TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Genomorientierte Bioinformatik

Prediction of protein structural features by machine learning methods

Andreas Kirschner

Vollständiger Abdruck der von der Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzende: Univ.-Prof. Dr. A. Kapurniotu

Prüfer der Dissertation:

1. Univ.-Prof. Dr. D. Frischmann
2. Univ.-Prof. Dr. I. Antes

Die Dissertation wurde am 01.10.2008 bei der Technischen Universität München eingereicht und durch die Fakultät Wissenschaftszentrum Weihenstephan für Ernährung, Landnutzung und Umwelt am 18.06.2009 angenommen.

to my grandmother

Abstract

Genome sequencing projects continue to reveal the building blocks of life, producing millions of amino acid sequences whose biological roles can be understood only when the structure and function of these proteins are elucidated. Although experimental structure determination methods become faster and cheaper and provide high quality insights, only computational structure prediction methods can satisfy the demand for structural data for the majority of proteins. Protein structures are predicted in various levels of detail: It is approached by the prediction in one-dimension which has the aim to detect local structural regularities like α-helices, β-sheets or backbone turns. The next higher level of detail involves prediction in two-dimensions where the protein contact map is a prominent representation.

Throughout this work an array of machine learning techniques is used to investigate sequence-structure relationships in proteins, while a strong focus lies on neural networks. One important advance made is the development of a novel bidirectional Elman-type recurrent neural network with multiple output layers (*MOLEBRNN*) capable of predicting multiple mutually dependent structural motifs. This computational architecture was successfully applied to develop the currently most accurate predictor of β-turns and solvent accessibility, two important structural and functional features of proteins. The advantage of the method introduced in this thesis when compared to other predictors is that it does not require any external input except for sequence profiles because interdependencies between different structural features are taken into account implicitly during the learning process.

Finally, the first method to identify interacting residues and α-helices in membrane proteins is presented. It is based on the analysis of co-evolving residues in predicted transmembrane regions and the use of neural networks. The neural network approach utilizes both input features commonly used for soluble proteins as well as those specific to membrane proteins only, such as a residue's position within the transmembrane segment or its orientation towards the hydro- or lipophilic environment. The predicted residue contacts were employed in a second step to identify contacting helices with high accuracy.

Acknowledgments

I would like to take this opportunity to thank a number of people who have helped me during the completion of this project.

First of all, I thank my advisor Prof. Dr. Dmitrij Frishman for his years of patience and guidance. He was the most important source of motivation. The way he is constantly generating new ideas is outstanding and inspirational. The trust he put into myself is especially mentioned which made it possible to freely develop methods. He was essential for the success of my dissertation.

I thank my coworker Angelika Fuchs. Her knowledge in biological relations introduced many meaningful aspects into my work. Her talent to brighten the work breaks was important for the whole research group.

I thank Dr. Pawel Smialowski. He was an important motivator who was improving the work by his qualitative questioning. We share many private preferences which led to lively conversations in the office.

I would like to thank Philip Wong. We did various side projects together.

I thank my coworker Thorsten Schmidt for his support during the teaching phases.

I thank Dr. Thomas Rattei who was constantly stimulating my teaching skills. He took over my last summer courses which allowed completion of the dissertation. Additionally he was the one to keep computers running in Weihenstephan.

I thank Prof. Dr. Hans-Werner Mewes for giving me the opportunity to work at his Bioinformatics Chair. I gained huge amounts of teaching experiences which trace back to the exercises we did together.

I thank Roland Arnold, Sindy Neuman, Martin Sturm and Patrick Tischler for many relaxing hours in the Mensa.

I thank my father-in-law, Klaus Hofmann and Johannes Mandel for proof reading the work.

I would also like to thank Daniel Berrar from the University of Ulster. Whenever I ended up in serious problems concerning machine learning topics, it was him that had the answer available.

Also my friends who allowed me moaning about all the work I have had to do and have tried to help me in ways only they could.

I would also like to thank my parents, they made this work possible.

I thank my wife Mildred.

Contents

Chapter 1

Introduction

1.1 Nomenclature

The fundamental notion 'structure' describes the composition or appearance of things that are somehow ordered. Structural biology is a scientific discipline that aims to uncover topologies and shapes in biological macromolecules like proteins. While structural biology traditionally is done in laboratories, the new branch, structural bioinformatics, is done by modeling molecule structures on silicon chips.

Proteins are macromolecules built of chained amino acids. They are essential in organisms and are engaged in all biological processes of cells. They catalyze biochemical reactions (enzymes), are responsible for the shape of biological compounds (structural proteins), take roles in cell signaling or ligand transport, are responsible for immune responses, can store molecules (Casein and Ovalbumin store amino acids), and are even responsible for cell mechanics (for example Actin and Myosin). Proteins occur individually or bound to other proteins or molecules where they form complexes. In general two different types of proteins are distinguished: globular proteins and membrane proteins. Globular proteins occur solute in water while membrane proteins are attached to cell or organelle membranes. Depending on the permanence of membrane association, membrane proteins aggregate mostly in water, they are insoluble.

Biochemists distinguish four different levels of the protein structure. The primary structure represents the sequence of amino acids in a protein, the secondary structure

represents local structural regularities, the tertiary structure is the three-dimensional structure of the protein while the quaternary structure involves protein-protein interactions and represents the three-dimensional structure of protein complexes. In three-dimensional space, where the tertiary and quaternary structures reside, a protein is fully defined after solving the positions of all atoms. On the one hand, the highest level of detail is obtained in this three-dimensional representation, while on the other hand its derivation demands time- and money-consuming methods in the wet laboratory. The cheaper and faster *in silico* approach requires very sophisticated methods which are still erroneous.

To support computational derivation of 3D-structures, less demanding structural features are desired, thereby accepting the drawback of being less informative. Such features reside in lower dimensions; they are one- and two-dimensional. One-dimensional structural features are for example, secondary structure, β-turn and residue solvent accessibility states, features that can be represented by a sequence of structural states associated to a corresponding residue. They reside in a single dimension because a single value is sufficient to position the states in space. A two-dimensional description of the protein structure is encoded in the contact-map. Close amino acid residue contacts, found in the native three-dimensional structure, are modeled in this two-dimensional coordinate system. Both axes of the coordinate system represent the amino acid sequence of a given protein. If two residues at positions i and j of the protein are in contact, the matrix element (i, j) reflects that property. (In Section 1.2.2 the contact-map is described in greater detail. There, Figure 1.2 shows an example.)

The research area 'machine learning' is embedded into the concept 'artificial intelligence', a branch in computer sciences. Machine learning deals with the development of algorithms and methods with the aim to allow machines to 'learn'. The machines learn rules during a training process involving examples, and the learned rules can then be applied to examples not available during training. Machine learning techniques are organized into branches where the two most common include supervised learning and unsupervised learning. Supervised learning searches for functions by which input data can be mapped to associated output values. The training

data consists of paired elements where input and output values are linked. During learning the algorithm induces a mapping function from the training data. With the found mapping function, examples with unknown output values can be linked to predicted output values. This thesis exclusively deals with supervised learning methods. Unsupervised methods are developed to find rules in data without output values. Here, the aim is to find similarities and dissimilarities between the examples in the data, similar examples are then grouped together.

Famous examples for supervised machine learning algorithms emerge from the concept of artificial neural networks (ANN). ANNs are graphical models inferred from biological neural networks. Being graph based, neural networks consist of interconnected neurons with defined input and output connections. Similar to what is thought to happen in biological neurons, artificial neurons accumulate the input stimuli by gathering the weighted sum over the input connections; if this accumulated input is larger than a certain threshold, the neuron output is activated which influences subsequent neurons. There are infinite possibilities to interconnect the neurons; experts can predefine the connections to model the full and functional neural network.

Having defined the most important terms, now they are put in context and related to this thesis: The goal of this thesis is the development of neural network based machine learning methods specifically tailored to the prediction of structural features of proteins. Two levels of structural complexity are considered, residing in one- and two-dimensional space and globular and membrane proteins are inspected. The task is to significantly contribute to research in structural biology.

1.2 Biological problem

Current life-science research is directed towards the understanding of full biological systems. With the availability of high throughput sequencing techniques, huge amounts of genetic data has been produced (Mewes *et al.*, 1997; Lander *et al.*, 2001; Sequencing and Consortium, 2005). As a consequence deep insights into genomes got available and the wish to understand biological systems arose. Towards under-

standing these biological systems the deciphering of the genetic messages in protein structures and functions is required. Although structural biologists are constantly developing faster and cheaper methods to measure three-dimensional structures from natively folded proteins, there exists a major discrepancy between the amount of available genetic DNA sequences and protein structures. To bridge this gap computational methods are developed. The most important requirement for the realization of structure prediction methods was found 35 years ago when Anfinsen (1973) discovered that the amino acid sequence is dictating the protein structure. Later this requirement was expanded as it was discovered that the native solution environment plays an import role for protein folding (Dill, 1990). Today structure prediction methods are so powerful that good results are obtained and far reaching conclusions get possible (Petrey and Honig, 2005).

To drive research, three-dimensional or tertiary structure is not always necessary, often other, less detailed structural descriptors are enough.

1.2.1 One-dimensional structure

The term 'one-dimensional structure' was invented by Rost and Sander (1994b) and terms projections of three-dimensional structures into one-dimensional sequences/arrays. One-dimensional structure projection has the advantage that users can easily get an overview about important structural elements. While full three-dimensional structure requires software for visualization, one-dimensional structure is viewed in ordinary text editors (Figure 1.1 shows one-dimensional structural features of the first SCOP (Murzin *et al.*, 1995) domain of example protein 1AKM from PDB. 1AKM plays a role in the urea cycle of humans). For example, the solvent accessibility of all residues in a protein can be projected onto a one-dimensional sequence with the characters 'B' and 'E' for buried and exposed. This declares the solvent accessibility state for each residue and gives a rudimentary overview about the residue locations relative to the protein surface (Line `AS1` in Figure 1.1). Secondary structure can be projected onto a sequence with characters 'H', 'E' and 'C' that code for the secondary structures α-helix, β-sheet and irregular, random coil (Line `SS` in Figure 1.1). The elements of the sequence are not required to be characters, float values

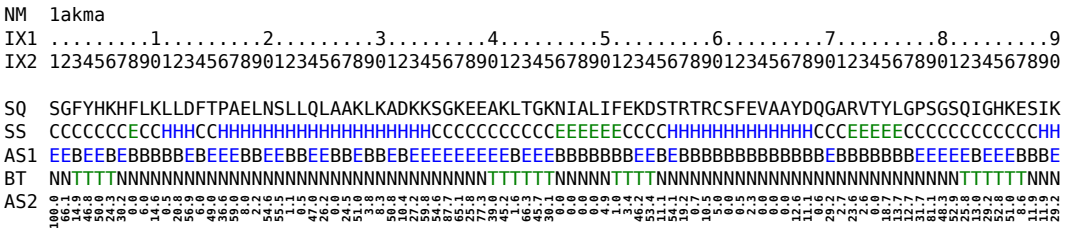

Figure 1.1: Example of a protein with associated one-dimensional structural features
Shown is the first SCOP domain (d1akma1) of pdb entry 1AKM, chain A. First line (`NM`) shows the protein name. Line `SQ` shows the peptide sequence. Line `SS` shows the secondary structure, where 'H' encodes the α-helix, 'E' encodes the extended sheet and 'C' is representing random, unregular coils. Line `AS1` encodes the solvent accessibility state for all residues, here 'E' represents 'exposed' and 'B' buried residues. In this plot, exposed residues have 25% of their surface area solvent accessible. Line `BT` encodes β-turns; the 'T' represents the turn state while 'N' is encoding a non-turn state. The last line `AS2` shows the percentage of solvent exposure for each residue.

are also allowed. For example, a float can describe the percentage of the residue surface that is solvent exposed (Line `AS2` in Figure 1.1). The following structural features can be projected onto 1D sequences by applying geometric rules on the full three-dimensional coordinates of the protein structure: secondary structure, solvent accessibility, presence of β-turn, type of β-turn if present, type of extended sheet (parallel or antiparallel sheet) if present, transmembrane state, direction of transmembrane segment, formation of crystal contact, amount of residue-residue contacts at the given position.

Predicted secondary structure is used for a variety of biologically related analysis and applications (examples in Pasta *et al.* (2003); Gamblin (2005) and Rigden *et al.* (2002)). Predicted solvent exposure of residues is also helpful for biologists (Feder and Bujnicki (2005) and Chmiel *et al.* (2005)). The prediction of transmembrane segments in proteins was for example used by Faça *et al.* (2008).

Sure enough, methods that predict one-dimensional protein features are prominently used within the domain of computational biology and bioinformatics. They are used for analysis purposes, to enhance new upcoming tools and to provide intermediary results. Especially, predicted secondary structure is often utilized. It can enhance sequence alignments (Simossis and Heringa, 2005) and it is used for homology detection (Ginalski *et al.*, 2003). Definitely it helps to improve three-dimensional structure prediction methods, *ab initio* methods (Bradley *et al.*, 2003; Cheng *et al.*,

2005) as well as template based methods (Chen *et al.*, 2006; Chivian *et al.*, 2003).

Additionally, predicted one-dimensional features help to predict other one-dimensional structural features. Examples will follow in greater detail throughout the thesis.

1.2.2 Two-dimensional structure

To represent the distances between all pairs of residues in a protein, the two-dimensional structure representation is used. To do so, the distances between any two residues i and j of a natively folded protein are measured. Next, a contact criteria is employed on these distances: If the distance between two residues i and j is less than a predefined threshold (often 8 Angstrom is used), this pair is in contact. Following this definition, a two-dimensional symmetric matrix is built. It encodes the contact state for all residue pairs ij by boolean matrix elements. Such a matrix is called contact-map. An example is shown in Figure 1.2.

The contact-map of a protein can be used for a variety of applications. Proteins can be superimposed and compared by their contact-map (Holm and Sander, 1996). The contact-maps have been proposed as intermediate between primary structure and tertiary structure where they have effectively served for 3D structure prediction (Bonneau *et al.*, 2002b; Ortiz *et al.*, 1999) and it was shown that contact-maps can be used to reconstruct the 3D coordinates of a protein (Vassura *et al.*, 2008).

A further property of contact-maps is that various structural elements form certain patterns when plotted in two-dimensions. Figure 1.1 shows that the target protein 1AKM has two long α-helices. These helices are represented by the continuous dots parallel to the diagonal (marked Ⓐ and Ⓑ in Figure 1.2). In α-helices, the residues i and $i + 4$ are in contact forming hydrogen bonds and these contacts are responsible for the described pattern. Similarly 1AKM chain A contains two β-sheet regions (positions 46-51 and 72-76) that, when visualized in a contact-map, form pattern Ⓒ: dots arranged in a diagonal that is shifted off the bisecting line. For β-sheets two topologies exist. They can interact in parallel and antiparallel manner. The β-sheets in our example are parallel bound. For antiparallel bound β-sheets the dot trace of the contact pattern would be oriented vertically on the bisecting

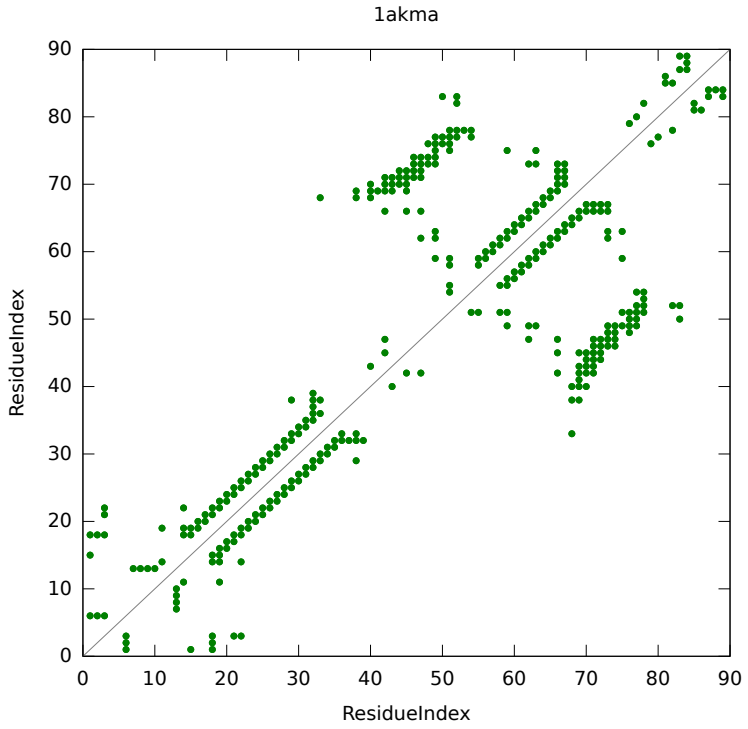

Figure 1.2: Example of a two-dimensional representation of a protein structure
Similar to Figure 1.1 the first SCOP domain of pdb entry 1AKM chain A is shown. If the spatial distance between the C_β atoms of two residues is less than 8 Angstrom a dot is plotted. Neighboring residues on the sequence are omitted in the plot as they all meet the contact criteria. Marked are patterns that emerge from significant structural features: Pattern Ⓐ and Ⓑ originate from α-helices, Ⓒ emerges from parallel β-sheets and the contact resulting from a β-turn is emphasized by Ⓓ. Note: The contact-map is symmetric.

line. The symbol Ⓓ in Figure 1.2 corresponds to a β-turn. Phenylalanine at residue position 51 (Phe51) forms a contact with the aspartic acid at position 54 (Asp54) which results in the β-turn. This contact is recognized and can be observed in the contact-map.

One application of contact-map prediction will be presented in this thesis: Predicted contacts between α-helical transmembrane residues are used to obtain helix-helix interaction patterns.

Membrane protein topologies are often interpreted as two-dimensional. The first dimension results from the sequence position and the second from the location in the environment which can be intra-cytoplasmic, extra-cytoplasmic and integral. Knowing the transmembrane regions allows visualization in a two-dimensional plot. Com-

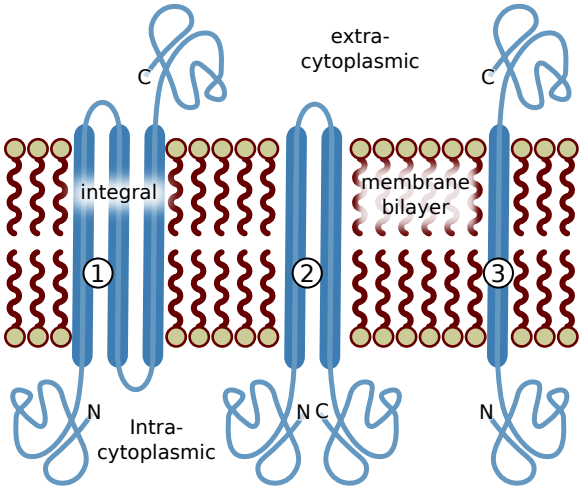

Figure 1.3: Possible topologies of membrane proteins.
The protein labeled with ① has three transmembrane regions. Its N-terminal end is located in the cytoplasma, its C-terminal end in the extra-cytoplasmic solution. The protein labeled with ② has two transmembrane regions, both located in the intra-cytoplasmic area and the rightmost protein ③ has a single transmembrane region.

pare Figure 1.3 where two-dimensional representations of three possible α-helical membrane proteins are shown.

In the process of this thesis another definition for two-dimensional membrane protein structure was developed and defined. Contrary to the side view onto the α-helical transmembrane protein (compare Figure 1.3), a view from above, from outside of the membrane is introduced. For this novel perspective of membrane proteins, not the locations of transmembrane segments are crucial, but the interactions between them. A graph based scheme is introduced that models transmembrane segments as vertexes and segment contacts as edges. This representation gives a better structural overview and allows further analysis and structural classification of membrane proteins.

1.2.3 Three-dimensional structure

The previously discussed protein domain is shown in cartoon representation in Figure 1.4. Structural biology deciphers protein structures mainly by two methods: X-ray crystallography and NMR spectroscopy. X-ray crystallography, as the name suggests, requires crystals for structure determination. The target protein

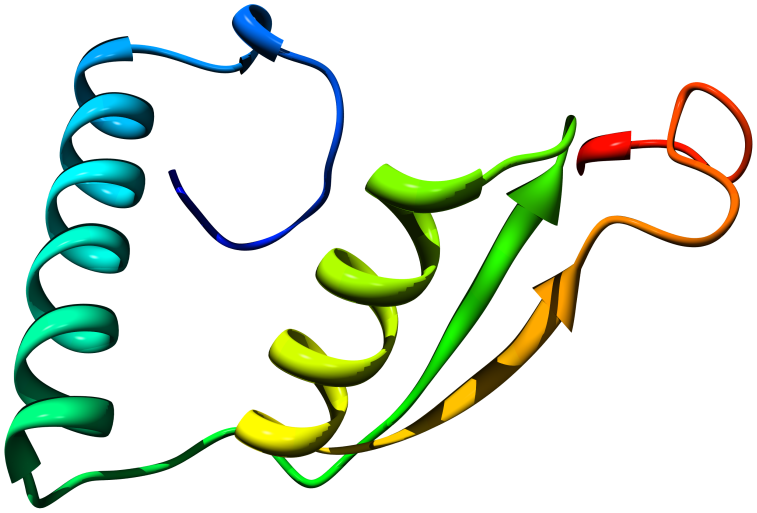

Figure 1.4: SCOP Domain 1akma1 of Protein 1AKM, chain A visualized in three-dimensions.

The protein backbone is colored from its N- to the C-terminal end, following a gradient from blue to red. Please note, the protein is chopped at the red α-helix, after its first SCOP domain.

is dissolved, then the solution conditions are changed gradually. This causes the proteins to gather and subsequently grow to larger crystals. The crystals are then exposed to X-ray where they scatter the X-ray beam into distinct patterns. From these patterns, also called 'reflections', the three-dimensional protein structure is determined by Fourier transformation. Structure determination by X-ray crystallography is limited due to the following reasons: Crystallization must result in large crystals to obtain good subsequent structure resolution. To this end the optimal solution conditions need to be determined which can be very time intensive. Another limitation of X-ray crystallization is, that the protein structure in the crystal may differ from the native structure.

The other important structure determining technique is NMR spectroscopy. NMR structure determination utilizes the magnetic properties of certain atom nuclei. Magnetic fields influence the nucleus spins of the atoms in a molecule. The scan for resonant absorption of electromagnetic radiation results in a spectrum which fingerprints a particular molecule. Applying Fourier transformation on this NMR spectrum achieves the protein structure. NMR spectroscopy is limited to small proteins due to problems in resolving overlapping and broad peaks in the spectrum. Less

frequently used is electron microscopy to determine protein structures. In particular large protein complexes or membrane proteins are analyzed by electron microscopy.

Although the amount of resolved protein structures is rising exponentially[1] the gap between known protein sequences and resolved structures is still diverging. Hence, the need for structure prediction software is still growing and mandatory. Additionally, prediction software is getting more and more accurate which makes them useful to understand phenomena in current molecular and cell biology and allows application for structure-based drug design (Petrey and Honig, 2005). To afford a significant contribution to biological research it is a demand on this dissertation to outperform previous methods. A development in computational biology only has impact in research if it is applied and this is solely achieved when the newly developed methods are better than available standards.

1.3 Computational solution

As deciphered protein structures are important in so many areas of biology and the task is extremely challenging, the prediction has been fascinating researchers for many years. Many attempts have been undertaken to achieve significant advances towards computationally solved protein structures. It was realized that accurate predictions in one and two dimensions are important intermediary steps and this is targeted by this thesis.

1.3.1 State of the art

Prediction for 1D structure can be distinguished in methods that rely on amino acid preferences, methods that exploit similar cases, and methods that purely rely on generalizations derived via machine learning.

2D structure prediction can be subdivided in methods that exploit co-evolution observed for neighboring residues, and again, machine learning methods are utilized.

[1]Protein Data Bank :: Yearly Growth of Total Structures :: http://www.pdb.org

Prediction of one-dimensional structure

Early attempts involved the analysis of amino acid preferences to gain insights into 1D structure. It was discovered that the 20 amino acids have different preferences to occur in different structural states (e.g. Chou and Fasman (1974, 1979)). For example, methionine, alanine, leucine, glutamate and lysine prefer to adopt helical conformations, similarly asparagine, aspartic acid, proline, serine and glycine tend to occur in β-turns. A comprehensive collection of such amino acid propensities responsible for their occurrence in various structural environments is found in the AAindex database (Kawashima *et al.*, 2008). In general, utilizing propensities for structure prediction pursues the following strategy: Starting from the N-terminal end of an amino acid sequence, an algorithm is consecutively passing along all residues towards the C-terminal end. All residues and their sequence neighborhood (which span a sequence window) are analyzed with respect to preferences for structural states. If the average preference of the window is larger than a predefined threshold, the winning structural state is assigned. In the early days only the single target sequence was considered for prediction while later approaches incorporate evolutionary related sequences to derive a weighted outcome (e.g. Fuchs and Alix (2005); Kloczkowski *et al.* (2002)).

The next class of methods utilizes databases with known structural assignments to derive similar cases for structure prediction. Segments of the target sequence are searched in the database and the structure of the best matching database sequence is taken as prediction. Prominent methods realizing this method were described from Yi and Lander (1993); Frishman and Argos (1996); Salamov and Solovyev (1997).

The third type of prediction methods employs big databases of known structure conformations together with machine learning methods like Artificial Neural Networks (ANN), Support Vector Machines (SVM) or Hidden Markov Models (HMM). The very first to utilize an ANN for secondary structure prediction were Qian and Sejnowski (1988). Highly accurate ANN methods for one-dimensional structure prediction are developed until now: Jones (1999); Pollastri *et al.* (2007); Pollastri and McLysaght (2005); Rost *et al.* (2004) predict secondary structure, Kaur and Raghava

(2004); Shepherd *et al.* (1999) predict β-turns, and Adamczak *et al.* (2005); Ahmad *et al.* (2003b) predict solvent accessibility. Among these ANN methods, two types of connection schemes are used which result in either feedforward neural networks or recurrent neural networks. While feed-forward networks are limited to local sequence segments, recurrent networks overcome this limitation and consider larger sequence neighborhood to predict local structural features. The recurrent network type is implemented in this thesis for prediction in 1D. Recently researchers try to replace ANN methods by SVM methods. These are thought to be unsurpassable but according to the no-free-lunch theorems (Wolpert, 2001) this can not be stated in general. Nevertheless the SVM has gained remarkable contributions in the field of one-dimensional structure prediction, see for example: Kim and Park (2004, 2003); Nguyen and Rajapakse (2005); Ward *et al.* (2003); Zhang *et al.* (2005).

Hidden Markov Models are also used to predict one-dimensional structure; mostly secondary structure (Aydin *et al.*, 2006; Martin *et al.*, 2006) and membrane protein topologies (Krogh *et al.*, 2001; Käll *et al.*, 2007) are predicted.

Prediction of two-dimensional structure

Probably the first method to explicitly predict residue contacts was published by Göbel *et al.* (1994), based on co-evolving residues. Co-evolution at residue level is observed when one residue in a protein mutates and causes a mutation of another residue. This second residue may be nearby in structure where it is influenced directly or it may reside somewhere else in the protein globule. However, the latter case is not fully understood, researchers refer to functional relationships (e.g. both residues are situated at interaction pockets) or non-obvious structural reasons emerging during protein folding. An example for co-evolving amino acids: A small, buried residue mutates to a larger residue; this causes a mutation of a nearby, large residue to a smaller one to clear potential steric stress. Any residue property can cause a co-evolution event to preserve the proteins structure or function. Göbel *et al.* (1994) presented a method that allows to quantify co-evolution of residue pairs by correlation coefficients. In a straight forward case, large values correspond to more likely contacts. Following that theory, the residue pairs can be ranked with respect

to the size of the calculated correlation coefficients. A predefined amount of highly ranked pairs is then selected for putative contacts.

Contact prediction methods based on residue co-evolution are continuously developed. Examples are Fleishman *et al.* (2004); Kundrotas and Alexov (2006); Olmea and Valencia (1997); Shindyalov *et al.* (1994).

As stated above, machine learning methods are also used to predict residue-residue contacts (Fariselli *et al.*, 2001b,a; Pollastri and Baldi, 2002; Punta and Rost, 2005a; Shao and Bystroff, 2003; Cheng and Baldi, 2007). Correlated mutations are used together with other features related to residue contacts and evaluated by machine learning methods which in general results in a higher predictive performance. Current contact prediction methods for soluble proteins gain precisions as high as 30% (Cheng and Baldi, 2007).

Although prediction of residue contacts in soluble proteins is quite advanced by now, there was no transfer of methods to contact prediction in membrane proteins.

1.3.2 Contributions of this dissertation

In the course of this thesis three methods have been developed that contribute to the area of protein structure prediction. Two methods predict one-dimensional structures of soluble proteins and one method was developed for the prediction of two-dimensional structures in membrane proteins. They have in common that all are neural network based. While contact prediction is done by a simple feedforward neural network, one-dimensional structure is predicted by a novel multi output layer Elman-type bidirectional recurrent neural network (*MOLEBRNN*). As it will be seen, the novel *MOLEBRNN* is able to simultaneously predict multiple structural aspects in 1D and, because of that, it is able to better tackle the prediction problems.

1.4 Organization of this thesis

Chapter 2 and 3 focus on the novel multi output layer Elman-type bidirectional recurrent neural network *MOLEBRNN* used for 1D structure prediction. In Chapter 2 *MOLEBRNN* is defined and its application on β-turn prediction is discussed.

MOLEBRNN is shown to outperform current β-turn and β-turn type prediction methods. This and the fact that *MOLEBRNN* advances current bidirectional recurrent neural networks are evaluated precisely. Due to the properties of *MOLEBRNN* it is concluded that the method has the potential to be preferably applied on related 1D prediction problems. These need not necessarily adhere to the domain of computational biology.

Chapter 3 discusses the extension of *MOLEBRNN* towards 1D solvent accessibility prediction. *SOPRANO* is introduced which predicts three representations of solvent accessibility in concert with β-turns and secondary structure. It is shown that the underlying *MOLEBRNN* is capable of outperforming current solvent accessibility predictors. Up-to-date secondary structure prediction performances are gained. Further it is emphasized that multiple prediction targets demand higher network complexity.

In Chapter 4 an approach is discussed which tries to exploit the information encoded in nucleotide codons for secondary structure prediction. In the beginning of this chapter various rationales are discussed which underline how the DNA sequence could contribute to structure formation. Concrete application of codon profiles reveals a slight gain in prediction performance of secondary structure.

In Chapter 5 *TMHcon* is introduced which is predicting 2D structure of membrane proteins. This method was developed together with Angelika Fuchs. Based on a neural network, including correlated mutations, common features of globular proteins and specific features intrinsic only to membrane proteins, *TMHcon* is the first method for contact prediction of residues in transmembrane helices. As stated above, its capability lies not only in the prediction of residue-residue contacts, it also integrates the predicted contacts and produces a novel membrane protein representation based on interacting transmembrane helices. This method and the novel membrane protein representation significantly contribute to structure deciphering of membrane proteins. The representation allows analysis of structural features leading to a novel scheme to classify the membrane protein fold space which is targeted in the near future. Membrane proteins with unknown structure may be arranged into that fold classification by *TMHcon*.

Chapter 6 summarizes the work presented, gives an overview on additional, but not successive experiments, brings the treated aspects into context and finalizes the thesis.

Chapter 2

Prediction in 1D :: Prediction of β-turn and β-turn types by *MOLEBRNN*

2.1 Introduction

β-turns are defined as reversals in direction of the polypeptide chain consisting of four consecutive amino acid residues, with the first and the last residue situated in close proximity to each other and the two central residues not being part of an α-helix (Venkatachalam, 1968). β-turns are classified into nine different types based on the dihedral angles of their two central residues (Hutchinson and Thornton, 1996). Local interactions in β-turns play an important role in initiating protein folding and stabilizing protein structure (Zimmerman and Scheraga, 1977). Approximately every fourth amino acid residue in globular proteins is found in a β-turn (Kabsch and Sander, 1983), and most of the β-turns are located on the protein surface (Rose *et al.*, 1985) where they are often involved in intra-molecular binding, cleavage, and posttranslational modification events. In particular, the role of β-turns in antigen recognition and antibody binding has been documented (see for example Hinds *et al.* (1991); Rini *et al.* (1993)). The binding specificity and sensitivity may depend on a particular turn subtype (Bach *et al.*, 1996; Li *et al.*, 1999).

The evolution of β-turn prediction methods closely followed the developments in the area of protein secondary structure prediction. Early approaches (Chou and Fasman, 1979; Hutchinson and Thornton, 1996) relied on β-turn type dependent position specific potentials for each residue derived from known three-dimensional structures of proteins. In particular it was found that β-turns tended to be enriched in hydrophilic residues owing to their frequent solvent exposure (Rose *et al.*, 1985). Zhang and Chou (1997) extended this simple approach by considering residue correlations between positions $1-4$ and $2-3$ in the turn tetra-peptides. Fuchs and Alix (2005) additionally weight β-turn propensities according to evolutionary conservation of respective residue positions.

The second major group of methods is based on machine intelligence algorithms that learn the mapping from the amino acid sequence to the residue β-turn propensity when trained on a database of known conformations. Neural networks have been widely used for this purpose, starting with the work of McGregor *et al.* (1989). Shepherd *et al.* (1999) applied a two-layer neural network architecture, with predicted secondary structure included as additional information at the second stage. The β-turn prediction accuracy was further boosted by utilizing *PSI-BLAST* derived position specific scoring matrices rather than single sequences as input for neural networks (Kaur and Raghava, 2003b) and the k-nearest neighbor algorithm (Kim, 2004). More recently, support vector machines (SVM) have become popular for sequence-based prediction of structural features, including β-turns (Zhang *et al.*, 2005). Here, too, significant improvement was achieved by utilizing multiple sequence information.

Different types of protein structural features are intrinsically interdependent. Some of them (e.g. α-helices and β-strands) are strictly mutually exclusive while others display a varying degree of correlation (e.g. solvent accessibility and β-turn propensity). It has long been realized that the incorporation of known or reliably predicted information about one structural property can help to improve the prediction accuracy of another target structural motif. Frishman and Argos (1997) enhanced secondary structure prediction by excluding from consideration residue positions with high β-turn propensity. Wood and Hirst (2005) reported a significant

gain in secondary structure prediction accuracy due to the utilization of predicted dihedral angles as additional input. Another structural feature exploited for secondary structure prediction is predicted solvent accessibility (Adamczak *et al.*, 2005). Conversely, Kaur and Raghava (2003b) used predicted secondary structure for more accurate prediction.

In all these cases the output of one or several prediction techniques was used as additional input for predicting a single feature of interest. Up until now there has not been an attempt to predict multiple structural features by a single method in a synergetic fashion. In this thesis a generic computational method is presented capable for the prediction of interdependent structural targets. A novel type of a recurrent neural network architecture is introduced which has multiple output layers: one for each prediction target. The introduced neural network is able to learn mutual dependencies between prediction targets by adjusting the interconnection weights so as to achieve the best possible prediction accuracy for all targets simultaneously. It is demonstrated that this technique is efficient for predicting β-turns and β-turn types in concert with secondary structure.

The initial considerations that lead to the development of the multi-target neural networks trace to the book 'Gödel Escher Bach' of Douglas Hofstadter (Hofstadter, 1999). Hofstadter describes findings in neurosciences that state that the brain stores knowledge in an interconnected fashion.

The here presented multi-target recurrent neural network conforms to the concept of 'multi-task learning' (MTL) (Caruana, 1997). The rationale for multi-task learning according to Caruana (1997) is that it can provide a data amplification effect, it allows for some targets to 'eavesdrop' on patterns discovered at other targets and the network is specially tailored towards underlying global functions. Essentially these three rationales can be attributed to the last one when obeying the following explanation: Consider two prediction targets that are related and both depend on a complicated hidden process that spans the observable prediction targets (Compare a Hidden Markov Model where a hidden Markov chain generates observable features). Further, consider the following example involving a single target prediction method; this method trained in an ordinary way is, to some extend, not able to detect the

underlying hidden process because this hidden process can not be sensed by one single aspect only. In the case of additional prediction targets more features of the hidden process are uncovered and a more global bias is induced. The hidden process is uncovered by the above mentioned data amplification effect, all prediction targets independently relate to the hidden process and thus introduce higher data density. The 'eavesdropping' rationale can also be deduced to the hidden process: in the case when all prediction targets are related and influence each other or depend on each other.

2.2 Material and methods

2.2.1 Dataset or known β-turns and other secondary structure elements

For β-turn prediction a standard dataset is used. It contains 426 protein sequences clustered at the 25% identity level and was described by Kaur and Raghava (2002). Structures of all amino acid chains were solved by X-ray crystallography with a resolution of at least 2.0Å and each contains at least one β-turn segment. The dataset contains the total of 95844 residues. β-turn states of residues were determined from the PDB (Berman *et al.*, 2000) coordinate files by *PROMOTIF* (Hutchinson and Thornton, 1996). Following the generally accepted definitions (Lewis *et al.*, 1973; Richardson, 1981) *PROMOTIF* identifies turns as any four consecutive residues such that the distance between the residues i and $i+3$ is less than 7Å and residues $i+1$ and $i+2$ are not part of an α-helix. Eight different β-turn types (I, II, $VIII$, I', II', $VIa1$, $VIa2$, and $VIba$) are distinguished based on the dihedral angles ϕ and ψ of the residues $i+1$ and $i+2$ (see table 2.1 which corresponds to the Table 1 in Hutchinson and Thornton (1996)). Both angles are allowed to deviate from ideal angles by $\pm 30°$, and one of the angles is allowed to deviate by $\pm 40°$. Any turns that do not conform to the definitions involving dihedral angles are grouped into type IV. The general turn class contains any subtypes.

Secondary structure assignments were obtained by *DSSP* (Kabsch and Sander,

β-Turn Type	$\phi(i+1)$	$\psi(i+1)$	$\phi(i+2)$	$\psi(i+2)$	**Remark**
I	-60°	-30°	-90°	0°	Predicted
II	-60°	120°	80°	0°	Predicted
$VIII$	-60°	-30°	-120°	120°	Predicted
I'	60°	30°	90°	0°	Predicted
II'	60°	-120°	-80°	0°	Predicted
$VIa1$	-60°	120°	-90°	0°	Residue at $i+2$ is a Prolin in cis conformation
$VIa2$	-120°	120°	-60°	0°	Residue at $i+2$ is a Prolin in cis conformation
$VIba$	-135°	135°	-75°	160°	
IV	Turns with angles not listed above				Predicted

Table 2.1: Definitions of ideal β-turn types taken from Hutchinson and Thornton (1996). The first column shows β-turn type names, the following four columns show the values of the dihedral angles ϕ and ψ for the residues $i+1$ and $i+2$. In the Remark column "Predicted" indicates those β-turn types that are predicted in this experiment.

1983). Following the usual practice the eight secondary structure types defined by *DSSP* were collapsed to three states as follows: H (α-helix), G (3_{10}-helix) and I (π-helix) became H, E (strand) and B (isolated β-bridge) became E, and S (bend), T (turn) and C (coil) became C.

2.2.2 Neural network architecture

In this thesis two novel types of neural networks are introduced which are referred to as *EBRNN* (Elman-type bidirectional recurrent neural network) and *MOLEBRNN* (multi output layer Elman-type bidirectional recurrent neural network). Both networks are derived from the Elman network (Elman, 1990), a standard feed-forward network (Figure 2.1, black nodes and connections) in which the nodes in the middle layer are connected via context nodes to themselves (Figure 2.1, green nodes and connections). These additional reverse connections are implemented by feeding the output activations from the middle layer to the so called context nodes. As a result, for each stimulus currently being propagated through the network, the middle layer activations from the previous input are also available. Thus, the standard Elman network only considers information about past events in time or, in the case of amino acid sequences, about residue positions on the left (upstream) from the current position. More recently, Baldi *et al.* (1999) extended a standard recurrent

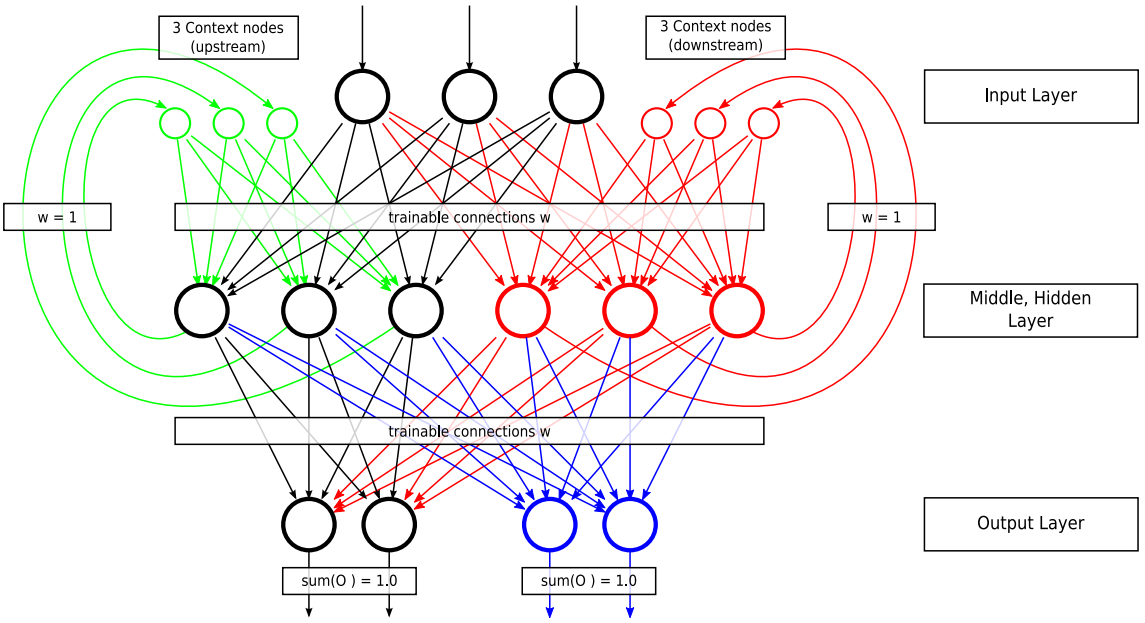

Figure 2.1: The architecture of the various neural network types considered in this work.

Black – a simple feed-forward network; green and black – the Elman recurrent network; green, black, and red – the Elman-type bidirectional recurrent neural network (*EBRNN*); green, black, red, and blue – the multiple output layer Elman-type bidirectional recurrent neural network (*MOLEBRNN*). See Materials & Methods for details.

neural network (RNN) by introducing a second context chain to store information about future (downstream) events. They call their network architecture bidirectional recurrent neural net (BRNN) because it captures both upstream and downstream information. The Elman-type bidirectional recurrent neural network (*EBRNN*) is obtained by adding a future context chain to the standard Elman network (red nodes and connections in Figure 2.1). Using the notation of Baldi and co-workers, the output prediction O_t of the *EBRNN* at a given sequence position t is:

$$O_t = (F_t, B_t)$$

The forward (upstream) context is encoded in the forward context chain F_t (black and green network in Figure 2.1), the backward (downstream) context is encoded in the backward context chain B_t (red network in Figure 2.1).

EBRNN differs from Baldi's BRNN in two respects. First, the recurrent property in BRNN is implemented by backward connections that bypass multiple hidden

layers while in *EBRNN* the recurrent connections follow the original Elman design such that the context nodes are activating the units of the nearest hidden layer. Secondly, compared to *EBRNN* network, Baldi *et al.* not only have the forward and backward context chains but also a third chain encoding the current external input which is being propagated without recurrent connections to the output layer. By contrast, in *EBRNN* the dependence of the output on the input signal is only implicitly encoded in the context chains according to

$$\begin{aligned} F_t &= \phi(F_{t-1}, I_t) \\ B_t &= \beta(B_{t+1}, I_t) \end{aligned}$$

where ϕ and β are functions modeled by the network chains that perform the mapping of the current network input I_t and previous network states F_{t-1} or subsequent network states B_{t+1}. Building upon the *EBRNN* architecture another novel neural network is introduced that is called multiple output layer Elman-type bidirectional recurrent neural network (*MOLEBRNN*). As the name implies, *MOLEBRNN* may have more than one output layer, each for a separate prediction target. In Figure 2.1 blue nodes and connections display a second output layer capable of predicting a second target. This architecture not only allows for simultaneous prediction of multiple target features (e.g. secondary structure, solvent accessibility, turns), but also leads to higher prediction accuracy for individual targets due to its ability to learn interdependencies between individual targets (see the Results section). *MOLEBRNN* is designed for the prediction of classification and regression targets. For each position t in a sequence of length T $(t = 1, ..., T)$ *MOLEBRNN* outputs a two-dimensional vector

$$O_t = \begin{pmatrix} (o_{1,1,t}, o_{2,1,t}, o_{3,1,t}) \\ (o_{1,2,t}, o_{2,2,t}) \\ o_{1,3,t} \end{pmatrix} \quad with$$

$$0 \leq o_{i,j,t} \leq 1 \quad and$$

$$\textstyle\sum_i^{N(j)} o_{i,j,t} = 1 \quad if N(j) > 1$$

where j is the index of a prediction target (e.g., turn, coil, helix). Within the prediction target j, i is an available class (e.g., turn, no turn) and $N(j)$ is the number of classes. If j is a classification target, then $N(j) > 1$, and $o_{i,j,t}$ is interpreted as the likelihood of the class membership. If $N(j) = 1$, the value of $o_{i,j,t}$ denotes the real valued prediction outcome for a regression problem, e.g. real value solvent accessibility prediction. For a classification target j the sum $\sum_i^{N(j)} o_{i,j,t} = 1$ which allows consistent interpretation of $o_{i,j,t}$ as the probabilities for a sequence position t to belong to the class i of the target j. This is achieved by applying individual softmax layers to the classification targets according to

$$o_{i,j,t} = \frac{exp(x_{i,j,t})}{\sum_k^{N(j)} exp(x_{k,j,t})}$$

where $x_{i,j,t}$ is the weighted sum over all input connections in the i-th output unit in the output layer of the j-th prediction target. In case of a regression problem, the output of a single unit within the layer is obtained by the standard sigmoidal activation function. As derived above, $o_{i,j,t}$ can be interpreted as the probability for a given sequence position t to belong to a given class i of a prediction target j.

In general, the class is predicted whose probability is maximal. However, in case of a two class problem with unbalanced class distribution, this may lead to sub-optimal performance. The so-called post-scaling (Lawrence *et al.*, 1998) is therefor utilized which involves the application of an adjustable threshold τ on the network output value of the positive class to filter the prediction outcome. In other words, the positive class (the presence of a β-turn) is only predicted if $o_{i,j,t}$ for this class is larger than τ. By modifying τ it is possible to control the balance between the sensitivity and the specificity of the predictor.

As originally suggested by Jones (1999) the network input data for each amino acid sequence is a position specific scoring matrix (PSSM) together with the gap score and information content of all residue positions as obtained from *PSI-BLAST* (Altschul *et al.*, 1997) searches against the NCBI's nonredundant protein sequence database (NR (Wheeler *et al.*, 2007); 4.8 million sequences, downloaded in March 2007). *PSI-BLAST* is applied such that it iterates three times and includes amino

acid sequences in the profile if their E-value is smaller than 1×10^{-4}. Predicted coiled-coil segments and low complexity segments of the NR database proteins were masked by *pfilt* (Jones and Swindells, 2002) and *SEG* (Wootton and Federhen, 1993), respectively. Each PSSM was normalized by the standard logistic function to obtain the value range $[0, 1]$.

2.2.3 Implementation details of *MOLEBRNN*

This section will present the most important details concerning the implementation of *MOLEBRNN*. The first part will introduce the used variables, the second part will describe the forward pass, the third part will describe the backward pass and finally the fourth part declares the weight updates. The forward pass is carried out during prediction, the network input is propagated through all layers to produce a network output. The backward pass is conducted during training, the network error is propagated backward through all layers and all nodes determine their contribution to the network error. During the weight update process the connection weights between the nodes are changed in response to the associated errors. The network error is minimized during training. The processing of an bidirectional Elman-type recurrent neural network with multiple output layers is outlined. As training algorithm BackPropagation Through Time (BPTT) (Williams and Zipser, 1995) was implemented and weight updates are performed utilizing the RProp protocol (Riedmiller and Braun, 1993) adjusted for recurrent networks. Weight update is performed after a full epoch is processed.

Notions

It is important to realize that a recurrent neural network is a sequence learner. As recurrent neural networks emerged from time series prediction the elements of a sequence S are identified by timepoints t. All timepoints of the training data consist of tuples $((s_e(t), s_o(t)))$, encoding for network input[1] e and outputs o. The input $s_e(t)$ itself is encoding a one-dimensional array that represent the network input features.

[1]The symbol 'e' is used to abbreviate the network 'entry'.

Variable	Description
S	a training or test sequence
t	a timepoint
$(s_e(t), s_o(t))$	all train sequences constitute tuples, the entry element $s_e(t)$ represents the input variables, while the output element $s_o(t)$ constitutes observed class labels
$s_e(t)$	a vector encoding all input features of a single instance in a sequence S at timepoint t
$s_e(t, i)$	a scalar value holding the value of feature i from an instance at timepoint t
$s_o(t)$	a vector encoding the target network outputs
$s_o(t, c, i)$	a scalar value holding the class value i of prediction target c at timepoint t
L	the set of layers in the network
$\|L\|$	the number of layers
$l(i)$	layer at index position i
$l(0)$	the output layer
$l(\|L\| - 1)$	the input layer
$c(l)$	returns all chains in the layer l
k	the number of prediction targets
$x(l, c, i, t)$	input of network unit i in chain c of layer l at timepoint t
$y(l, c, i, t)$	output of network unit i in chain c of layer l at timepoint t
$u(l, c)$	this function returns all network units in layer l and chain c
$u(l, c, i)$	this function returns a specific unit i in chain c of layer l
E	the total network error
$e_{c,t}$	the error of a single output chain c at timepoint t
W	the set of connection weights
$w[i, j]$	the weight for a connection *from* unit j *to* unit i
$\sigma(x) = (1 + e^{-x})^{-1}$	the sigmoidal activation function
$\sigma'(x) = \sigma(x)(1 - \sigma(x))$	the derivative of the sigmoidal activation function

Table 2.2: Summary of the notions used to describe the network algorithms.

The output at a specific timepoint $s_o(t)$ is multi-dimensional. For *MOLEBRNN* the various prediction targets are encoded in the first dimension and the value found for a target, in the second dimension. A nominal class target is modeled by as many network units as classes exist. The unit representing an observed class is trained to take the value 1.0 while all other units aim for 0.0. A real valued regression target is modeled by a single output unit which is trained to take the target value. An example in the context of protein one-dimensional structure prediction would yield a sequence S that represents a protein with the residue PSSM features constituting the input elements $s_e(t)$ and $s_o(t)$ some prediction targets like secondary structure or real value solvent accessibility rSA. The secondary structure target can take the three values (H, E, C) which are modeled in *MOLEBRNN* by three different network output units. If a specific instance of the training set is in helical state (H) then the unit representing this state should output 1.0 after training, while the other units

responsible for the values (C, E) should yield 0.0. The rSA represents a real value regression target which is modeled by a single network output unit.

The training involves the presentation of training sequences to the network. An epochwise operation (Williams and Zipser, 1995) is implemented: One full presentation of all training sequences to the network is called epoch. During the epoch a global network error is recorded that summarizes the individual errors of all single sequences. When all training sequences are presented to the network the error is used to deduce the weight updates.

The network implementation allows for multiple individual layers. A layer l is defined by the unit connections ('node' is a synonym for 'unit') of the network. Recurrent connections are only allowed within a layer and not between the layers. The layers are connected by feed-forward connections. The network output layer is denoted as $l(0)$. The network input layer is denoted by $l(|L|-1)$, $|L|$ denotes the number of layers. The hidden layers are indexed by values > 0 and $< |L|-1$. To allow the bidirectional setup, the hidden layers contain *two* chains (see section 2.2.2) called *downstream* and *upstream* chains. To allow k output targets, the output layer $l(0)$ contains k chains. The input layer $l(|L|-1)$ contains a single chain that is connected to both, *downstream* and *upstream* hidden layer nodes. The connection weights w are denoted $w[i, j]$ which encodes the connection from unit j to unit i.

A network unit has multiple inputs and one output. The inputs of an *hidden* layer unit come from the outputs of all units in the previous layer that are in the current chain (these are the feedforward connections, not time depending) and from the outputs of all units in the current chain of the current layer (these are the recurrent connections that introduce time dependency). There are no feedforward or recurrent connections between the chains of the hidden layers!

The input and output for unit i in chain c and layer l at timepoint t is denoted $x(l, c, i, t)$ and $y(l, c, i, t)$. The unit output y is derived from the input when applying the sigmoidal activation function $\sigma(x) = (1 + e^{-x})^{-1}$.

A summary of all variables introduced is found in table 2.2.

Forward pass

The propagation of the network input to the output layer is shown in Listing 2.1.

For timepoints t where the unit output $y(l, c, i, t)$ is not available (when $t = -1$ or $t = |S|$) $y(l, c, i, -1) = y(l, c, i, |S|) = 0.0$. There is only a single chain in the first layer. All requests on specific chains from nodes in the second layer are ignored.

Backward pass

Neural networks in general are trained by minimizing the total network error E. The total error E is the sum of all errors $e_{c,t}$ for a single output chain c and a single instance (at timepoint t) available from the training dataset. This error $e_{c,t}$ when using the softmax activation function is given by the cross entropy (table 2.2 gives the overview on the variables used):

$$e_{c,t} = \sum_{j \in u(0,c)} s_o(t, c, j) ln\left(y(0, c, j, t)\right)$$

where $u(0, c)$ are the units in output chain c, $s_o(t, c, j)$ holds the observed target class value j at timepoint t for chain c and $y(0, c, j, t)$ holds the output of unit j in the output layer 0 of chain c at timepoint t.

Essentially the total error E of a neural network is a function of the connection weights W: $E(W)$. During training, E is minimized with respect to W. The alteration of W requires to estimate the partial derivative $\frac{\partial E}{\partial W}$ that answers: How does a change of W influence the error E? This derivate makes a statement on the global network. When breaking it down to the individual units, it can be estimated how much the output of each unit contributes to the total error ($\frac{\partial E}{\partial Y}$), it can be estimated how much the input of each unit contributes to the output ($\frac{\partial Y}{\partial X}$) and how much each of the input weights contribute on all unit inputs ($\frac{\partial X}{\partial W}$). Essentially for all nodes the partial derivative is solved

$$\frac{\partial E(t)}{\partial W} = \frac{\partial E(t)}{\partial Y(t)} \frac{\partial Y(t)}{\partial X(t)} \frac{\partial X(t)}{\partial W}$$

This derivative is time depending in recurrent units, the non time depending gradient

```
1  { Initial definitions: }
2  σ(x) = (1 + e^{-x})^{-1}
3  y(l, c, u, −1) = y(l, c, u, |S|) = 0
4  c(|L| − 1, i) = c(|L| − 1, 0)
5
6  Reset all unit outputs y to 0.
7  for t := 0 to |S| − 1
8  l := |L| − 1
9  begin
10   for all u in u(l, 0)
11   begin
12     y(l, 0, u, t) := s(t, u)
13   end
14 end
15 c := downstream
16 for t := 0 to |S| − 1
17 begin
18   for l := |L| − 2 downto 1
19   begin
20     for i in u(l, c)
21     begin
22       x(t, l, c, i) := Σ_{j∈u(l+1,c)} y(l + 1, c, j, t) × w[u(l, c, i), u((l + 1, c, j)]+
                          Σ_{j∈u(l,c)} y(l, c, j, t − 1) × w[u(l, c, i), u(l, c, j)]
23       y(t, l, c, i) := σ(x(t, l, c, i))
24     end
25   end
26 end
27 c := upstream
28 for t := |S| − 1 downto 0
29 begin
30   for l := |L| − 2 downto 1
31   begin
32     for i in u(l, c)
33     begin
34       x(t, l, c, i) := Σ_{j∈u(l+1,c)} y(l + 1, c, j, t) × w[u(l, c, i), u((l + 1, c, j)]+
                          Σ_{j∈u(l,c)} y(l, c, j, t + 1) × w[u(l, c, i), u(l, c, j)]
35       y(t, l, c, i) := σ(x(t, l, c, i))
36     end
37   end
38 end
39 for t := 0 to |S| − 1
40 begin
41   for c in c(0)
42   begin
43     for i in u(0, c)
44     begin
45       x(0, c, i, t) := Σ_{c_p∈c(1)} Σ_{j∈u(1,c_p)} y(1, c_p, j, t) × w[u(0, c, i), u(1, c_p, j)]
46       y(0, c, i, t) := σ(x(0, c, i, t))
47     end
48   end
49 end
```

Listing 2.1: *MOLEBRNN* forward pass

- In line 3 the sequence borders are initialized with 0.
- In line 4 the rules to access the chains in the first layer $l(|L|-1)$ are defined: the input layer $|L|-1$ contains just a single chain responsible for *downstream* and *upstream* response.
- Line 8 defines to process the first layer in the following.
- Iterations following lines 15 and 27 is responsible for the downstream and upstream chains.
- First term in line 22 is responsible for the previous layer at current timepoint, while the second term for the output within the current layer at previous timepoint. When incorporating the previous layer, only nodes from the same chain c are considered. If the previous layer is the input layer $l(|L|-1)$ that contains just a single chain, the nodes in this single chain are incorporated no matter what the chain in the current layer is.
- c_p in line 45 endodes a chain in a previous layer. This can be nothing but *downstream* and *upstream*, only the output layer allows other chains.

in a *downstream* chain is derived from

$$\frac{\partial E}{\partial w_{i,j}} = \sum_{t=1}^{|S|-1} \delta_i(t) x_j(t-1)$$

that is in turn used to update all weights.

The backpropagation through time is realized in Listing 2.2.

Weight update

Having computed the non time dependent version of the error gradient $\frac{\partial E}{\partial W}$ allows weight update:

$$\Delta W = -\eta \frac{\partial E}{\partial W}$$

where η is the learning rate.

The weight update process is depicted in Listing 2.3.

2.2.4 Ensembles of neural networks

A two layer neural network ensemble is used to recognize β-turns in protein sequences. In the first layer (sequence-to-structure) multiply aligned amino acid sequences encoded in form of a position specific scoring matrix (PSSM) are mapped into structural states of individual residue positions. The output of this first layer is then fed into a second, structure-to-structure layer. Such two-stage approach was pioneered in bioinformatics by Qian and Sejnowski (1988) and has been frequently used for secondary structure prediction (Rost and Sander, 1993; Jones, 1999; Adamczak *et al.*, 2005). It is particularly efficient in detecting correlations between neighboring residues and smoothing predicted secondary structure segments (Rost and Sander, 1993). In the implementation of the predictor, the first layer consists of five independently trained neural networks with different numbers of hidden nodes. Each network performs sequence to structure mapping. The second layer, consisting of just one network, averages the outcomes of the first layer and accounts for residue correlations.

To compare *MOLEBRNN* neural network design with *EBRNN* in terms of

```
1  Reset all partial derivatives δ to 0
2  for t := 0 to |S| − 1
3  begin
4    for c in c(0)
5    begin
6      for i in u(0, c)
7      begin
8        δ(0, c, i, t) = y(0, c, i, t) − s_o(t, c, i)
9      end
10   end
11 end
12 c = downstream
13 for t := |S| − 1 downto 0
14 begin
15   for l := 1 to |L| − 2
16   begin
17     for i := u(l, c)
18     begin
           δ(l, c, i, t)  =  σ′(x(l, c, i, t))+
                             (
19                           Σ_{j∈u(l−1,c)} δ(l − 1, c, j, t) × w[u(l − 1, c, j), u(l, c, i)]+
                             Σ_{j∈u(l,c)} δ(l, c, j, t + 1) × w[u(l, c, j), u(l, c, i)]
                             )
20     end
21   end
22 end
23 c = upstream
24 for t := 0 to |S| − 1
25 begin
26   for l := 1 to |L| − 2
27   begin
28     for i := u(l, c)
29     begin
           δ(l, c, i, t)  =  σ′(x(l, c, i, t))+
                             (
30                           Σ_{j∈u(l−1,c)} δ(l − 1, c, j, t) × w[u(l − 1, c, j), u(l, c, i)]+
                             Σ_{j∈u(l,c)} δ(l, c, j, t − 1) × w[u(l, c, j), u(l, c, i)]
                             )
31     end
32   end
33 end
```

Listing 2.2: *MOLEBRNN* backward pass

- First term in line 19 corresponds to the derivative of the activation function $\sigma'(x)$. Second term is responsible for incorporating the unit error contribution from following layer $l-1$ at current timepoint t and the third term is responsible for the error contribution on units in the current layer from timepoint $t+1$. Note: this is always available as time during BPTT is running backwards (see line 13).

```
1  for c in c(0)
2  begin
3    for i in u(0,c)
4    begin
5      for c_p in c(1)
6      begin
7        for j in u(1,c_p)
8        begin
9          ∂E/∂w[u(0,c,i),u(1,c_p,j)] := Σ_{t=0}^{|S|-1} δ(0,c,i,t)y(1,c_p,j,t)
10         w[u(0,c,i),u(1,c_p,j)] := w[u(0,c,i),u(1,c_p,j)] + η ∂E/∂w[u(0,c,i),u(1,c_p,j)]
11       end
12     end
13   end
14 end
15 c = downstream
16 for l := 1 to |L| - 2
17 begin
18   for i in u(l,c)
19   begin
20     for j in u(l+1,c)
21     begin
22       ∂E/∂w[u(l,c,i),u(l+1,c,j)] := Σ_{t=0}^{|S|-1} δ(l,c,i,t)y(l+1,c,j,t)
23       w[u(l,c,i),u(l+1,c,j)] := w[u(l,c,i),u(l+1,c,j)] + η ∂E/∂w[u(l,c,i),u(l+1,c,j)]
24     end
25     for j in u(l,c)
26     begin
27       ∂E/∂w[u(l,c,i),u(l,c,j)] := Σ_{t=1}^{|S|-1} δ(l,c,i,t)y(l,c,j,t-1)
28       w[u(l,c,i),u(l,c,j)] := w[u(l,c,i),u(l,c,j)] + η ∂E/∂w[u(l,c,i),u(l,c,j)]
29     end
30   end
31 end
32 c = upstream(*@@*)
33 for l := 1 to |L| - 2
34 begin
35   for i in u(l,c)
36   begin
37     for j in u(l+1,c_p)
38     begin
39       ∂E/∂w[u(l,c,i),u(l+1,c,j)] = Σ_{t=0}^{|S|-1} δ(l,c,i,t)y(l+1,c,j,t)
40       w[u(l,c,i),u(l+1,c,j)] := w[u(l,c,i),u(l+1,c,j)] + η ∂E/∂w[u(l,c,i),u(l+1,c,j)]
41     end
42     for j in u(l,c)
43     begin
44       ∂E/∂w[u(l,c,i),u(l,c,j)] = Σ_{t=|S|-2}^{0} δ(l,c,i,t)y(l,c,j,t+1)
45       w[u(l,c,i),u(l,c,j)] := w[u(l,c,i),u(l,c,j)] + η ∂E/∂w[u(l,c,i),u(l,c,j)]
46     end
47   end
48 end
```

Listing 2.3: *MOLEBRNN* weight update process

- c_p in line 5 holds the chains in the previous layers which are exclusively *downstream* and *upstream*.
- In lines 15 and 32 the procession on the *downstream* and *upstream* hidden layer chains start.
- lines 22 and 23 describe the weight update for feedforward connections to the previous layer $l+1$ while lines 27 and 28 describe the weight update for recurrent connections within the current layer l.

Setup name	Description	Prediction targets
Ensemble EBRNN	Five EBRNNs in the first layer and one *EBRNN* in the second layer are used.	Separate instances for secondary structure, β-turn and β-turn types I, II, $VIII$ and IV.
Ensemble EBRNN SSP	The single second layer *EBRNN* gets input from the five first-layer *EBRNN*s and, additionally, from a secondary structure predictor *PSIPRED* (Jones, 1999).	Separate instances for secondary structure, turn and turn types I, II, $VIII$ and IV.
Ensemble MOLEBRNN	Five *MOLEBRNN*s in the first layer and one *MOLEBRNN* in the second layer are used.	One instance to predict secondary structure, turn and turn types I, I', II, II', $VIII$ and IV simultaneously.

Table 2.3: Network setups used in this study.

Layer	# Input Nodes	# Hidden Nodes	Output Layer Definition
Sequence-to-Structure	22	20	In all networks, 3-state secondary structure, the generic β-turn and the individual β-turn types serve as prediction targets resulting in 7 output layers with 2 nodes (turns/non-turns) and one output layer with 3 nodes (3-state secondary structure)
Sequence-to-Structure	22	30	
Sequence-to-Structure	22	40	
Sequence-to-Structure	22	50	
Sequence-to-Structure	22	60	
Structure-to-Structure	85	5	
Structure-to-Structure	85	10	
Structure-to-Structure	85	20	

Table 2.4: The structures of the used multi output layer Elman-type bidirectional recurrent neural networks.

β-turn and β-turn type prediction accuracy three different setups are employed: *Ensemble EBRNN*, *Ensemble EBRNN SSP*, and *Ensemble MOLEBRNN*. The networks are additionally outlined in table 2.3. Both *EBRNN*s and *MOLEBRNN*s employed in this work are precisely defined in tables 2.4 and 2.5 where the numbers of nodes in all network layers are listed.

2.2.5 Performance measures

The accuracy of all predictions reported here was determined by full sevenfold cross-validation. For β-turn prediction the following four parameters are used to measure the performance of the classifiers and to compare the *MOLEBRNN* results to reported values: i) Q_{total}, the general prediction accuracy for two classes, ii) Q_{pred},

Layer	**# Input Nodes**	**# Hidden Nodes**	**# Output Nodes**	**Targets**
Sequence-to-structure	22	20	2	β-turn, β-turn types I, II, $VIII$ and IV
Sequence-to-structure	22	20	3	Secondary structure
Sequence-to-structure	22	30	2	β-turn, β-turn types I, II, $VIII$ and IV
Sequence-to-structure	22	30	3	Secondary structure
Sequence-to-structure	22	40	2	β-turn, β-turn types I, II, $VIII$ and IV
Sequence-to-structure	22	40	3	Secondary structure
Sequence-to-structure	22	50	2	β-turn, β-turn types I, II, $VIII$ and IV
Sequence-to-structure	22	50	3	Secondary structure
Sequence-to-structure	22	60	2	β-turn, β-turn types I, II, $VIII$ and IV
Sequence-to-structure	22	60	3	Secondary structure
Ensemble EBRNN	10	10	2	β-turn, β-turn types I, II, $VIII$ and IV
Ensemble EBRNN SSP	13	10	2	β-turn, β-turn types I, II, $VIII$ and IV
Ensemble EBRNN	15	10	3	Secondary structure

Table 2.5: The structures of the used Elman-type bidirectional recurrent neural networks.

the percentage of correct elements in the positively predicted set (also called positive predictive value or precision), iii) Q_{obs}, the percentage of observed positive elements predicted (also called recall or sensitivity), and iv) MCC, the Matthews correlation coefficient with the value range of $[-1, 1]$ where the values 0, +1, and -1 indicate a random prediction, the best possible prediction, and a reverse prediction, respectively. The values of these four performance measures will depend on the threshold τ chosen. In 2.3 Results and discussion typically the measures Q_{total}, Q_{pred}, Q_{obs} and MCC are presented with τ adjusted such that the maximal MCC (labeled by $maxMCC$) is obtained. In some cases predictive performance is discussed for a different τ such that there is a trade-off between high MCC and high Q_{total} (labeled by $tradeoff$). In both cases the name of the network architecture is labeled by *'maxMCC'* or *'tradeoff'* to indicate the choice of (e.g., *Ensemble MOLEBRNN maxMCC*). Further, threshold curves are presented where the performance measures are plotted versus τ.

The *ROC* graph is another popular performance indicator. It visualizes the relative trade-off between benefits (true positives) and costs (false positives) of each prediction. It is a two-dimensional graph where true positive rate is plotted versus the false positive rate when varying the threshold τ (Fawcett, 2004). To facilitate

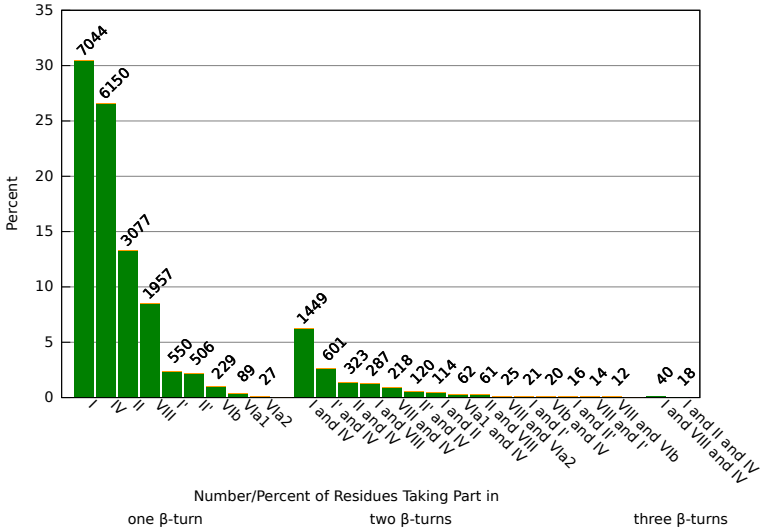

Figure 2.2: Composition of β-turns as found in the dataset.
Shown are the percentages and the counts of individual β-turn types and their combinations in β-turns.

comparison between different classifiers considered in this work, the information in the two-dimensional *ROC* graph is compressed to a single scalar value, the area under the ROC curve (*AUC*). *AUC* is usually interpreted as the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2004).

2.3 Results and discussion

2.3.1 Analysis of prediction target data

This study started by analyzing the interdependencies between the localization of β-turns, different types of β-turns, and other secondary structure elements. Each turn is of length four, and its type is defined based on the dihedral angles of the two central amino acid residues (recall Table 2.1). Thus, by definition, each residue can be part of at most three different β-turns (see Figure 2.2). In the dataset of 95844 residues, described in Section 2.2.1, there are 23134 residues, or 24%, in any of the individual β-turn types, making the β-turn training dataset significantly unbalanced. As described in Material and methods, the general target encompasses all individual

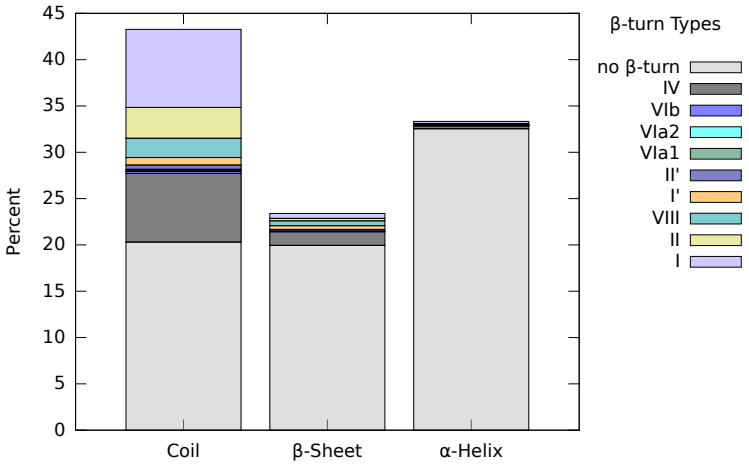

Figure 2.3: Occurrence of β-turns within secondary structure elements.

β-turn types but from Figure 2.2 follows that nearly 60% of β-turns are of the type I and IV.

The occurrence of β-turns in different secondary structure elements is shown in figure 2.3. α-helices are nearly β-turn free due to the definition of β-turns that does not allow the two central residues to be part of an α-helix. The β-turn content of strands is also very low, represented for the most part by the turn type IV involving untypical (irregular) dihedral angles (see table 2.1). As expected, most β-turns are found in coils where they account for more than a half of all residues.

There are thus clear interdependencies between the secondary structure elements considered here. Turns generally avoid α-helices and, except for the turn type IV, β-strands, while coil has a strong statistical preference to the β-turns of type I and IV. This situation motivates the development of an integrated method that, when predicting individual structural targets, would not only learn the presence or absence of a single structural element, but would also gain additional information from other structural targets being predicted for a given residue. The interdependence between multiple secondary structure elements have already been used by other groups. For example, Kaur and Raghava (2003b) and Fuchs and Alix (2005) show that consideration of predicted secondary structure improves β-turn prediction. *Vise versa*, knowledge about β-turns helps to improve secondary structure prediction (Frishman

Network type	Q_{total}	Q_{pred}	Q_{obs}	MCC	AUC
Single EBRNN	75.0%	49.1%	64.4%	0.394	0.793
Ensemble EBRNN	76.4%	51.3%	66.0%	0.424	0.814
Ensemble EBRNN SSP	76.5%	51.4%	66.4%	0.427	0.822
Single MOLEBRNN	75.3%	49.7%	70.5%	0.428	0.819
Ensemble MOLEBRNN	77.9%	53.9%	66.0%	0.448	0.832

Table 2.6: Performance measures of β-turn predictors.

and Argos, 1997).

In contrast to these previous attempts, to the knowledge of the author, this is the first approach to model such interdependencies via a single network. A *BRNN* sequence learner is embodied instead of a feed-forward neural network. *BRNN*s have the advantage of being able to tackle sequence prediction problems globally while feed-forward networks require short sequence windows and can thus only consider local structural context.

2.3.2 Predictive performance of *EBRNN* and *MOLEBRNN*

Prediction of β-turns

In Table 2.6 the performances are presented, derived from all network types developed for predicting β-turns from protein sequences. Layered architectures – *Ensemble EBRNN* and *Ensemble MOLEBRNN* – display significantly better performance compared to single networks in terms of Q_{total} (overall prediction accuracy) as well as MCC (Matthews correlation coefficient) and AUC (area under the ROC curve). Incorporation of the secondary structure predicted by *PSIPRED* only has a marginal effect on the performance of the *Ensemble EBRNN*. The latter already reaches the MCC value of 0.424 even without secondary structure, while a previously reported method (Fuchs and Alix, 2005) achieved $MCC = 0.410$ when *PSIPRED* predictions are taken into account. The performance of the *Ensemble MOLEBRNN* – $Q_{total} = 77.9\%$ and $MCC = 0.448$ – is the highest reported in literature so far (table 2.7). In particular, *Ensemble MOLEBRNN* outperforms the BETATURN technique (Zhang *et al.*, 2005) which is based on support vector machines.

Method Name	Reference	Method Type	Q_{total}	Q_{pred}	Q_{obs}	MCC
Ensemble MOLEBRNN	This work	*MOLEBRNN*	77.9%	53.9%	66.0%	0.45
Ensemble EBRNN SSP	This work	BRNN	76.5%	51.4%	66.4%	0.43
Ensemble EBRNN	This work	BRNN	76.4%	51.3%	66.0%	0.42
BETATURN	Zhang *et al.* (2005)	SVM	77.3%	53.1%	67.0%	0.45
BETATPRED2	Kaur and Raghava (2003a)	MLP	75.5%	49.8%	72.3%	0.43
COUDES	Fuchs and Alix (2005)	Propensities	74.8%	48.8%	69.9%	0.42
KNN	Kim (2004)	KNN	75.0%	46.5%	66.7%	0.40
BTPRED	Shepherd *et al.* (1999)	MLP	76.0%	50.9%	63.0%	0.40
Chou-Fasman	Chou and Fasman (1979)	Propensities	74.3%	47.7%	54.3%	0.34

Table 2.7: Comparison of β-turn prediction methods.

The performance of all methods is reported for the same test dataset to allow direct comparison. Values for *BTRED* and *Chou-Fasman* are taken from Kaur and Raghava (2002). Abbreviations: BRNN –bidirectional recurrent neural network, SVM – support vector machines, MLP – Multi Layer Perceptron, KNN – k-nearest neighbors.

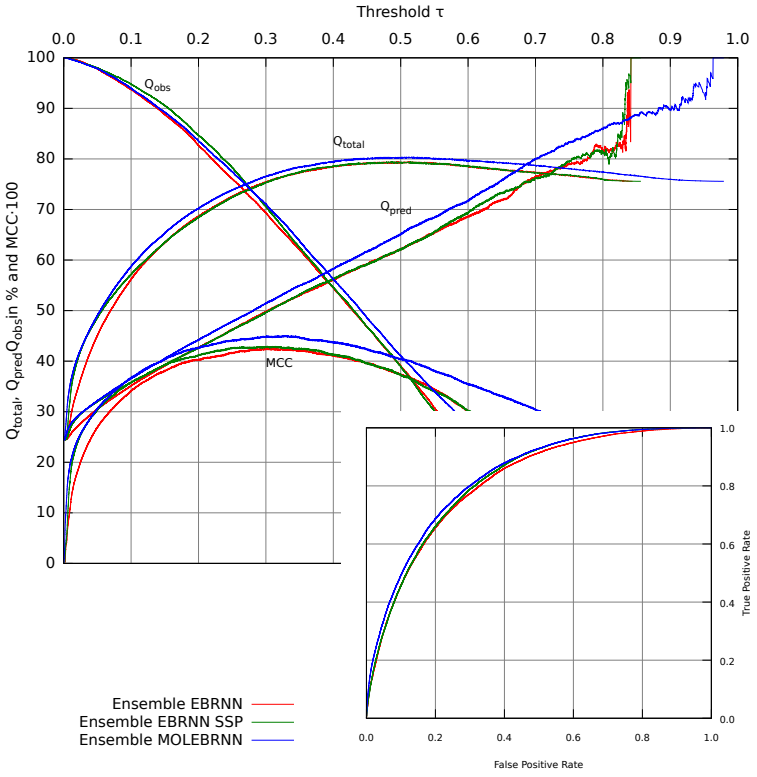

Figure 2.4: β-turn prediction performance of *Ensemble EBRNN*, *Ensemble EBRNN SSP*, and *Ensemble MOLEBRNN*.

A more detailed comparison of the *MOLEBRNN* with the ordinary *EBRNN* with and without secondary structure considered is shown in figure 2.4. The threshold curves together with the ROC graph and the high *AUC* (Table 2.6) clearly demon-

Turn Type	Method	τ	Q_{total}	Q_{pred}	Q_{obs}	MCC
I	*Ensemble MOLEBRNN maxMCC*	0.168	82.5%	28.9%	56.9%	0.317
I	*Ensemble MOLEBRNN tradeoff*	0.200	85.4%	31.7%	48.7%	0.314
I	*COUDES*	-	84.5%	30.8%	50.0%	0.309
I	*BETATURNS*	-	74.5%	22.1%	74.1%	0.29
II	*Ensemble MOLEBRNN maxMCC/tradeoff*	0.293	96.2%	50.2%	25.2%	0.339
II	*COUDES*	-	91.0%	22.2%	52.8%	0.302
II	*BETATURNS*	-	93.5%	25.5%	52.8%	0.29
VIII	*Ensemble MOLEBRNN maxMCC*	0.024	53.4%	4.7%	80.5%	0.109
VIII	*Ensemble MOLEBRNN tradeoff*	0.072	93.0%	8.0%	19.0%	0.076
VIII	*COUDES*	-	90.7%	6.9%	18.7%	0.071
VIII	*BETATURNS*	-	96.5%	7.2%	2.8%	0.02
IV	*Ensemble MOLEBRNN maxMCC*	0.130	72.3%	20.1%	63.8%	0.236
IV	*Ensemble MOLEBRNN tradeoff*	0.204	85.2%	26.0%	29.3%	0.194
IV	*COUDES*	-	84.9%	20.7%	17.7%	0.109
IV	*BETATURNS*	-	67.9%	18.6%	72.0%	0.23
I'	*Ensemble MOLEBRNN maxMCC/tradeoff*	0.382	98.8%	59.3%	21.9%	0.356
I'	*COUDES*	-	94.4%	11.6%	51.8%	0.226
II'	*Ensemble MOLEBRNN maxMCC/tradeoff*	0.079	98.6%	12.7%	16.3%	0.137
II'	*COUDES*	-	94.6%	4.6%	32.8%	0.106

Table 2.8: Comparison of β-turn prediction methods.

strate the advantage of *Ensemble MOLEBRNN* over both *Ensemble EBRNN* and *Ensemble EBRNN SSP*. As expected, precision continuously improves with higher τ values. However, for very high τ values the Q_{pred} curves become unstable due to low number of instances available at such high thresholds τ. For all three curves Q_{total} reaches its maximum at τ values around 0.5. MCC is maximal at values around $\tau = 0.3$, close to the general occurrence of residues in the β-turn state which is around 24%.

Prediction of β-turn types

Ensemble MOLEBRNN also clearly outperforms *EBRNN* and the state-of-the-art methods developed by others in predicting individual β-turn types. In table 2.8 the performance of *Ensemble MOLEBRNN* is shown both for the threshold values optimized to achieve the best possible MCC (labeled *maxMCC*) as well as for those

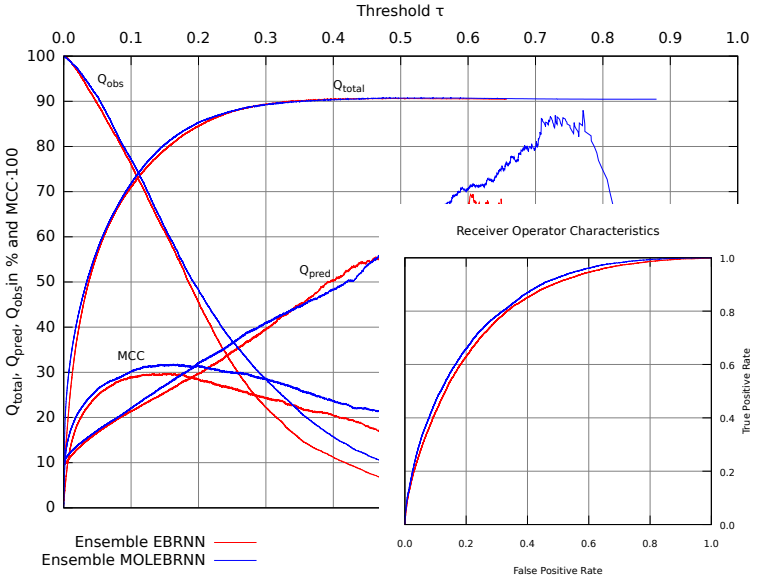

Figure 2.5: Threshold and *ROC* graph for the β-turn type I prediction.

that provide a reasonable tradeoff between Q_{total} and MCC based on the threshold curves (labeled *tradeoff*, shown in figure 2.5 for the β-turn type I). In some cases the *maxMCC* and *tradeoff* values correspond to the same τ value. As seen in table 2.8, for β-turn types I and II *Ensemble MOLEBRNN tradeoff* performs better than *COUDES* (Fuchs and Alix, 2005) and *BETATURNS* (Kaur and Raghava, 2004) both in terms of Q_{total} and MCC. For the β-turn type I' and II' it also outperforms *COUDES* while a comparison with BETATURN is not possible as the latter method does not predict these β-turn types. For the β-turn types $VIII$ and IV *Ensemble MOLEBRNN tradeoff* is better than *COUDES*, but compared to *BETATURNS* it achieves worse Q_{total} on β-turn $VIII$ and worse MCC on β-turn IV. At the same time, for these two turn types *Ensemble MOLEBRNN maxMCC* achieves worse Q_{total} than both *COUDES* and *BETATURNS* (type $VIII$ only), but its MCC is significantly better. The final version of the *MOLEBRNN* software allows the user to select one of the two modes – *maxMCC* or *tradeoff* – dependent on whether better precision or better recall is desired. In all cases *Ensemble MOLEBRNN* is characterized by better AUC values than other methods. To further illustrate the advantages of *Ensemble MOLEBRNN* over *Ensemble EBRNN*, in figure 2.5 one set

of threshold curves is exemplified together with the ROC graph for the β-turn type I. For all measures used in this work the *Ensemble MOLEBRNN* is clearly better than the *EBRNN* setup. Similar tendencies are observed for the β-turn types II, $VIII$ and IV (data not shown). The AUCs for β-turn type I prediction are 0.804 for *Ensemble EBRNN* and 0.821 for *Ensemble MOLEBRNN*. Kaur and Raghava (2004) obtain for β-turn type I an AUC of 0.746.

2.3.3 Performance for different combinations of prediction targets

Experiments were conducted to find out how interdependencies between different combinations of prediction targets influence the performance of the multiple output architecture. To allow a direct comparison all performance measures were derived from a sequence-to-structure model with 30 hidden nodes trained on the same data. All experiments were carried out five times using random initializations of connection weights in the neural networks, and the average outcomes were recorded. As seen in figure 2.6, *EBRNN* (i.e. *MOLEBRNN* with a single output) performs worst in predicting β-turns. The addition of output layers for the six β-turn subtypes (I, II, I', II', IV and $VIII$) results in increased performance. If secondary structure is used as an output layer together with the β-turn target the performance is further increased. The combined use of β-turns, β-turn types, and secondary structure targets in eight output layers results in the best performance. This finding illustrates the power of *MOLEBRNN*. Although the number of hidden nodes and the input data stays constant the network manages to incorporate further knowledge during training through additional output layers. Beyond each individual target function, *MOLEBRNN* takes into account related targets to learn the problem of recognizing residue secondary structure states globally. β-turns are a cumulative representation of all β-turn types, and secondary structure indirectly correlates with β-turns. There exists an hierarchy of the secondary structure elements, with β-turn types at the most detailed level and secondary structure at the most general level. Curves in figure 2.6 imply that global features have a higher impact on prediction accuracy that local

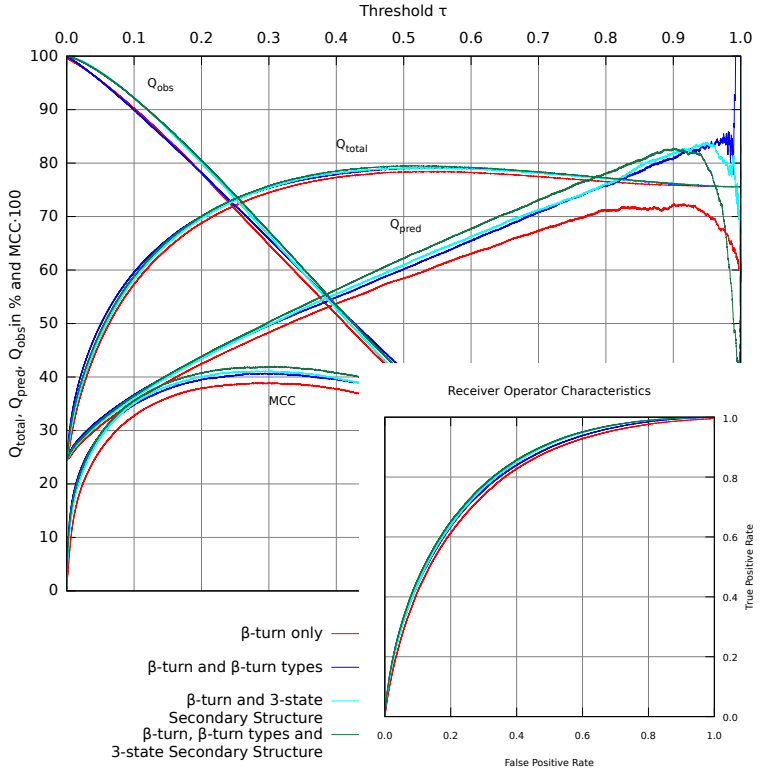

Figure 2.6: Performance of *MOLEBRNN* on different combinations of target classes.

ones. Thus, the secondary structure output layer influences the prediction accuracy stronger than β-turn types. One reason for the weaker influence of β-turn types could be the seldom occurrence of many turn types in the dataset. The knowledge about the generic β-turn target is mostly obtained from the types I and IV while the four rare types II, I', II' and $VIII$ do not contribute much. This is not the case for secondary structure. All three secondary structure elements occur with nearly equal frequency and a strong connection between coil and β-turns is found.

2.4 Conclusions

The data presented here demonstrate a better performance of recurrent neural networks compared to feed-forward neural networks when applied to β-turn prediction. The *MOLEBRNN* architecture is able to learn multiple aspects of protein structure and achieves the best performance reported so far both in distinguishing β-turns

from non-β-turns and in predicting specific β-turn types. It is interesting to note that while the SVM-based methods have recently surpassed neural networks in terms of prediction accuracy, the success of the *MOLEBRNN* technique demonstrates that the potential of neural networks has not yet been exhausted. The online version of *MOLEBRNN* at http://webclu.bio.wzw.tum.de/predator-web/ can be applied to predict β-turns in proteins.

Chapter 3

Prediction in 1D :: Prediction of multiple structural motifs in soluble proteins

3.1 Introduction

Prediction of one-dimensional structural motifs from amino sequences constitutes one of the most traditional and established areas of bioinformatics. The accuracy of prediction tools has been steadily increasing over the past three decades, primarily due to two factors: availability of ever increasing training sets of known structures and the growing sophistication of algorithms applied to extract knowledge from sequences. For example, in the early work of Chou and Fasman (1974) simple statistical propensities of amino acid occurrence in helices and strands extracted from just 19 structures known at that time were used to predict protein secondary structure with 50-60% accuracy (Mount, 2004), while the accuracy of today's state-of-the art tools based on sophisticated machine learning techniques trained on thousands of diverse structures is approaching 80% (Pollastri and McLysaght, 2005; Adamczak *et al.*, 2005; Jones, 1999). Likewise, prediction of solvent accessibility improved from 58% (Rost and Sander, 1994a) to 78-79% (Pollastri *et al.*, 2007; Nguyen and Rajapakse, 2005). Other structural features of interest that can be predicted from

sequence alone include β-turns, structural class (α, β, or mixed), trans-membrane helices, chain flexibility, and disordered regions, to name just a few.

A somewhat under-appreciated aspect of one-dimensional structure prediction is that many protein features are in fact strongly inter-correlated. For example, the location of β-turns where protein backbone abruptly changes its direction when reaching the protein surface is obviously correlated with solvent accessibility, while the very presence of isolated β-turns at a particular sequence site automatically excludes α-helical or β-strand conformation. Knowledge of one structural motif should thus be helpful for predicting other motifs. Indeed, it has been shown that predicted secondary structure is an informative feature for the prediction of solvent accessibility (Garg *et al.*, 2005) and *vice versa* (Adamczak *et al.*, 2005). Consideration of predicted β-turns leads to increased secondary structure prediction accuracy (Frishman and Argos, 1995) and again, an improvement for the reverse application was also reported (Kaur and Raghava, 2003b). Another example where such trivial dependencies have been successfully applied involves the prediction of β-sheet types (parallel or anti-parallel) given predicted secondary structure (Zimmermann *et al.*, 2007). Whenever various structural aspects are interdependent, the utilization of one predicted feature to improve the prediction of another feature becomes possible.

In this chapter the question is investigated whether a recurrent neural network (RNN) able to predict various structural features simultaneously can outperform one-dimensional structure prediction methods trained for a single prediction target. It is hypothesized that a neural network designed to predict multiple interdependent aspects and optimized during training for all these aspects simultaneously captures important interdependencies intrinsic to protein structures and may show superior performance when compared to single-target neural network methods. Specifically, the aim in this work is to develop a method to predict solvent accessibility, secondary structure and β-turns that exploits inter-correlations between these three features. In the previous Chapter 2 an improvement in the prediction of β-turns and β-turn types was shown by including a variety of related prediction targets in the RNN output layer.

Currently web services exist that accomplish the task of predicting multiple

structural features by running a variety of independent prediction methods concurrently and reporting all prediction results combined on a single page. Such services are *Distill*, *PredictProtein*, *PSIPRED Server* or *Jpred 3* introduced from Baú *et al.* (2006), Rost *et al.* (2004), McGuffin *et al.* (2000) and Cole *et al.* (2008). These methods are retrained on a regular basis which is required to cope with the increasing amount of available protein structures. Opposite to this, the here described predictor requires retraining of just a single instance.

Similar to the β-turn prediction in Chapter 2 the structural features selected for this work have in common that they all are usually predicted from sequence profiles obtained from multiple sequence alignments. Prominent solvent accessibility predictors from Jones (1999); Kaur and Raghava (2003b); Adamczak *et al.* (2005) are all using the raw *PSI-BLAST* PSSM output and fed this, after some preprocessing, into machine learning algorithms, in the case of this three works, neural networks were used and these were organized in a topology that is used similar to that proposed by Qian and Sejnowski (1988): The output of an initial machine learning prediction is fed into a second machine learning layer that smooths prediction and correlates neighbored states. Here a method with similar architecture is described where all first layer outputs are also fed into the second layer.

In this chapter the stages are described that lead to the development of a method able to predict multiple one-dimensional features. The three structural motifs β-turns, solvent accessibility and three state secondary structure are predicted and the prediction performances are measured. In the beginning the performances on the individual structural motifs are analyzed when predicted simultaneously. Then it is analyzed whether there is a higher network requirement for multiple prediction targets and how performance increases when compared to single target prediction. In the last result section a strict comparison to available methods in terms of solvent accessibility and secondary structure is shown. It is concluded that that the here described *MOLEBRNN* based method outperforms to-date solvent accessibility predictions and keeps up with current secondary structure prediction methods. The final, download-able version is trained on a large nonredundant dataset comprising proteins of known structure and dubbed *SOPRANO*, the

solubility predictor applying neural networks with multiple outputs. SOPRANO is available at http://webclu.bio.wzw.tum.de/soprano.

3.2 Material and methods

3.2.1 Protein set

Dataset used for method development

The development of this method is based on two different datasets. Essentially the collection of nonredundant protein chains pdb_select (Hobohm *et al.*, 1992) is used to compile the data. For method optimization and obtainment of initial performance measures a dataset that was compiled from pdb_select of June 2000 (Hobohm *et al.*, 1992) was used. This dataset was also used by Pollastri *et al.* (2002a) and their protocol for preprocessing was transfered to this method: Chains were removed with backbone breaks, missing atom descriptions, or format errors. This dataset is called pdb_select-2000 in the following. The pdb_select-2000 dataset was also used for comparison to Pollastri *et al.* (2002a) and Adamczak *et al.* (2004). The final version of the predictor (*SOPRANO*) is trained on pdb_select of October 2007. Again the data is preprocessed in a similar way. In the following this dataset used to build the final version will be called pdb_select-2007.

Dataset based on Cuff and Barton (1999), used for comparison

For comparison of the here developed method to a variety of other predictors the protein chains originally compiled by Cuff and Barton (2000) where also deployed. This chains were selected very rigorously based on protein structure qualities and redundancy criteria. Ahmad *et al.* (2003a) suggested to remove those chains that were shorter than 30 residues and thus obtained a subset of the original Cuff and Barton (2000) set of 502 protein chains. As many solvent accessibility predictors are validated on the 502 protein chains, for validation and comparison purposes this set is also deployed here. It is referred to as cb502 in the following.

Dataset based on *SABLE*, used for comparison

The four control sets of Adamczak *et al.* (2004), S156, S135, S163 and S149 which consist of 603 proteins altogether were also used for comparison. The protein list was published in 2002 and non of which are homologue to the train set of Adamczak *et al.* (2004) compiled from a nonredundant snapshot of PDB in August 2001. Because it is newer, the train set covers a larger protein space than pdb_select-2000 dataset does and theoretically this 603 test proteins of Adamczak *et al.* (2004) should be unrelated to the pdb_select-2000 set. An analysis of the homology showed that 56 proteins in the test data of Adamczak *et al.* (2004) align at 35% sequence identity or better. These 56 proteins are removed and from that, four test sets similar to those from Adamczak *et al.* (2004) were obtained with the sets S156, S135, S163 and S149 now containing 138, 126, 147 and 136 proteins respectively.

Dataset based on Naderi-Manesh *et al.* (2001), used for comparison

Finally the Naderi-Manesh *et al.* (2001) dataset was utilized. It consists of 215 homology reduced protein chains which were also used by a variety of other methods. Especially Pollastri *et al.* (2007), to my knowledge the best method to-date, report performance measures on the Naderi-Manesh dataset, the reason why it is included here.

3.2.2 Definition of prediction targets

In order to exploit the power of *MOLEBRNN* the following variety of prominent prediction targets is defined. All of them are predicted simultaneously by *MOLEBRNN*.

Real value solvent accessibility

The number of water molecules that can adhere a residue in a three-dimensional protein structure defines the solvent accessibility. *DSSP* (Kabsch and Sander, 1983) can be used to calculate this number. Obviously residues of different sizes allow different numbers of water molecules attached such that a relative solvent accessibility

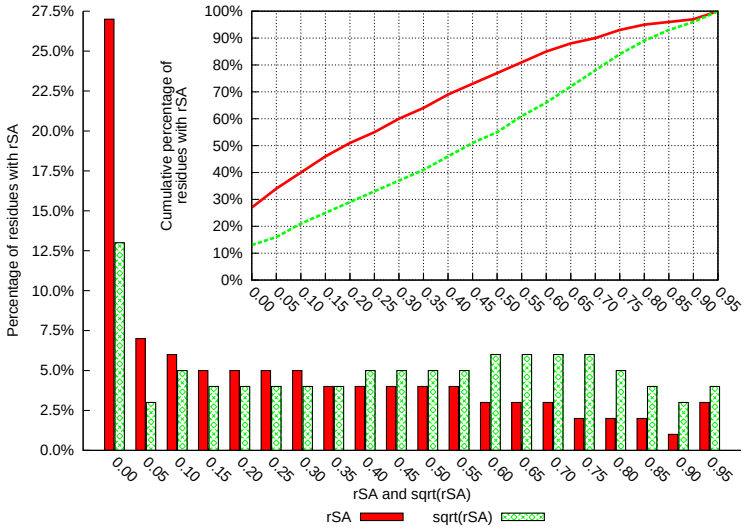

Figure 3.1: Distribution and cumulative distribution of solvent accessibility in pdb_select June 2000 dataset.

Shown are the raw rSA distribution and the distribution obtained from $sqrt(rSA)$. $\mu(rSA) = 0.310$, $\mu(sqrt(rSA)) = 0.468$, $\sigma^2(rSA) = 0.287$, $\sigma^2(sqrt(rSA)) = 0.303$, $\sigma(rSA) = 0.536$, $\sigma(sqrt(rSA)) = 0.550$.

$(rSA(i))$ has to be defined, that is independent of the residue type:

$$rSA(i) = \frac{aSA(i)}{max_j(aSA(j))}$$

where $aSA(i)$ is the absolute solvent accessibility of an amino acid residue i as obtained from *DSSP*, j is the type of the residue and $max_j(aSA(j))$ holds the maximal observable aSA for a given residue type. The $max_j(aSA(j))$ values where taken from Ahmad *et al.* (2003a). $rSA(i)$ takes values between 0.0 and 1.0 declaring the fraction of residue enclosure. Figure 3.1 shows the distribution of rSA as observed in pdb_select-2000 dataset. As can be seen in the figure, rSA values are not uniformly distributed: many residues are observed with low rSA while just very few residues are found with large rSA values. To smooth the distribution rSA is transformed by the square root to $sqrt(rSA)$. The obtained distribution is also contained in Figure 3.1. It can be seen that $sqrt(rSA)$ values occur much more uniformly distributed. This novel target was developed in order to bias the method for large rSA values. The $sqrt(rSA)$ predictions can be converted back to the original relative solvent accessibility by squaring them and obtain $sqrt(rSA)^2$ which equals rSA. rSA and

$sqrt(rSA)$ as they are defined here are prediction targets that are to be predicted by regression methods.

Discretized two-class solvent accessibility

As common in residue solvent accessibility prediction, not only real valued solvent accessibility is predicted by regression but additionally rSA is discretized to obtain two-class problems. Here the two classes define whether a residue is buried or exposed. Depending on the threshold applied on rSA for discretization, different amounts of residues in either state are obtained. rSA is discretized here with thresholds $0.05, 0.1, ..., 0.5$ and thus ten different two-class prediction targets are obtained.

Secondary structure

Secondary structure is defined equivalently to the 'real time' evaluation server EVA (Eyrich *et al.*, 2001). It is obtained from PDB (Berman *et al.*, 2000) entry files by *DSSP* (Kabsch and Sander, 1983) and the eight *DSSP* secondary structure elements were reduced to three states following the rules: $[GHI]$ become H, a helix representation, $[EB]$ become E, the extended conformation and $[CST]$ become C, representing irregular coil structures. These rules are considered as the most difficult ones.

β-turns

Similar to the previous Chapter 2 β-turns are again included as prediction targets. The β-turn was defined by Lewis *et al.* (1973) as four consecutive residues where the first and the last residue have a distance in space of less than 7Å. The central two residues are not allowed to be part of an α-helix. Essentially they form a tight turn in the protein backbone. The β-turn states were obtained from PDB entry files by *PROMOTIF* (Hutchinson and Thornton, 1996).

3.2.3 Feature space

Equivalent to Chapter 2 and as described by Jones (1999) the raw position specific scoring matrices (PSSM) generated from *PSI-BLAST* (Altschul *et al.*, 1997) serve as solely input to the predictor. The profiles were generated by blasting against a modified sequence database. This sequence database was generated by processing NCBIs non-redundant protein database (NR, Wheeler *et al.* (2007), downloaded in October 2007) with *cd-hit* (Li and Godzik, 2006) to obtain sequence clusters at 98% homology level. The homology reduced database has a size of 3.6 Mio sequences and has the advantage of a reduced search time while prediction performance should not be affected. Further the protein sequences were preprocessed by removing low complexity regions found by *SEG* (Wootton and Federhen, 1993) and coiled coil structures found by *pfilt* (Jones and Swindells, 2002). The *PSI-BLAST* search was performed by iterating three times and setting the e-value cutoffs H for profile sequences and E for output sequences to identical values 0.1×10^{-3}. The raw PSSM values were standardized by the standard logistic function to obtain a value range of $[0, 1]$.

3.2.4 Multiple output layer Elman-type bidirectional recurrent neural network *MOLEBRNN*

In this work again the novel neural network *MOLEBRNN* was utilized that is introduced in section 2.2.2. *MOLEBRNN* (multi output layer Elman-type bidirectional recurrent neural network) is displayed, together with the composing networks, in figure 2.1.

Again an ensembles setup implemented in stacked networks is used to detect correlations between neighboring residues and to smooth predicted structural segments. This stacking is realized with the common two-stage approach introduced in computational biology by Qian and Sejnowski (1988). A first layer (often called sequence-to-structure layer) maps the input features (see above) into the target structural states and a second layer, the structure-to-structure layer, uses this initial mapping as input to further optimize prediction. Similar to Chapter 2 on β-turn

prediction, the first layer consists of independently trained *MOLEBRNN*s with 30, 40, 50 and 60 hidden units and their outcome is fed into a second layer *MOLEBRNN* with 10 hidden units which performs the described correlating and smoothing actions. The full ensemble of predictors is called *SOPRANO* (the solubility predictor applying neural networks with multiple outputs).

3.2.5 Evaluation of prediction

Due to the multiple prediction targets the prediction method has to be evaluated by a variety of measures.

Evaluation of regression

Real value prediction targets are evaluated by Pearson's correlation coefficient (PCC), mean absolute error (MAE), and root mean square deviation ($RMSD$). The Pearson's correlation is defined as the ratio of the covariance between the predicted and observed real value target to the product of their standard deviations. The mean absolute error states the average absolute difference between the predicted and observed real value targets. Both measures are prominently used and defined in greater detail in Ahmad *et al.* (2003a). The root mean square deviation ($RMSD$) is given by

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n}(x_{1,i} - x_{2,i})^2}{n}}$$

where $x_{1,i}$ and $x_{2,i}$ are the observed and predicted real valued targets at residue position i, n states the number of all residues in the test sets, independent to proteins.

Evaluation of classification

For the nominal prediction targets the performance measures *accuracy*, *precision*, *recall*, *gain* and the Matthews correlation coefficient (MCC) were used. The *accuracy* states the percentage of correctly predicted residues in any class. To compute *precision* and *recall* a positive class needs to be defined, where here the solvent exposed stated of a residue is used. The *precision* gives the percentage of residues that are correctly predicted as solvent exposed from those residues that are predicted

as such. The *recall* states the percentage of residues that are correctly detected from all solvent exposed ones. For three-state secondary structure prediction the term Q_3 is a synonym for the *accuracy*. Given the confusion matrix for a two class nominal prediction target which is distinguishing the true positives (TP), the true negatives (TN), the false positives (FP) and the false negatives (FN), *accuracy*, *precision* and *recall* are defined in the equations:

$$\begin{aligned} accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\ precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \end{aligned}$$

The Matthews correlation coefficient (MCC) is given again from the confusion matrix following:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

It considers both classes and all values from the confusion matrix and thus is considered as best performance measure (Baldi *et al.*, 2000). Its value range is $[-1, 1]$; large absolute values indicate good accordance between observed and predicted values while values around 0 indicate bad prediction.

The *gain* given from

$$\begin{aligned} gain &= \frac{precision}{r} \qquad\qquad with \\ r &= \frac{TP+FN}{TP+TN+FP+FN} = \frac{Positives}{N} \end{aligned}$$

states the improvement the classifier achieves when compared to random prediction. It is understood as a factor by which the prediction from the classifier is better than random guessing.

For β-turns the performance measures MCC and *accuracy* are analyzed. As only 25% of all residues form β-turns, post-scaling was applied on the prediction outcome. Post-scaling as described by Lawrence *et al.* (1998), improves performance in case of

rSA **Type**	*PCC*	*MAE*	*RMSD*
rSA	0.688	0.159	0.208
$sqrt(rSA)^2$	0.686	0.155	0.213
$sqrt(rSA)$	0.718	0.167	0.211

Table 3.1: Performance of regression-based real-value solvent accessibility predictions. Shown are the Pearson's correlation coefficient (PCC), the mean absolute error (MAE) and the root mean square deviation ($RMSD$).

unbalanced classes. A β-turn is only predicted if the network output responsible for this structural target is larger than a adjustable threshold. Varying this threshold influences MCC and the *accuracy*. Throughout the work MCC and *accuracy* of the β-turn target is shown for the threshold where a maximal MCC is obtained. (Compare $maxMCC$ in Chapter 2.)

3.3 Results and discussion

3.3.1 Performance of motif prediction

In this section results are shown that were obtained from a *MOLEBRNN* that was trained to simultaneously predict secondary structure, rSA, $sqrt(rSA)$, three solvent accessibility states and β-turns as defined in Material & methods. The reported results were obtained from seven-fold cross-validation on pdb_select-2000 dataset.

Real valued solvent accessibility regression

As described in Material & methods section 3.2.2, two types of real valued solvent accessibility definitions were used: the general accepted rSA value and the modified $sqrt(rSA)$ version. In order to measure whether the modified square rooted version performs as prediction target the predicted $sqrt(rSA)$ value is reverted to $sqrt(rSA)^2$ which is again comparable to the original rSA. All obtained evaluation values are shown in table 3.1. In terms of $RMSD$ the rSA prediction target is predicted best. Also PCC is larger when compared to the $sqrt(rSA)^2$ target. The comparison of the MAE values indicates that the rescaled $sqrt(rSA)^2$ target is predicted best. The target $sqrt(rSA)$ is predicted with best PCC but high MAE. A property

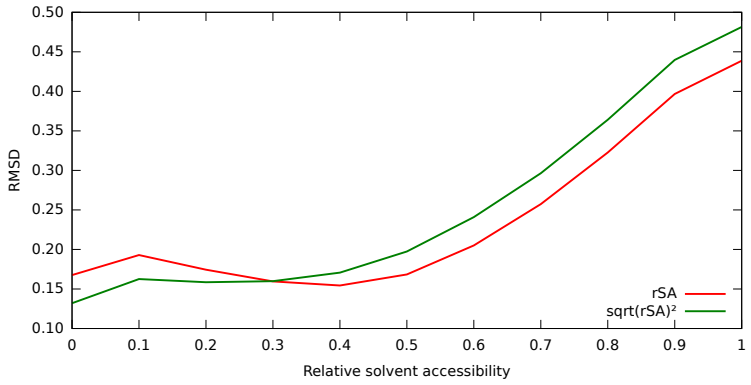

Figure 3.2: Dependency of prediction error ($RMSD$) on solvent accessibility.
rSA and $sqrt(rSA)^2$ targets plotted for comparison.

Threshold	Accuracy	Precision	Recall	MCC	Gain
0.15	79.4764	0.814	0.851	0.569484	1.36
0.20	78.8077	0.795	0.822	0.571862	1.46
0.25	78.4006	0.775	0.792	0.568193	1.57

Table 3.2: Summary of performance measures for solvent accessibility two-class classification.
The threshold column indicates the relative solvent accessibility used to separate into the two classes buried and exposed. *Accuracy*, *precision*, *recall*, MCC and *gain* are defined in Material & methods section.

of the measures MAE and $RMSD$ is that they are distribution dependent. If a distribution has high variance, MAE and $RMSD$ are larger and vise versa. This behavior is observed for the $sqrt(rSA)$ target that has a higher variance (σ) than rSA (see figure 3.1) and thus the MAE and $RMSD$ are larger by definition. Another aspect distinguishing the rSA and $sqrt(rSA)^2$ targets is shown in figure 3.2. The prediction of rSA performs better for exposed residues, it is worse for buried ones when compared to the $sqrt(rSA)^2$ target.

Multiplexed two-class solvent accessibility classification

The performance measures listed in this section are obtained from the same *MOLEBRNN* as those in the previous sections. The results from the seven-fold jack-knife validation are summarized in table 3.2. From figure 3.1 it follows that the thresholds 0.20 and 0.25 are the most informative as they create almost equal amounts of buried and exposed residues. The classification at these thresholds is

Amino Acid	Accuracy	Precision	Recall	*MCC*	Gain
W	70.2	54.1	35.5	0.248	1.23
C	78.6	45.7	35.6	0.276	1.16
K	87.2	91.3	94.5	0.290	1.09
R	77.3	83.2	87.9	0.340	1.22
Y	69.0	66.8	51.9	0.350	1.35
F	76.8	57.8	45.4	0.364	1.26
E	85.3	88.3	94.8	0.413	1.19
D	81.0	84.7	91.7	0.425	1.27
P	75.0	78.4	85.5	0.425	1.37
N	78.8	81.6	91.2	0.431	1.32
L	78.1	63.4	52.4	0.432	1.31
Q	81.1	84.3	92.3	0.432	1.28
I	80.0	59.3	53.9	0.436	1.26
H	73.1	76.1	78.7	0.444	1.42
G	74.2	74.1	82.2	0.473	1.47
V	79.4	68.0	59.0	0.492	1.37
S	76.6	76.9	88.1	0.496	1.46
T	76.6	78.5	84.0	0.506	1.47
M	78.0	69.1	66.9	0.512	1.43
A	78.2	71.6	79.6	0.562	1.52

Table 3.3: Performance measures for individual amino acids when predicting solvent accessibility states buried and exposed with an rSA threshold of 0.20.
The rows are sorted according to increasing MCC.

hardest as there is no valuable *a priori* knowledge about the class distributions. This consideration explains why the accuracy at these thresholds is so low compared to most other thresholds. On the other hand MCC and *gain* are highest at threshold 0.20 which indicates that compared to random prediction, the classifier is improving prediction the most.

To decode some further aspects on prediction performance it is analyzed whether the different types of amino acids are predicted with different accuracy. The analysis results are shown in Table 3.3 and Figure 3.3. Table 3.3 shows average performance measures for all 20 amino amino acids when solvent accessibility is defined as two-class problem and the rSA threshold to distinguish the buried and exposed state is 0.20. It can be seen that there are great differences in prediction performance. While the SA-state of Alanine, Methionine or Threonine is predicted with high MCC, only low values are obtained for Tryptophan, Cysteine and Lysine. Looking deeper into the properties of the various amino acids reveals the following: There is a clear dependency between the maximal solvent accessibility value for amino acids

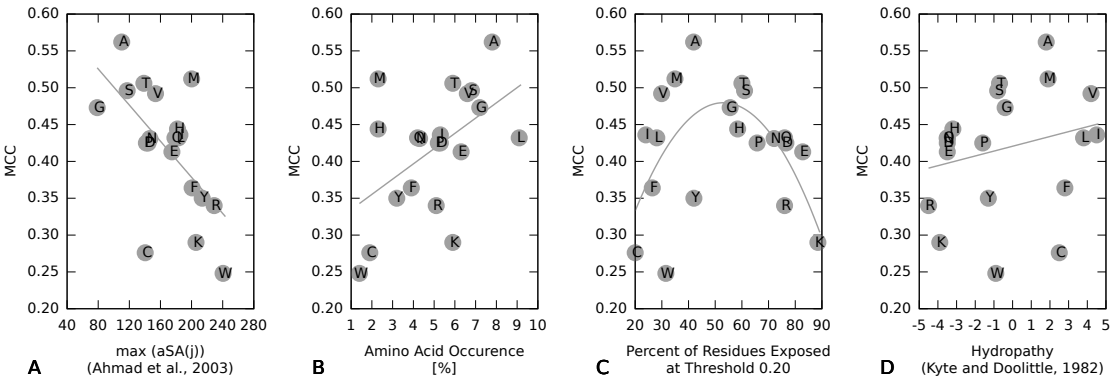

Figure 3.3: Dependency of *MCC* prediction performance on amino acid properties.
Panel A shows *MCC* over the maximal *aSA* per residue, panel B shows the *MCC* over amino acid occurrence in proteins, panel C shows a reverse quadratic dependency between residue exposure in percent and *MCC* and panel D shows how prediction performance depends on the Kyte and Doolittle (1982) residue hydropathy index.

and the associated *MCC* prediction performance (see panel A in Figure 3.3). If $max(aSA(j))$ for amino acids is high, the prediction performance on those residues is low. This can be explained by the fact that large residues can interact with a large number of other residues thus interaction pattern is difficult and can involve many long range interactions which are not tackled to enough extend to gain good performance. On the other hand, small residues interact with just few other protein elements allowing the machine learner to evaluate the associated interaction patterns successfully. The next clear dependency is found between the amino acid occurrence and the *MCC* (visualized in panel B of Figure 3.3). The more often a specific type of amino acid occurs the better its prediction performance. Again this finding can be explained to some extend. If many residues of a specific type occur in a dataset the residue's properties can be learned to a greater extend and this results in a better performance on those frequent residues. Further it has to be noted that mainly high *MCC* values are obtained for amino acids thats class distribution is balanced (panel C in Figure 3.3). Around half of the residues alanine or glycine are exposed when a SA-threshold of 0.25 is applied which in turn maximizes performance in terms of *MCC*. In panel D the dependency of prediction performance on Kyte and Doolittle (1982) hydropathy index is shown. Compared to other articles (Wang *et al.*, 2007; Ahmad *et al.*, 2003a) the performance in this work does not depend that much on

residue hydropathy. There may be two reasons to describe this phenomenon: Firstly, the different index used here to describe the residue hydropathy could influence the analysis and secondly it could be that the performance of *MOLEBRNN* simply correlates much better to the other residue properties analyzed in Figure 3.3.

β-turn prediction

MOLEBRNN in the given setup is able to predict β-turns with an accuracy of 76.9% and gains an MCC of 0.44. This performance measures compare to the previous chapter on β-turn and β-turn type prediction (compare Table 2.6) where an accuracy of 77.9% and an MCC of 0.45 are gained. This time performance is slightly lower which is explained by the missing β-turn sub classes that helped prediction in the previous case.

3-state secondary structure

The secondary structure target is predicted with a residue based Q_3 performance of 77.8% on the pdb_select-2000 dataset after seven-fold cross-validation as described above. The average protein based Q_3 measure is 78.3% and SOV (Zemla *et al.*, 1999) being 73.9%.

3.3.2 Network requirements with increasing class complexity

Choosing the number of hidden nodes in neural networks is crucial for optimal network performance. If the network is designed too small or too large either its accuracy or its ability to generalize is reduced (Swingler, 1996). When the number of nodes is not sufficient, the network is not able to learn the target function. If too many nodes are chosen the network is prone to overfit which again has a negative impact on generalization. In the case of multiple output layers it is dealt with the flow of complex information where it is inevitable to optimize this network parameter. Figure 3.4 shows how increased class complexity influences network size requirements. The *EBRNN* shown in this figure exclusively predicts the solvent

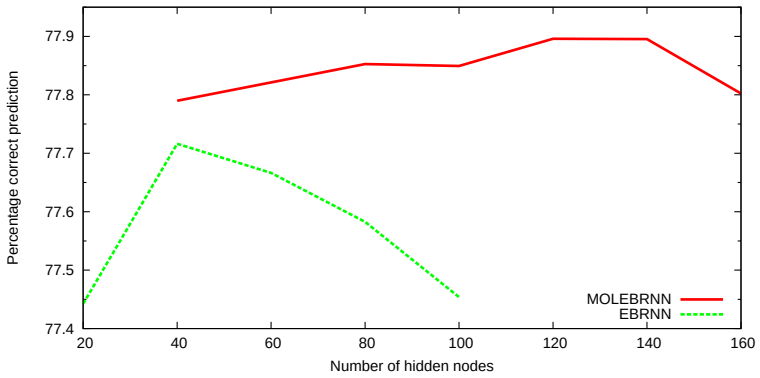

Figure 3.4: Optimization of hidden nodes in the recurrent neural networks.
The accuracy for solvent accessibility two-class prediction with threshold 25% is shown over the number of hidden nodes in the network.

accessibility two-class problem for threshold 25% while the *MOLEBRNN* predicts all targets, rSA, $sqrt(rSA)$, eleven two-class solvent accessibility targets and three-state secondary structure. *EBRNN* performs best with 40 hidden nodes and *MOLEBRNN* requires 120 to 140 hidden nodes. As expected from theory (Swingler, 1996), the size requirement for an optimal *MOLEBRNN* is raised significantly. Further it is observed that performance does not depend that much on the number of hidden nodes. Where the *EBRNN* has a small range for optimal hidden nodes (low variability around 40 hidden nodes) the *MOLEBRNN* is not affected that much from the hidden nodes. Figure 3.4 further shows that *MOLEBRNN* is gaining higher accuracy compared to *EBRNN*. While *EBRNN* saturates at an accuracy of 77.71%, *MOLEBRNN* is reaching 77.9% with 120 hidden nodes. This performance increase for *MOLEBRNN* is observed for all targets and is maximal for the rSA regression target where *EBRNN* gains a *PCC* of 0.667 and *MOLEBRNN* reaches a *PCC* as high as 0.679.

3.3.3 Comparison to other methods

Solvent accessibility prediction

Solvent accessibility is an important structural feature to describe three-dimensional protein structure and thus has been predicted various times in the past (Pollastri *et al.*, 2002a; Ahmad *et al.*, 2003a; Adamczak *et al.*, 2004; Garg *et al.*, 2005; Wang

Dataset	Remark	Method	PCC	MAE	RMSD
S163	full dataset	*SABLE-wa*	0.65	15.5	21.2
S163	full dataset	*MOLEBRNN*	0.6856	15.34	20.29
S163	147 protein chains	*MOLEBRNN*	0.6838	15.37	20.32
S156	full dataset	*SABLE-wa*	0.64	15.7	21.3
S156	full dataset	*MOLEBRNN*	0.6743	15.67	20.56
S156	138 protein chains	*MOLEBRNN*	0.6703	15.73	20.65
S135	full dataset	*SABLE-wa*	0.67	15.3	20.9
S135	full dataset	*MOLEBRNN*	0.6954	15.04	19.89
S135	126 protein chains	*MOLEBRNN*	0.6947	15.03	19.88
S149	full dataset	*SABLE-wa*	0.65	15.8	21.4
S149	full dataset	*MOLEBRNN*	0.6714	15.8	20.69
S149	136 protein chains	*MOLEBRNN*	0.6741	15.73	20.63

Table 3.4: Comparison to *SABLE*.
Shown are the performance measures for the four datasets used in Adamczak *et al.* (2004). See 3.2.1 Material & methods section for detailed information on the datasets.

et al., 2007). In this section comparisons are shown that rank *MOLEBRNN* to other prominent methods available. Essentially three comparison procedures are embodied.

Comparison based on the *SABLE* dataset

The first comparison procedure undertaken measures the difference of *MOLEBRNN* to *SABLE* described in Adamczak *et al.* (2004). The datasets described in 3.2.1 Material & methods are used to obtain the measures discussed in the following. For comparison see Table 3.4 where the Pearson's correlation coefficient (PCC), the mean absolute error (MAE), and the root mean square deviation ($RMSD$) are shown. For all four datasets, $MOLEBRNN$ is gaining higher prediction performance which is conserved for all measures. Only S149 is predicted with equal MAE. As expected from theory the predictions on all "full datasets" (first two lines for all datasets in Table 3.4) are better than the predictions on the redundancy removed sets (last lines). The 56 proteins in the *SABLE* test sets which are similar to some of the here used training proteins are responsible for this improved prediction. When they are removed, the performance is not declining substantially and *MOLEBRNN* is still capable to surpass *SABLE* .

Target	Method	Citation	*PCC*	*MAE*	*RMSD*
rSA	*MOLEBRNN*	this work	0.6833	0.1518	0.2017
$sqrt(rSA)$	*MOLEBRNN*	this work	0.7136	0.1655	0.2091
$sqrt(rSA)^2$	*MOLEBRNN*	this work	0.6813	0.1482	0.2068
rSA	*SVM-cabins*	Wang *et al.* (2007)	0.66	15.1	-
rSA	*Two-stage SVR*	Nguyen and Rajapakse (2006)	0.66 (0.68 [a])	15.7	-
rSA	*SARpred*	Garg *et al.* (2005)	0.65	15.9	-
rSA	*MLR*	Wang *et al.* (2005)	0.64	16.2	-
rSA	*RVP-Net*	Ahmad *et al.* (2003a)	0.48	18.8	-

[a] The original publication states these two values. Not clear which one applies.

Table 3.5: Comparison of real value solvent accessibility regression with the cb502 dataset.

Comparison based on the Cuff and Barton (1999) dataset

Although the cb502 dataset from Cuff and Barton (1999) is aged, it is still a prominent evaluation set used for comparison in a variety of structure prediction applications. As it was already generated in 1999 it does not cover the currently available protein structure space. Nevertheless the performance measures obtained from *MOLEBRNN* on cb502 dataset are also presented. To become comparable a seven-fold cross-validation scheme is adopted. In Table 3.5 the measures for *MOLEBRNN* obtained from the cross-validation are shown. Compared to the other methods, *MOLEBRNN* is first to obtain a Pearson's correlation coefficient of 0.68. Further the novel $sqrt(RSA)^2$ target is predicted very well obtaining a very low MAE of 0.148 with a PCC of 0.68.

The performance of *MOLEBRNN* and recent methods on the discretized two-class solvent accessibility targets is listed in Table 3.6. Again, *MOLEBRNN* clearly outperforms all recent methods, it best predicts the cb502 proteins in terms of both measures: prediction accuracy and MCC.

MOLEBRNN is able to predict β-turns with an accuracy of 75.6% and an MCC of 0.43.

Secondary structure in the cb502 dataset is predicted with a residue based Q_3 as high as 77.2%. The protein based averaged Q_3 and SOV measures are 76.6% and 72.4% respectively.

Threshold	Method	Citation	Accuracy	*MCC*
0.15	*MOLEBRNN*	this work	78.7924%	0.566062
0.20	*MOLEBRNN*	this work	78.319%	0.566033
0.25	*MOLEBRNN*	this work	78.0514%	0.558545
0.25	*PSIMLRacc2*	Qin *et al.* (2005)	77.7%	0.548
0.20	*SARpred*	Garg *et al.* (2005)	76.4%	0.53
0.30	*SARpred*	Garg *et al.* (2005)	76.6%	0.52
0.25	*JNET*	Cuff and Barton (2000)	76.2%	-

Table 3.6: Comparison of discretized two-class solvent accessibility prediction with the cb502 dataset.

Comparison using the Naderi-Manesh *et al.* (2001) dataset

Finally, a comparison is given to methods where performance measures are available on the Naderi-Manesh (Naderi-Manesh *et al.*, 2001) dataset. To our knowledge, the best to date solvent accessibility prediction method, evaluated against the Naderi-Manesh dataset is *PaleAle* published in (Pollastri *et al.*, 2007). For the comparison the proceeding was similarly to Pollastri *et al.* (2007). The most recent pdb_select dataset from October 2007, pdb_select-2007 (see section 3.2.1 in Material and methods) is used where 193 proteins were removed which were homologue to the Naderi-Manesh proteins. Pdb_select-2007 proteins were considered as homologue if a Smith-Waterman local alignment between a pdb_select-2007 and a Naderi-Manesh protein is longer than 50 residues and contains at least 30% identical residues. The *MOLEBRNN* machinery was then retrained with the obtained subset of pdb_select-2007. The performance measures listed in Table 3.7 were obtained when applying *MOLEBRNN* on the 215 Naderi-Manesh proteins. Additionally, Table 3.7 compares *MOLEBRNN* to other methods including *PaleAle* from Pollastri *et al.* (2007). The table lists *MOLEBRNN* ahead of the best performing solvent accessibility predictors. All methods are surpassed significantly. The top performing *PaleAle* which uses large ensembles of bidirectional recurrent neural networks together with a second, filtering layer is toped by *MOLEBRNN*. Please note, that *MOLEBRNN* is not compared to *PaleAle_H* (Pollastri *et al.*, 2007). This method gains an accuracy of 86.0% on the Naderi-Manesh dataset by incorporating templates of known

Method	**Citation**	rSA **threshold**	**Accuracy [%]**
MOLEBRNN	this work	0.15	80.3%
MOLEBRNN	this work	0.20	79.7%
MOLEBRNN	this work	0.25	79.5%
PaleAle	Pollastri *et al.* (2007)	0.25	79.2%
NETASA	Ahmad and Gromiha (2002)	0.25	70.3%
Two-stage SVM	Nguyen and Rajapakse (2005)	0.25	78.1%
PP	Gianese *et al.* (2003)	0.25	71.6%

Table 3.7: Comparison of *MOLEBRNN* to methods tested on the Naderi-Manesh dataset (Naderi-Manesh *et al.*, 2001).

structures. *MOLEBRNN* method is considered as *ab initio*, not requiring template information and applicable on any soluble proteins.

Secondary structure, present in an additional output layer is predicted with Q_3 and SOV accuracies of 79.1% and 76.0% respectively.

β-turns are predicted with an MCC of 0.454 and *accuracy* being 77.9%.

rSA is predicted with a Pearson's correlation coefficient (PCC) of 0.71 and an mean absolute error (MAE) of 0.148. For $sqrt(rSA)$ a PCC of 0.743 is obtained together with MAE 0.159 and $sqrt(rSA)^2$ is predicted with PCC 0.709 and MAE 0.141.

Comparison of secondary structure prediction performances

In this section the secondary structure prediction performance measures obtained from all datasets used are summarized and confronted to popular methods. We experienced that secondary structure prediction algorithms are difficult to compare. Most methods are trained on different datasets with varying protein numbers. The EVA (Eyrich *et al.*, 2001) server has stopped evaluating methods in March 2003 offering only 232 evaluation proteins. The common secondary structure performance measures Q_3 and SOV are listed in Table 3.8 where a comparison to other methods and associated datasets is shown. All these methods are currently available for prediction and EVA considers them as most powerful. According to Table 3.8, *MOLEBRNN* compares quite well and obtains a Q_3 score as high as 79.1% on the

Method	**Dataset**	**Citation**	Q_3[a]	*SOV*
MOLEBRNN	pdb_select-2000[c]	this work	78.3%	73.9%
MOLEBRNN	cb502[c]	this work	76.6%	72.4%
MOLEBRNN	Naderi-Manesh[d]	this work	79.1%	76.0%
PSIPRED	187 CATH T-level proteins	Jones (1999)	76.0%	73.5%
PORTER	2171 proteins from pdb_select of December 2003	Pollastri and McLysaght (2005)	79.0%[b]	75.0%
SABLE2	603 PDB proteins published between January and December 2002	Adamczak *et al.* (2005)	77.3%[b]	-

[a] By default the averaged protein based Q_3 measure is listed
[b] This is a residue based Q_3 measure
[c] Measures obtained from seven-fold cross-validation
[d] Measures obtained when training on reduced pdb_select-2007 dataset, see text

Table 3.8: Comparison of Secondary Structure Performance to popular methods.

Naderi-Manesh proteins. Still, it can not be claimed that *MOLEBRNN* is really and significantly outperforming the other secondary structure prediction methods. Therefor, the variance in the performance is too high ranging from 76.6% to 79.1% given the cb502 and the Naderi-Manesh datasets which indicates high, but non comparable performance. At least *MOLEBRNN* is best applied for solvent accessibility and allows the simultaneous prediction of secondary structure with performance not lower than other methods. The most convincing feature of *MOLEBRNN* is the evaluation procedure applied. There are not much methods available that had been put to the acid test.

3.4 Conclusion

This chapter showed that *MOLEBRNN* could successfully be extended from the β-turn and secondary structure prediction system introduced in Chapter 2 to a one-dimensional structure predictor additionally applicable on multiple solvent accessibility problems. The method is integrated into the software package SOPRANO (solubility predictor applying neural networks with multiple outputs). The solvent accessibility prediction performance is superior to all to-date methods available. Secondary structure prediction performance is highly comparable to current developments. In this chapter it is shown that *MOLEBRNN* has a higher requirement

for hidden network nodes when class complexity increases. The finding in Chapter 2 persists whereupon the performance of *MOLEBRNN* is increased compared to a single target *EBRNN*.

By applying *SOPRANO* it is possible to simultaneously predict multiple one-dimensional protein features singly based on *PSI-BLAST* generated sequence profiles. The method is applicable on genome scale. Its runtime is solely depending on the execution time of *PSI-BLAST* and this time consuming step is required for most other methods and essential for any protein analysis. The method is available for download at http://webclu.bio.wzw.tum.de/soprano.

Chapter 4

Prediction in 1D :: Secondary structure prediction utilizing novel codon profiles

4.1 Introduction

As discussed in the introduction to the thesis, it follows from Anfinsen (1973) that protein structures and functions are solely determined by the peptide sequence. Nevertheless in many biological systems, chaperones are required to assist protein folding. During the folding they help to avoid aggregation or misfolding (Ellis, 1996; Johnson and Craig, 1997). Hence, the chaperones introduce a mechanism to extend beyond the Anfinsen (1973) theory. Additionally it was found that codon usage bias influences folding (Frydman *et al.*, 1994; Kolb *et al.*, 1994; Komar *et al.*, 1999). It is said that folding *in vivo* is co-translational and a vectorial process (Komar *et al.*, 1999) and due to that the ribosome itself as well as the translation process are affecting the folding mechanism and pathways (Komar *et al.*, 1999). It was found that rare codons are placed at particular sequence positions to slow down translation and provide enough time for folding. Rare codons tend to occur at turn, loop and domain linker positions (Thanaraj and Argos, 1996) and codon signals were found at N- and C-terminal ends of secondary structure segments (Brunak and Engelbrecht,

1996; Oresic and Shalloway, 1998).

From all these relations found between codon biases and protein folding and protein structure in this thesis it is concluded that the incorporation of codon profiles could enhance prediction of secondary structure. If amino acid profiles are used as sole input data for a predictor all the biases encoded in the codons and relating to secondary structure are hidden. The information provided by codon profiles is enriched over the amino acid profiles and in theory, covers these.

To the knowledge of the author there exist no secondary structure prediction algorithms that utilize codon profiles as input data. Until now, the sole input to structure prediction methods originate from peptide sequences. Secondary structure prediction methods based on peptide sequences are highly sophisticated in exploiting all information encoded in peptide sequences. Evolutionary information is derived from large amounts of related sequences by utilizing *PSI-BLAST* (Jones, 1999), condensed amino acid representations were introduced to provide specialized information on residue properties (hydrophobicity and amino acid volumes are used for example by Adamczak *et al.* (2005)) and Pollastri and McLysaght (2005) even utilize the residue symbols B, U, X and Z in their profiles.

The development described in this chapter was started by calculating codon secondary structure propensities and compared these propensities to the amino acid secondary structure propensities. This analysis revealed differences in the propensities of many codons and the associated amino acids and it was continued to develop a secondary structure prediction method based on the codon profile.

The aim is to extend the information in the sequence representations by exploiting codon usage bias information and to improve secondary structure prediction accuracy. To evaluate improvements obtained from the codon profiles the predictor was initially evaluated on standard amino acid profiles that constitutes a prediction baseline. The performance on the codon profiles is then compared to the baseline. Finally both profiles were merged and again evaluated the performance.

4.2 Material and methods

4.2.1 Protein set

For the prediction with codon profiles the pdb_select25 (Hobohm and Sander, 1994) database from March 2006 was used to compile the set of test and train protein chains. This database contains 3080 entries with 459963 residues and was compiled from PDB (Berman *et al.*, 2000) by clustering all sequences to obtain a set where sequence identity between all remaining proteins is lower than 25%. As this data is way to large for training a classifier and contains too poor quality structures, a filtering procedure was applied. All NMR structures were removed where chains are shorter than 100 residues, and X-ray structures that have resolutions less than 2 Angstrom. This resulted in a dataset containing 798 protein chains with 162676 residues. In the following this dataset is referred to as pdb_select-filtered.

4.2.2 Reduction of *DSSP* 8 secondary structure states to 3 target states

The gold standard secondary structures were computed by *DSSP* (Kabsch and Sander, 1983) from the PDB entry files. Application of *DSSP* on a PDB entry results in 8 different structure units. Only three types of secondary structures were used for the predictions therefor again the structure states alpha helix (H), 3_{10}-helix (G) and phi-helix (I) were merged to H, extended sheet (E) and beta-bridge (B) were merged to E and bend (S), turn (T) and coil (C) to C.

4.2.3 Representations of protein sequences

To obtain sequence profiles for all proteins and hence include evolutionary relationships *PSI-BLAST* (Altschul *et al.*, 1997) was applied. The number of *PSI-BLAST* iterations was set to 3. Sequences with E-values better than 0.1×10^{-4} were included in the profiles. EBI's nonredundant coding sequence database (EMBL-CDS, Kulikova *et al.* (2007)) taken in July 2006 was used as blast database. This database contains 4.667.241 nonredundant nucleotide sequences and was downloaded from

ftp://ftp.ebi.ac.uk/pub/databases/embl/cds/. The nucleotide sequences were translated to peptide sequences by the use of *transeq*, a tool from the EMBOSS package (Rice *et al.*, 2000). The standard genetic code was used. As discussed in the scope of secondary structure prediction by Jones (1999) a raw blast database is prone to errors that result from the iterative behavior of *PSI-BLAST*. Repetitive sequences that often occur in low-complexity regions may get incorporated into intermediary profiles. This leads to a search drift and unrelated sequences are found and incorporated in the blast profile. To avoid this drift the blast database was filtered the same way Jones (1999) did. First all low-complexity regions were removed with *SEG* (Wootton and Federhen, 1996), and secondly all coiled-coil regions predicted with *pfilt* (Jones and Swindells, 2002) were removed. Finally the database contains protein sequences were all problematic sequence segments are masked. *PSI-BLAST* searches should end up with homologue sequences of high specificity.

Amino acid sequence profile

For each protein, the application of iterative *PSI-BLAST* results in a position specific scoring matrix (PSSM). This matrix is of size $n \times m$ where n is the length of the protein in scope and m is the number of standard amino acids namely $m = 20$. All entries in this matrix are integers in range $[-8, 8]$. For neural network classifiers the input values should comply with ranges $[0, 1.0]$, hence the original PSSM-entries were transformed by applying of the standard logistic function which is

$$y = \frac{1}{1 + e^{-x}}$$

where x is the original entry in the PSSM and y is the transformed entry in range $[0, 1.0]$. This sequence representation by amino acid profiles will be called PSSM-profile in the remainder of this chapter.

Codon score profile

As stated above the translated EMBL nonredundant coding sequence database (EMBL-CDS) was used as blast database. Special sequence identifiers were de-

veloped for all sequences of the EMBL-CDS database which is the CRC32 checksum generated from the raw nucleotide sequences. These identifiers were also used to identify the peptide sequences resulting from the translation of EMBL-CDS. Having done so, the *PSI-BLAST* search results in related peptide sequences with accession identifiers from the corresponding coding sequences in the original EMBL-CDS nucleotide sequence database. For all peptide sequences in the *PSI-BLAST* output their coding sequence can be aligned to the protein query sequence. From this multi-sequence nucleotide alignment amino acid frequencies and scores can be calculated in a similar way as the method *PSI-BLAST* does to compute the position specific scoring matrices (PSSM). For the here described development the PSSM computation was altered so that it is applicable on the 61 codons.

Similar to *PSI-BLAST*, the scores are computed by considering the background frequencies $P(m)$ of the 61 codons. This background was obtained from the EBI nonredundant coding sequence database mentioned above.

The score for a codon $S(i, m)$ at a given position i within a protein chain is calculated from its logodds following

$$S(i, m) = log_2 \frac{Q(i, m)}{P(m)}$$

where $S(i, m)$ is the score for a codon m at sequence position i. $Q(i, m)$ is the observed frequency for position i and codon m. $P(m)$ is the frequency, the codon m occurs in the background model. $Q(i, m)$ is defined in a very similar way as done for *PSI-BLAST*. This includes the weighting scheme for all the homologue coding sequences found. Here the procedure introduced from Henikoff and Henikoff (1994) was implemented. This weighting of the aligned sequences results in a more difficult computation of $Q(i, m)$ because the number of sequences used to calculate the frequencies is no longer properly defined. Please refer to Altschul *et al.* (1997) for a precise discussion. To derive the scores, Altschul *et al.* (1997) introduce so called 'target frequencies' that are altered for the here developed purposes. The 'target frequencies' encode the probabilities that a given amino acid is substituted by any other amino acid or stays conserved while an evolutionary process occurs. Altschul

et al. (1997) define it

$$q(i,j) = P(i)P(j)e^{\lambda_u s(i,j)}$$

where $q(i,j)$ is the 'target frequencies' for the mutation of amino acid i into amino acid j. $P(i)$ and $P(j)$ are the background frequencies for amino acids i and j. It is required that the $q(i,j)$ sum to 1.0. The parameter λ_u is adjusted to be a unique positive number to fulfill this requirement. λ_u is kept constant after adjustment. The values $s(i,j)$ are the substitution parameters taken from PAM (Schwartz and Dayhoff, 1978) or BLOSUM (Henikoff and Henikoff, 1992) matrices.

The 'target frequencies' are redefined in the following manner to cope with 61 codons

$$q(m,n) = P(m|aa_m)P(n|aa_n)P(aa_m)P(aa_n)e^{\lambda_u s(aa_m,aa_n)}$$

Here, m and n are the codons that mutate into each other. aa_m and aa_n are the amino acids coded from codons m and n. $P(m|aa_m)$ is the probability codon m is used to code for amino acid aa_m. Similarly $P(n|aa_n)$ is the probability for codon n to encode amino acid aa_n. Further, $P(aa_m)$ and $P(aa_n)$ are the probabilities that amino acids m and n appear in the background model, $s(aa_m, aa_n)$ is the substitution score taken from BLOSSUM62 substitution matrix (Henikoff and Henikoff, 1992) and λ_u is taken from Altschul *et al.* (1997) and fixed at 0.27. The parameter λ_u is optimized such that the 20x20 'target frequencies' for all possible amino acid substitutions sum up to 1.0. The conditional probabilities $P(m|aa_m)$ sum to 1.0 for all codons that encode a particular amino acid m. This causes the 61x61 $q(m,n)$ 'target frequencies' also sum to 1.0.

The redefined 'target frequencies' are subsequently used in the formulas 4 and 5 of Altschul *et al.* (1997) to derive related position specific matrix scores.

Similar to the *PSI-BLAST* PSSM, the codon score profile contains the Shannon entropy (Shannon, 1997) to encode the variability of the codons at all sequence positions.

The 61 codon scores are filtered by the above described standard logistic function while the Shannon entropy is used unfiltered as an input for the classifier.

4.2.4 Comparison of secondary structure propensities

The secondary structure propensities of codons and amino acids are defined as conditional probabilities $P(SS|Codon)$ and $P(SS|aa)$. These probabilities were not derived from single sequences, but multiple sequence alignments were used which had been obtained from *PSI-BLAST* searches in the above described translated nucleotide sequence databases. This is required because PDB entries can not be mapped reliably to the underlying nucleotide sequences.

To compare the codon propensities with the amino acid propensities a parameter is introduced that is called enrichment E in the following. The enrichment is calculated for all 61 codons plus the 'gap' symbol following

$$E(Codon, SS) = \frac{P(SS|Codon)}{P(SS|aa_{Codon})}$$

where $P(SS|Codon)$ is the propensity for a specific $Codon$ to occur in secondary structure state SS and $P(SS|aa_{Codon})$ is the propensity for the amino acid aa_{Codon} that is encoded from $Codon$. If the enrichment takes the value 1.0 there is no difference between coded amino acid and codon. The more the enrichment deviates from 1.0 the larger is the difference between the codon propensity and its associated amino acid. An example taken from the real values plotted in the Results & discussion Figure 4.2 shows: The enrichment of CTC to form an 'Extended Sheet' is 1.054 compared to the coded amino acid leucine (L). This means that CTC is occurring in extended sheet conformation more often with factor 1.054 compared to its amino acid leucine. On the other hand occurrence of CTC in random coils and alpha helices is reduced by the factors 0.972 and 0.987.

The enrichment is separately computed for the three secondary structure states H, E, C. To uncover those codons with highest deviation in all secondary structure states the 'absolute enrichment' aE is defined which is independent from the three secondary structure states:

$$aE(Codon) = \sum_{i\ in\ \{H,E,C\}} |E(Codon, i) - 1.0| \times 100\%$$

4.2.5 Neural network architecture

For this test again the Elman-type bidirectional recurrent neural network *EBRNN* as described in Chapter 2 was used. The network input is the sequence information encoded in the profiles together and the network output is a probability vector encoding the predicted probabilities for the three target secondary structure states.

Similar to Jones (1999) and the previously described one-dimensional structure prediction methods again a two layered architecture was employed. A sequence-to-structure layer consisting of different combinations of *EBRNN*s trained on the various input data was used and the output of all predictors in the first layer were fed into a structure-to-structure layer.

4.2.6 Evaluation of prediction performance

For parameter optimization and estimation of prediction accuracies a five-fold cross-validation was applied; $^4/_5$ of the proteins were used for training while the remaining set of $^1/_5$ was used for method testing.

The standard performance measures for secondary structure prediction were computed, namely the Q_3 measure together with the segment overlap measure (SOV) (Zemla *et al.*, 1999).

4.3 Results and discussion

4.3.1 Analysis of codon input data

The differences of secondary structure propensities between amino acids and their corresponding coding codons were analyzed. The propensities for 20 amino acids derived from the *PSI-BLAST* sequence alignments are shown in Figure 4.1.

Then the secondary structure propensities for all 61 codons were measured and compared to the corresponding amino acid propensities. Each codon propensity is compared to the propensity from the coded amino acid by the enrichment E (see Material & Methods). The enrichment together with the individual codon occurrences are shown in Figure 4.2. It is observed, that 46 codons do have an absolute

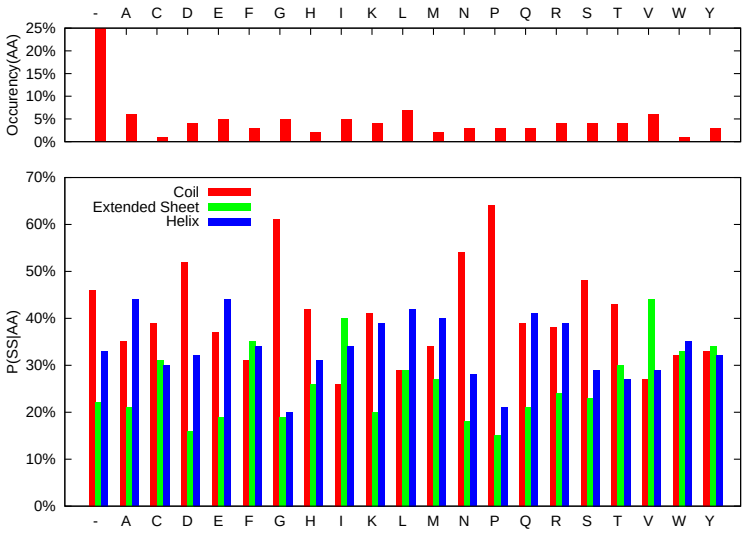

Figure 4.1: Secondary structure propensities for the 20 amino acids derived from the *PSI-BLAST* alignments.

The top plot shows the occurrences of all amino acids in the pdb_select-filtered dataset (Around 25% of the alignment positions are gap positions.). The bottom plot shows the secondary structure propensities.

enrichment compared to the coded amino acid of at least 5%. Proline, known to be a strong coil former and extended sheet and alpha helix breaker (compare Figure 4.1), is coded of the four codons CCT, CCC, CCA and, CCG. The analysis shows, that the codon propensity of CCG is enriched for coding an alpha helix by 1%. Leucine is known to be a strong helix former. By looking at the codon level one observes, that the Leucine codon CTC is enriched by 1.5% to form beta sheets. Its occurrence in helices is suppressed by 0.7%. The isoleucine codon ATA is forming alpha helices more often than the amino acid phenylalanine while the helix propensity of isoleucine is lower than that of phenylalanine. From the absolute enrichments together with the occurrence plots on top of Figure 4.2 it is concluded that those codons with low enrichment occur more frequently.

4.3.2 Performance improvement from codon profile

To be able to measure an improved performance using the codon profile, first the bidirectional network was trained and optimized on amino acid profiles only. The results of the recurrent neural networks on the amino acid profile are shown in

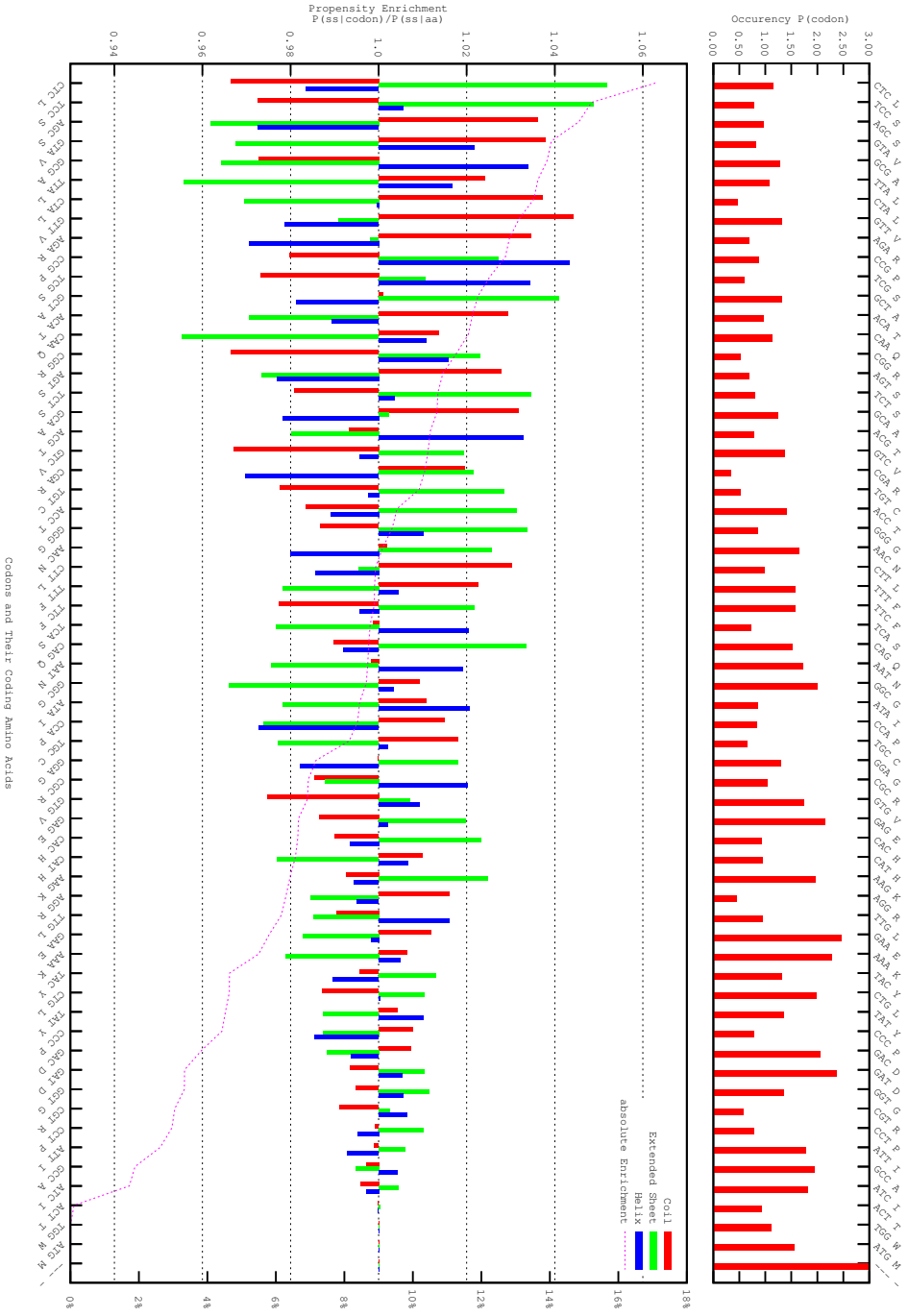

Figure 4.2: Secondary structure propensity enrichments for the 61 codons compared to the propensities from the coded amino acids.

The figure shows the enrichments of secondary structure propensities compared to the baseline of coded amino acid propensity. The x-axis shows all individual codons together with their coded amino acid. The line 'absolute Enrichment' shows the sum of all deviations, its values are printed on the right y-axis. The codons are sorted by the absolute enrichment (see Material & methods). Methionine and tryptophan do not have a deviation because only a single codon encodes these amino acids.

Feature Set	**#Input-Nodes**	**#Hidden-Nodes**	**#Output-Nodes**	Q_3	*SOV*
amino acid only profile	22	70	3	75.98%	71.36%
codon only profile	63	86	3	76.22%	71.77%
amino acid and codon profiles	85	95	3	76.43%	71.80%
structure-to-structure	9	20	3	77.77%	75.03%

Table 4.1: *EBRNN* performances of the analyzed classifiers.
For all feature sets, amino acid only profile, codon only profile and a combined set with amino acid and codon profiles Q_3 and *SOV* measures are listed. The last line lists the performance of a structure-to-structure classifier when trained on the output of the classifiers in lines 1-3. The lines are sorted according to Q_3 performances.

Table 4.1. Next the codon profile were utilized and the prediction algorithm was optimized and trained. The performance measures from the codon profile are also shown in Table 4.1.

A comparison of the Q_3 measure (76.22%) with the best Q_3 from the amino acid only approach (75.98%) shows, that there is an improvement of 0.2%. This shows, that there is some more information in the codon profile compared to the amino acid only profile.

Finally both profiles were combined to see, whether this gives some further improvement. Again, Table 4.1 shows the results of an optimized *EBRNN* on both, codon and amino acid profiles.

Again a slight improvement is obtained. Compared to the codon profile, the combined feature set improves Q_3 performance by 0.2%. Compared to the amino acid only profile, Q_3 is enhanced by 0.4%. Similar to the Q_3 measure, *SOV* is also improved from the codon profiles. While a *SOV* measure of 71.36% is obtained from the amino acid only profile, performance increases to 71.77% if the codon profile is used and saturates at that level when the merged profile is used.

Table 4.1 also clarifies that more complex input codings demand a higher complexity of the *EBRNN*. This behavior is expected and was discussed from Swingler (1996). While the amino acid only profile gets along with 70 hidden nodes, the optimal network trained on the codon profile requires 86 hidden nodes. Finally, the network trained on the merged profiles requires 90 nodes for optimal performance.

Neural networks are 'black box methods', and therefor it cannot be explained why only 4 more nodes are sufficient for the merged profile while the amino acid profile requires 70 nodes and the codon profile requires 86 nodes.

4.3.3 Performance of two layered predictor

In order to produce the best performing secondary structure predictor all optimal sequence-to-structure classifiers with their respective setups listed in Table 4.1 were combined. The output of these classifiers, they produce posterior probabilities for the three secondary structure states H, E and, C, is merged and supplied to the structure-to-structure *EBRNN* classifier. On the pdb_select-filtered dataset, this classifier then gains Q_3 and SOV measures of 77.77% and 75.03% respectively. Although these measures seem quite impressive the developed method does not outperform other to-date secondary structure predictors. But since it was claimed that codon profiles include much more information to aid prediction, nevertheless other methods could not be outperformed significantly. The development of this method was subsequently withdrawn.

4.4 Conclusion

The initial analysis uncovered that codon secondary structure propensities differ from the secondary structure propensities of associated amino acids. This analysis revealed that sequence representation solely based on amino acid data masks details from underlying codons. Similarly to the findings discussed in Section 4.1 (Thanaraj and Argos, 1996; Brunak and Engelbrecht, 1996; Oresic and Shalloway, 1998) secondary structure dependencies in the codon level were found.

Knowing about that dependencies a secondary structure predictor was designed. To this end a sequence representation which encodes codon scores in the manner of a position specific scoring matrix was developed. Three types of prediction systems, based solely on amino acid profiles, based solely on codon profiles and based on a combination of both profiles, were developed. The development involved searching for the best *EBRNN* architecture that maximizes accuracy for all input profiles

independently. Based on the secondary structure prediction accuracies of these three predictors the different sequence representations were compared.

It was found that the lowest performance is obtained from the amino acid only profile, the next best sequence representation was the codon profile while best performance was gained by the combined setup involving the merged amino acid and codon profiles.

Nevertheless, performance improvement was too low to significantly boost prediction in a two layered setup. The performance gained from the structure-to-structure *EBRNN* is not sufficient to compete with currently available secondary structure prediction software. The secondary structure prediction evaluation server *EVA* (Eyrich *et al.*, 2001) lists Q_3 performances of 79.9% for *Porter* (Pollastri and McLysaght, 2005) and 77.9% for *SSpro4* (Pollastri *et al.*, 2002b), while here only 77.8% were gained. Hence, it was decided not to spend more time on the development of a codon based secondary structure prediction method. A better method would be obtained with more sophisticated machine learning algorithms that raise prediction performance with the amino acid profiles to a level comparable to others. Then an extension to the codon profile could surpass other methods.

Chapter 5

Prediction in 2D :: Prediction of contacts in membrane proteins

5.1 Introduction

Integral membrane proteins constitute ∼20-30% of the genome (Frishman and Mewes, 1997; Krogh *et al.*, 2001; Wallin and von Heijne, 1998) and are involved in a large variety of essential cellular functions such as metabolite transport, regulation, host interaction and motility. The importance of membrane proteins is further highlighted by the fact that 50% of all current drugs are targeted against this class of proteins (Klabunde and Hessler, 2002). Two structural architectures are known for polytopic membrane proteins: the α-helix bundle and the β-barrel. While proteins of the first type are found in all cellular membranes, the latter class is observed only in the outer membrane of Gram-negative bacteria, mitochondria and chloroplasts. Due to this restriction in structural complexity it seems obvious that structure prediction of membrane proteins should be clearly easier to tackle compared to water-soluble proteins. However, so far no method could be developed which accurately predicts the 3D structure of a membrane protein based on its sequence alone. To some extent this may be caused by the fact that membrane proteins in general are significantly larger than globular proteins (Elofsson and von Heijne, 2007). Additionally, with more high-resolution structures of α-helical membrane proteins becoming available, it could be observed that α-helical membrane

proteins comprise a clearly higher structural diversity than initially expected. They can contain non-α-helical elements such as 3_{10}-helices, Π-helices or intrahelical kinks (Riek *et al.*, 2001), or may include reentrant loops which cross the membrane only halfway and them turn back to the side where they entered the membrane (Viklund *et al.*, 2006). Helices can also be much longer and much more tilted than anticipated, as was observed for example in the case of the ClC chloride channel (Dutzler *et al.*, 2002). More irregular structures were found in the membrane-water interface region such as interfacial helices positioned roughly parallel to the membrane surface (Granseth *et al.*, 2005).

While the problem of 3D structure prediction of membrane proteins is still far from being solved, several sub-tasks resulting in 2D or so-called 2.5D structure predictions (Elofsson and von Heijne, 2007) have been addressed with more success. Recent HMM-based topology prediction methods which incorporate evolutionary information, such as *HMMTOP2* (Tusnády and Simon, 2001) or *polyPhobius* (Käll *et al.*, 2007) are able to predict the correct number of transmembrane helices and the correct orientation in the membrane in close to 70% of all cases (Elofsson and von Heijne, 2007). Several methods have been proposed to identify non-canonical structural features such as kinks (Yohannan *et al.*, 2004) and reentrant loops (Viklund *et al.*, 2006; Lasso *et al.*, 2006). An even higher number of approaches have been published dealing with the problem of predicting the degree of lipid-exposure of each residue in a transmembrane helix (Beuming and Weinstein, 2004; Pilpel *et al.*, 1999; Adamian and Liang, 2006). Using an empirical scoring function, which combines lipophilicity and residue conservation, an accuracy of 88% was obtained with one of the most recent methods (Adamian and Liang, 2006).

However, less attention has been given to one of the most commonly addressed 2D structure prediction problems in soluble proteins – the prediction of residue-residue contacts. For soluble proteins predicted contacts were successfully utilized for 3D structure prediction (Bonneau *et al.*, 2002b; Ortiz *et al.*, 1999). Additionally, predicted contacts can be used to identify incorrect models generated by a threading method (Olmea *et al.*, 1999) or to estimate folding rates (Punta and Rost, 2005b). A variety of methods for contact prediction in soluble proteins have been developed,

which either rely on correlated mutational behavior of residues (Fleishman *et al.*, 2004; Göbel *et al.*, 1994; Kundrotas and Alexov, 2006; Olmea and Valencia, 1997; Shindyalov *et al.*, 1994) or use machine approaches (Fariselli *et al.*, 2001b,a; Pollastri and Baldi, 2002; Punta and Rost, 2005a; Shao and Bystroff, 2003; Cheng and Baldi, 2007) and genetic programming (MacCallum, 2004). Within several editions of the CASP experiment, contact prediction methods have been evaluated as an independent category (Graña *et al.*, 2005; Izarzugaza *et al.*, 2007; Moult *et al.*, 2003, 1997, 1999). Average prediction accuracies for the best performing contact prediction groups of the CASP6 experiment ranged between 16% and 23% and were clearly superior to contacts derived from predicted 3D structures (Graña *et al.*, 2005).

Angelika Fuchs has recently conducted the first analysis of correlated mutations in polytopic membrane proteins (Fuchs *et al.*, 2007). In this study she was able to show that co-evolving residues alone are not sufficient to predict helix-helix contacts, but that these residues still carry a strong signal for the detection of interacting transmembrane helices due to their frequent occurrence in close sequence neighborhood to helix-helix contacts (Fuchs *et al.*, 2007).

This chapter presents the first neural-network based approach specifically developed for the prediction of helix-helix contacts in α-helical membrane proteins. Based on the work of Angelika Fuchs (Fuchs *et al.*, 2007) on correlated mutations **Angelika Fuchs and me jointly developed the here described extension of the helix-helix contact predictor in membrane proteins**. It is difficult to decipher the respective contributions but a breakdown is attempted in the affected sections.

The predictor integrates sequence profiles, correlated mutations, protein topology, sequence separation and predicted scores for lipid-exposure. Using the best performing predictor, an average prediction accuracy of 25.9% is obtained for the $L/5$ highest scoring predictions (L being the combined length of the transmembrane segments of the protein) based on a dataset of 62 membrane proteins with available 3D structure. It is further demonstrated how the predicted contacts can be utilized to identify interacting transmembrane helices distant in sequence, which is an important step in discrimination of different helix architectures of membrane pro-

teins. Based on a simple selection procedure, which requires several predicted residue contacts to rate a given helix pair as interacting, incorrectly predicted helix-helix contacts are removed and that processing allows prediction of interacting helices with a sensitivity of 53.1% and a specificity of 86.3%. This approach clearly outperforms the results earlier obtained with correlated mutations alone (Fuchs *et al.*, 2007).

5.2 Material and methods

5.2.1 Dataset

A non-redundant dataset of membrane proteins with solved structure was constructed using the Protein Data Bank of Transmembrane Proteins (PDBTM, Tusnády *et al.* (2005)) and the dataset provided by the Stephen White laboratory at UC Irvine (http://blanco.biomol.uci.edu/Membrane_proteins_xtal.html) as of September 17, 2007 (further referred to as the White dataset). Starting with the non-redundant set of PDB chains containing α-helical transmembrane segments obtained from the PDBTM, an initial dataset of those proteins was created, whose structure was solved by X-ray with a resolution of less than 3.5Å and which contained at least three transmembrane segments according to the PDBTM annotation. Since this initial set consisting of 50 PDB chains was lacking several prominent membrane proteins with solved structures such as rhodopsin, it was subsequently enriched with sequences from the White dataset. To this end, first all chains with less than three transmembrane segments in their PDBTM entry from the White dataset were eliminated. Additionally, all sequences with at least 40% sequence identity to another sequence with better resolution (either within the White dataset or in the initial dataset) or with a resolution worse than 4Å were removed. Both the moderate threshold for sequence identity and the relaxed threshold for structural resolution at this step are concessions needed to be made due to the limited number of available membrane protein structures. The remaining 12 sequences were merged with the sequences from the initial dataset to form the final set of 62 protein chains originat-

pdb entry	**chain**	**pdb entry**	**chain**	**pdb entry**	**chain**	**pdb entry**	**chain**	**pdb entry**	**chain**
1aig	l	1bcc	c	1eys	m	1fft	a	1fft	c
1fx8	a	1jb0	a	1jb0	l	1kqf	c	1l7v	a
1m0k	a	1orq	c	1pw4	a	1q16	c	1qle	c
1rh5	a	1u19	a	1vf5	a	1vf5	b	1xio	a
1xme	a	1yew	b	1yew	c	1zcd	a	2a65	a
2a79	b	2acz	c	2acz	d	2agv	a	2axt	a
2axt	b	2axt	c	2axt	d	2b2f	a	2b76	c
2b76	d	2bg9	a	2bhw	a	2bl2	a	2bs2	c
2c3e	a	2cfp	a	2evu	a	2exw	a	2f93	a
2fbw	c	2fbw	d	2fyn	a	2gfp	a	2gif	a
2gsm	a	2hi7	b	2hyd	a	2ic8	a	2jaf	a
2nmr	a	2nr9	a	2nwl	a	2o9d	a	2oau	a
2onk	c	2uuh	a						

Table 5.1: The identifiers of the used PDB protein chains.

ing from 52 PDB structures (Table 5.1). Exact transmembrane segment positions and the in/out topology for each protein were obtained from the recently developed TOPDB (Tusnády *et al.*, 2008), which contains comprehensive topology information derived both from literature and public databases for a large number of membrane proteins. For two cases (2UUH chain A and 1ORQ chain C) no entry could be found in the TOPDB, therefor transmembrane positions for these proteins from the PDBTM and the in/out topology from *OPM* (Lomize *et al.*, 2006) were obtained.

The final dataset includes proteins with three up to thirteen transmembrane segments with close to 25% of all sequences (15 out of 62) containing ten or more transmembrane segments. Despite the liberal threshold of 40% sequence identity used for the construction of the dataset, the pairwise sequence identity in the final dataset is low, with less than 2.5% of all possible sequence pairs having a sequence identity above 30% and less than 0.5% having a sequence identity above 35%. Since the aim is to predict structural contacts within the transmembrane parts of each protein, the pairwise sequence identities among proteins after aligning only their concatenated transmembrane segments were also evaluated. Naturally, the obtained values are slightly higher due to the hydrophobic nature of transmembrane segments. Still, less than 5.5% of all protein pairs had a sequence identity of higher than 35%.

The preparation of the datasets was conducted by Angelika Fuchs.

5.2.2 Contact definition

The same helix-helix contact definition was used as described in Fuchs *et al.* (2007). Briefly, two residues within different transmembrane segments were considered in contact if the minimal distance between side chain or backbone atoms was less than 5.5Å. While contact prediction methods developed for soluble proteins mostly use C_β-distances and a contact threshold of 8Å, this contact definition incorporating side chain atoms is considered to be more appropriate for membrane proteins with their regular α-helix bundle structure. The difficulty of the contact prediction problem for membrane proteins is not influenced by the choice of contact criterion since the number of observed contacts remains basically the same. Using the contact criterion the observed contact density (the number of observed contacts divided by the number of possible pairs) was 0.021 while the usage of the C_β contact criterion resulted in a contact density of 0.020 for the dataset of 62 membrane proteins.

Contacts were defined by Angelika Fuchs.

5.2.3 Contact density

To estimate the optimal number of contacts to predict per protein, the dependency between the number of transmembrane residues and the amount of helix-helix contacts was investigated. The observed contact density within the transmembrane parts of the 62 transmembrane proteins was compared to the corresponding values derived for soluble proteins taken from the 25% homology threshold list of the pdb_select database from October 2007 (Hobohm and Sander, 1994). Two different subsets of pdb_select were used, one comprising all 3652 pdb_select proteins belonging to the SCOP (Andreeva *et al.*, 2008) classes all-alpha, all-beta, alpha and beta (a/b), alpha and beta (a+b) and multidomain proteins and one subset consisting of all-alpha proteins only. In any case, contacts were calculated according to the definition given above. For every dataset linear functions were fitted. They are describing the dependency of the number of observed contacts on the length of the protein (for membrane proteins only the transmembrane parts were considered). The following (rounded) dependencies between the number of considered residues

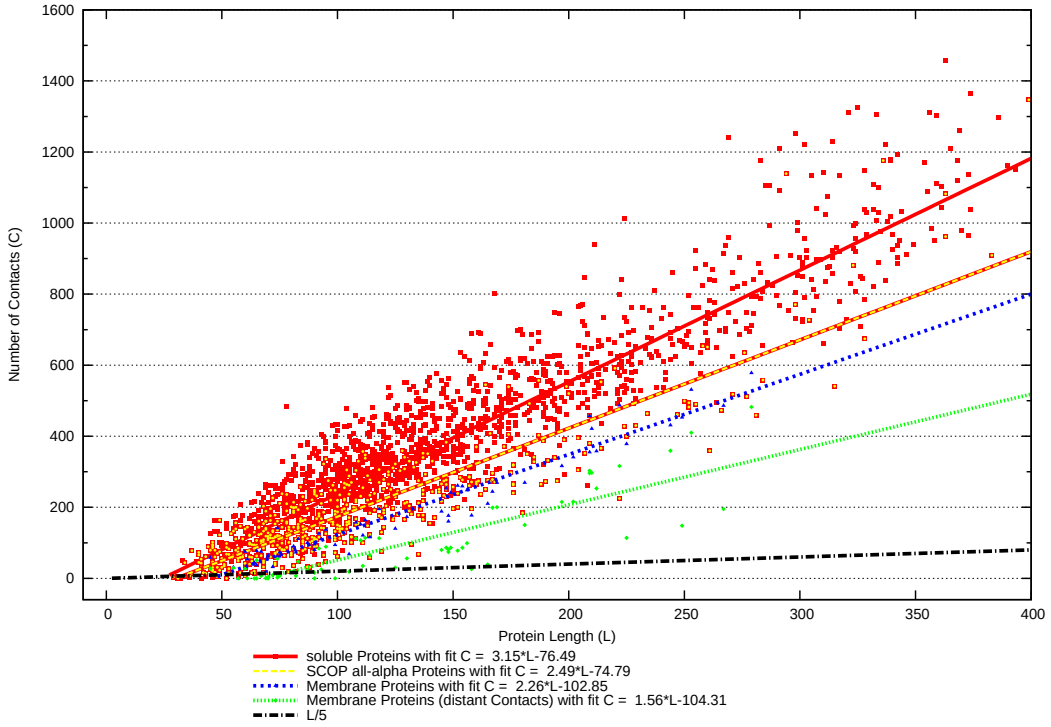

Figure 5.1: Contact density (number of contacts depending on protein length) of membrane proteins compared to soluble proteins.

The amount of contacts for any type of proteins is linearly proportional to the protein length. Most contacts are found for soluble proteins. All-alpha proteins have fewer contacts than soluble proteins in general, but still more contacts than the analyzed membrane proteins. Disregarding contacts between adjacent transmembrane helices, even fewer contacts are observed. The fitted curves represent contact functions that can be used for the selection of an appropriate amount of contacts (see Section 5.2.3). In comparison to the observed number of contacts, the $L/5$ line shows the commonly used selection criterion, which is quite stringent.

L and the amount of observed contacts C were found: For soluble proteins in general $C = 3.15L - 76.5$, for all-alpha proteins $C = 2.5L - 75$ and for the membrane proteins $C = 2.25L - 100$ (Figure 5.1).

The procedure resulting in the definition of contact densities and associated function fitting was iteratively carried out by both authors.

5.2.4 Input features

The prediction of spatial contacts between two amino acid residues is generally based on the analysis of multiple sequence features. These features can be divided into out-of-context features defined for single residues without any contact related infor-

mation, features targeting properties related to residue pairs in contact, and features that describe global properties of the proteins. Contact prediction is then derived by mapping these features onto the contact state of the residues under observation. Over the last years, machine learning algorithms have become a method of choice to obtain such mapping in an automated fashion (Fariselli *et al.*, 2001b,a; Pollastri and Baldi, 2002; Punta and Rost, 2005a; Cheng and Baldi, 2007). The better the chosen features relate to the contact information of two residues, the better the mapping and thus the better the predictive performance of the developed algorithm. Accordingly, for the membrane protein contact prediction problem, prominent features were selected. Some of these are used for globular protein contact prediction and others included various features that are available for membrane proteins only.

Out-of-context features

The out-of-context features describing individual residues are: windowed PSSM (position-specific scoring matrices) profiles, the position of each residue within the transmembrane helix (cytosolic side of the membrane, hydrophobic core or extracellular side), and the orientation of its side chain, i.e. whether the residue is facing towards the lipophilic membrane or the protein interior. The PSSM profiles were obtained using *PSI-BLAST* (Altschul *et al.*, 1997) searches against the NCBI's unfiltered NR database (Wheeler *et al.*, 2008), with three iterations and the inclusion of related database sequences into the profile with an E-value threshold of 1×10^{-4}. The raw profiles from *PSI-BLAST* contained scores for all residue positions representing their amino acid preferences. These scores were transformed by the standard logistic function to obtain values in the range [0..1]. In order to include information about adjacent residues as well, a window of five residues to the left and five residues to the right was employed together with the central target residue. An additional feature was included to indicate whether the window was not built properly due to missing data (i.e. at the end of the protein sequences). The position of each residue within the transmembrane helix was encoded by two distinct features. First, a boolean vector of length S was used to represent each transmembrane helix divided into a set of S fragments of equal size. The values of the vector were initialized with 0 and

the value at vector index $s = \lfloor \frac{S}{N} \times i \rfloor$ was set to 1 with N representing the length of the transmembrane helix, i being the position of the described residue within the transmembrane helix numbered from 1 to N from the N- to C-terminal end and the function $f(x) = \lfloor x \rfloor$ returning the largest integer which is less or equal the real number x. Based on preliminary optimization experiments, the parameter S was fixed at $S = 7$. Second, a boolean vector of size three was used, to encode whether a residue lies close to the extracellular side of the membrane, close to the cytoplasm or within the hydrophobic core of the helix. A region of seven residues was used to define both the extracellular or cytoplasmic side of the helix. The side chain orientation of each residue was calculated using *LIPS* (Adamian and Liang, 2006), a method for the prediction of transmembrane helix orientation with the reported accuracy of close to 90%. *LIPS* defines seven helical surfaces called faces which are identified based on the average lipophilicity and the conservation of residues within each face. Large *LIPS* scores indicate that a particular face is oriented towards the membrane while low scores indicate an orientation towards the hydrophilic membrane protein interior. The helix orientation was encoded in a boolean vector of length seven with the elements in the vector representing the seven helical faces ordered by increasing average lipophilicity. The vector is initialized by zeros. If a residue is member of the helical face with the i-th highest *LIPS* score, this i-th element is set to 1 in the boolean vector. A single residue can participate in up to three helical faces, as defined by Adamian and Liang (2006).

Features of residue pairs

To represent properties pertinent to paired residues, two features were considered: sequence distance between the residues and predicted correlated mutation rates indicating co-evolving residues. The distance between two residues was encoded by a boolean vector of length eight corresponding to sequence separations of less than 25, 50, 75, 100, 150, 200, 300 residues, or more. For a given pair of residues having a sequence separation corresponding to the vector element i, not only this vector element was set to 1 but also all vector elements at positions $\leq i$.

Residue co-evolution was calculated using three different prediction methods.

The algorithm *McBASC* (Olmea and Valencia, 1997) was applied in two variations, using either the *McLachlan* (McLachlan, 1971) or the *Miyata* (Miyata *et al.*, 1979) substitution matrix, and the modified version of the *OMES* algorithm (Fodor and Aldrich, 2004). Multiple sequence alignments used for the calculation of correlation scores were obtained from the *PSI-BLAST* alignments. First, all positions were removed from the full length *PSI-BLAST* alignment which did not correspond to any transmembrane segment of the PDB sequence resulting in an alignment representing only the transmembrane parts of the reference sequence. Following the procedure described in Fuchs *et al.* (2007), sequences thought to be inappropriate for the prediction of correlated positions were discarded. The raw correlation scores were standardized individually for all proteins following the formula $y = \frac{x-min}{max-min}$, where x is the raw correlation score and min and max are the minimal and maximal scores observed for a given protein and algorithm. Applying this type of standardization conserves relative scores but makes results from different proteins comparable. As Fuchs *et al.* (2007) previously established, co-evolution in membrane proteins occurs much more often at residue pairs in close vicinity to an actual helix-helix contact than at the contact positions themselves. Therefor not only the correlation scores found for the pair of residues i and j under observation were included, but also adjacent residue pairs with a window size of 5 centered around the positions i and j, respectively were encoded.

Global features

Two global protein features were considered in the neural network: protein length and the number of transmembrane helices. Both descriptors are again encoded as boolean vectors using the same strategy as described for the sequence distance. The protein length vector has a size of five elements corresponding to protein lengths of less than 100, 200, 400, 800 or more residues. The vector describing the number of transmembrane helices has a length of ten encoding proteins with 3, 4, 5, 6, 7, 8, 9, 10, 11 or 12 and more transmembrane regions.

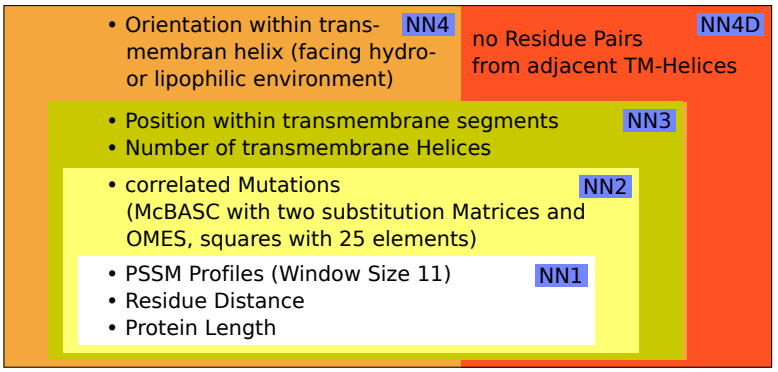

Figure 5.2: Input features used for the prediction of helix-helix contacts of membrane proteins.

The feature complexity increases from *NN1* to *NN4*. All feature sets enclose all less complex feature sets. The *NN4* set contains all applied features. *NN4-D* contains all the features of *NN4*. Here only residue pairs are considered that are distant in sequence, lying on non neighboring α-helices only.

Combination of features

Each input vector representing a residue pair contains out of context features for both participating residues, residue pair features, and global features of the particular protein. To estimate the importance of the various features, input vectors of increasing complexity were constructed and thereupon iteratively improved prediction performance (Figure 5.2). Starting with an input vector consisting of only those features available also for soluble proteins (*NN1* and *NN2*, without and with correlated mutations, respectively), membrane protein specific features were gradually added (*NN3*: position within transmembrane segment and total number of transmembrane helices; *NN4*: side chain orientation). The *NN4* implementation was additionally evaluated with a dataset that did not include instances with residue pairs from sequentially adjacent helices (termed *NN4-distant* or *NN4-D*) in order to find out how the predictive performance depends on short range contacts between neighboring helix pairs. Throughout this work, *NN4*, the neural network based on the full set of input features, is synonymously also referred to as *TMHcon*, the name of the final version of the contact predictor which is also available on-line (http://webclu.bio.wzw.tum.de/tmhcon/) together with the predictor *NN4-D*.

The following features were developed and incorporated by myself: PSSM profiles, residue distance, protein lengths, position within the transmembrane segments

and the number of transmembrane segments. Angelika Fuchs provided all correlated mutations and came up with the idea about the *LIPS* scores. The idea to *NN4-D* came from myself. The association of features to networks was organized by Angelika Fuchs.

5.2.5 Neural network architecture and training

Similar to many contact prediction methods for globular proteins, feed-forward neural networks specially trained for data with biased class distributions was used. Every network consisted of the same number of input nodes as features available. It had two output nodes which represent the two prediction classes 'contact' (positive class) and 'nonContact' (negative class). The number of hidden nodes was varied in order to optimize prediction performance, and finally an architecture with 90 hidden nodes was chosen. Generally, during each training iteration of a neural network, called epoch, a set of instances is presented to the neural network, the average error on the given set is estimated and this error is used to calculate the weight update for all node connections. The presentation and weight update process is repeated until a defined stop criterion is reached. The contact prediction network was trained such that for each epoch all positive (contact) instances of the training proteins were selected and for all these proteins an equal number of negative instances was randomly sampled. The training was iterated over 200 epochs.

The neural network was implemented and trained by myself.

5.2.6 Measuring contact prediction performance

To assess the prediction performance of the neural networks the leave-one-protein-out jackknife cross-validation was applied. This tests the method on a single protein while all other proteins are used as training set. Performance measures are obtained for the test protein and the procedure is repeated for all proteins. The overall prediction performance is calculated by averaging the individually obtained performance results leading to an accurate assessment of method performance. Following common practice the number of predicted contacts was chosen based on the length of the

protein L. Since the aim is to predict helix-helix contacts within the transmembrane helices of a protein only, L was calculated as the sum of the lengths of all transmembrane helices of a given protein. Reported contact prediction accuracies are based on the $L/5$ highest scoring residue pairs, a threshold commonly used in contact prediction assessment (Izarzugaza *et al.*, 2007). From this number of predicted contacts the prediction accuracy (fraction of correctly predicted contacts out of all predicted contacts) was calculated. Additionally, the coverage (fraction of correctly predicted contacts out of all observed contacts) was calculated. In order to investigate the position of predicted contacts with respect to observed helix-helix contacts, a 'δ-Analysis' (Ortiz *et al.*, 1999) was embodied which calculates the fraction of correlations between residues i and j given an observed contact between residues in the interval $i-\delta, i+\delta$ and $j-\delta, j+\delta$. A value of $\delta = 4$ was used to determine the fraction of predicted contacts where both participating residues lie within one helix turn of residues forming an inter-helical contact.

Performances were measured by procedures of both authors. They applied individual versions, resulting in double checked values.

5.2.7 Identification of interacting helices

To apply the obtained contact predictions for the problem of identifying interacting pairs of transmembrane helices, a dataset of helix-helix pairs from the dataset of 62 membrane proteins was constructed. From the total number of 1486 helix pairs, 714 helix pairs were considered to be in contact since they contained at least one residue pair less than 5.5Å away from each other. Using this dataset of helix pairs interacting transmembrane helices were predicted based on the number of predicted contacts for every helix pair. To this end, the initial list of predicted contacts was compiled based on two different strategies, either using the protein length based $L/5$ criterion or employing the formula for the number of observed contacts found for a given number of residues described in Section 5.2.3. Several thresholds for the required number of predicted contacts for a positive prediction were evaluated by calculating the sensitivity and specificity of each obtained prediction. The significance of each prediction was calculated based on a chi-square test. For comparison reasons, inter-

acting helices were also predicted with the previously developed method *HelixCorr* (Fuchs *et al.*, 2007), which uses solely correlated mutations for the prediction process.

Starting with the development of Fuchs *et al.* (2007), the here required sophistication for the identification of interacting helices was jointly developed by both authors.

5.3 Results and discussion

5.3.1 Prediction of helix-helix contacts using neural networks with increasing complexity

Machine learning techniques have been applied for the prediction of amino acid contacts in soluble proteins for more than five years (Fariselli *et al.*, 2001b,a; Pollastri and Baldi, 2002; Punta and Rost, 2005a; Cheng and Baldi, 2007). Here the first application of neural networks for the specific problem of predicting helix-helix contacts in membrane proteins is introduced. Using contact data derived from 62 membrane proteins with solved structure, five neural networks were trained for the prediction of helix-helix contacts. While four of these networks were developed in order to analyze the influence of different input features on the resulting prediction, the neural network *NN4-D* included the same input features as the network *NN4*, but was trained only on long-range contacts lying on non-neighboring transmembrane helices. Such long-range contacts are particularly important for the discrimination between membrane protein folds resulting from different helix packing in α-helix bundles and therefor, the aim was to especially predict them with high reliability.

Influence of different input features on the prediction of helix-helix contacts

Following the strategy reported for the first contact map predictions using neural networks in globular proteins (Fariselli *et al.*, 2001b,a), neural networks of increasing complexity were constructed by incorporating an increasing number of input features (see Figure 5.2). While the first two neural networks (*NN1* and *NN2*) included

Predictor	$L/5$			Contact density formula		
	Acc[a] [%]	**Acc** ($\|\delta\| = 4$)[b] [%]	**Cov**[c] [%]	**Acc** [%]	**Acc** ($\|\delta\| = 4$) [%]	**Cov** [%]
NN1	17.2	65.2	2.3	10.5	61.2	10.6
NN2	18.9	68.4	2.6	11.4	65.4	11.6
NN3	23.5	78.7	3.2	15.7	70.8	15.8
NN4	25.9	78.5	3.5	15.8	70.7	16.0
NN4-D	14.8	50.2	3.9	10.0	46.0	10.1

[a] Prediction accuracy: Fraction of correctly predicted contacts out of the total number of predicted contacts.
[b] Prediction accuracy ($|\delta| = 4$): Fraction of predicted contacts lying within one helix turn of an observed contact.
[c] Coverage: Fraction of correctly predicted contacts out of the total number of observed contacts.

Table 5.2: Contact prediction with neural networks of increasing complexity.
Contact prediction accuracy, accuracy ($|\delta| = 4$) and coverage are reported based both on the selection of the $L/5$ highest scoring residue pairs (L being the length of the concatenated transmembrane segments), and after selecting the expected number of contacts derived using the contact formula for membrane proteins describing the observed number of contacts in dependence on the number of participating residues (see Materials & methods).

only sequence features also available for soluble proteins (e.g. sequence profiles, sequence separation, protein length and correlated mutations), membrane protein specific features were incorporated in the neural networks *NN3* and *NN4* (position of each residue within a transmembrane helix, number of transmembrane helices and orientation of each residue). This step-wise procedure reveals the contribution of individual feature sets, in particular those not available for soluble proteins and therefor missing in earlier studies on contact prediction with neural networks.

In agreement with publications on contact prediction for soluble proteins, the $L/5$ highest scoring contact pairs for every protein were selected and the accuracy and the coverage are reported based on this procedure. Additionally the delta-accuracy ($|\delta| = 4$) was calculated. As described above, this measure describes the fraction of predicted contacts that are found within one helix turn of an observed contact and therefor lie in close sequence neighborhood to an actual helix-helix contact (Table 5.2). As seen in Table 5.2, prediction accuracy increases by more than 8% with the addition of more and more input features. While the incorporation of correlated mutations leads to an improvement of 1.6% accuracy, the most significant increase in prediction accuracy of 4.6% is achieved with the first addition of membrane protein specific features in *NN3*. The incorporation of *LIPS* scores in *NN4* leads to a further

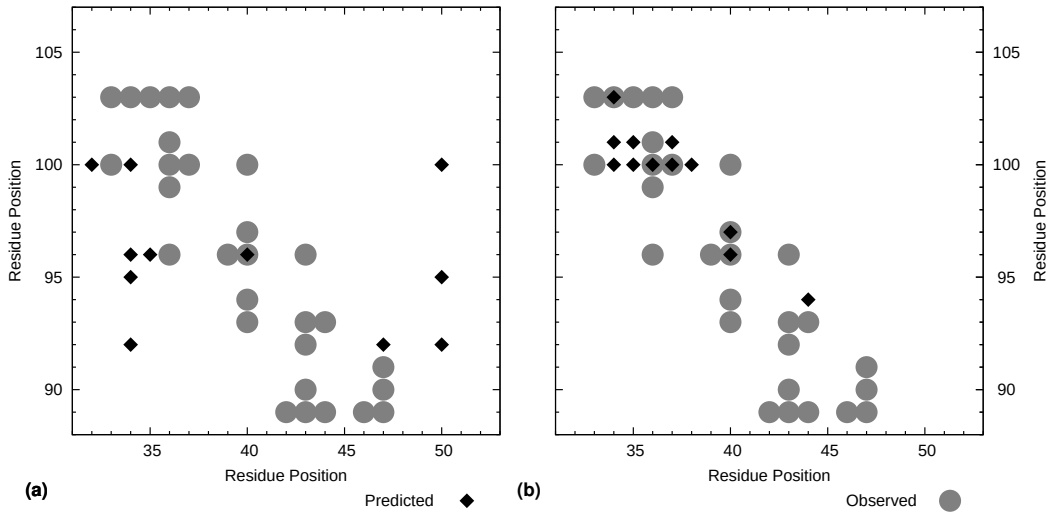

Figure 5.3: Observed and top $L/5$ predicted contacts between transmembrane helix 1 and transmembrane helix 2 of the protein 1VF5, chain A.

(a) Predictions with *NN2*. (b) Predictions with *NN3*. *NN3* includes information about each residue's position within the transmembrane helix and therefor is aware of the helix orientation. While *NN2* has problems to detect the antiparallel character of the helix interaction and predicts many contacts off diagonal, *NN3* exclusively predicts contacts that capture the anti-parallel interaction pattern.

improved prediction accuracy of 25.9%. Since the number of analyzed predictions is equal for all neural networks, the coverage increases accordingly. The same trend can be observed for the accuracy ($|\delta| = 4$), which increases by more than 13% from *NN1* (65.2%) towards *NN4* (78.5%). Interestingly, the observed value is basically constant between *NN3* and *NN4*. In both cases around 78% of all predicted contacts are found in close sequence neighborhood to an observed helix-helix contact. Since the number of predicted contacts located close to an actual contact stays the same while the number of correctly predicted contacts increases from *NN3* towards *NN4*, the addition of *LIPS* scores seems to be helpful in determining the exact position of helix-helix contacts, which are otherwise only located slightly misplaced from the correct position.

Since the most remarkable increase in prediction accuracy is obtained from *NN2* towards *NN3* with the inclusion of a feature group defining each residue's position within the transmembrane helix, the predictions of *NN3* was investigated in greater detail. As can be seen from the example in Figure 5.3, the given relative position of each residue within the transmembrane helix seems to aid the neural network

in detecting the parallel or antiparallel interaction pattern of two transmembrane helices and therefor constrains predicted contacts. Figure 5.3a illustrates observed and the top $L/5$ predicted contacts for residues on transmembrane helices 1 and 2 from cytochrome B6 (1VF5 Chain A) when using *NN2*. Here, contacts are predicted for the given two transmembrane helices, but the algorithm is not able to detect in which orientation the two helices are positioned relative to each other, resulting in a significant deviation of the predicted contacts from the known ones. In contrast, *NN3* (Figure 5.3b) is able to deduce information on the helix orientation, and thus the predicted contacts lie on the correct diagonal of the contact map. The neural network is constrained by the transmembrane residue positions: a residue near the extracellular membrane surface cannot contact a residue near the cytoplasmic membrane surface.

The feature contributions were jointly analyzed by both authors.

Dependence of the contact prediction performance on the number of transmembrane helices

For the best performing neural network *NN4* it was further analyzed how the prediction success depended on the number of transmembrane segments within a protein. Therefor the set of 62 membrane protein was grouped into subsets of proteins with a similar number of transmembrane segments and calculated the prediction accuracy and coverage for every subset (Table 5.3). As expected, prediction accuracy decreases for large proteins. For proteins with eight or more transmembrane helices prediction accuracies of close to 20% are obtained, while proteins with less than eight transmembrane segments prediction accuracies of 25% or more were achieved. Interestingly, the fraction of predicted contacts in close vicinity to observed contacts (accuracy ($|\delta| = 4$) is largely independent of the protein size since in proteins having more than ten transmembrane helices contacts are still detected with an accuracy ($|\delta| = 4$) of more than 80% which is even slightly above the mean value found for all proteins (78.5%, Table 5.2). However, the best contact predictions are obtained for proteins with seven transmembrane helices. These proteins typically belong to the class of G-protein coupled receptors (GPCRs) whose structures largely resemble

TMS	$N(Proteins)$	$N(Contacts)$	$L/5$		
			Acc[a] [%]	**Acc** ($\|\delta\| = 4$)[b] [%]	**Cov**[c] [%]
3-4	19	260	33.1	77.7	7.8
5-6	17	359	25.1	72.4	4.2
7	7	201	40.3	93.5	5.0
8-10	7	242	19.0	71.9	2.6
>10	12	549	20.9	80.1	2.2

[a] Prediction accuracy: Fraction of correctly predicted contacts out of the total number of predicted contacts.
[b] Prediction accuracy ($|\delta| = 4$): Fraction of predicted contacts lying within one helix turn of an observed contact.
[c] Coverage: Fraction of correctly predicted contacts out of the total number of observed contacts.

Table 5.3: Contact prediction using *NN4/TMHcon* for subsets of membrane proteins grouped according to their number of transmembrane helices.
Reported prediction accuracies, accuracies ($|\delta| = 4$) and coverage are based on the selection of the $L/5$ highest scoring residue pairs.

the canonical α-helix bundle structure with only few helix-helix contacts between sequentially distant transmembrane helices (Palczewski *et al.*, 2000), facilitating contact prediction for these targets.

Angelika Fuchs performed the here described analysis.

Dependency of the contact prediction performance on the number of selected contacts

The dependency of prediction quality on the number of predicted contacts was also investigated. Figure 5.4 illustrates how the obtained prediction accuracy and the coverage depend on the cutoff for the number of analyzed contacts. While *NN2* performs better than *NN1*, as do the two neural networks with membrane protein specific input features *NN3* and *NN4* compared to *NN1* and *NN2*, the improvement of *NN4* compared to *NN3* is varying with the number of selected contacts. While for large numbers of predicted contacts *NN3* and *NN4* perform more or less with equal accuracy and coverage, the highest improvement of prediction accuracy due to addition of *LIPS* scores as input features in *NN4* is obtained for small numbers of predicted contacts ($L/3$ or less). The same can be observed from Table 5.2 where the quality measures are reported for predictions with the number of predicted contacts determined using a contact formula. This formula was derived from available

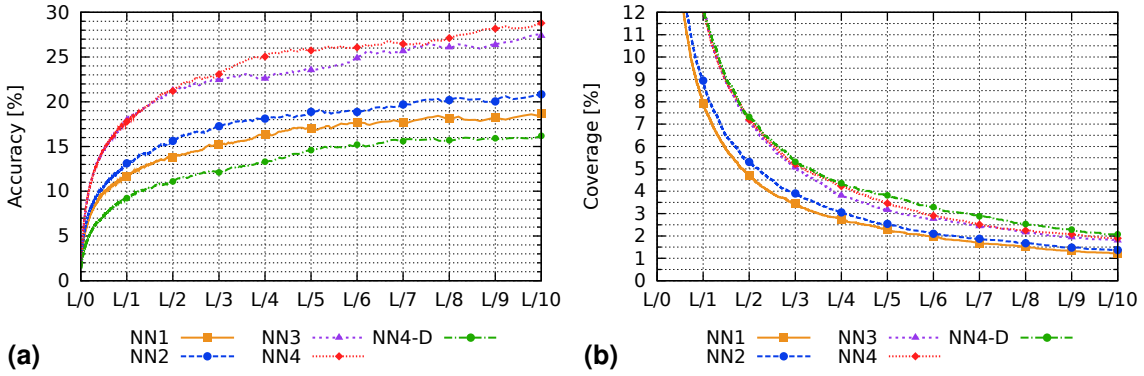

Figure 5.4: Contact prediction accuracy (a) and coverage (b) of different neural networks as a function of the number of predicted contacts (L/X).

As expected, the accuracy is increasing while the coverage is decreasing for more stringent criteria. Both performance curves increase steadily for the *NN1*-*NN3* architectures while the improvement of *NN4* is less clear cut. L/X scaling for *NN4-D* is not comparable to the other architectures since the number of possible residue pairs and the number of observed contacts is different.

membrane protein structures describing the number of expected contacts for a given number of participating residues (see section 5.2.3 in Materials & methods). A higher number of predicted contacts lead to an obvious decrease in prediction accuracy in favor of an increased coverage. While the increase in prediction accuracy from *NN1* towards *NN3* is still clearly visible, *NN3* and *NN4* perform with more or less equal accuracy and coverage.

This analysis was carried out by myself.

Contact prediction in membrane proteins compared to soluble proteins

It is well known that the prediction of intra-molecular amino acid contacts gets increasingly difficult with decreasing contact density (fraction of observed contacts among the total number of possible residue pairs, Punta and Rost (2005a)). This is the reason why contact predictions for large proteins are generally less successful than predictions for small proteins (Fariselli *et al.*, 2001b; Pollastri *et al.*, 2002b), and why all-alpha soluble proteins, whose contact density is roughly only a half of the contact density found for all-beta proteins (Punta and Rost, 2005a) were consistently found to pose special difficulties for the prediction. To compare the success of the contact prediction for membrane proteins at least at a very basic level to the results obtained

for soluble proteins, the contact density for the membrane proteins in the dataset was calculated. Figure 5.1 shows the dependency of the number of observed contacts in a protein on the protein length for four different types of proteins: soluble proteins, soluble proteins in the SCOP class all-alpha, the 62 membrane proteins (only transmembrane segments considered), and the 62 membrane proteins where residue pairs lying on neighboring helices were not considered. For all four datasets linear fits were calculated. While all-alpha soluble proteins were found to possess slightly fewer contacts than soluble proteins in general, as was reported earlier (Punta and Rost, 2005a), the number of observed contacts within membrane proteins was found to be even more reduced compared to soluble proteins in general and all-alpha soluble proteins in particular. When residue pairs on neighboring helices were ignored, the number of observed contacts was further decreased significantly, indicating that the prediction of helix-helix contacts in membrane proteins is at least of comparable difficulty to the prediction of intra-molecular contacts within all-alpha soluble proteins, if not more difficult. Thus, compared to prediction accuracies reported for all-alpha soluble proteins (20% for a $L/10$ prediction based on 30 proteins at a sequence separation of 8 (MacCallum, 2004), 24% for a $L/2$ prediction based on 131 proteins and a sequence separation of 6 Punta and Rost (2005a)), the here described contact predictor for membrane proteins has equal quality to state-of-the-art methods for soluble proteins. This is also true for the prediction of long-range contacts. Using the neural network *NN4-D*, which predicts only contacts between non-neighboring transmembrane helices, a prediction accuracy of 14.8% (Table 5.2) can be obtained. Reported values for all-alpha soluble proteins with a sequence separation of at least 24 amino acids range between comparable values of 13.5% ($L/2$ prediction, Punta and Rost (2005a)) and 15.3% ($L/10$ prediction, MacCallum (2004)).

Both authors worked on this analysis.

5.3.2 Prediction of interacting helices

After demonstrating the capability of the method to predict helix-helix contacts in membrane proteins with equal accuracy to state-of-the-art methods for soluble proteins, the potential application of these predicted contacts for another structural

problem in membrane proteins was further investigated: the identification of interacting helices. With more and more 3D structures of membrane proteins being available, it is now common understanding that alpha-helical membrane proteins may deviate remarkably from simple helix bundle structures. A study on helix-packing arrangements proposed a possible number of 1,500,000 different folds for a membrane protein with seven transmembrane helices (Bowie, 1999). Recent studies trying to classify the naturally occurring membrane protein fold space suggested a limited number of ∼250-500 different membrane protein folds (Martin-Galiano and Frishman, 2006; Oberai *et al.*, 2006). However, the difficulty of membrane protein structure determination has led to the estimation that three more decades will be required to obtain a structural representation of 90% of the current membrane protein sequence space (Oberai *et al.*, 2006). Therefor, the reliable prediction of helix interaction patterns may be a valuable tool to distinguish membrane proteins of different folds without knowing their structure or to assign a new protein sequence to a known membrane protein fold. Based on the dataset of 62 proteins used for the contact prediction, a dataset of 1486 helix pairs was compiled. In this dataset 714 helix pairs were rated as interacting since they contained at least one helix-helix contact in the corresponding 3D structure. To predict interacting helices using the obtained contacts two different strategies were employed and compared.

The initial idea was to select the $L/5$ highest scoring contact pairs (with L being defined as the sum of the transmembrane segments' lengths) and every helix pair is predicted as interacting it if has at least one predicted contact. However, as can be seen from Figure 5.1, the solely sequence length dependent threshold $L/5$ is much too restrictive to obtain a number of contacts typical for an alpha-helical membrane protein. Additionally it was observed that the number of contacts per helix pair predicted by *NN4/TMHcon* tends to increase with the number of observed contacts per helix pair (Figure 5.5). After selecting predicted contacts based on the contact density formula introduced in the Material & methods section, helix pairs with more than 5 actual helix-helix contacts were found to have on average 15 predicted contacts (median: 8) while helix pairs with only a small number of helix-helix contacts between one and five had nine predicted contacts on average (median:

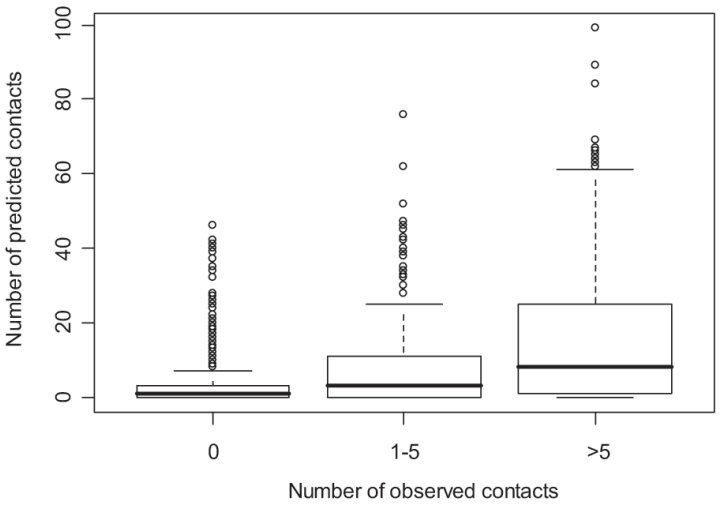

Figure 5.5: Dependency of the number of predicted contacts on the number of observed contacts.

Helices with a given number of observed helix-helix contacts (0 1-5, or more than 5) were grouped. The number or predicted contacts increases in average with the number of observed contacts.

3).

The second prediction strategy for interacting helices was developed based on this observation. The initial number of predicted contacts for every protein is derived from the contact density formula. Afterwards, a threshold of required contacts for an interacting helix pair is applied to remove wrongly predicted interacting helices. Similar to the approach introduced for the *HelixCorr* method (Fuchs *et al.*, 2007), this contact threshold can be used to achieve predictions of increasing specificity at the cost of decreasing sensitivity.

Using these two strategies (termed length-based prediction and contact-based prediction) interacting helix predictions for all four neural networks could be obtained and specificity, sensitivity, accuracy and significance based on a chi-square test for each prediction was calculated (Table 5.4). Since the contact-based predictions made at different thresholds of required contacts are hard to compare, always those those predictions measures are presented where the specificity is closest to 80% and 90%. As can be seen from Table 5.4, the contact-based selection resulted in a

Method	Threshold [a]	$N(predicted)$[b]	Neighboring [%] [c]	Accuracy	Sensitifity	Specificity	P-value
HelixCorr	C7	462	0.381	0.662	0.429	0.798	7.36×10^{-21}
	C11	292	0.428	0.743	0.304	0.903	2.35×10^{-23}
NN1	$L/5$	359	0.604	0.721	0.363	0.870	1.76×10^{-25}
	C4	494	0.579	0.713	0.493	0.816	2.71×10^{-36}
	C9	336	0.732	0.774	0.364	0.902	4.43×10^{-34}
NN2	$L/5$	327	0.651	0.761	0.349	0.899	2.28×10^{-30}
	C3	531	0.571	0.718	0.534	0.806	5.06×10^{-42}
	C7	366	0.724	0.795	0.408	0.903	2.05×10^{-43}
NN3	$L/5$	380	0.697	0.761	0.405	0.882	1.97×10^{-36}
	C4	565	0.593	0.729	0.577	0.802	1.04×10^{-50}
	C10	373	0.796	0.796	0.416	0.902	8.39×10^{-45}
NN4 (*TMHcon*)	$L/5$	413	0.654	0.780	0.451	0.882	3.72×10^{-46}
	C4	587	0.571	0.722	0.594	0.789	5.10×10^{-51}
	C9	397	0.756	0.804	0.447	0.899	8.48×10^{-51}
NN4-D	C7	324	-	0.580	0.438	0.807	1.76×10^{-18}
	C10	212	-	0.665	0.329	0.899	3.49×10^{-21}
NN4/NN4-D	C9/C10	552	0.543	0.748	0.578	0.820	2.09×10^{-56}
	C9/C15	485	0.619	0.781	0.531	0.863	2.24×10^{-58}

[a] Predicted contacts used for the identification of interacting helices were selected with two different procedures. $L/5$ corresponds to the length based selection of predicted contacts while CX describes the number X required contacts for an interacting helix pair after compiling an initial list of contact predictions using the contact density formula for membrane proteins described in Materials & methods.
[b] The number of predicted interacting helices.
[c] The percentage of neighboring helix pairs out of the total number of predicted interacting helices.

Table 5.4: Prediction of interacting transmembrane helices using helix-helix contacts predicted by neural networks of increasing complexity.

For comparison, results obtained with *HelixCorr*, a method using only correlated mutations for the identification of interacting helices, are also reported.

more significant prediction for all of the four neural networks than the length-based selection.

The parameter optimization and subsequent analysis was done by Angelika Fuchs.

Prediction performance of neural networks with increasing complexity

A comparison of the performance of the different neural networks produced similar results to those obtained in the analysis of predicted helix-helix contacts. Predictions based on the same selection strategy showed a clear increase in accuracy, sensitivity

and p-value at the same specificity level with increasing complexity of the used neural network. For example, using length-based ($L/5$) selection, all four neural networks resulted in a prediction of interacting helices with a specificity between 87% and 89%, while the prediction accuracy increased from 72% towards 78%. At the same time, the sensitivity increased by 9%, and the p-value decreased from 1.76×10^{-25} to 3.72×10^{-46} (Table 5.4). The same can be observed using a contact-based selection strategy. When predictions with the same specificity (for example 90%) were compared, again accuracy and sensitivity increased (3% and 8%, respectively, in the case of 90% specificity) while the p-value decreased (from 4.43×10^{-34} towards 8.48×10^{-51}, again for predictions with 90% specificity).

Angelika Fuchs performed the here presented analysis.

Prediction of interacting helices distant in sequence

For every prediction also the fraction of predicted interacting helices that are neighboring in sequence was calculated. Despite all deviations from the canonical alpha-bundle structure found in membrane proteins, neighboring helices still have a clearly higher probability for interaction with each compared to non-neighboring helix pairs (80.5% compared to 37.9% for non-neighboring helix pairs in the dataset). Therefor, a primitive way of predicting interacting helices in membrane proteins would be to predict all neighboring helices as interacting and non-neighboring helices as not interacting. While this prediction method would lead to a high prediction accuracy of 80.5% in the case, its subsequent application for the discrimination of different membrane protein folds would be impossible, since no differences in the helix packing of proteins with the same number of transmembrane helices could be determined. Optimally, one would therefor wish to obtain predictions with a small fraction of neighboring helices (possibly close to the naturally occurring fraction of 39.9% in the dataset), to get a maximum of information about the specific fold of the protein. A comparison of *NN1* and *NN2* (Table 5.4) reveals that the incorporation of correlated mutations as input feature results in predictions of higher sensitivity and accuracy at equal specificity with a slightly smaller fraction of neighboring helices in the set of predicted helices using the contact-based selection (with 90% speci-

ficity 73.2% neighboring helices with *NN1* and 72.4% with *NN2*). The additionally detected interacting helices are therefor primarily long distance helix pairs, implying that co-evolving residues are generally independent of sequence separation (see also the discussion of *HelixCorr* results below). In contrast, the first incorporation of membrane protein specific features (residue position within the transmembrane helix as well as the total number of transmembrane helices) within *NN3* resulted in a strong increase of the number of neighboring helices in the prediction (at 90% specificity 79.6% with *NN3* compared to 72.4% with *NN2*). This demonstrates a general tendency of the neural network to learn about the helix-bundle structure of membrane proteins from basic membrane protein specific input features. The addition of *LIPS* scores within *NN4* reduces the fraction of neighboring helices again to a final value of 75.6% for the prediction with 90% specificity. Since the fraction of falsely predicted non-interacting neighboring helices decreases at the same time (from 17.2% with *NN3* towards 15.3% with *NN4*), the inclusion of *LIPS* scores (the predicted orientation of each residue towards the membrane or the protein interior) seems to prevent the incorrect prediction of those amino acid residues as being in the contact state which would originally be well positioned on neighboring helices to form a contact in a perfect helix bundle structure.

In order to increase the fraction of non-neighboring helices in the final prediction a neural network was trained especially on long-range contacts by omitting all helix-helix contacts from neighboring helices from the training set (*NN4-D*). Using contacts predicted by this neural network and selected according to the contact formula derived for non-neighboring helices (see Materials & methods) a prediction of distant interacting helices was obtained. Due to the increased difficulty of predicting contacts on non-neighboring helix pairs resulting from the smaller contact density (Figure 5.1), the sensitivity and accuracy of this prediction was clearly lower than those obtained for the full dataset (Table 5.4). However, at 80% specificity still 43.8% of all distant interacting helices could be correctly predicted. More than 32% of these interacting helices were predicted with close to 90% specificity. To enhance the original *NN4/TMHcon* prediction with long distant interactions the helix pairs predicted from *NN4-D* were included into the single *NN4/TMHcon* prediction. Af-

ter adding all helix pairs with at least 10 predicted contacts (corresponding to the 90% specificity prediction of *NN4-D*), the significance of the prediction increased to 2.1×10^{-56} (Table 5.4). While still 57.8% of all interacting helices were predicted with a specificity of 82%, the fraction of neighboring helices decreased to only 54.3%. This prediction was further improved by raising the threshold of required contacts for *NN4-D*, corresponding to the increased difficulty of long-range contact prediction. With 15 required contacts a final prediction with a significance of 2.2×10^{-58}, a sensitivity of 53.1% and a specificity of 86.3% was obtained. The fraction of neighboring helices was only 61.9%, a clear improvement compared to the original *TMHcon* prediction.

To come up with the optimal combination of *NN4* with *NN4-D*, both authors discussed the processing jointly. The here shown analysis was performed by Angelika Fuchs.

5.3.3 Application of *TMHcon* to three membrane proteins with recently solved structure

To test the *TMHcon* prediction ability under 'real-life' conditions, the newly developed method was applied to three membrane proteins whose structure was solved after the construction of the data set: the site-2 protease (3B4R Chain B, Feng *et al.* (2007)), the sodium-potassium pump (3B8E Chain A, Morth *et al.* (2007)) and the plasma membrane proton pump (3B8C Chain A, Pedersen *et al.* (2007)). None of these proteins had more than 30% sequence identity to any of the proteins in the 62 protein data set. Transmembrane helix positions determined from the 3D structure were obtained from PDBTM. Additionally transmembrane helices were predicted by Phobius (Käll *et al.*, 2007) to simulate the case when no protein structure is available. While Phobius predicted transmembrane helix number and position consistent with the PDBTM annotation in the case of 3B8C, one transmembrane helix was not detected in the case of 3B4R, and two were missing in the case of 3B8E. Subsequently, the helix-helix contacts were predicted both with *TMHcon* and *NN4-D* and the derived contacts were used to predict the helix-helix interaction patterns

for all three proteins. The same prediction parameters were chosen as in the most significant earlier prediction. This requires at least 9 predicted contacts by *TMHcon* or 15 predicted contacts by *NN4-D* in order to predict a helix pair as interacting.

While for all three proteins an average prediction accuracy ($L/5$) for helix-helix contacts close to 20% was obtained for transmembrane helices taken from the PDBTM, this value decreased to only 13% in case transmembrane helices were predicted with *Phobius*. However, the fraction of predicted contacts within one helix turn of an observed contact was remarkably high both for transmembrane helices taken from PDBTM and predicted by *Phobius*, resulting in an even higher accuracy ($|\delta| = 4$) than in the original data set (87.1% for *Phobius*, 86.3% for PDBTM). Therefor, the majority of all predicted contacts were found on actual interacting helices (Figure 5.6a) regardless of the method used for determining transmembrane helix positions. Accordingly, the predicted helix interaction patterns closely resemble the actually observed patterns (Figure 5.6b).

Angelika Fuchs carried out the analysis. Myself provided the raw predictions.

5.3.4 Comparison to other contact prediction methods

To further assess the benefit of the here described contact prediction method which is specifically developed for membrane proteins, the obtained prediction results were compared to predictions obtained using available state-of-the-art contact prediction methods. Despite the fact that these predictors were developed exclusively for soluble proteins, they might still be capable of detecting the contact pattern originating from the alpha-helical bundle structures of membrane proteins. Accordingly, predictions were obtained for the set of 62 membrane proteins using the contact predictor PROFcon (Punta and Rost, 2005a), a neural network based predictor ranking among the best performing methods in the CASP6 competition, as well as using SVMcon (Cheng and Baldi, 2007), a contact map predictor based on support vector machines, one of the top predictors in the CASP7 experiment. Since both methods returned predicted contacts for the full length sequence of each protein, obtained predictions were filtered by removing all contacts lying outside the transmembrane parts of the protein or within the same transmembrane helix. From the remaining contacts, the

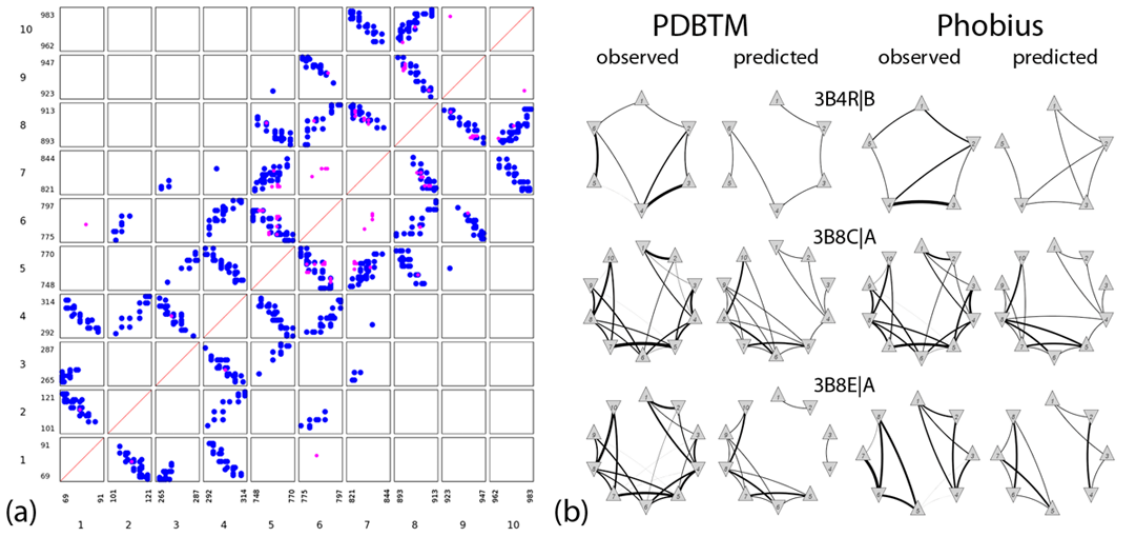

Figure 5.6: Prediction of helix-helix contacts and interacting helices for three membrane proteins newly added to the PDB (3B4R Chain B, 3B8C Chain A, 3B8E Chain A).
(a) Contact map of 3B8E (Chain A). Shown are only the transmembrane parts of the protein as given by the PDBTM database. Observed helix-helix contacts are depicted in blue, the $L/5$ best predicted contacts by *TMHcon* are shown in pink. Only six predicted contacts are found on non-interacting transmembrane helix pairs. (b) Observed helix interactions and predicted helix interactions for all three proteins based on transmembrane segments predicted by *Phobius* or extracted from PDBTM. Circles represent transmembrane helices while connection lines represent an interaction between the participating helices. The thickness of the connection line in the case of the observed helix interactions corresponds to the number of observed helix-helix contacts. Interacting helices were predicted by compiling an initial contact list from predictions obtained with *TMHcon* and *NN4-D* using the contact formula for membrane proteins or distant contacts in membrane proteins, respectively and selecting all helix pairs with at least 9 helixhelix contacts predicted by *TMHcon* or 15 helix-helix contacts from *NN4-D*. Bold connections lines in the case of predicted helix interactions indicate interacting helices predicted both by *TMHcon* and *NN4-D*.

top $L/5$ scoring ones were selected for every protein and every method (Table 5.5).

Using PROFcon, predictions could be obtained for 43 proteins out of the total set of 62 proteins. However, since PROFcon restricts the number of returned contacts to $2L$, the number of proteins with predicted contacts within their transmembrane helices was only 24. Based on the $L/5$ selection criterion, an average contact prediction accuracy of 4.2% was obtained for these 24 proteins. The accuracy ($|\delta| = 4$) was found to be 36.8%. Despite these low values, PROFcon was still able to produce comparable results to *TMHcon* in individual cases with a maximum prediction accuracy of 21% and a accuracy ($|\delta| = 4$) of 98% obtained for the ammonia channel AmtB (2NMR| Chain A). Using SVMcon, predictions were obtained for all 62 proteins. The average prediction accuracy was 9.3% and the delta prediction accuracy

Predictor	$N(Proteins)$	$N(Contacts)$	$L/5$		
			Acc[a] [%]	**Acc** ($\|\delta\| = 4$)[b] [%]	**Cov**[c] [%]
HelixCorr	62	4822	10.8	51.9	4.4
PROFCon	24	503	4.2	36.8	0.2
SVMCon	62	1600	9.3	55.8	1.3
TMHcon	62	1611	25.9	78.5	3.5

[a] Prediction accuracy: Fraction of correctly predicted contacts out of the total number of predicted contacts.
[b] Prediction accuracy ($|\delta| = 4$): Fraction of predicted contacts lying within one helix turn of an observed contact.
[c] Coverage: Fraction of correctly predicted contacts out of the total number of observed contacts.

Table 5.5: Contact predictions for 62 membrane proteins using external contact predictors or *TMHcon*.

Reported prediction accuracies, accuracies ($|\delta| = 4$) and coverage are based on the selection of the $L/5$ highest scoring residue pairs.

($|\delta| = 4$) was 55.8%, resulting in total in a clearly superior prediction compared to PROFcon without reaching the prediction accuracies obtained with *TMHcon*. Again the obtained prediction quality was significantly differing among proteins with eight proteins having a prediction accuracy of 20% or more while 23 proteins were found with no correctly predicted contact at all. The best prediction using SVMcon was obtained for the sensory rhodopsin II with a prediction accuracy of 31% and an accuracy ($|\delta| = 4$) of 97%. Based on these results it is clear that the development of a membrane protein specific contact predictor is necessary since currently available contact predictors are not able to predict contacts within transmembrane helix over a large set of proteins.

The neural network based predictions were also compared to predictions with the earlier developed *HelixCorr* method (Fuchs *et al.*, 2007). *HelixCorr* is a consensus approach for detecting co-evolving residues in transmembrane helices in order to predict helix-helix contacts and interacting helices. This method was applied to the dataset of 62 membrane proteins using the same alignments consisting of concatenated transmembrane segments which were also used for deriving correlated mutations used as input feature for the neural networks. To improve prediction accuracy, *HelixCorr* includes a filtering step, where all predicted co-evolving residue pairs on helices with fewer correlations than a given threshold are removed. In agreement with Fuchs *et al.* (2007) 5 correlated pairs were required as filter threshold for the

helix-helix contact prediction. Similar to the previous work which was based on the careful analysis of 14 membrane proteins with solved structure, a prediction accuracy for helix-helix contacts of roughly 10% (Table 5.5) was obtained. *HelixCorr* is easily outperformed by even the most basic neural network reaching a prediction accuracy of 17% (Table 5.2). The same is true for the prediction accuracy ($|\delta| = 4$) where the most basic neural network achieves a more then 10% higher quality score than *HelixCorr* (64.6% compared to 51.9%). This observation is consistent with reported results for soluble proteins where the prediction of intra-molecular contacts was improved by at least 7% after using a neural network instead of correlated mutations alone (Fariselli *et al.*, 2001a). It was also analyzed how well *HelixCorr* performs in the prediction of interacting helices compared to *TMHcon* based on the 62 protein data set. Again, the increase in prediction quality from *HelixCorr* towards *TMHcon* is quite remarkable. In order to obtain comparable results between *HelixCorr* and the neural networks the above mentioned filter threshold of *HelixCorr* was varied and here the results for those filter thresholds are reported when prediction is resulting in a specificity close to 80% and close to 90% (Table 5.4). In both cases *TMHcon* predictions of basically equal specificity to comparable *HelixCorr* predictions resulted in a clearly higher sensitivity and accuracy. An increase in accuracy of up to 6% (*HelixCorr* with 7 required contacts (C7) was observed when comparing to *TMHcon* with 4 required contacts (C4)). In sensitivity an increase of up to 16% (again *HelixCorr/C7* compared to *TMHcon/C4*) was observed. The significance of the prediction increased from 7.4×10^{-21} to 5.1×10^{-51}. However, it must be noted, that the fraction of neighboring helix pairs is significantly lower in the case of *HelixCorr* compared to any prediction obtained by a neural network (42.8% with *HelixCorr/C11* compared to maximal 79.6% with *NN3/C10*). While neural networks tend to learn that neighboring transmembrane helices have a higher probability for interacting with each other, co-evolving residues are much more independent of this fact. Since in the predictions more importance is attached to higher specificity than sensitivity, resulting in a limited number of predicted interacting helix pairs. This leads to an enrichment of neighboring helices in the prediction of the neural networks. In contrast, the prediction from *HelixCorr* with a fraction of close to 40%

neighboring helices resembles nearly perfectly the naturally occurring fraction of 39.9% neighboring helices in the total set of interacting helices (285 out of 714).

Using the predictions of myself, Angelika performed the required analyses.

5.4 Conclusion

The experimental determination of membrane protein structures is still a difficult and time-consuming process. Computational methods for the prediction of structural features of membrane proteins are therefor required to close the gap between available sequence and structure data of membrane proteins. While a large number of algorithms are already available for the prediction of membrane protein topology or the prediction of lipid-exposed surfaces (for reviews see Elofsson and von Heijne (2007); Punta *et al.* (2007)) here the first method is presented which uses neural networks for the prediction of helix-helix contacts. Based on a data set of 62 membrane proteins with solved structure a prediction accuracy of close to 26% was obtained with the best predictor *TMHcon*, which therefor performs with equal accuracy to available contact predictors for soluble proteins. Furthermore, it is demonstrated that predicted contacts can be a valuable tool for the detection of interacting helices. Since recent membrane protein structures have shown that membrane proteins can adopt folds of much broader variety than originally expected, the identification of interacting helices can lead to the distinction of different helix architectures or to the assignment of a membrane protein sequence to a related fold. The application of *TMHcon* to three proteins with only recently solved structure such as the sodium-potassium pump, resulted in the prediction of a helix interaction pattern closely resembling the observed pattern. *TMHcon* is available for download and off-line usage at http://webclu.bio.wzw.tum.de/tmhcon.

The website was designed by Angelika Fuchs, myself implemented the final program version.

Chapter 6

Conclusion

The goal of this thesis was the development of prediction algorithms specially tailored for structural features of proteins. In the course of this thesis, one- and two-dimensional structural features of proteins had been considered. Prediction methods were developed for globular proteins and proteins embedded in cell membranes. This chapter summarizes the most important contributions, discusses limitations and, provides an outlook to potential future work.

6.1 Summary

The developments for this thesis center around machine learning, in particular there is a strong focus on artificial neural network models. In the Chapters 2 and 3 a novel type of recurrent neural network is introduced and applied on one-dimensional structure prediction. The network was developed by integrating various approaches from related works and is finally arisen as visualized in Figure 2.1. The novelty becomes apparent in the combination of ordinary recurrent neural networks with multi-task learning and their subsequent application on structure prediction. Although multi-task learning was already introduced ten years ago (Caruana, 1997) it has never been used to extend recurrent neural networks and it was never applied in the area of computational biology. The most common approaches for one-dimensional structure prediction employed in computational biology are setups where problems are tackled in a sequential way. The classic application is to use predicted secondary

structure in subsequent predictions. Opposite to that, the methodology introduced here allows prediction of multiple structural features in a synergetic fashion and even more: higher prediction performance is expected as already hypothesized from Caruana (1997).

In Chapter 4 one experiment on secondary structure prediction is described that was canceled due to insufficient performance. Compared to a variety of other experiments conducted during the course of this thesis, the codon profiles described there were the most promising yet. For a list of other canceled experiments see section 6.2.

In Chapter 5 a method is introduced that is again based on a neural network. This time a single neural network is used to predict contacts of residues in trans-membrane helices of membrane proteins. The introduced neural network approach outperforms an older development of Fuchs *et al.* (2007) that solely relied on correlated mutations. The neural network can outperform the original development, thanks to its ability to integrate additional features; here at most 595 features were used. Moreover in Chapter 5 not only an algorithm is introduced but also a scheme is designed that allows visualization of interacting trans-membrane helices in a graphical representation. A quick overview on membrane protein topologies is enabled by that representation as well as a subsequent application to organize membrane proteins into structural classes. The graphical representation offers possibilities to measure the differences between two trans-membrane proteins and via this distance metric the proteins can get classified in a hierarchical way similar to the SCOP classification (Murzin *et al.*, 1995).

6.2 List of canceled secondary structure prediction experiments

Many experiments on secondary structure prediction had to be aborted due to insufficient performance. The following list gives an overview of the experiments carried out which aimed for the integration of various additional data.

6.2.1 Meta models with predictions from *PREDATOR*

PREDATOR (Frishman and Argos, 1997) internally computes seven types of propensities, three of them constitute locally derived likelihoods for the three classes C, E and H, obtained from a k-nearest neighbor classifier (Aha *et al.*, 1991), three are estimated non-locally, based on patterns of potentially hydrogen bonded residues describing propensities for the formation of parallel, anti-parallel β-sheets and α-helices and the seventh value holds a propensity for turn formation. These were used for the following two-layered architecture with the sequence-to-propensities layer performed by *PREDATOR* and the propensities-to-structure constructed from a meta classifier like k-nearest neighbor or multi-layer perceptron – performance gain was not enough.

6.2.2 Consideration of taxonomic characteristics

To this end a bachelor thesis was arranged. The student Michael Lidschreiber implemented various ways to perform the prediction with taxonomically separated data. The rationale therefor is that the different species bare slightly different setups for secondary structure (Lidschreiber, 2005), and that the taxa of all target proteins are known. Three clades eukaryots, prokaryots and viruses were considered. All train data was separated into these clades, including the blast databases, and specific secondary structure predictors were developed – to-date algorithms could not be surpassed.

6.2.3 Inclusion of predicted *FunCat* classifications

All training proteins were automatically classified into *FunCat* (Ruepp *et al.*, 2004) categories. All these categories were then modeled in the feature space of a classifier – evaluation revealed no gain in performance.

6.2.4 Integration of predicted secondary structure content

The rationale for this purpose is that an *a priori* knowledge about the content of secondary structures in a protein could bias a classifier to better detect the secondary

structure state of residues in an local environment. Actually a significant gain in performance was noticed when observed secondary structure content was included in the classifier feature space – with predicted secondary structure content, the performance was neutralized.

6.2.5 Integration of predicted contact order

The contact order (Plaxco *et al.*, 1998) is defined as the average sequence separation between contacting residues in the native state. It strongly correlates with the folding rate of a protein. *Ab initio* 3D structure prediction programs like Rosetta have problems predicting structures of proteins with high contact order (Bonneau *et al.*, 2002a). The rationale for using it as a feature for secondary structure prediction comes again from the idea to bias the classifier to detect more 'complex' proteins and utilize that information to better discover local structural features – no change in prediction performance was obtained.

6.2.6 Integration of correlated mutations

Similar to Frishman and Argos (1996) the integration of long-range residue interaction was attempted. The three algorithms successfully utilized in chapter 5 were used to detect correlated mutations in proteins and therefrom obtain residue-residue contacts. Then new features were developed for all residues in a protein. The features encoded whether a particular residue is mutating in a correlated way with another residue in this protein and what sequence separation is observed to this other residue. These new features were merged with the original PSSM values and used for training and prediction – the classifiers could not extract meaningful information from this setup.

6.2.7 Model of hydrogen bonding patterns

Another way to integrate long range interactions was to integrate hydrogen bonding patterns. Each residue can form two hydrogen bonds: One bond can be established between the hydroxy-group which is a hydrogen bond donor and the other bond

can be established with the nitrogen as bond acceptor. Knowledge of the hydrogen bonding patterns of a protein reveals the secondary structure. DSSP (Kabsch and Sander, 1983) for example uses the hydrogen-bonding patterns to assign secondary structure, hence the hydrogen-bonding patterns are sufficient but not necessary for secondary structure assignment. For all residues the hydrogen bond patterns were predicted and these predicted bonds were then used as additional input features – when using observed hydrogen bonds, accuracies of 90% and beyond were obtained, but predicted ones did not help.

6.2.8 Integration of profile derivatives

This technique relates to image recognition tasks that include derivatives to detect edges or ridges in pictures (Lindeberg, 1996). This can be obtained by subtracting neighbored color intensities and high values then obviously indicate large changes. Such processing was adopted for PSSM profiles and introduced into the ordinary features. Actually a slight performance gain was observed. But again not enough to outperform other methods.

6.2.9 Integration of missed protein cleavage sites

Peptide mass fingerprinting is a technique to identify currently expressed proteins in whole cells. Trypsin is used to cleave the proteome and mass spectrometry (MS) is subsequently used to obtain the masses of the obtained cleaved protein segments. *A priori* knowledge about the sequences of the proteom and the trypsin cleavage sites allows protein identification. When performing cleavage experiments one observes potential cleavage sites that should be detected from trypsin, but are missed. It is hypothesized that these sites are inaccessible in natively folded proteins and hence may serve as determinants for one-dimensional structural features. Huge amounts of cleavage data was obtained from Jürgen Cox (Max Planck Institute of Biochemistry) and experiments were conducted to exploit them for structure prediction – no gain in performance was derived.

6.3 Discussion

This thesis applies machine learning techniques to predict structure-based features of proteins. In the following the most prominent issues are summed up and brought in context.

The structural features predicted in this work are important for a large variety of applications. For many tasks the knowledge about these features are quite sufficient and allow to come to conclusions. Two examples that point out this statement follow now: Personal communication with Roland Arnold who is working on a predictor which allows the discovery of type III and IV secreted proteins revealed that he and his colleagues are analyzing the secondary structure and solvent accessibility of protein segments responsible for secretion. By doing so they try to uncover structural patterns detected by the secretion system. The second example shows that predicted secondary structure was successfully used to describe protein complexes. Philip Wong discovered that proteins participating in protein complexes are constrained in secondary structure content (Wong *et al.*, submitted).

An important aspect discovered in this thesis is that the algorithms are crucial for a successful method development. The algorithms developed here for one-dimensional structure prediction managed to surpass straight forward approaches although merely being based on standard input data. The discussion of canceled experiments in Section 6.2 and the description in Chapter 4 reveal that additional descriptors were less successful to surpass current methods. Herein possibilities arise to continue development, for example: The utilization of the codon profile described in Chapter 4 together with the more sophisticated *MOLEBRNN* might even surpass *SOPRANO* introduced in Chapter 3.

By contrast the improvement that *TMHcon* described in Chapter 5 obtains compared to the original version of Fuchs *et al.* (2007) is possible because the used neural network offers to integrate a multitude of data. Here a combination of applied algorithm and utilization of meaningful data is responsible for the success.

Another important aspect associated with the input data is: Throughout the work no positive effect of global protein descriptors used for the prediction of lo-

cal features was detected. The features based on the taxonomy, *FunCat*, predicted secondary structure content and, the contact order (Section 6.2) are non-varying within a given protein and hence considered as global. No positive effect emerged when these features were included. For *TMHcon* two global features were utilized: The length of the proteins and the number of trans-membrane helices. No special tests were conducted with these features but it is strongly believed that their contribution to the performance is not measurable.

Although not directly approached in this work, it should be noted that method assembly is crucial when providing downloadable software. When developing a prediction system, cross-validation is the method of choice to get insights into performances. The cross-validation procedure in general is well described and comparatively straight forward, but whenever the full prediction system has emerged and the individual parts are assembled, a significant deliberation is required to ensure that the final product has equal performance as the development revealed.

A literature search uncovers that multi-task learning did not play a big role in machine learning after being introduced. Caruana (1997) and related work of Suddarth and Kergosien (1990); Suddarth and Holden (1991) used artificial data to analyze the influence of multiple prediction targets. It was not applied on real world data for a long while. Today, greater interest is put on multi-task learning. Four articles relating to multi-task learning have been submitted to the International Conference on Machine Learning 2008 (ICML2008), one of these presenting a neural network approach (Collobert and Weston, 2008) dealing with natural language processing (NLP) and the other dealing with HIV Therapy Screening (Bickel *et al.*, 2008) which is directing towards applications in computational biology. The work presented in Chapters 2 and 3 is the first one to settle multi-task learning in computational biology and structural bioinformatics. Similar to natural language processing, prediction of one-dimensional protein structures offers a great variety in prediction targets which is best modeled by multi-task algorithms. There is hope that *MOLEBRNN* will be applied in various domains of science and thus will bring forth research on sequence learning.

6.4 Final conclusions

The work carried out for this thesis is centered around machine learning algorithms, particularly round neural networks, applied to predict one- and two-dimensional protein structures. The task was to contribute to structural biology by establishing methods in the bioinformatics domain.

To reach these goals a variety of attempts were undertaken. Initial developments are summarized in section 6.2. All these developments remained far from the goal to significantly contribute to structural biology. The first meaningful approach is discussed in Chapter 4 but the achieved results were not enough to alert the public. During this initial progress the integration of additional data played the major role.

The goals of this thesis were reached when the focus was put on algorithm development. Soon after implementing the multi output layer Elman-type bidirectional recurrent neural network (*MOLEBRNN*), an integrated β-turn, β-turn type and secondary structure predictor was invented that was thought to appeal to 3^{rd} parties (Kirschner and Frishman, 2008)[1]. Continual development with *MOLEBRNN* enabled insights in the capacities resting on the concept of multi-task learning and allowed the extension towards *SOPRANO* (solvent accessibility predictor applying neural networks with multiple outputs), the best method to-date for prediction of solvent accessibility in proteins (Kirschner and Frishman, submitted)[2]. Besides not only solvent accessibility prediction with *SOPRANO* is to be mentioned but also its ability to predict secondary structure and β-turns, which is outstanding and comparable to current methods, if not superior.

Parallel to the development of predictors for one-dimensional protein structures, work was done together with Angelika Fuchs aiming at the prediction of two-dimensional protein structures. As Angelika already had done much preparatory work we soon came up with *TMHcon* (Fuchs *et al.*, 2008)[3], a method to predict helix-helix contacts of trans-membrane proteins. This method is the first one to utilize a neural network for contact prediction in membrane proteins. Additionally, a

[1] `http://webclu.bio.wzw.tum.de/predator-web`
[2] `http://webclu.bio.wzw.tum.de/soprano`
[3] `http://webclu.bio.wzw.tum.de/tmhcon`

novel representation for the interactions of trans-membrane helices was introduced.

The professional work on one-dimensional protein structure prediction offered the possibility to collaborate with two teams and contribute predicted secondary structure for downstream analysis (Smialowski *et al.*, 2006; Wong *et al.*, submitted).

Appendices

Publications

Parts of this thesis have appeared in the following publications:

P. Smialowski, T. Schmidt, J. Cox, A. Kirschner and D. Frishman. "Will my protein crystallize? A sequence-based predictor." *Proteins*, 62(2):343–355, **2006**.

A. Kirschner and D. Frishman. "Prediction of beta-turns and beta-turn types by a novel bidirectional Elman-type recurrent neural network with multiple output layers (MOLEBRNN)." *Gene*, 422(1-2):22–29, **2008**.

A. Fuchs, A. Kirschner and D. Frishman. "Prediction of helix-helix contacts and interacting helices in polytopic membrane proteins using neural networks." *Proteins*, in Press, **2008**.

P. Wong *et al.* "An evolutionary characterization of mammalian protein complex organization." **submitted**.

A. Kirschner and D. Frishman. "Accurate prediction of protein solvent accessibility by a neural network architecture trained to simultaneously recognize multiple structural features." **submitted**.

Bibliography

R. Adamczak, A. Porollo and J. Meller. "Accurate prediction of solvent accessibility using neural networks-based regression." *Proteins*, 56(4):753–767, **2004**.

R. Adamczak, A. Porollo and J. Meller. "Combining prediction of secondary structure and solvent accessibility in proteins." *Proteins*, 59(3):467–475, **2005**.

L. Adamian and J. Liang. "Prediction of transmembrane helix orientation in polytopic membrane proteins." *BMC Struct Biol*, 6:13, **2006**.

D. W. Aha, D. Kibler *et al.* "Instance-Based Learning Algorithms." *Machine Learning*, 6(1):37–66, **1991**.

S. Ahmad and M. M. Gromiha. "NETASA: neural network based prediction of solvent accessibility." *Bioinformatics*, 18(6):819–824, **2002**.

S. Ahmad, M. M. Gromiha and A. Sarai. "Real value prediction of solvent accessibility from amino acid sequence." *Proteins*, 50(4):629–635, **2003**a.

S. Ahmad, M. M. Gromiha and A. Sarai. "RVP-net: online prediction of real valued accessible surface area of proteins from single sequences." *Bioinformatics*, 19(14):1849–1851, **2003**b.

S. F. Altschul *et al.* "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucleic Acids Res*, 25(17):3389–3402, **1997**.

A. Andreeva *et al.* "Data growth and its impact on the SCOP database: new developments." *Nucleic Acids Res*, 36(Database issue):D419–D425, **2008**.

C. B. Anfinsen. "Principles that govern the folding of protein chains." *Science*, 181(96):223–230, **1973**.

Z. Aydin, Y. Altunbasak and M. Borodovsky. "Protein secondary structure prediction for a single-sequence using hidden semi-Markov models." *BMC Bioinformatics*, 7:178, **2006**.

A. Bach *et al.* "Type II' to type I β-turn swap changes specificity for integrins." *J. Am. Chem. Soc.*, 118:293–294, **1996**.

P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen and H. Nielsen. "Assessing the accuracy of prediction algorithms for classification: an overview." *Bioinformatics*, 16(5):412–424, **2000**.

P. Baldi, S. Brunak, P. Frasconi, G. Soda and G. Pollastri. "Exploiting the past and the future in protein secondary structure prediction." *Bioinformatics*, 15(11):937–946, **1999**.

D. Baú *et al.* "Distill: a suite of web servers for the prediction of one-, two- and three-dimensional structural features of proteins." *BMC Bioinformatics*, 7:402, **2006**.

H. M. Berman *et al.* "The Protein Data Bank." *Nucleic Acids Res*, 28(1):235–242, **2000**.

T. Beuming and H. Weinstein. "A knowledge-based scale for the analysis and prediction of buried and exposed faces of transmembrane domain proteins." *Bioinformatics*, 20(12):1822–1835, **2004**.

S. Bickel, J. Bogojeska, T. Lengauer and T. Scheffer. "Multi-task learning for hiv therapy screening." In A. McCallum and S. Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 56–63. Omnipress, **2008**.

R. Bonneau, I. Ruczinski, J. Tsai and D. Baker. "Contact order and ab initio protein structure prediction." *Protein Sci*, 11(8):1937–1944, **2002**a.

R. Bonneau *et al.* "De novo prediction of three-dimensional structures for major protein families." *J Mol Biol*, 322(1):65–78, **2002**b.

J. U. Bowie. "Helix-bundle membrane protein fold templates." *Protein Sci*, 8(12):2711–2719, **1999**.

P. Bradley *et al.* "Rosetta predictions in CASP5: successes, failures, and prospects for complete automation." *Proteins*, 53 Suppl 6:457–468, **2003**.

S. Brunak and J. Engelbrecht. "Protein structure and the sequential structure of mRNA: alpha-helix and beta-sheet signals at the nucleotide level." *Proteins*, 25(2):237–252, **1996**.

R. Caruana. "Multitask Learning." *Machine Learning*, 28(1):41–75, **1997**.

C.-C. Chen, J.-K. Hwang and J.-M. Yang. "(PS)2: protein structure prediction server." *Nucleic Acids Res*, 34(Web Server issue):W152–W157, **2006**.

J. Cheng and P. Baldi. "Improved residue contact prediction using support vector machines and a large feature set." *BMC Bioinformatics*, 8:113, **2007**.

J. Cheng, A. Z. Randall, M. J. Sweredoski and P. Baldi. "SCRATCH: a protein structure and structural feature prediction server." *Nucleic Acids Res*, 33(Web Server issue):W72–W76, **2005**.

D. Chivian *et al.* "Automated prediction of CASP-5 structures using the Robetta server." *Proteins*, 53 Suppl 6:524–533, **2003**.

A. A. Chmiel, J. M. Bujnicki and K. J. Skowronek. "A homology model of restriction endonuclease SfiI in complex with DNA." *BMC Struct Biol*, 5:2, **2005**.

P. Y. Chou and G. D. Fasman. "Prediction of protein conformation." *Biochemistry*, 13(2):222–245, **1974**.

P. Y. Chou and G. D. Fasman. "Prediction of beta-turns." *Biophys J*, 26(3):367–383, **1979**.

C. Cole, J. D. Barber and G. J. Barton. "The Jpred 3 secondary structure prediction server." *Nucleic Acids Res*, **2008**.

R. Collobert and J. Weston. "A unified architecture for natural language processing: deep neural networks with multitask learning." In A. McCallum and S. Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 160–167. Omnipress, **2008**.

J. A. Cuff and G. J. Barton. "Evaluation and improvement of multiple sequence methods for protein secondary structure prediction." *Proteins*, 34(4):508–519, **1999**.

J. A. Cuff and G. J. Barton. "Application of multiple sequence alignment profiles to improve protein secondary structure prediction." *Proteins*, 40(3):502–511, **2000**.

K. A. Dill. "Dominant forces in protein folding." *Biochemistry*, 29(31):7133–7155, **1990**.

R. Dutzler, E. B. Campbell, M. Cadene, B. T. Chait and R. MacKinnon. "X-ray structure of a ClC chloride channel at 3.0 A reveals the molecular basis of anion selectivity." *Nature*, 415(6869):287–294, **2002**.

R. J. Ellis. "Discovery of molecular chaperones." *Cell Stress Chaperones*, 1(3):155–160, **1996**.

J. L. Elman. "Finding Structure in Time." *Cognitive Science*, 14(2):179–211, **1990**.

A. Elofsson and G. von Heijne. "Membrane protein structure: prediction versus reality." *Annu Rev Biochem*, 76:125–140, **2007**.

V. A. Eyrich *et al.* "EVA: continuous automatic evaluation of protein structure prediction servers." *Bioinformatics*, 17(12):1242–1243, **2001**.

P. Fariselli, O. Olmea, A. Valencia and R. Casadio. "Prediction of contact maps with neural networks and correlated mutations." *Protein Eng*, 14(11):835–843, **2001**a.

P. Fariselli, O. Olmea, A. Valencia and R. Casadio. "Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations." *Proteins*, Suppl 5:157–162, **2001**b.

T. Fawcett. "ROC Graphs: Notes and Practical Considerations for Researchers." *Machine Learning*, 31, **2004**.

V. M. Faça *et al.* "Proteomic analysis of ovarian cancer cells reveals dynamic processes of protein secretion and shedding of extra-cellular domains." *PLoS ONE*, 3(6):e2425, **2008**.

M. Feder and J. M. Bujnicki. "Identification of a new family of putative PD-(D/E)XK nucleases with unusual phylogenomic distribution and a new type of the active site." *BMC Genomics*, 6(1):21, **2005**.

L. Feng *et al.* "Structure of a site-2 protease family intramembrane metalloprotease." *Science*, 318(5856):1608–1612, **2007**.

S. J. Fleishman, O. Yifrach and N. Ben-Tal. "An evolutionarily conserved network of amino acids mediates gating in voltage-dependent potassium channels." *J Mol Biol*, 340(2):307–318, **2004**.

A. A. Fodor and R. W. Aldrich. "Influence of conservation on calculations of amino acid covariance in multiple sequence alignments." *Proteins*, 56(2):211–221, **2004**.

D. Frishman and P. Argos. "Knowledge-based protein secondary structure assignment." *Proteins*, 23(4):566–579, **1995**.

D. Frishman and P. Argos. "Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence." *Protein Eng*, 9(2):133–142, **1996**.

D. Frishman and P. Argos. "Seventy-five percent accuracy in protein secondary structure prediction." *Proteins*, 27(3):329–335, **1997**.

D. Frishman and H. W. Mewes. "Protein structural classes in five complete genomes." *Nat Struct Biol*, 4(8):626–628, **1997**.

J. Frydman, E. Nimmesgern, K. Ohtsuka and F. U. Hartl. "Folding of nascent polypeptide chains in a high molecular mass assembly with molecular chaperones." *Nature*, 370(6485):111–117, **1994**.

A. Fuchs, A. Kirschner and D. Frishman. "Prediction of helix-helix contacts and interacting helices in polytopic membrane proteins using neural networks." *Proteins*, in Press, **2008**.

A. Fuchs *et al.* "Co-evolving residues in membrane proteins." *Bioinformatics*, 23(24):3312–3319, **2007**.

P. F. J. Fuchs and A. J. P. Alix. "High accuracy prediction of beta-turns and their types using propensities and multiple alignments." *Proteins*, 59(4):828–839, **2005**.

T. C. Gamblin. "Potential structure/function relationships of predicted secondary structural elements of tau." *Biochim Biophys Acta*, 1739(2-3):140–149, **2005**.

A. Garg, H. Kaur and G. P. S. Raghava. "Real value prediction of solvent accessibility in proteins using multiple sequence alignment and secondary structure." *Proteins*, 61(2):318–324, **2005**.

G. Gianese, F. Bossa and S. Pascarella. "Improvement in prediction of solvent accessibility by probability profiles." *Protein Eng*, 16(12):987–992, **2003**.

K. Ginalski *et al.* "ORFeus: Detection of distant homology using sequence profiles and predicted secondary structure." *Nucleic Acids Res*, 31(13):3804–3807, **2003**.

U. Göbel, C. Sander, R. Schneider and A. Valencia. "Correlated mutations and residue contacts in proteins." *Proteins*, 18(4):309–317, **1994**.

E. Granseth, G. von Heijne and A. Elofsson. "A study of the membrane-water interface region of membrane proteins." *J Mol Biol*, 346(1):377–385, **2005**.

O. Graña *et al.* "CASP6 assessment of contact prediction." *Proteins*, 61 Suppl 7:214–224, **2005**.

S. Henikoff and J. G. Henikoff. "Amino acid substitution matrices from protein blocks." *Proc Natl Acad Sci U S A*, 89(22):10,915–10,919, **1992**.

S. Henikoff and J. G. Henikoff. "Position-based sequence weights." *J Mol Biol*, 243(4):574–578, **1994**.

M. G. Hinds *et al.* "Synthesis, conformational properties, and antibody recognition of peptides containing beta-turn mimetics based on alpha-alkylproline derivatives." *J Med Chem*, 34(6):1777–1789, **1991**.

U. Hobohm and C. Sander. "Enlarged representative set of protein structures." *Protein Sci*, 3(3):522–524, **1994**.

U. Hobohm, M. Scharf, R. Schneider and C. Sander. "Selection of representative protein data sets." *Protein Sci*, 1(3):409–417, **1992**.

D. R. Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid*. Basic Books, 20 anv edition edition, **1999**.

L. Holm and C. Sander. "Mapping the protein universe." *Science*, 273(5275):595–603, **1996**.

E. G. Hutchinson and J. M. Thornton. "PROMOTIF - a program to identify and analyze structural motifs in proteins." *Protein Sci*, 5(2):212–220, **1996**.

J. M. G. Izarzugaza, O. Graña, M. L. Tress, A. Valencia and N. D. Clarke. "Assessment of intramolecular contact predictions for CASP7." *Proteins*, 69 Suppl 8:152–158, **2007**.

J. L. Johnson and E. A. Craig. "Protein folding in vivo: unraveling complex pathways." *Cell*, 90(2):201–204, **1997**.

D. T. Jones. "Protein secondary structure prediction based on position-specific scoring matrices." *J Mol Biol*, 292(2):195–202, **1999**.

D. T. Jones and M. B. Swindells. "Getting the most from PSI-BLAST." *Trends Biochem Sci*, 27(3):161–164, **2002**.

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

L. Käll, A. Krogh and E. L. Sonnhammer. "Advantages of combined transmembrane topology and signal peptide prediction–the Phobius web server." *Nucleic Acids Res*, 35(Web Server issue):W429–W432, **2007**.

H. Kaur and G. P. S. Raghava. "An evaluation of beta-turn prediction methods." *Bioinformatics*, 18(11):1508–1514, **2002**.

H. Kaur and G. P. S. Raghava. "BTEVAL: a server for evaluation of beta-turn prediction methods." *J Bioinform Comput Biol*, 1(3):495–504, **2003**a.

H. Kaur and G. P. S. Raghava. "Prediction of beta-turns in proteins from multiple alignment using neural network." *Protein Sci*, 12(3):627–634, **2003**b.

H. Kaur and G. P. S. Raghava. "A neural network method for prediction of beta-turn types in proteins using evolutionary information." *Bioinformatics*, 20(16):2751–2758, **2004**.

S. Kawashima *et al.* "AAindex: amino acid index database, progress report 2008." *Nucleic Acids Res*, 36(Database issue):D202–D205, **2008**.

H. Kim and H. Park. "Protein secondary structure prediction based on an improved support vector machines approach." *Protein Eng*, 16(8):553–560, **2003**.

H. Kim and H. Park. "Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor." *Proteins*, 54(3):557–562, **2004**.

S. Kim. "Protein beta-turn prediction using nearest-neighbor method." *Bioinformatics*, 20(1):40–44, **2004**.

A. Kirschner and D. Frishman. "Prediction of beta-turns and beta-turn types by a novel bidirectional Elman-type recurrent neural network with multiple output layers (MOLEBRNN)." *Gene*, 422(1-2):22–29, **2008**.

A. Kirschner and D. Frishman. "Accurate prediction of protein solvent accessibility by a neural network architecture trained to simultaneously recognize multiple structural features." **submitted**.

T. Klabunde and G. Hessler. "Drug design strategies for targeting G-protein-coupled receptors." *Chembiochem*, 3(10):928–944, **2002**.

A. Kloczkowski, K.-L. Ting, R. L. Jernigan and J. Garnier. "Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence." *Proteins*, 49(2):154–166, **2002**.

V. A. Kolb, E. V. Makeyev and A. S. Spirin. "Folding of firefly luciferase during translation in a cell-free system." *EMBO J*, 13(15):3631–3637, **1994**.

A. A. Komar, T. Lesnik and C. Reiss. "Synonymous codon substitutions affect ribosome traffic and protein folding during in vitro translation." *FEBS Lett*, 462(3):387–391, **1999**.

A. Krogh, B. Larsson, G. von Heijne and E. L. Sonnhammer. "Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes." *J Mol Biol*, 305(3):567–580, **2001**.

T. Kulikova *et al.* "EMBL Nucleotide Sequence Database in 2006." *Nucleic Acids Res*, 35(Database issue):D16–D20, **2007**.

P. J. Kundrotas and E. G. Alexov. "Predicting residue contacts using pragmatic correlated mutations method: reducing the false positives." *BMC Bioinformatics*, 7:503, **2006**.

J. Kyte and R. F. Doolittle. "A simple method for displaying the hydropathic character of a protein." *J Mol Biol*, 157(1):105–132, **1982**.

E. S. Lander *et al.* "Initial sequencing and analysis of the human genome." *Nature*, 409(6822):860–921, **2001**.

G. Lasso, J. F. Antoniw and J. G. L. Mullins. "A combinatorial pattern discovery approach for the prediction of membrane dipping (re-entrant) loops." *Bioinformatics*, 22(14):e290–e297, **2006**.

S. Lawrence, I. Burns, A. Back, A. C. Tsoi and C. L. Giles. *Neural Networks: Tricks of the Trade*, volume 1524 of *Lecture Notes in Computer Science*, chapter Neural Network Classification and Prior Class Probabilities, page 545. Springer Berlin / Heidelberg, **1998**.

T. J. Lewis *et al.* "Possible Recoil Effects in (16O, 15N) Transitions About 10 MeV Above the Coulomb Barrier." *Phys. Rev. C*, 8(2):678–683, **1973**. Original title = Possible Recoil Effects in (^{16}O, ^{15}N) Transitions About 10 MeV Above the Coulomb Barrier.

S. Z. Li *et al.* "Type I beta-turn conformation is important for biological activity of the melanocyte-stimulating hormone analogues." *Eur J Biochem*, 265(1):430–440, **1999**.

W. Li and A. Godzik. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences." *Bioinformatics*, 22(13):1658–1659, **2006**.

M. Lidschreiber. "Taxonomy based Protein Secondary Structure Prediction." Bachelor Thesis, **2005**.

T. Lindeberg. "Edge detection and ridge detection with automatic scale selection." *International Journal of Computer Vision*, 30(2):77–116, **1996**.

M. A. Lomize, A. L. Lomize, I. D. Pogozheva and H. I. Mosberg. "OPM: orientations of proteins in membranes database." *Bioinformatics*, 22(5):623–625, **2006**.

R. M. MacCallum. "Striped sheets and protein contact prediction." *Bioinformatics*, 20 Suppl 1:i224–i231, **2004**.

J. Martin, J.-F. Gibrat and F. Rodolphe. "Analysis of an optimal hidden Markov model for secondary structure prediction." *BMC Struct Biol*, 6:25, **2006**.

A. J. Martin-Galiano and D. Frishman. "Defining the fold space of membrane proteins: the CAMPS database." *Proteins*, 64(4):906–922, **2006**.

M. J. McGregor, T. P. Flores and M. J. Sternberg. "Prediction of beta-turns in proteins using neural networks." *Protein Eng*, 2(7):521–526, **1989**.

L. J. McGuffin, K. Bryson and D. T. Jones. "The PSIPRED protein structure prediction server." *Bioinformatics*, 16(4):404–405, **2000**.

A. D. McLachlan. "Tests for comparing related amino-acid sequences. Cytochrome c and cytochrome c 551 ." *J Mol Biol*, 61(2):409–424, **1971**.

H. W. Mewes *et al.* "Overview of the yeast genome." *Nature*, 387(6632 Suppl):7–65, **1997**.

T. Miyata, S. Miyazawa and T. Yasunaga. "Two types of amino acid substitutions in protein evolution." *J Mol Evol*, 12(3):219–236, **1979**.

J. P. Morth *et al.* "Crystal structure of the sodium-potassium pump." *Nature*, 450(7172):1043–1049, **2007**.

J. Moult, K. Fidelis, A. Zemla and T. Hubbard. "Critical assessment of methods of protein structure prediction (CASP)-round V." *Proteins*, 53 Suppl 6:334–339, **2003**.

J. Moult, T. Hubbard, S. H. Bryant, K. Fidelis and J. T. Pedersen. "Critical assessment of methods of protein structure prediction (CASP): round II." *Proteins*, Suppl 1:2–6, **1997**.

J. Moult, T. Hubbard, K. Fidelis and J. T. Pedersen. "Critical assessment of methods of protein structure prediction (CASP): round III." *Proteins*, Suppl 3:2–6, **1999**.

D. W. Mount. *Bioinformatics: Sequence and Genome Analysis, Second Edition.* Cold Spring Harbor Laboratory Press, New York, second edition edition, **2004**.

A. G. Murzin, S. E. Brenner, T. Hubbard and C. Chothia. "SCOP: a structural classification of proteins database for the investigation of sequences and structures." *J Mol Biol*, 247(4):536–540, **1995**.

H. Naderi-Manesh, M. Sadeghi, S. Arab and A. A. M. Movahedi. "Prediction of protein surface accessibility with information theory." *Proteins*, 42(4):452–459, **2001**.

M. N. Nguyen and J. C. Rajapakse. "Prediction of protein relative solvent accessibility with a two-stage SVM approach." *Proteins*, 59(1):30–37, **2005**.

M. N. Nguyen and J. C. Rajapakse. "Two-stage support vector regression approach for predicting accessible surface areas of amino acids." *Proteins*, 63(3):542–550, **2006**.

A. Oberai, Y. Ihm, S. Kim and J. U. Bowie. "A limited universe of membrane protein families and folds." *Protein Sci*, 15(7):1723–1734, **2006**.

O. Olmea, B. Rost and A. Valencia. "Effective use of sequence correlation and conservation in fold recognition." *J Mol Biol*, 293(5):1221–1239, **1999**.

O. Olmea and A. Valencia. "Improving contact predictions by the combination of correlated mutations and other sources of sequence information." *Fold Des*, 2(3):S25–S32, **1997**.

M. Oresic and D. Shalloway. "Specific correlations between relative synonymous codon usage and protein secondary structure." *J Mol Biol*, 281(1):31–48, **1998**.

A. R. Ortiz, A. Kolinski, P. Rotkiewicz, B. Ilkowski and J. Skolnick. "Ab initio folding of proteins using restraints derived from evolutionary information." *Proteins*, Suppl 3:177–185, **1999**.

K. Palczewski *et al.* "Crystal structure of rhodopsin: A G protein-coupled receptor." *Science*, 289(5480):739–745, **2000**.

S. Y. Pasta, B. Raman, T. Ramakrishna and C. M. Rao. "Role of the conserved SRLFDQFFG region of alpha-crystallin, a small heat shock protein. Effect on oligomeric size, subunit exchange, and chaperone-like activity." *J Biol Chem*, 278(51):51,159–51,166, **2003**.

B. P. Pedersen, M. J. Buch-Pedersen, J. P. Morth, M. G. Palmgren and P. Nissen. "Crystal structure of the plasma membrane proton pump." *Nature*, 450(7172):1111–1114, **2007**.

D. Petrey and B. Honig. "Protein structure prediction: inroads to biology." *Mol Cell*, 20(6):811–819, **2005**.

Y. Pilpel, N. Ben-Tal and D. Lancet. "kPROT: a knowledge-based scale for the propensity of residue orientation in transmembrane segments. Application to membrane protein structure prediction." *J Mol Biol*, 294(4):921–935, **1999**.

K. W. Plaxco, K. T. Simons and D. Baker. "Contact order, transition state placement and the refolding rates of single domain proteins." *J Mol Biol*, 277(4):985–994, **1998**.

G. Pollastri and P. Baldi. "Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners." *Bioinformatics*, 18 Suppl 1:S62–S70, **2002**.

G. Pollastri, P. Baldi, P. Fariselli and R. Casadio. "Prediction of coordination number and relative solvent accessibility in proteins." *Proteins*, 47(2):142–153, **2002**a.

G. Pollastri, A. J. M. Martin, C. Mooney and A. Vullo. "Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information." *BMC Bioinformatics*, 8:201, **2007**.

G. Pollastri and A. McLysaght. "Porter: a new, accurate server for protein secondary structure prediction." *Bioinformatics*, 21(8):1719–1720, **2005**.

G. Pollastri, D. Przybylski, B. Rost and P. Baldi. "Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles." *Proteins*, 47(2):228–235, **2002**b.

M. Punta and B. Rost. "PROFcon: novel prediction of long-range contacts." *Bioinformatics*, 21(13):2960–2968, **2005**a.

M. Punta and B. Rost. "Protein folding rates estimated from contact predictions." *J Mol Biol*, 348(3):507–512, **2005**b.

M. Punta *et al.* "Membrane protein prediction methods." *Methods*, 41(4):460–474, **2007**.

N. Qian and T. J. Sejnowski. "Predicting the secondary structure of globular proteins using neural network models." *J Mol Biol*, 202(4):865–884, **1988**.

S. Qin, Y. He and X.-M. Pan. "Predicting protein secondary structure and solvent accessibility with an improved multiple linear regression method." *Proteins*, 61(3):473–480, **2005**.

P. Rice, I. Longden and A. Bleasby. "EMBOSS: the European Molecular Biology Open Software Suite." *Trends Genet*, 16(6):276–277, **2000**.

J. S. Richardson. "The anatomy and taxonomy of protein structure." *Adv Protein Chem*, 34:167–339, **1981**.

M. Riedmiller and H. Braun. "RPROP Algorithm." *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591, **1993**.

R. P. Riek, I. Rigoutsos, J. Novotny and R. M. Graham. "Non-alpha-helical elements modulate polytopic membrane protein architecture." *J Mol Biol*, 306(2):349–362, **2001**.

D. J. Rigden *et al.* "PrfA protein of Bacillus species: prediction and demonstration of endonuclease activity on DNA." *Protein Sci*, 11(10):2370–2381, **2002**.

J. M. Rini *et al.* "Crystal structure of a human immunodeficiency virus type 1 neutralizing antibody, 50.1, in complex with its V3 loop peptide antigen." *Proc Natl Acad Sci U S A*, 90(13):6325–6329, **1993**.

G. D. Rose, L. M. Gierasch and J. A. Smith. "Turns in peptides and proteins." *Adv Protein Chem*, 37:1–109, **1985**.

B. Rost and C. Sander. "Prediction of protein secondary structure at better than 70% accuracy." *J Mol Biol*, 232(2):584–599, **1993**.

B. Rost and C. Sander. "Combining evolutionary information and neural networks to predict protein secondary structure." *Proteins*, 19(1):55–72, **1994**a.

B. Rost and C. Sander. "Conservation and prediction of solvent accessibility in protein families." *Proteins*, 20(3):216–226, **1994**b.

B. Rost, G. Yachdav and J. Liu. "The PredictProtein server." *Nucleic Acids Res*, 32(Web Server issue):W321–W326, **2004**.

A. Ruepp *et al.* "The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes." *Nucleic Acids Res*, 32(18):5539–5545, **2004**.

A. A. Salamov and V. V. Solovyev. "Protein secondary structure prediction using local alignments." *J Mol Biol*, 268(1):31–36, **1997**.

R. M. Schwartz and M. O. Dayhoff. "Origins of prokaryotes, eukaryotes, mitochondria, and chloroplasts." *Science*, 199(4327):395–403, **1978**.

C. Sequencing and A. Consortium. "Initial sequence of the chimpanzee genome and comparison with the human genome." *Nature*, 437(7055):69–87, **2005**.

C. E. Shannon. "The mathematical theory of communication. 1963." *MD Comput*, 14(4):306–317, **1997**.

Y. Shao and C. Bystroff. "Predicting interresidue contacts using templates and pathways." *Proteins*, 53 Suppl 6:497–502, **2003**.

A. J. Shepherd, D. Gorse and J. M. Thornton. "Prediction of the location and type of beta-turns in proteins using neural networks." *Protein Sci*, 8(5):1045–1055, **1999**.

I. N. Shindyalov, N. A. Kolchanov and C. Sander. "Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations?" *Protein Eng*, 7(3):349–358, **1994**.

V. A. Simossis and J. Heringa. "PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information." *Nucleic Acids Res*, 33(Web Server issue):W289–W294, **2005**.

P. Smialowski, T. Schmidt, J. Cox, A. Kirschner and D. Frishman. "Will my protein crystallize? A sequence-based predictor." *Proteins*, 62(2):343–355, **2006**.

S. Suddarth and A. Holden. "Symbolic-neural systems and the use of hints for developing complex systems." *International Journal of Man-Machine Studies*, 35(3):291–311, **1991**.

S. C. Suddarth and Y. L. Kergosien. *Neural Networks*, eurasip workshop 1990 sesimbra, portugal, february 15-17, 1990 proceedings Rule-injection hints as a means of improving network performance and learning time, pages 120–129. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, **1990**.

K. Swingler. *Applying Neural Networks: A Practical Guide*. Morgan Kaufmann, **1996**.

T. A. Thanaraj and P. Argos. "Ribosome-mediated translational pause and protein domain organization." *Protein Sci*, 5(8):1594–1612, **1996**.

G. E. Tusnády, Z. Dosztányi and I. Simon. "PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank." *Nucleic Acids Res*, 33(Database issue):D275–D278, **2005**.

G. E. Tusnády, L. Kalmár and I. Simon. "TOPDB: topology data bank of transmembrane proteins." *Nucleic Acids Res*, 36(Database issue):D234–D239, **2008**.

G. E. Tusnády and I. Simon. "The HMMTOP transmembrane topology prediction server." *Bioinformatics*, 17(9):849–850, **2001**.

M. Vassura *et al.* "FT-COMAR: fault tolerant three-dimensional structure reconstruction from protein contact maps." *Bioinformatics*, 24(10):1313–1315, **2008**.

C. M. Venkatachalam. "Stereochemical criteria for polypeptides and proteins. V. Conformation of a system of three linked peptide units." *Biopolymers*, 6(10):1425–1436, **1968**.

H. Viklund, E. Granseth and A. Elofsson. "Structural classification and prediction of reentrant regions in alpha-helical transmembrane proteins: application to complete genomes." *J Mol Biol*, 361(3):591–603, **2006**.

E. Wallin and G. von Heijne. "Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms." *Protein Sci*, 7(4):1029–1038, **1998**.

J.-Y. Wang, H.-M. Lee and S. Ahmad. "Prediction and evolutionary information analysis of protein solvent accessibility using multiple linear regression." *Proteins*, 61(3):481–491, **2005**.

J.-Y. Wang, H.-M. Lee and S. Ahmad. "SVM-Cabins: prediction of solvent accessibility using accumulation cutoff set and support vector machine." *Proteins*, 68(1):82–91, **2007**.

J. J. Ward, L. J. McGuffin, B. F. Buxton and D. T. Jones. "Secondary structure prediction with support vector machines." *Bioinformatics*, 19(13):1650–1655, **2003**.

D. L. Wheeler *et al.* "Database resources of the National Center for Biotechnology Information." *Nucleic Acids Res*, 35(Database issue):D5–12, **2007**.

D. L. Wheeler *et al.* "Database resources of the National Center for Biotechnology Information." *Nucleic Acids Res*, 36(Database issue):D13–D21, **2008**.

R. J. Williams and D. Zipser. "Gradient-based learning algorithms for recurrent networks and their computational complexity." In Y. Chauvin and D. E. Rumelhart, editors, *Back-propagation: Theory, Architectures and Applications*, pages 433–486. Lawrence Erlbaum Publishers, Hillsdale, N.J., **1995**.

D. Wolpert. "The supervised learning no-free-lunch theorems." *Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, **2001**.

P. Wong *et al.* "An evolutionary characterization of mammalian protein complex organization." **submitted**.

M. J. Wood and J. D. Hirst. "Protein secondary structure prediction with dihedral angles." *Proteins*, 59(3):476–481, **2005**.

J. Wootton and S. Federhen. "Statistics of local complexity in amino acid sequences and sequence databases." *Computers & chemistry*, 17(2):149–163, **1993**.

J. C. Wootton and S. Federhen. "Analysis of compositionally biased regions in sequence databases." *Methods Enzymol*, 266:554–571, **1996**.

T. M. Yi and E. S. Lander. "Protein secondary structure prediction using nearest-neighbor methods." *J Mol Biol*, 232(4):1117–1129, **1993**.

S. Yohannan, S. Faham, D. Yang, J. P. Whitelegge and J. U. Bowie. "The evolution of transmembrane helix kinks and the structural diversity of G protein-coupled receptors." *Proc Natl Acad Sci U S A*, 101(4):959–963, **2004**.

A. Zemla, C. Venclovas, K. Fidelis and B. Rost. "A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment." *Proteins*, 34(2):220–223, **1999**.

C. Zhang and K. Chou. "Prediction of beta-turns in proteins by 1-4 & 2-3 Correlation Model." *Biopolymers*, 41:673–702, **1997**.

Q. Zhang, S. Yoon and W. J. Welsh. "Improved method for predicting beta-turn using support vector machine." *Bioinformatics*, 21(10):2370–2374, **2005**.

S. S. Zimmerman and H. A. Scheraga. "Local interactions in bends of proteins." *Proc Natl Acad Sci U S A*, 74(10):4126–4129, **1977**.

O. Zimmermann, L. Wang and U. H. E. Hansmann. "BETTY: prediction of beta-strand type from sequence." *In Silico Biol*, 7(4-5):535–542, **2007**.

List of Figures

List of Tables

Listings