Institutet, Medical Biochemistry and Biophysics: http://broccoli.mfn.ki.se/pps_course_96/ss_960723_1.html (loaded Feb. 2002) 2001.

[KM97] Kubat M, Matwin S: Addressing the curse of imbalanced training sets: One-sided selection, in Fiesher D (ed.), Procedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufman, San Francisco 1997, pp. 179-186.

[KS83] Kabsch W, Sander C: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers 22 (1983) 2577-2637.

[KSM92] Katz W, Snell J, Mericel M: Artificial neural networks, Methods in Enzymology 210 (1992) 610-632.

[Leh79] Lehninger A: Biochemistry, Worth Publisher, INC. (1979)

[LGG01] Luscombe N, Greenbaum D, Gerstein M: What is bioinformatics? A proposed definition and owerview of the field, Methods of Information in Medicine 4 (2001) 346-358.

[LGT98+] Liò P, Goldman N, Thorne J, Jones D: PASSML: Combining evolutionary inference and protein secondary structure prediction, Bioinformatics 14 (1998) 726-733.

[MBJ01] McGuffin L, Brysson K, Jones D: What are the baselines for protein fold recognition, Bioinformatics 17 (2001) 63-72.

[McP99] McPherson P: Regulatory role of SH3 domain-mediated protein-protein interactions in synaptic vesicle endocytosis, Cell Signal 11 (1999) 229-238.

[MF98] Melo F, Feytmans E: Assessing protein structures with a non-local atomic interaction energy, Journal of Molecular Biology 227 (1998) 1141-1152.

[Mit97] Mitchell T: Machine Learning, The McGraw-Hill Companies, Inc. New York, 1997.

[MP43] McCulloch W, Pitts W: A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics 5 (1943) 115-133.

[NHH00] Ng P, Henikoff J, Henokof S: PHAT: a transmembrane-specific substitution matrix, Bioinformatics 16 (2000) 760-766.

[NKV89] Niemi M, Korhonen K, Virtanen I: Solu- ja molekyylibiologia, Welin+Göös, toinen laitos, 1989 (in Finnish).

[NW70] Needleman S, Wunsch D: A general method applicable to search for similarities in amino-acid sequence of two proteins, Journal of Molecular Biology 48 (1970) 443-453.

[OAX97+] Ogura H, Agata H, Xie M, Odaka T: A study of learning splice sites of DNA sequence by neural networks, Computers in Biology and Medicine 27 (1997) 67-75.

[PBB90+] Petersen S, Bohr H, Bohr J, Brunak S, Cotteril R, Fredholm H, Lautrup B, Training neural networks to analyze biological sequences, Trends in Biotechnology 8 (1990) 304-308.

[PDB] The protein data bank (PDB): http://www.rcsb.org/pdb/

[PF90] Prevelige P, Fasman G: Chou-Fasman prediction of the secondary structure of proteins, in Fasman (ed.), Prediction of protein structure and the principles of protein conformation, Plenum Press, New York and London 1990, pp. 391-416.

[PHD02] PHD prediction server, http://www.public.iastate.edu/~pedro/pprotein_query.html (loaded Feb 2002), 1994.

[QS88] Qian N, Sejnowski T: Predicting the secondary structure of globular proteins using neural network models, Journal of Molecular Biology, 202 (1988) 865-884.

[RE97] Rice D, Eisenberg D: A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence, Journal of Molecular Biology 267 (1997) 1026-1038.

[Ros62] Rosenblatt F: Principles of neurodynamics, Spartan Books, New York 1962.

[Ros96] Rost B: PHD, Methods in Enzymology 266 (1996) 525-539

[Ros97] Rost B: Protein structure prediction in 1D, 2D, and 3D, in: (eds.) Schleyer PvR, Allinger NL, Clark T, Gasteiger J, Kollman PA, Schaefer HF III and Schreiner PR, The Encyclopaedia of Computational Chemistry, 3 (1997) 2242-2255.

[Ros97b] Rost B: A neural network prediction of protein secondary structure, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. G4.1:1-9.

[Ros97c] Rost B: Learning from evolution to predict protein structure, in Olsson B, Lund D, Narayanam A (eds.), Bio-computing and emergent computation, Springer, Heidelberg 1997.

[RS94] Rost B, Sander C: Combining evolutionary information and neural networks to predict protein secondary structure, Proteins 19 (1994) 55-72.

[RSR93] Ruggiero C, Sacile R, Rauch G: Peptides secondary structure prediction with neural networks: a criterion for building appropriate learning sets, IEEE Transactions on Bio-Medical Engineering 40 (1993) 1114-1121.

[RSS97] Rost B, Schneider R, Sander C: Protein fold recognition by prediction-based threading, Journal of Molecular Biology 270 (1997) 471-480.

[SaSo97] Salamov A, Solovyev V: Protein secondary structure prediction using local alignments, Journal of Molecular Biology 268 (1997) 31-36.

[SGT99] Shepherd A, Gorse D, Thornton J: Prediction of the location and type of β-turns in proteins using neural networks, Protein Science 8 (1999) 1045-1055.

[Sho95] Shokat K: Tyrosine kinases: modular signaling enzymes with tunable specifications, Chemistry & Biology 2 (1995) 509-514.

[Sie99] Siermala M: Polyproliini II-sekundaarirakenteen ennustaminen neuroverkoilla, M.Sc. thesis, Department of Computer Sciences, University of Tampere, 1999 (in Finnish).

[SK98] Sicheri F, Kuriyan J: Structures of src-family tyrosine kinases, Current Opinion in Structural Biology 7 (1997) 777-785.

[SML99] Selbig J, Mevissen T, Lengauer T: Decision tree-based formation of consensus protein secondary structure prediction, Bioinformatics 15 (1999) 1039-1046.

[SO97] Suominen I, Ollikka P: Yhdistelmä-DNA-tekniikan perusteet. Opetushallitus, 1997 (in Finnish).

[SS97] Snyder E, Stormo G: Neural networks for identification of protein coding regions in genomic DNA sequences, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. G4.1:1-9.

[Ste96] Sternberg M: Protein Structure prediction, Oxford University Press, Oxford, 1996.

[Swi96] Swingler K: Applying neural networks, Academic Press, London, 1996.

[Sza97] Szapo Z: Polyproline helices, http://www.cryst.bbk.ac.uk/pps97/assignments/projects/szabo/index.htm (loaded Mar 2002), 1997.

[Tay97] Taylor J: The historical background, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. G4.1:1-9.

[Tur97] Turpeenoja L: Biokemiaa, virtsa-aineesta lääkemaitoon, Opetushallitus, toinen laitos, 1997 (in Finnish).

[VJP01+] Viikki K, Juhola M, Pyykko I, Honkavaara P: Evaluating training data suitably for decision tree induction, Journal of Medical System, 25 (2001).

[UTY98+] Ulmanen I, Tenhunen J, Ylänne J, Valste J, Viitanen P: Geeni, Söderström, 1998 (in Finnish).

[Wer97] Werbos P: A traditional roadmap of artificial neural network capabilities, in Fiesler E, Beale R (eds), Handbook of Neural Computation, IOP Publishing and Oxford University Press, Amsterdam, Oxford 1997, pp. A2.3:1-6.

[WM00] Wu K, McLarty J: Neural networks and genome information, in Konopka (ed.), Methods in computational biology and biochemistry, Elsevier, Amsterdam, 2000.

[WWS98] Williams J, Wierenga R, Saraste M: Insights into src kinase functions: structural comparisons, Trends in Biochemical Sciences 23 (1998) 179-184.