UNIVERSITÀ DEGLI STUDI DI FIRENZE
Facoltà di Ingegneria - Dipartimento di Sistemi e Informatica

Dottorato di Ricerca in
Ingegneria Informatica e dell'Automazione
XVIII Ciclo

Prediction of Structure and Function of Proteins and Ligands by means of Neural and Kernel Methods for Structured Data

Alessio Ceroni

Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Control Engineering

Ph.D. Coordinator
Prof. Edoardo Mosca

Advisors
Prof. Paolo Frasconi

Prof. Giovanni Soda

Anno Accademico 2004–2005

Abstract

In the post-genomics era the new challenge posed by proteomics is the translation of the millions of already sequenced genes into high-resolution protein three-dimensional structures. The study of these protein structures would then eventually lead to a better understanding of the molecular mechanisms that define life. Machine learning can play an important role in this process. Using the evidence coming from known native structures, the structural and functional features of the other proteins can be rapidly predicted without the need of costly equipment. For this purpose, the use of neural and kernel methods for structured data is explored here on several predictive tasks in computational biology and biochemistry. A two-stages SVM-BRNN architecture for sequence translation tasks is applied to the problem of secondary structure (SS) prediction. The proposed approach, coupled with a Viterbi decoder that enforce constraints in the output sequences, is tested on a significant dataset and demonstrates state-of-the-art performances. The same architecture is then applied to the identification of protein metal binding sites (MBS), an important problem that has so far received scarce attention by the machine learning community. The proposed approach greatly improves the coverage of predicted MBS compared to pattern search methods, and additionally provide accurate prediction of disulfide bonded cysteines. An improved bidirectional recursive neural network architecture is introduced to solve sequence learning tasks in the presence of explicit knowledge about interacting positions. This architecture, tested on the SS prediction task using predicted contacts between residues, provides interesting directions of future investigation, while additional work is needed to advance the current state-of-the-art. An efficient structure reconstruction procedure is presented that can be used to reconstruct native structures when incomplete (coarse) information is available. While the proposed approach is able

to build protein models matching the available characteristics of the native conformation, the algorithm is not yet sufficiently reliable to correct the unavoidable errors of the coarse-grained contacts predictor. Several probability product kernels are used to learn an energy function for scoring protein models generated by structure prediction algorithms. Differently from the commonly used statistical potentials, that are built from native structures only, this discriminant approach can effectively extract information from decoys to output improved rankings of predicted models. Finally, a new decomposition kernel for three-dimensional structures is described that can exploit the information contained in the spatial arrangement of molecule atoms. This kernel is used to predict the biological activity of chemical compounds, reaching higher performances than previously published methods on the same data, thus proving to be a powerful tool for QSAR analysis and drug discovery.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgements

Creating a predictor for computational biology problem is a long and daunting task: it requires a lot of work for data preparation and the creation of complex architectures for machine learning. It is thus almost impossible to perform successful experiments without the collaboration of other researches. For this reason, I must thank a good bunch of colleagues (and friends) for their help. I put them in the order their contribution is found in this work.

First, I must thank dr. Andrea Passerini (DSI, Firenze) for having performed most of the experiments on metal binding site predictions, thus testing the two stages architecture on this task. A big thank goes to prof. Gianluca Pollastri (UCD, Dublin), for having run its Distill software to predict fine-grained contact maps, for providing me its coarse-grained contact map predictions, and for being my host in Dublin for a few enjoyable days. Many thanks to dr. Lawrence Kelley (Imperial, London) for providing me the protein decoys generated with its Phyre predictor. I must also thank dr. Fabrizio Costa and Sauro Menchetti (DSI, Firenze) for their implementation of the Weighted Decomposition Kernel. Finally, I must sincerely thank my supervisor prof. Paolo Frasconi for being the "gray eminence" behind all this work.

Most of the member of my laboratory at DSI figure in this list. I should also thank them for the hours of useful discussions and four hands programming that made this work possible.

Chapter 1

Introduction

During the 90s several genome projects have been initiated with the aims to map the DNA content of various species. The introduction of ad-hoc instrumentation and bioinformatics tools finally lead to the discovery of millions of genes with their nucleotide sequences, with new genomes mapped every year. Nature has evolved a biological process to transform the genetic information into large molecules with complex three-dimensional structures known as *proteins*. Proteins provide most of the machinery needed for the function and the structure of living beings. Proteins carry on their tasks by interacting each other and with the other molecules that compose and enters the organism. All these interactions are mediated by protein three-dimensional structure, a stable characteristic that arise from the folding of its chain of *amino-acids*. It is then becoming increasingly important to understand how the combination of amino-acids into a sequence generates those complex structures and how those structures interact with their environment to carry out the fundamental tasks they are assigned to by evolution.

Despite the efforts taken for the study of protein structure, there is still no fast and accurate method available for the estimation of a protein final conformation from its amino-acid sequence. The determination of a protein structure is still a complex process that involves several months of works by expert biochemists to interpret the data resulting from various experiments. In this context, there is a strong demand for predictive methods that can exploit available knowledge from already determined structures to estimate the structural characteristics of a protein from it sequence.

If characterization of a protein structure is a necessity, even more need

is felt for the study of protein functions. All the processes taking place in a organism involves the interaction of one protein with another molecule, that can be anything from a metal ion to another protein. Realizing the mechanisms of action of the proteins can lead to understanding the processes that define life. From this knowledge, one can hope that cures for the most severe diseases affecting humans can be found. For this reason, methods that predict the functional characteristics of proteins, such as the presence of suitable parts for binding certain molecules, or the ability of a small chemical to target some protein and affect its functions, hence the capability of the compound to act as a medication for some disease, are particularly sought.

All the types of molecules, from small chemicals to large proteins, are characterized by their constituent atoms, the chemical interactions between these atoms, and their spatial disposition. Some or all these levels of a molecule structure can be available to generate a decision on the properties of the molecule. Suitable methods are then needed to extract all the relevant information from these structured data in order to solve the predictive tasks needed in computational biology. The definition of supervised methods for learning structural and functional properties of biologically relevant molecules from their structure is the main focus of this work. To this purpose, the use of neural and kernel methods for structured data will be presented on several predictive tasks in computational biology.

The work is organized as follows. In Part I, the biological foundations of protein structure and function are overviewed. In Chap. 2 the formation of polypeptide chains, the folding process and the resulting hierarchy of structural levels are explained. In this context, the currently available methodologies for structure determination and prediction are presented. In Chap. 3 the mechanisms of interaction of proteins with metal ions and small chemicals are discussed, also showing some of the available methodology for the identification of binding sites and the prediction of biological activity of small chemicals.

Part II discusses the role of supervised learning methods for the construction of predictive tools for computational biology tasks. In Chap. 4 the problem of learning from molecule structures is proposed in the framework of learning from relational data. A classification of the different tasks that are faced in this work is then explained, making a distinction based on the structure of the inputs in the various problems. In Chaps. 5 and

6 the neural and kernel methods for structured data that are used in this work are explained in detail. These chapters also show two contributions of this work to the field, a recursive neural network architecture for learning from sequences enriched with long-term interactions, and a kernel function for learning from three-dimensional data structures.

In Parts III, IV and V various predictive tasks are considered. For each problem, a specific architecture is proposed for learning the input/output mapping and the results of experiments are reported. The problems are divided according to the classification proposed in Chap. 4. In Part. III, input data is in the form of sequences, and the problem of secondary structure prediction[1] (Chap. 7) and metal binding site prediction[2] (Chap. 8) are considered. Both problems are tackled using a two-stages architecture that exploit the advantages of both support vector machines and recursive neural networks. In Part. IV, input data is in the form of contact matrices. The problem of secondary structure prediction is further discussed in Chap. 9 where input data is enriched with predicted contact maps to test an alternative way of improving predictive accuracy[3]. In Chap. 10, coarse contact information is used to reconstruct the three-dimensional structure for a specific class of proteins[4]. In Part. V inputs are in the form of set of coordinates (three-dimensional structures). Chap. 11 explore the use of kernels for ranking decoys generated by structure prediction methods[5], while in Chap. 12 the kernel for three-dimensional structures is applied to the prediction of molecular activity of small chemicals and tested on two large datasets[6]. Finally, in Chap. 13 some conclusions are drawn and future directions of work are considered.

[1] This work has been published in Ceroni et al. (2003a)

[2] This chapter is based on the work made for an article that is under submission.

[3] This work has been published in Ceroni et al. (2005b)

[4] This work has been published in Ceroni and Frasconi (2004)

[5] This work has been published as a tecnical report (Ceroni et al., 2005a) and is currently under submission to an international journal.

[6] This works constitute the basis for an article that is currently in preparation.

Part I

Proteins

Chapter 2

Protein Structure

Proteins carry out most of the basic functions of life at the molecular level. They are designed to bind every conceivable molecule from simple ions to large complex molecules. Proteins catalyze an extraordinary range of chemical reactions, provide structural rigidity to the cell, control flow of material through membranes, regulate the concentrations of metabolites, act as sensors and switches, cause motion, and control gene function.

Amino-acids are the building blocks of proteins. Nature has evolved a single chemical linkage, the peptide bond, to connect amino-acids into a linear, unbranched chain (Sec. 2.1). This linear chain fold (Sec. 2.2) in a complex three-dimensional structure (Sec. 2.3) under the forces of non-covalent interactions between regions of the sequence of amino-acids in a unique shape that is responsible of the protein behavior .

One of the major areas of biological research today is the study of how proteins, constructed from only 20 different amino-acids, carry out the incredible array of diverse tasks that they do. A key concept in understanding how proteins work is that function is derived from three-dimensional structure, and three-dimensional structure is specified by amino-acid sequence. Even if proteins three-dimensional structures can be experimentally determined (Sec. 2.4), structure determination methods are slow and difficult to automate. Therefore, it is becoming increasingly important to predict protein tertiary structure from its amino-acid sequence using insights obtained from already known structures (Sec. 2.5).

2.1 Protein Synthesis

Proteins are synthesized inside cells. The instructions used by cells to build proteins are written in the DNA. During the process of *transcription*, the genetic information is copied from DNA into messenger ribonucleic acid (mRNA). The mRNA is a single chain of nucleotides that codifies a protein in the form of a comma-less code. Each triplet of nucleotides is called a *codon*: of the 64 possible codons in the genetic code, 61 specify one of the 20 possible amino-acids and three are stop codons. The AUG codon for methionine (an amino-acid) is the most common start codon. Most amino-acids are encoded by more than one codon, which allow robustness to mutations. The uninterrupted sequence of codons in mRNA from a start codon to a stop codon is called a *reading frame*. During the process of *translation* the information contained in a open reading frame is interpreted by the transfer RNA (tRNA) with the aid of ribosomal RNA (rRNA) and its associated proteins that form the *ribosome*. The ribosomes assemble the amino-acids in the order specified by the mRNA to form the polymeric linear chain of a protein.

The amino-acids can be considered as the alphabet in which proteins are written. Each amino-acid is formed by a single carbon atom (C_α) bonded to four different chemical groups: an amino (NH_2) group, a carboxyl (COOH) group, a hydrogen (H) atom, and one variable group, called *side chain* or R group. All 20 amino-acids have this same general structure, but their side-chain groups vary in size, shape, charge, hydrophobicity, and reactivity. The amino-acids in a protein chain are linked together by peptide bonds. The peptide bond is formed by a reaction between the amino group of one amino-acid and the carboxyl group of another. The repeated amide N, C_α, and carbonyl C atoms of each amino acid residue form the *backbone* (or *main chain*) of a protein molecule from which the various side-chain groups project. This leaves at opposite ends of the chain a free amino group (the N-terminus) and a free carboxyl group (the C-terminus).

Many terms are used to denote the chains formed by polymerization of amino-acids. A short chain of 20-30 amino-acids linked by peptide bonds and having a defined sequence is a *peptide*. Longer peptides (up to 4000 residues) are referred to as *polypeptides*. Finally, the term protein identifies a polypeptide (or a complex of polypeptides) that has a stable three-dimensional structure.

2.2 Protein Folding

The newly created chain elongating from the ribosome assumes its final conformation during the process of *folding*. In general, all the molecules of any protein specie adopt a single conformation, called the *native* state, which is the most stably folded form of the molecule. A cellular system prevents misfolded proteins from forming. The polarity of amino-acid side chains is one of the forces responsible for shaping the final three-dimensional structure of proteins. During folding in the watery environment of a cell the amino acids with polar side groups tend to remain on the surface of proteins; by interacting with water, they make proteins soluble in aqueous solutions. In contrast, amino-acids with non-polar side groups avoid water and aggregate to form the water insoluble core of proteins.

The realization that the amino-acid sequence of a protein determines its folding came from in vitro studies on protein unfolding and re-folding (Anfinsen, 1973). Thermal energy from heat, extremes level of pH, and various chemicals can disrupt the weak non-covalent bonds that stabilize the native conformation of a protein. Many proteins that are completely unfolded by chemical forces can renature (re-fold) into their native state when the denaturing reagents are removed. During renaturation, all the bonds that stabilize the native conformation are re-formed. Because this renaturation requires no cofactors or other proteins, protein folding is deemed to be a self-assembly process. The observation of such reversible denaturation and renaturation provided a clue that the information for folding a protein lies in its sequence.

Folding of proteins in vitro is an inefficient process, with only a minority of unfolded molecules undergoing complete folding within a few minutes. Clearly, in vivo most protein molecules must rapidly fold into their correct shape. More than 95% of the proteins present within cells have been shown to be in their native conformation, despite high protein concentrations which usually cause proteins to precipitate in vitro. The explanation for the cell's remarkable efficiency in promoting protein folding probably lies in chaperones, a family of proteins found in all organisms from bacteria to humans. Chaperones are located in every cellular compartment, bind a wide range of proteins, and may be part of a general protein-folding mechanism.

2.3 Hierarchical Structure of Proteins

After folding a protein assume a stable conformation that is commonly described in terms of four hierarchical levels of organization.

2.3.1 Primary Structure

Each piece of the linear chain that forms a protein backbone consists of one of the 20 amino-acid existing in nature. Therefore, a single protein chain can be represented as a string of letters from a 20 elements alphabet. This sequence is called the *primary structure* of the protein and each position is called a *residue*. If the similarity between two protein sequences from different organisms is significant, then the proteins are *homologs* of one another, they probably share a similar structure and carry out similar functions. Sequence similarity suggests an evolutionary relationship between proteins, that is, they evolved from a common ancestor. We can therefore describe homologous proteins as belonging to the same "family" and build a "molecular" taxonomy by tracing the lineages from comparisons of sequences. Closely related proteins have the most similar sequences; distantly related proteins have only faintly similar sequences.

Sequence similarity is computed by string alignment algorithms that search the maximum local alignment between two protein sequences (Smith and Waterman, 1981; Altschul et al., 1997). A collection of aligned sequences from different proteins searched in a large database of primary structures (like TrEMBL, Bairoch and Apweiler, 1999) is called a *multiple alignment*. Once the multiple alignment has been computed, a profile is obtained by counting the frequency of each amino-acid at every position in the sequence. These position specific frequencies contain information about the conservation of each position during the evolution of the protein family, an indication of the importance of that residue for the structure and function of the protein. Multiple alignment profiles have been introduced (Rost and Sander, 1993b) in the context of SS prediction because they convey more information than the bare amino-acid string about the native protein structure. The conformation of every position of the sequence is influenced by the whole content of amino-acids of the protein, therefore the same group of linked amino-acids can appear in different configurations if its spatial context varies (such misleading patterns are called *chameleons*). On the

other side, the structure is more conserved than sequence during evolution (Abagyan and Batalov, 1997) and sequence positions that are well conserved in different organisms may identify important residues for the structure and function of the protein.

2.3.2 Secondary Structure

All the observed proteins present local regularities in their three-dimensional structure formed and maintained by hydrogen bonds that are referred to as the protein's *secondary structure*. Without any stabilizing interactions, a polypeptide assumes a random-coil structure. However, when stabilizing hydrogen bonds form between certain residues the backbone assumes one of three geometric arrangements: an α helix, which is a spiral, rod-like structure; a β sheet, a planar structure composed of alignments of two or more β strands, which are relatively short, fully extended segments of the backbone; a turn, which is a U-shaped four-residue segment. In an average protein, 60 percent of the polypeptide chain exists as α helices and β sheets; the remainder of the molecule is in random coils and turns.

In a α helix the carbonyl oxygen of each peptide is hydrogen-bonded to the amide hydrogen of the amino-acid four residues toward the C-terminus. The peptide backbone twists into a helix having 3.6 amino-acids per turn. The stable arrangement of amino-acids in the α helix holds the backbone as a rod-like cylinder from which the side chains point outward.

The β sheet consists of laterally packed β strands. Each β strand is a short nearly fully extended part of the polypeptide chain. Hydrogen bonding between backbone atoms in adjacent β strands, within either the same or different polypeptide chains, forms a β sheet. In a pleated sheet, adjacent β strands can be oriented anti-parallel or parallel with respect to each other. In both arrangements of the backbone, the side chains project from both faces of the sheet.

Turns are compact, U-shaped secondary structures composed of three or four residues, stabilized by a hydrogen bond between their end residues. They are located on the surface of a protein, forming a sharp bend that redirects the polypeptide backbone back toward the interior. Without turns, a protein would be large, extended, and loosely packed. The backbone may also contain long bends, or loops. In contrast to turns, which exhibit a few defined structures, loops can be formed in many different ways.

A group of adjacent amino-acids sharing the same conformation are members of a *segment* of secondary structure. Segments of secondary structure are well defined and stable aggregations of amino acids which strongly influence the chain's folding and which usually carry out specific functions inside the protein.

Many proteins contain one or more *motifs* built from particular combinations of secondary structures. A motif is defined by a specific combination of secondary structures that has a particular topology and is organized into a characteristic three-dimensional structure. The presence of the same motif in different proteins with similar functions clearly indicates that during evolution these useful combinations of secondary structures have been conserved.

2.3.3 Tertiary Structure

Tertiary structure refers to the overall conformation of a polypeptide chain, that is, the three-dimensional arrangement of all the amino acids. In contrast to secondary structure, which is stabilized by hydrogen bonds, tertiary structure is stabilized by hydrophobic interactions between non-polar side chains and, in some proteins, by disulfide bonds (see below). These stabilizing forces hold the α helices, β strands, turns, and random coils in a compact internal scaffold. Thus, a protein's size and shape is dependent not only on its sequence but also on the number, size, and arrangement of its secondary structures.

The tertiary structure of large proteins is often subdivided into distinct globular or fibrous regions called domains. Structurally, a domain is a compactly folded region of a polypeptide. These discrete regions are well distinguished or physically separated from other parts of the protein, but connected by the polypeptide chain. Domains sometimes are defined in functional terms based on observations that the activity of a protein is localized to a small region along its length. The functional definition of a domain is less rigorous than its structural definition. However, if the three-dimensional structure of a protein has not been determined, identification of functional domains can provide useful information about the protein. Because the activity of a protein usually depends on a proper three-dimensional structure, a functional domain consists of at least one and often several structural domains. The organization of tertiary structure into domains fur-

ther illustrates the principle that complex molecules are built from simpler components. Like secondary-structure motifs, tertiary-structure domains are incorporated as modules into different proteins, thereby modifying their functional activities. The modular approach to protein architecture is particularly easy to recognize in large proteins, which tend to be a mosaic of different domains and thus can perform different functions simultaneously. Large collections of structural domains with functional annotations are available for the study of new proteins (PROSITE, Falquet et al. 2002; ProDom, Bru et al. 2005).

2.3.4 Disulfide Connectivity

Cysteines are important amino-acids because they can form covalent bonds, known as disulfide bonds, that play a fundamental role in the stabilization of the native conformation of protein structures. The side chain of a cysteine contains a reactive sulfhydryl group SH, which can oxidize to form a disulfide bond $S-S$ to a second cysteine. Regions within a protein chain or in separate chains sometimes are cross-linked covalently through disulfide bonds. Although disulfide bonds are rare in intra-cellular proteins, they are commonly found in extracellular proteins, where they help maintain the native, folded structure. This is a consequence of the oxidative environment of the ER cell compartment in which secreted proteins fold, that permit the formation of disulfide bonds.

2.3.5 Higher-Order Structures

The highest level of the protein hierarchy of structures is occupied by multimeric proteins, containing two or more polypeptide chains, or subunits, that are held together by non-covalent bonds. Quaternary structure describes the number and relative positions of this subunits in a multimeric protein. However, in a fashion similar to the hierarchy of structures that make up a protein, proteins themselves are part of a hierarchy of cellular structures. Proteins can associate into larger structures termed macromolecular assemblies. Macromolecular assemblies in turn combine with other cell biopolymers like lipids, carbohydrates, and nucleic acids to form complex cell organelles, and so on.

2.4 Determination of Protein Structure

Protein function is derived from protein structure. Thus, to figure out how a protein works, its three-dimensional structure must be known. Thanks to several genome sequencing projects, the entire DNA sequence of many organisms has been experimentally determined. Inside each genome the positions of genes have been discovered and from these identified genes the primary sequences of millions of proteins have been extracted. Using the technique of recombinant DNA, an engineered bacteria can be forced to produce large amounts of any given protein from its genetic sequence. From the purified solution of the desired protein the three-dimensional structure can than be experimentally determined using two alternative methods: x-ray cristallography and nuclear magnetic resonance.

2.4.1 Cristallography

The use of x-ray crystallography to determine the three-dimensional structures of proteins was pioneered in the 1950s. In this technique beams of x-rays are passed through a crystal grown from the solution containing copies of the target protein. The atoms in the protein crystal scatter the x-rays, which produce a diffraction pattern of discrete spots when they are intercepted by photographic film. Such patterns are extremely complex and their interpretation involves elaborate distance geometry calculations (Crippen and Havel, 1988) and opportune chemical modifications of the protein (such as binding of heavy metals). The final solution is a complete protein three-dimensional structure detailed at atomic level with resolutions as low as 1 Å. However, although some proteins readily crystallize, obtaining crystals of others requires a time-consuming trial-and-error effort to find just the right conditions, that takes month to finish and is still impossible to automate.

2.4.2 NMR

The three-dimensional structures of small proteins containing up to about 200 amino-acids can be studied with nuclear magnetic resonance (NMR) spectroscopy. In this technique, a concentrated protein solution is placed in a magnetic field and the effects of different radio frequencies on the resonances of different atoms are measured. The behavior of every atom is in-

fluenced by neighboring atoms in adjacent residues, with the closely spaced residues more perturbed than distant residues. Therefore, from the magnitude of the effect, the distances between residues can be calculated. These distances are then used to generate a model of the three-dimensional structure of the protein. Although NMR does not require crystallization of a protein, a definite advantage, this technique is limited to proteins smaller than about 20 kDa. However, NMR analysis can also be applied to single protein domains, which tend to be small enough for this technique and can often be obtained as stable structures. Unfortunately, the analysis of distances generated from NMR spectra is a critical and time-consuming step in NMR analysis. Since there exists no deterministic algorithm to build a unique model structure from this constraints, human intervention is necessary and results in a trial and error process that can takes as long as the creation of a crystal for x-ray analysis.

2.5 Prediction of Protein Structure

Experimental methods for structure determination are slow and difficult to automate, therefore they cannot be applied at a genomic scale. Genome sequencing projects resulted in millions of known protein sequences (Swiss-Prot and TrEMBL, Bairoch and Apweiler, 1999) but only for few thousands of them a structure has been deposited in the Protein Data Bank (Berman et al., 2002). It is therefore becoming increasingly important to derive protein tertiary structure *ab initio* from its amino-acid sequence using insights obtained from already known structures. According to Anfinsen's thermodynamic hypothesis (Anfinsen, 1973), a protein's native fold is the one having lowest free energy and it depends only on the sequence and the external environment (Berg et al., 2002). Unfortunately, computing and minimizing the potential function from first principles is infeasible for present computers and therefore predictive methods are a very important approach in order to obtain an approximation of the three-dimensional structure associated with known sequences.

Predictive methods for protein structure are traditionally divided into three main classes: comparative/homology modeling (that can be applied if the target protein has sufficient sequence similarity with existing structures), fold-recognition/threading (that requires similarity at the level of structure),

and ab initio (for new folds). The importance of structural genomics for the success of molecular biology rose the need of an independent and reliable assessment of the performances of predictive methods. The various CASP and CAFASP rounds (Zemla et al., 2001; Baker and Sali, 2001) has taken a picture of the evolution of the field, demonstrating the success of homology modeling methods in augmenting the structural knowledge to families of homologous proteins from single known representatives, while documenting the poor performances of the other methods in producing reliable structures other than for small chains.

2.5.1 Homology Modeling

As data concerning protein sequences and three-dimensional structures accumulates, the concept that similar sequences fold into similar secondary and tertiary structures is confirmed: structure is more conserved than is sequence (Abagyan and Batalov, 1997). This is the pillar for the success of homology modeling, whose principal idea is to model the unknown structure of a target protein based on the template of a resolved homolog. The major drawback of this methodology is to limit the predictable structures to a small percentage of known protein sequences with available homologs in the PDB. However, for levels of at least 50% sequence identity, this class of methods is able to output accurate structures (Baker and Sali, 2001) with resolutions even comparable to experimentally determined structures for high values of homology.

The basic assumption of homology modeling is that target and template have identical backbones shapes. Once a good alignment between the two sequences is made, the problem is to correctly place the side chains of the target into the backbone of the template. Side chains conformations are built based on similar environments in known structures (Bower et al., 1997). Libraries of observed side-chain orientations (*rotamers*) are searched by comparing fragments of residues extracted from PDB to the backbone of the template around each amino-acid. The best match constitutes the final orientation of the target's side-chain. Further improvements to the three-dimensional structure are then made using molecular dynamics, a protocols that simulate proteins folding with molecular mechanics potential functions.

2.5.2 Fold Recognition and Threading

Proteins with no resolved homologous sequences cannot be predicted using homology modeling algorithms. However, the PDB contains thousands of pairs of proteins with very similar structure but less than 25% pairwise sequence identity (*remote homologs*). If a correct structural alignment between the target and a remote homolog is given, one could build the 3D structure of the target by (remote) homology modelling based on this template. A popular class of remote homology modelling methods are the *threading* algorithms, whose basic idea is to thread the target sequence onto the known template structure and then select between various tentatives by evaluating the fitness of the model structures through some kind of potential function. Detection of remote homologs is the most critical part of these methods: the alignment between target and template can relies on both primary structure (Altschul et al., 1997; Karplus et al., 1998) or predicted sequences of structural features (Kelley et al., 2000) like secondary structure or solvent accessibility (a measure of the surface of a residue exposed to the surrounding water molecules). Knowledge-based potentials are then built to capture the structural propensities of amino acids (such as preferences for secondary structure formation, hydrophobicity, and polarity) and used to assess whether the target sequence fits into the structural environment of the template (Brooks et al., 1983; Pearlman et al., 1995; Ponder and Case, 2003). These propensities can be also extracted from known structures by training mean-force potentials on observed distances between pairs of amino-acids (Sippl, 1995).

2.5.3 Ab-initio

Detection of remote homologs is a daunting task and still most of the proteins do not belong to any known fold class. Various *de-novo* predictive methods for new folds have been proposed, but the results of various CASP rounds (Zemla et al., 2001) have seen just a bunch of methods emerge in which Rosetta (Bradley et al., 2003) was the leader. In 3D structure prediction the size of the conformational space of proteins that must be sampled to find the correct fold is vast and cannot be tackled without simplified assumptions. Rosetta is based on a picture of protein folding in which short segments of the chain independently sample discrete distributions of local conformations dependent to their composition. Folding to the native state then occurs when these segments have relative orientations and conformations that allow

low free energy nonlocal interactions to form throughout the protein. The distribution of local structures sampled by a given segment during folding is approximated by the distribution of local structures adopted by such segments in known protein structures. During a Rosetta folding simulation, each nine- and three-residue segment of the protein chain flickers between the different local structures that are consistent with its local sequence, while the nonlocal interactions are slowly optimized using a Monte Carlo search procedure. A low-resolution model of the nonlocal interactions dominated by hydrophobic burial and strand pairing is used until near the end of the simulation, when a rotamer-based explicit side-chain model with Lennard Jones interactions is introduced. A total of 1,000 independent simulations are carried out (starting from different random number seeds) for each query sequence, and the resulting structures are clustered using root mean square deviation. These cluster centers are then rank-ordered according to the size of the clusters they represent, with the cluster centers of largest clusters representing the highest confidence models.

Chapter 3

Protein Function

The function of nearly all proteins depends on their ability to bind other molecules, or *ligands*, which may by small molecules (like metal ions and small compounds) or macro-molecules (such as lipids, DNA strands, and other proteins). Almost every chemical reaction in a cell is catalyzed by a class of proteins called *enzymes*. Catalysts increase the rates of reactions that are already energetically favorable by lowering the activation energy. Enzymes are essential to sustain life because most chemical reactions in biological cells would occur too slowly, or would lead to different products, without enzymes. A malfunction of a single critical enzyme can lead to a severe disease. As catalysts of chemical reactions, enzymes must first bind tightly and specifically to their target molecules, called *substrates*.

The native conformation and activities of some proteins require the presence of a prosthetic group, a small non-peptide molecule or metal that binds tightly to a protein, keeping the protein in a fixed conformation and participating in binding ligands. The activity of numerous enzymes also depends on the presence of a prosthetic group, commonly referred to as a co-enzyme. Many co-enzymes act to lower the activation energy of biochemical reactions by forming a covalent intermediate with a substrate. Conversely, inhibitors are naturally occurring or synthetic molecules that decrease or abolish enzyme activity. Suicide inhibitors bind enzymes very tightly, effectively deactivating them. Many drugs and poisons act by inhibiting enzymes.

Biologically active substances interact in most cases with biomolecules, triggering specific molecular mechanisms which finally leads to a certain biological response. Each chemicals that enter an organism can thus alter

its internal state and possibly lead to disease, or have a positive effect and cure a preexisting illness. Discovering a new chemical that can be used to cure a specific disease (without causing another) is the main scope of pharmaceutic research.

3.1 Binding Molecules

Two properties of a protein characterize its interaction with ligands: *affinity* refers to the strength of binding between a protein and a ligand; *specificity* refers to the ability of a protein to bind one molecule in preference to other molecules. Both properties depend on the structure of the ligand *binding site* on the protein, which is designed to fit its partner like a mold. For high-affinity and highly specific interactions to occur, the shape and chemical surface of the binding site must be complementary to the ligand molecule.

The binding sites are located at the surface of the protein molecule and are determined by geometrical arrangements and physico-chemical properties of the side-chains of the residues that form the binding pocket. The amino acids that make the active site do not need to be adjacent in the linear polypeptide sequence; rather, folding of the molecule results in juxtaposition of these amino acids, forming a space in which the substrate sits.

Active sites in enzymes consist of two functionally important regions: one that recognizes and binds the substrate (or substrates), and one that catalyzes the reaction once the substrate has been bound. In some enzymes, the catalytic site is part of the substrate-binding site; in others, the two sites are structurally as well as functionally distinct.

3.2 Binding Metals

A significant fraction (from one third to a half) of all known proteins is believed to bind metal ions as cofactors in their native conformation (Thomson and Gray, 1998). Metal ions in proteins perform multiple tasks: they help stabilizing protein structure, induce conformational changes and assist protein function as cofactors in enzymes. Metallo-proteins participate in the most important biological processes including respiration and oxygen photosynthesis and were the first class of enzymes to appear in cells (Degtyarenko, 2000).

Metal binding sites are characterized by the bonded ion (or ions), by the protein atomic groups involved in binding, by the coordination number (number of bonds to the ion) and by their geometry (for a detailed analysis of metal binding sites in PDB structures see Harding 2004). Metals with a prominent biological role include alkali (K^+, Na^+), alkaline earth (Mg^{2+}, Ca^{2+}) and several transition metals, most importantly Mn, Fe, Cu, Zn and Cd.

Different metal groups exhibit different modalities of binding (Degtyarenko, 2000): alkali and alkaline earth metals bind to proteins predominantly through electrostatic interactions, while transition metals tend to form coordinate covalent bonds with protein atoms. The protein atomic groups involved in binding metals can be part of residues side-chains (mostly cysteines, histidines, methionine, aspartic acids, asparagine, glutamic acids and glutamine), backbone carbonyl oxygens and amides, N-terminal and C-terminal free groups. Coordination numbers vary considerably, ranging from a minimum of 1 to a maximum of about 8. Although most metals have strong preference for a specific coordination number (e.g. 6 for Ca^{2+}, 4 for Zn), they also show a fairly high degree of variability (Ca^{2+} can have from 4 to 7 ligands, Zn from 3 to 5).

Beside the amino acids that participate directly to the chemical bond, other residues can be important for structural stabilization of the binding site or for assisting enzyme activity. Geometry of a binding site results from optimization of the energetic interactions between the involved residues and the bonded metals and between the residues themselves. However, even binding sites that share the same metal, coordination number and bonded residues, can display different geometries due, for example, to a different spacing of the residues along the protein sequence. To make the situation even more intricate, different metals can fit into the same binding site (though with different affinities).

3.3 Detecting Binding Sites From Sequence

Homologous proteins in different organisms are usually evolutionary transformations of common ancestors with which they share most of their functions. The first tool available for functional study of a protein with unknown structure is therefore sequence alignment. The availability of homologs with

known functional domains (ProDom, Bru et al., 2005) is an important indication of possible binding sites.

The mechanisms of ligand binding are diversified and of difficult generalization. However, specific sequences of amino-acids involved in the formation of binding sites have been re-used by evolution in very different contexts. These gapped sequences of residues are strongly conserved and have been identified in many non-homologous proteins. Collections of identified functional patterns (ProSite, Hulo et al., 2004) in the form of regular expressions are used to search for putative sites in new sequences. Although those patterns are very precise indications of active sites, their coverage of newly sequenced proteins is insufficient (Passerini and Frasconi, 2004).

Unfortunately, the binding of a particular ligand to a certain protein can be regulated not only from physical and chemical properties but also from the availability of the ligand through a specific pathway involving different enzymes. Therefore, the actual functions of a protein could not only derive from its amino-acid sequence and predictive methods that do not take into account the biochemical context would not be sufficient for protein function studies.

3.4 Detecting Binding Sites From Structure

Proteins are macro-molecules with thousands of atoms many of which can possibly interact with metals and other molecules. Therefore, even if the protein three-dimensional structure is known, the identification of binding sites on it is still an open problem. The recognition of active sites on a known structure can be conducted by geometrical search of pockets on the protein surface in which the exposed side-chains are conformationally and chemically favorable for binding (Liang et al., 1998; Peters et al., 1996). However, the difficult modeling of general properties of binding sites is a strong limitations to the use of such techniques.

The simulation of ligand binding of a known molecule to a known protein structure is another approach that is widely used for functional annotation. The methods for docking ligands into proteins (Goodsell and Olson, 1990; Morris et al., 1998) perform a global optimization of the ligand conformation to minimize an energy that model affinity of the compound to the putative binding site. Ligand structures are flexible to enable rearrangements of

molecular atoms to accommodate inside the binding pocket.

NMR is also used for locating binding sites and determining their precise conformation (Roberts, 1999). The NMR analysis of protein solution with molecules of interests is used to identify the binding sites by detection of the changes in chemical shifts from non-bonded state and from the presence of particular peaks in NOE spectra. However, the same problems that affect protein structure determination by NMR defy the automation of this process and limit the use of NMR techniques for detection of binding sites.

3.5 Prediction of Activity of Small Compounds

Discovering a new drug to cure a specific disease is the primary goal of pharmaceutic research, but drug discovery is still a long and expensive process. In the initial and most critical phase, for each tentative compound a biological experiment must be conducted in order to verify if the molecule produce the desired response. The number of chemical compounds that have been synthesized or can be synthesized using combinatorial chemistry is extremely large and a complete screening is never practically feasible. Moreover, chemists are not only interested in finding the active molecule but also in studying what part of it leads to the desirable behavior, so that more effective compounds can be synthesized.

In the early 1960s, the pioneering work of Hansch et al. (1962) demonstrated that the biological activity of a chemical compound can be computed as a function of its physico-chemical properties. From these findings, many different techniques have been proposed to correlate biological response with molecular properties of the target compound through quantitative structure activity/property relationships analysis (for a throughly review of QSAR/QSPR techniques, see Sec. 12.1). These methods have been increasingly used to speed up the process of drug discovery by virtually screening all known compounds using statistical models built on those molecule that have already been tested. Even if, QSAR methods are not 100% accurate, they can be used to limit the number of assays performed in vitro to the most probable candidates, thus decreasing the total cost of the drug discovery process.

Part II

Learning From Data Structures

Chapter 4

Learning from Protein and Ligand Structure

In Chaps. 2 and 3 the problems of predicting the structure and function of proteins and ligands have been exposed in their cruciality. Inside the fields of structural genomics, proteomics and drug design there are many tasks that already benefit of a machine learning approach to the prediction of structural and functional features of proteins and ligands (see Baldi and Brunak, 2001, for a review).

Biomolecules are rich sources of information. Proteins have many hierarchical levels of structure that highlight different aspects and properties of their three-dimensional conformation. Ligands are chemical compounds with complex shapes and varied compositions. Proteins and ligands are the instances from which learning machines must be trained. Therefore, the problem of extracting all the relevant information from these richly structured data must be tackled in order to solve the predictive tasks needed in computational biology.

In this work all the prediction problems are considered as specific instantiations of the general framework of learning from parts and relations of structured data (Sec. 4.1). The various tasks are divided into three categories on the basis of the structure of the input data: learning from sequence (Sec. 4.2), learning from contacts (Sec. 4.3) and learning from coordinates (Sec. 4.4). This classification is inspired by the taxonomy of predictive methods for protein structures referred by Rost (1998) and it is extended

to comprise the prediction of ligand properties. These two categorizations are orthogonal, in the sense that what is here an input to the predictive method in the classical taxonomy is its output. The division adopted in this work is intended to represent the increasing level of complexity in the input structure from one-dimensional to three-dimensional data.

4.1 Learning from Parts and Relations

All the predictive methods described in this work are supervised learning algorithms in which a training set $\mathcal{D} = \{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle\} \subset \mathcal{X} \times \mathcal{Y}$ of input/output pairs is used to learn a mapping $f : \mathcal{X} \to \mathcal{Y}$ from a input pattern $\boldsymbol{x} \in \mathcal{X}$ to a output label $\boldsymbol{y} \in \mathcal{Y}$ by choosing the parameters $\boldsymbol{\theta}$ of the decision function $f(\boldsymbol{x}; \boldsymbol{\theta})$, whose class $\mathcal{F}$ is a characteristic of the specific algorithm. The various problems of prediction of structural and functional properties of proteins and ligands can be reduced to the supervised learning setting by choosing suitable representations of molecular data into input patterns and target properties into output labels.

In the relational model (Codd, 1970; Date, 1995; Lavrač and Džeroski, 2001) a datum is represented as a pair $\langle P, T \rangle$ formed by a set P of *entities* (or *parts*) and a set $T \subseteq 2^P$ of *relations* between these parts, each element of T being a set of related entities in P. Both entities and relations can have attributes. The representation of a data into entities and relations can then be mapped into a hyper-graph $\langle V, E \rangle$ with a vertex for each entity ($V \equiv P$) and a hyper-edge connecting each group of related entities ($E \equiv T$).

Not all supervised learning algorithms are able to learn the I/O map directly from structured (graphical) inputs. Early sub-symbolic learning methods were based on the limiting assumption that input data was vectorial ($\mathcal{X} \equiv \mathbb{R}^k$). Structured inputs can be reduced to vectorial representation by extracting their *features*, each feature being a real-valued representation of the possible states accessible to a specific *attribute* of the input data (*propositionalization*, Kramer et al. 2001a). However, propositionalization is usually a time and space consuming procedure: enumerating all the possible topological connections between the parts of a data structure can lead to a combinatorial explosion of the number of features, especially if higher order relations (paths of length ≥ 2 in the hyper-graph) are considered. Recursive graphical models (see Chap. 5 for a review of the recursive neural

networks architectures that are used in this work) were the first class of subsymbolic supervised learning methods that could process graphical inputs, through the recursive computation of the decision function. On the other side, kernel machines (see Chap. 6) reduce the computational costs of propositionalization by implicitly mapping input data to feature vectors using a kernel function. In fact, various kernel functions have been proposed for computing the similarity between graphical structures (see Gärtner, 2003; Gärtner et al., 2004, for a review).

The representation of target properties into output labels is more constrained by the capabilities of available learning algorithms. Every supervised learning algorithm can handle scalar outputs , that can be either discrete (*classification*, $\mathcal{Y} \subset \mathbb{N}$) or real-valued (*regression*, $\mathcal{Y} \equiv \mathbb{R}$)). A scalar label can be associated either to the whole molecule, thus allowing the prediction of global properties of the target structure (e.g. the activity of a ligand), or to distinct parts of the molecule (e.g. the secondary structure class of a protein residue). In this second case, a learner that output only scalar values would be forced to predict all labels independently for each part of the same structure. Conversely, recursive graphical models are able to learn a IO-isomorph map from a input data to a output structure that has the same topology of the input and a scalar label in each vertex. Recursive models are thus able to capture the correlation between output labels at distinct part of the data structure. Tsochantaridis et al. (2004) proposed a support vector machine approach for learning a decision function on the joint $\mathcal{X} \times \mathcal{Y}$ space. During testing the decision $\boldsymbol{y} = \arg\max_{\boldsymbol{y} in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ is computed by searching for the most confident output structure given the input. This method allow the structure of the output to be different to the structure of the input. However, the number of output patterns that needs to be explored for each input data during both training and testing limit the use of this approach.

4.2 Learning from Sequence

The only information available to ab-initio methods for structure prediction is the sequence of amino-acids that form the target protein. A protein primary structure can be represented as a string of symbols in a 20 letters alphabet and effectively used by methods that perform string processing,

e.g. kernel machines based on the Spectrum Kernel (Leslie et al., 2002). If multiple alignment profiles are used every position of the protein sequence is encoded by a vector of 20 real numbers and string methods do not apply. In the entity-relationship model outlined in Sec. 4.1, the primary structure can be represented as an entity set formed by the elements of the chain and a relation set defined by their sequential ordering. The sequence of amino-acids can thus be mapped into an undirected acyclic graph whose nodes contain the encoded residues and their attributes.

In structural genomics both protein-wise and single-residue predictions tasks are found. Protein-wise classification problems comprise the prediction of protein sub-cellular localization (Nair and Rost, 2005), protein fold classification (Jaakkola et al., 2000; Leslie et al., 2002) and many other important tasks. In this work, only problems in which a property is predicted from sequence for each residue are considered. To this end, in the following chapters a hybrid SVM-BRNN architecture is described for the prediction of protein secondary structure (Chap. 7) and metal binding sites (Chap. 8).

4.3 Learning from Contacts

The folding of a protein chain is a consequence of the chemical interactions of protein atoms with themselves and with solvent atoms. In a folded structure hydrophobic amino-acids are densely packed in the three-dimensional space to escape the interactions with water molecules. The packing of residues can be captured by computing their relative distances and it results in a collection of values $\{d_{i,j}\}$ called the *distance matrix*. The distance matrix is a redundant representation of the protein structure (for a proteins of n residues the distance values are n^2 while the coordinates are $3n$) that is independent of the coordinates reference system. Ab-initio methods could in principle focus on the prediction of the distances between residues instead of their coordinates. In reality, a simplified version of this task has been proposed in which the distances are discretized and the problem is to predict the *contact maps* at various resolutions (Fariselli et al., 2001; Pollastri and Baldi, 2002; Pollastri et al., 2003). The contact map for a protein of n residues is a $n \times n$ boolean matrix in which a *contact* $c_{i,j}$ at a certain resolution δ implies that corresponding residues i and j are distant in space less that δ. The three-dimensional structure of a protein can be reconstructed from its contact

map with good accuracy even in presence of strong Gaussian random noise (Vendruscolo et al., 1997). A reduced contact prediction tasks has also been proposed in which contacts are defined between segments of secondary structure. A smaller $k \times k$ *coarse-grained* contact map is thus defined for a protein with k segments. This matrix can in turn be used to output a low resolution reconstruction of a protein structure as it is outlined in Chap. 10.

The predicted contact map can contain additional information (other than the sequence of amino-acids) about the native conformation of the protein and can be used as an input to other prediction tasks. The matrix of contacts induce a relation between residues that are distant in sequence but close in space. These *long-range* relations define the spatial context of a residue and can be used to identify the allowed conformation for a fragment of residues, thus reducing the problem of chameleons. The representation of a protein structure by a sequence of amino-acids and a contact matrix is a graphical structure in which the vertices encode for chain elements and the edges connect pairs of residues in contact. In Chap. 9 the graphical representation of a protein chain with long-range interactions is used to improve the prediction of secondary structure.

The chemical structure of a molecule in the form of atoms and bonds is another type of informative graphical representation that can be used to learn input/output mappings. As mentioned before, the chemical structure of a protein is quite peculiar and can simply be represented by its sequence of amino-acids. The actual representation of all atomic bonds is not informative since it can be subsumed by the primary structure. An exception is constituted by disulphide bonds, whose pattern is not easily induced by the disposition of cysteines along the sequence but arise from their proximity in the folded structure (other than the specific intra-cellular environment in which the folding takes place). In Vullo and Frasconi (2004) the prediction of disulphide connectivity is configured as a preference learning task. A scoring function is trained on all possible disulphide patterns to give the highest rank to the correct topology. Each example is formed by a set of vertices encoding for the cysteines and their environment, and a set of edges connecting the cysteines in one of the allowed topologies. This method relies on a accurate prediction of the bonding state of cysteines, a problem that here is addressed in Chap. 8.

The representation with atom and bonds of ligand molecules is more

informative than for proteins since their chemical structure is more diversified. A graphical representation of a molecule is easily derived in which the nodes are the molecule atoms and the edges are the bonds between them. In this context many methods for graph classification and regression have been proposed to solve both QSAR and QSPR problems (Deshpande et al., 2003; Horváth et al., 2004; Swamidass et al., 2005). In Chap. 12 a weighted decomposition kernel (see Sec. 6.6) for graphs is used to predict the properties of ligands and compared to a novel decomposition kernel for three-dimensional structures that is presented in this work.

4.4 Learning from Coordinates

Both proteins and ligands are strongly characterized by their three-dimensional structure which affect their ability to perform specific functions. The physical and chemical properties of a molecule are a function of the shape of the electron cloud surrounding each atom, that is a consequence of the interactions between the atom and its neighbors.

Threading methods use potentials computed on tentative protein models to choose the predicted protein structure (see Sec. 2.5). These mean-force potentials (Sippl, 1995) are function of the (discretized) distances between pairs of residues and are trained using known native structures without supervision on decoys. Alternatively, the matrix of distances can be used in conjunction with probability product kernels (Sec. 6.3) for learning to rank decoys in a supervised setting as shown in Chap. 11.

The distance matrix representation of a three-dimensional structure capture only pair-wise interactions between atoms. Relations between higher numbers of molecule atoms can model more complex structural characteristics such as bond angles, dihedral angles, polyhedra, etc. In Sec. 6.7 a kernel that compares the shapes formed by groups of molecule atoms is presented and then used to address the problem of predicting the activity and the properties of small chemicals (Chap. 12).

Chapter 5

Learning Data Structures with Neural Networks

Neural networks (NN) are a popular class of supervised learning methods originated from the early works on perceptrons (Rosenblat, 1958) and multi-layered perceptrons (Rumelhart et al., 1986). Each neuron of the network implements a I/O transition function that is computed as a non-linear function (typically a tanh) of its activation value, which in turn is a weighted combination of the neuron inputs. In a multi-layered perceptron, the neurons are collected into layers and connected such that the outputs of neurons at i-th layer are the inputs of neurons at $(i+1)$-th layer. The neurons and their dependencies form a graphical structure, and for this reason neural networks are considered as graphical models. Network weights are optimized by minimization of an error function that measures the difference between outputs and targets. Error minimization is performed using a gradient descent approach called back-propagation. Gradient computation impose a constraint on the model graphical structure which must be directed and acyclic. Moreover, feed-forward neural networks are limited to fixed-size input vectors.

Recursive neural networks (RNN) have been later introduced (Frasconi et al., 1998) to overcome the limitation of NNs to vector inputs and extend the neural models to relational data. RNN graphical structure contains cyclic connection between recursive layers (*states*), but its unfolding over the input structure is a DAG and therefore can still be trained by back-

propagation (details are given in Sec. 5.1). Standard RNN architectures are limited to input graphs that are directed ordered acyclic graphs (DOAGs). For this reason contextual RNN (CRNN) have been introduced (see Sec. 5.2 for an overview), where the input undirected graph is split into several DOAGs. This class contains bidirectional RNN (BRNN) for sequence translation (Sec. 5.3) and a novel architecture, dubbed interaction-enriched BRNN (IEBRNN), that is introduced here to model explicit long-term interaction between sequence positions (Sec. 5.4).

5.1 Recursive Neural Networks

Recursive neural networks are a generalization of feed-forward neural networks capable of processing structured data as directed ordered acyclic graphs, where a input vector and a discrete or real output label are associated with each vertex (Frasconi et al., 1998). The key idea is to replicate a NN for each node of the DOAG and consider as input to the network both the atomic information represented by the input vectors and the structured information derived by the outputs of all the networks instantiated for each child node. The process of replicating a NN is called network unfolding and, as a result of this procedure, a single large network is obtained having shared weights and whose topology matches that of the input graph.

At each node v, the NN outputs a vector encoding of the whole subgraph induced by vertices reachable from v. Data processing takes place in recursive fashion, traversing the DOAG in post-order, using a transition function t such that $X(v) = t(X(ch[v]), I(v))$ where $X(v) \in \mathbb{R}^N$ denotes the state vector associated with node v, $I(v) \in \mathbb{R}^M$ is the input vector associated with node v and $X(ch[v]) \in \mathbb{R}^{C_v N}$ is a vector obtained by concatenating the components of the state vectors contained in the C_v children of v. The transition function $t : \mathbb{R}^{C_v N} \times \mathcal{I} \to \mathbb{R}^N$ maps states at v's children and the inputs at v into the state vector at v. A frontier state $X_0 = 0$ is used as the base step of recursion. A feed-forward neural network is used to model the transition function t according to the scheme:

$$\begin{aligned} a_j(v) &= \theta^{in}_{j,0} + \sum_{h=1}^{M} \theta^{in}_{j,h} I_h(v) + \sum_{k=1}^{C_v} \sum_{l=1}^{N} \theta^{rec}_{j,k,l} X_l(ch_k[v]) \qquad (5.1) \\ X_j(v) &= tanh(a_j(v)) \ , \quad j = 1, \ldots, N \end{aligned}$$

where $X_j(v)$ denotes the j-th component of the state vector at vertex v, $I_h(v)$

is the h-th component of the input vector at v and $\boldsymbol{\theta}^{in}$, $\boldsymbol{\theta}^{rec}$ are adjustable weights. Proceeding in this fashion, the state vector $X(r)$ at the root r of the tree encodes the whole data structure and can be used for subsequent processing (i.e when learning a mapping from the input structure and a global output label). The prediction $f(v) \in \mathbb{R}^Q$ at node v is then computed by the output network as:

$$f_q(v) = \theta^{out}_{q,0} + \sum_{j=1}^{N} \theta^{out}_{q,j} X_j(v) \ , \quad q = 1, \ldots, Q \tag{5.2}$$

where $\boldsymbol{\theta}^{out}$ are the weights of the output network. A suitable error function is then used regarding of the problem, i.e. classification or regression. Minimizing the error leads to find a value for the parameters of the RNN and to discover a vector state representation of input structures. Minimization is achieved by a variant of the gradient descend back-propagation algorithm (Goller and Kuechler, 1996).

5.2 Contextual Recursive Neural Networks

Recursive models put a causality assumption on data processing: structures are processed bottom-up according to a reverse topological order of the nodes. Therefore, the state variables associated to these nodes and their outputs depend only on the sub-structures induced by their children. The above assumption imposes some restrictions on the amount of contextual information that can be tackled and extensions of these models for dealing with more general undirected structures have been proposed (Baldi et al., 1999; Pollastri and Baldi, 2002; Vullo and Frasconi, 2003). A more general assumption is considered here: the input space $\mathcal{X}$ is contained in the class of bounded-degree undirected graphs. In this case, there is no concept of causality and the computational scheme described in Frasconi et al. (1998) cannot be directly applied. The strategy consists in splitting graphical processing into a set of causal "dynamics", each one computed over a plausible orientation of input graph $\boldsymbol{x}$. More formally, assume $\boldsymbol{x} = \langle V, E \rangle \in \mathcal{X}$ has a single connected component. A set of spanning DAGs $G_1, \ldots, G_m$ with $G_i = \langle V, E_i \rangle$ is identified such that:

- the undirected version of G_i is $\boldsymbol{x}$
- $\forall\ v, u \in V\ v \neq u,\ \exists\ i :\ \langle v, u \rangle \in E_i^\star$ the transitive closure of E_i

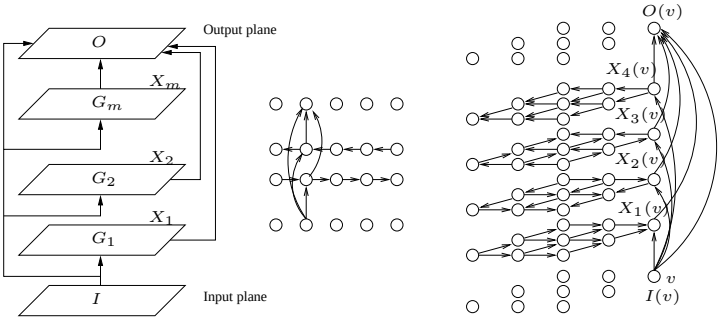


Figure 5.1. (left): Contextual RNNs, dependencies among input, state and output variables. (center and right): processing of undirected sequences and grids with contextual RNNs (only a subset of connections are shown).

and for each G_i, introduce a state variable X_i computed in the usual way.

Fig. 5.1 (left) shows a compact description of the set of dependencies among the input, state and output variables. Connections run from vertices of the input structure (layer I) to vertices of the spanning DAGs and from these nodes to the nodes of the output structure (layer O). Using weight-sharing, the overall model can be summarized by $m+1$ distinct neural networks implementing the output function $O(v) = f(X_1(v), \ldots, X_m(v), I(v))$ and the m transition functions $X_i(v) = t_i(X_i(ch_1[v]), \ldots, X_i(ch_k[v]), I(v))$. Learning can then proceed by gradient-descent (back-propagation) due to the acyclic nature of the underlying graph.

Fig. 5.1 (center) shows that an undirected sequence can be spanned by two sequences oriented in opposite directions, thus obtaining bidirectional recurrent neural networks (Baldi et al., 1999) or bi-recursive neural networks (Vullo and Frasconi, 2003), if generic undirected graphs are considered. In case of two-dimensional grids, which can be seen as spanned by four directed grids oriented from each cardinal corner (Fig. 5.1, right), the corresponding model is called 2D DAG-RNNs (Pollastri and Baldi, 2002).

5.3 Bidirectional Recursive Neural Networks

In several interesting prediction tasks (e.g. protein secondary structure in molecular biology) both input $\boldsymbol{x}$ and output $\boldsymbol{y}$ are in the form of sequences. In this kind of sequential supervised learning problems, sometimes also referred to as *sequence translation* task (see e.g. Dietterich, 2002, for a review), the objective of learning consists of approximating the probabilistic depen-

dency between sequence pairs by a function f that maps an input sequence $\boldsymbol{x}$ into a corresponding output sequence $f(\boldsymbol{x})$. The bidirectional version of RNN has been successfully applied to various problems involving biological sequences (Baldi et al., 1999; Pollastri et al., 2002b,c; Hawkins and Bodén, 2005). BRNNs are a special case of contextual RNNs with two state transition functions: for each position t of the input sequence, two real vectors $X_F(t) \in \mathbb{R}^N$ and $X_B(t) \in \mathbb{R}^N$ are introduced, called the *forward* and the *backward* states, respectively. Intuitively the forward state at t, $X_F(t)$, contains context information about the "past" sub-sequence $I(1), \ldots, I(t)$; similarly, $X_B(t)$ contains context information about the "future" sub-sequence $I(t), \ldots, I(T)$. These two vectors are expected to contain all the information about the input sequence that is needed to make a prediction at position t.

A standard BRNN can be interpreted as the combination of two discrete-time dynamical systems defined by two recursive nonlinear update equations in which $X_F(t)$ explicitly depends on $X_F(t-1)$ and $X_B(t)$ on $X_B(t+1)$. An output function finally calculates $f(t)$ from $X_F(t)$ and $X_B(t)$. The mapping between the input sequence and the output sequence in a BRNN is thus *global* in the sense that the output $f(t)$ at every position $t = 1, \ldots, T$ depends on BRNN inputs $I(\tau)$ at *every* other position $\tau = 1, \ldots, T$. A softmax function can then be used to guarantee that $f(t)$ can be interpreted as the (conditional) probabilities of the output given the input sequence.

The main difficulty with this class of neural networks is due to the lack of generally efficient algorithms for solving numerical optimization. In particular, error minimization is known to fail in the presence of long-range dependencies between distant sequence positions (Bengio et al., 1994; Hochreiter et al., 2001). Interesting remedies to vanishing gradients have been suggested in the literature (Hochreiter and Schmidhuber, 1997; Gers et al., 2002) but their effectiveness in realistic large scale supervised learning tasks has not been elucidated so far. The severity of this problem clearly depends on the specific application domain. For some tasks, dependencies are *mainly* local and sub-optimal solutions can be still quite useful in practice. Note that the difficulty of dealing with long-range dependencies is not necessarily inherently associated with neural networks. Other learning algorithms may equally suffer computational complexity problems.

5.4 Interaction-enriched BRNNs

As stated in the previous section, learning a sequence mapping in the presence of implicit long-range interactions is difficult. If these interactions could be made explicit, through specification of (even tentative) interacting sequence positions, sequential translation problems would become less difficult. In this extended setting, input sequences are enriched with interaction relations: the input portion of the data is now described by an undirected graph $\boldsymbol{x} = \langle V, E^s \cup E^i \rangle$ where V is the set of nodes forming the input sequence, E^s are the edges connecting nodes in sequential relation and E^i is the set of edges representing interactions between distant sequence positions. In this case the objective is to learn a function f that can exploit both sources of information: the sequential relations E^s and the additional relational information consisting of the interactions E^i.

Here, an extension to the family of BRNNs is proposed to solve interaction enriched sequential supervised learning problems. The resulting architecture is called Interaction Enriched BRNN (Fig. 5.2). As in standard BRNN, for each position t, a forward state $X_F(t) \in \mathbb{R}^N$ and a backward state $X_B(t) \in \mathbb{R}^N$ are defined. In an IEBRNN additional "shortcut" dependencies are introduced so that states $X_F(t)$ and $X_B(t)$ depend explicitly also on states at interacting positions, as specified by E^i. In order to construct the generalized dynamics for an IEBRNN two directed graphs are defined from $\boldsymbol{x}$ as follows:

$$\begin{aligned} E_F &\doteq \{(s,t) : \{s,t\} \in E^s \cup E^i, s < t\} \\ E_B &\doteq \{(s,t) : \{s,t\} \in E^s \cup E^i, s > t\}. \end{aligned} \tag{5.3}$$

Now a predecessor graph $G_F = \langle V, E_F \rangle$ with forward oriented edges and a successor graph $G_B = \langle V, E_B \rangle$ with backward oriented edges are defined. It is immediate to see that G_F and G_B are acyclic. Moreover, for each given vertex t, the set $\{(t,s) : s \in V\}$ of edges incident on t can be sorted in increasing order of s. Also, all graphs are assumed to have a degree smaller than a fixed constant K. Define

$$\ell_{t,j} = \begin{cases} j\text{-th vertex in the sorted adjacency list of } t \text{ in } G_F & \text{if } t \text{ has at least } j \text{ parents in } G_F \\ 0 & \text{otherwise} \end{cases}$$

and

$$r_{t,j} = \begin{cases} j\text{-th vertex in the sorted adjacency list of } t \text{ in } G_B & \text{if } t \text{ has at least } j \text{ parents in } G_B \\ T+1 & \text{otherwise} \end{cases}$$

The IEBRNN is then based on the following non-causal dynamics:

$$\begin{aligned} X_F(t) &= t_F\left(I(t), X_F(t-1), X_F(\ell_{t,1}), \ldots, X_F(\ell_{t,K}); \theta_F\right) \\ X_B(t) &= t_B\left(I(t), X_B(t+1), X_B(r_{t,1}), \ldots, X_B(r_{t,K}); \theta_B\right) \end{aligned} \tag{5.4}$$

with boundary conditions $X_F(0) = X_B(T+1) = 0$. In the above recursions, t_F and t_B are the forward and backward state transition function, respectively. They are parametric functions with adjustable parameters $\boldsymbol{\theta}_F$ and $\boldsymbol{\theta}_B$ that are determined by learning. The transition functions can be realized by feed-forward neural networks with $M + (K+1)N$ inputs and N sigmoidal outputs (with no internal hidden layer). Outputs are then computed as follows:

$$f(t) = t_{out}\left(X_F(t), X_B(t); \boldsymbol{\theta}_{out}\right) \tag{5.5}$$

where t_{out} is also a parametric function with adjustable parameters $\boldsymbol{\theta}_{out}$.

The computations described by Eqs. 5.4 and 5.5 can be graphically depicted as shown in Figure 5.2. Nodes in the diagram represent input, state, and output vectors at different sequence positions. Arcs represent arguments to transition and output functions. Dotted arcs represent the first argument of functions t_F and t_B. Solid arcs going left-to-right and right-to-left represent the second argument of functions t_F, t_B and are also found in the computation of standard BRNNs. Thin arcs are shortcut inherited from the interaction graphs associated with the input sequence. Finally, dashed arcs represent the arguments of the output function t_O.

Assuming Q classes, the outputs are then computed using a soft-max function:

$$\begin{aligned} a_q(t) &= \sum_{j=1}^{N} \theta^{out}_{F,j,q} X_{F,j}(t) + \sum_{j=1}^{N} \theta^{out}_{B,j,q} X_{B,j}(t) \quad q = 1, \ldots, Q \\ f_q(t) &= \frac{e^{a_q[t]}}{\sum_{r=1}^{Q} e^{a_r[t]}}. \end{aligned} \tag{5.6}$$

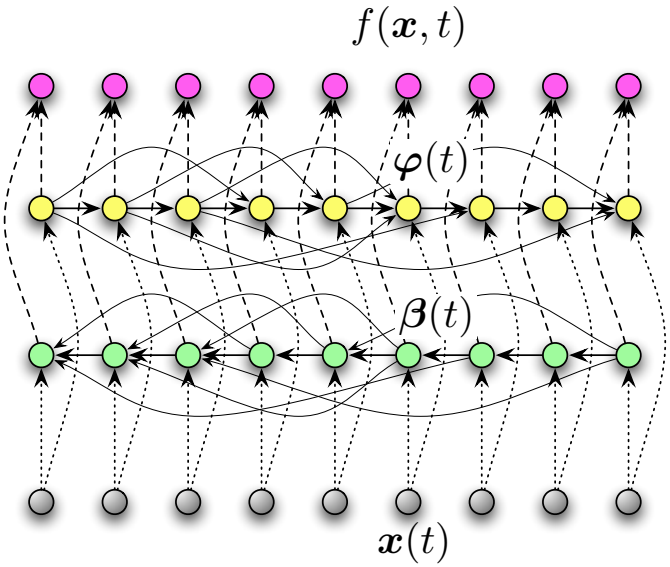


Figure 5.2. Graphical representation of the message passing in the IEBRNN architecture.

The final decision is then taken as

$$y(t) = \arg\max_q f_q(t) \tag{5.7}$$

The use of normalized exponentials allows the interpretation of $f_q(t)$ as the conditional probability $\Pr(y(t) = q|\boldsymbol{x})$ that the predicted class at position t is q, given the input $\boldsymbol{x}$. Discriminant training is carried out following the maximum likelihood approach: a likelihood function of the parameters and the training set is obtained thanks to the probabilistic interpretation of the outputs. Assuming a training set $\mathcal{D}$ of i.i.d sequences and denoting by $\boldsymbol{\theta}$ the whole set of parameters:

$$\ell(\mathcal{D}; \boldsymbol{\theta}) \doteq \log p(\mathcal{D}|\boldsymbol{\theta}) \propto \sum_{i=1}^{|\mathcal{D}|} \sum_{t=1}^{T_i} \log p(y_i(t)|\boldsymbol{x}_i, \boldsymbol{\theta}) \tag{5.8}$$

that can be rewritten as

$$\ell(\mathcal{D}; \boldsymbol{\theta}) \quad = \quad \sum_{i=1}^{|\mathcal{D}|} \sum_{t=1}^{T_i} \sum_{q=1}^{Q} z_{i,q}(t) \log f_q(t; \boldsymbol{x}_i, \boldsymbol{\theta}) \tag{5.9}$$

where $z_{i,q}(t) = 1$ if $y_i(t) = q$ and 0 otherwise. Gradient computation for likelihood optimization can be carried out analytically using back-propagation

on the unfolded architecture (see Figure 5.2) and taking into account the fact that parameters of the transition functions and output function are shared across different sequence positions. The details are highlighted below.

5.4.1 Gradient Computation

For simplicity only the forward states $X_F(t) \in \mathbb{R}^N$ are considered since the computation for backward states follows a similar scheme. Also, to avoid a cumbersome notation the calculations are shown only on a single sequence. For a generic component $j = 1, \ldots, N$ of the forward state vector let

$$\begin{aligned} a_j(t) &= \theta_{F,j,0} + \sum_{h=1}^{M} \theta^{in}_{F,j,h} I_h(t) + \sum_{l=1}^{N} \theta^{so}_{F,j,l} X_{F,l}(t-1) + \\ &\quad + \sum_{k=1}^{K} \sum_{l=1}^{N} \theta^{ie}_{F,j,l,k} X_{F,l}(\ell_{t,k}) \\ X_{F,j}(t) &= \tanh(a_j(t)) \end{aligned} \tag{5.10}$$

where $\theta^{in}_{F,j,h}$ are weights connecting input units to forward state units, $\theta^{so}_{F,j,l}$ are weights connecting previous to present states (serial order), and $\theta^{ie}_{F,j,l,k}$ are weights associated with connections defined by interacting states (the F subscript stands for "forward" weights). Define

$$\delta_j(t) \doteq \frac{\partial \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})}{\partial a_j(t)} \tag{5.11}$$

where $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})$ is the likelihood of the generic sequence in the data set given the entire vector of network weights $\boldsymbol{\theta}$. Therefore:

$$\begin{aligned} \delta_j(t) = \Bigg(& \sum_{q=1}^{Q} \theta_{Y,q,j}(z_q(t) - f_q(t)) + \sum_{l=1}^{N} \theta^{so}_{F,l,j} \delta_l(t+1) + \\ & + \sum_{\tau, h:\ell_{\tau,h}=t} \sum_{l=1}^{N} \theta^{ie}_{F,l,j} \delta_l(\tau) \Bigg) \cdot \tanh'(a_j(t)) \end{aligned} \tag{5.12}$$

where $\theta_{Y,q,j}$ are the output weigths, $z_q(t) = 1$ if $y(t) = q$ and $z_q(t) = 0$ otherwise. Gradients for the weight connecting two units are finally computed by multiplying the δ of the receiving unit by the activation of the sending unit. Thus for example

$$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})}{\partial \theta^{in}_{F,j,l}} = \sum_t \delta_j(t) I_l(t) \tag{5.13}$$

where the summation over sequence indices t is needed to take into account weight sharing.

Chapter 6

Learning Data Structures with Kernel Machines

Kernel machines (Schölkopf and Smola, 2002) are a class of learning methods based on a non-linear mapping of inputs to features induced by a positive definite kernel (see Sec. 6.1). Support vector machines are a popular kernel method used to learn the parameters $\boldsymbol{\theta}$ of the decision function $f(\boldsymbol{x};\boldsymbol{\theta})$ computed as a weighted combination of the kernel between $\boldsymbol{x}$ and the training examples (Sec. 6.2). SVMs define the training phase as a quadratic optimization problem, thus obtaining a unique optimal set of parameters in a time that is a polynomial function of the number of training examples.

Suitable kernel functions have been defined for dealing with structured data. Probability product kernels (Sec. 6.3) improves the propositionalized representation of input data by integrating the product between the probabilities of its attributes. Fisher kernels (Sec. 6.4) use generative models to represent the structure of the input data as a set of real-valued parameters. Both probability product and Fisher kernels are used in this work to learn a ranking between decoys generated by threading methods (Chap. 11). Decomposition kernels (Sec 6.5) are a vast class of kernel functions that relies on a decomposition in parts of structured objects and a composition of kernels between these parts. A novel decomposition kernel specific for solid structures is defined in Sec. 6.7. The three-dimensional decomposition kernel is then used for learning structure/activity and structure/property relationships from datasets of molecules (Chap. 12) and compared to a

weighted decomposition kernel for graphs (Sec. 6.6).

6.1 Kernel Machines

A symmetric real function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel iff, for all $n \in \mathbb{N}$ and $x_1, \ldots, x_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$, it holds:

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0 \tag{6.1}$$

If K is a positive definite kernel than for Mercer theorem a feature map $\phi : \mathcal{X} \to \mathcal{H}$ into the Hilbert space $\mathcal{H}$ exists such that $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ (Aronszajn, 1950). The Hilbert space $\mathcal{H}$ is called the *feature space*.

The attractiveness of kernel methods comes from the fact that the dot product $\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$ is usually computed without explicitly performing the transformation from input to features. This implies that the time complexity of the kernel computation is independent of the dimension of the feature space. For example, the Gaussian kernel $e^{\gamma \|\vec{x} - \vec{x}'\|^2}$ between two input vectors $\vec{x}$ and $\vec{x}'$ induce a map from the input space R^n to an infinite dimensional feature space (Cristianini and Shawe-Taylor, 2000), but the time needed to compute the kernel is only given by the calculation of the distance between the two vectors in the input space.

Kernels can also be combined to generate novel kernel functions. Any polynomial of positive kernels is itself a positive definite kernel. As a result of kernels composition, a new feature space is defined that has higher dimensionality of the original ones.

6.2 Support Vector Machines

Support vector machines (Cortes and Vapnik, 1995; Cristianini and Shawe-Taylor, 2000) are a class of kernel methods motivated by Vapnik principle of structural risk minimization (Vapnik, 1998). Support vector machines search an optimal decision function $f : \mathcal{X} \to \mathbb{R}$ that minimizes a regularized risk functional over a training set $\{\langle \boldsymbol{x}_i, y_i \rangle\}$. The class of decision functions used by SVM has the following form:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{i=1}^{m} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) \tag{6.2}$$

where y_i are the outputs observed in training examples, $\boldsymbol{\theta} = \{\alpha_i\}$ is a set of non-negative real-valued parameters and K is a kernel function. This decision function correspond to the distance of $\phi(\boldsymbol{x})$ from the separating hyper-plane $\vec{w} = \sum_{i=1}^{m} \alpha_i y_i \phi(\boldsymbol{x}_i)$ in the feature space induced by the kernel. Vapnik (1998) demonstrated that the generalization error of this class of decision functions is bounded by a measure that is inversely dependent on the *margin* $\frac{1}{\|\vec{w}\|}$ of the hyper-plane (minimum distance of the hyper-plane from data points). Hence, the optimal set of parameters is the one that maximize the margin and SVM are also called *maximum margin classifiers*.

In the typical setting, the kernel function K is fixed and learning consists of determining the coefficients α_i as the (unique) solution of the following quadratic programming problem with linear constraints:

$$\begin{aligned} \max_{\alpha_1,\ldots,\alpha_m} \quad & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \quad 1 \leq i \leq m \\ & \sum_{i=1}^{m} y_i \alpha_i = 0. \end{aligned} \tag{6.3}$$

This convex optimization task is a kernelized dual version of the minimization of the inverse margin:

$$\begin{aligned} \min \quad & \|\vec{w}\|^2 + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & y_i(\vec{w} \cdot \phi(\boldsymbol{x}_i)) \geq 1 - \xi_i \quad 1 \leq i \leq m \\ & \xi_i \geq 0 \end{aligned} \tag{6.4}$$

where the slack variables ξ_i model decision errors.

The parameter $C > 0$ in both problems controls *regularization* by quantifying the trade-off between fitting the data perfectly (low decision error $\sum_{i=1}^{m} \xi_i$) and choosing a smooth solution that generalizes well to future data (large margin $\frac{1}{\|\vec{w}\|}$). Large value of C reflect the belief that data contain little noise and can be safely fitted. In the limit of very large C, no restriction is imposed on the coefficients α_i and data will be fitted as well as possible.

6.3 Kernels for Distribution of Properties

As mentioned in Sec. 4.1, propositionalization transform a complex data $\boldsymbol{x}$ into a vector of features $\vec{x}$, each feature representing the states accessible to

a single attribute of $\boldsymbol{x}$. A simple kernel between two vectors $\vec{x}, \vec{x}' \in \mathbb{R}^k$ is computed as the linear dot product $\vec{x} \cdot \vec{x}'$. More complex feature spaces can be defined by composition with other kernel functions such as the polynomial $(\vec{x} \cdot \vec{x}' + 1)^d$ and the Gaussian $e^{\gamma \|\vec{x} - \vec{x}'\|^2}$.

If the distribution of states accessible to each attribute is either known or estimated, the kernel between two examples can be evaluated by integrating the product of the two corresponding distributions (Jebara et al., 2004). If the number of states is finite, a probability product kernel based on multinomial frequencies can be defined. Given the data $\boldsymbol{x}$ and an attribute η_i, let $p_i(j)$ be the observed frequency of value j in $\eta_i(\boldsymbol{x})$, the dot product between two features is:

$$K_i(x, x') = \sum_{j=1}^{m_i} p_i(j)^\rho p_i'(j)^\rho \tag{6.5}$$

where m_i is the number of distinct values for η_i. Setting $\rho = 1/2$ results in a discrete version of the Bhattacharyya kernel. Histogram intersection kernels (Barla et al., 2003; Odone et al., 2005) can also be used to compute the kernel between two distributions. Histogram intersection kernels have been introduced in the context of image classification and builds on similar ideas, by replacing products with the minimum operator:

$$K_i(x, x') = \sum_{j=1}^{m_i} min\{p_i(j), p_i'(j)\} \tag{6.6}$$

For both kernels the contributions of multiple attributes can either be summed or multiplied:

$$K(x, x') = \prod_{i=1}^{n} \big(1 + K_i(x, x')\big) \tag{6.7}$$

$$K(x, x') = \sum_{i=1}^{n} K_i(x, x') \tag{6.8}$$

6.4 Fisher Kernels

Fisher kernels were first introduced by Jaakkola and Haussler (1999) in the context of remote homology detection. If a probability model of input data is provided, the Fisher vector $\phi(\boldsymbol{x})$ associated with the input $\boldsymbol{x}$ is defined as the gradient of the log-likelihood of $\boldsymbol{x}$ with respect to the model parameters

$\boldsymbol{\gamma}$:

$$\phi(\boldsymbol{x}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{x}; \boldsymbol{\gamma}) \tag{6.9}$$

The Fisher kernel between two instances $\boldsymbol{x}$ and $\boldsymbol{x}'$ is then defined as the dot product between their associated Fisher vectors $\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}')$.

The parameters $\boldsymbol{\gamma}$ are usually estimated using generative probabilistic models such as Hidden Markov Models (Rabiner, 1989) and the resulting $\ell(\boldsymbol{x}; \boldsymbol{\gamma})$ is the log-likelihood of $\boldsymbol{x}$ given the training set. Using HMMs a probability model can be defined for structured data such sequences and trees. Therefore, the Fisher kernel can be considered as the first example of a kernel function for structured data.

6.5 Decomposition Kernels

The class of decomposition kernels has been formalized by Haussler (1999) to compute dot products between structured data. An R-decomposition on a set $\mathcal{X}$ is a triple $\mathcal{R} = \langle \vec{\mathcal{X}}, R, \vec{K} \rangle$ where $\vec{\mathcal{X}} = \langle \mathcal{X}_1, \ldots, \mathcal{X}_D \rangle$ is a D tuple of non empty subsets of $\mathcal{X}$, R is a finite part-hood relation on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_D \times \mathcal{X}$, and $\vec{K} = \langle K_1, \ldots, K_D \rangle$ is a D tuple of positive definite kernel functions $K_d : \mathcal{X}_D \times \mathcal{X}_D \to \mathbb{R}$. $R(\vec{x}, \boldsymbol{x})$ is true iff $\vec{x}$ is a tuple of parts of $\boldsymbol{x}$, i.e. $\vec{x}$ is a *decomposition* of $\boldsymbol{x}$. For any $\boldsymbol{x} \in \mathcal{X}$, let $R^{-1}(\boldsymbol{x}) = \{\vec{x} \in \vec{\mathcal{X}} : R(\vec{x}, \boldsymbol{x})\}$ denote the set of all possible decompositions of $\boldsymbol{x}$. The decomposition kernel is then defined as the multi-set kernel (Gärtner et al., 2002) between the decompositions:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\substack{\vec{x} \in R^{-1}(\boldsymbol{x}) \\ \vec{x}' \in R^{-1}(\boldsymbol{x}')}} \prod_{d=1}^{D} K_d(x_d, x'_d) \tag{6.10}$$

One commonly used family of decomposition kernels consists of all-sub-structures kernels, which count the number of common sub-structures in two structured objects. In this case $D = 1$ and $\mathcal{R} = \langle \mathcal{X}, R, \delta \rangle$, where $R(x_1, \boldsymbol{x})$ is true iff x_1 is a substructure of $\boldsymbol{x}$, while δ is the exact matching kernel: $\delta(x_1, x'_1) = 1$ if $x_1 \equiv x'_1$ and 0 otherwise. The all-sub-structure kernel can be efficiently computed by performing the decomposition only implicitly, such to avoid the combinatorial explosion of $|R^{-1}(\boldsymbol{x})|$. Examples of such efficient kernels are the Spectrum Kernel (Leslie et al., 2002), that counts all the common k-mer in two input strings, and the kernel proposed in

Gärtner et al. (2003), that counts the common labels between all possible paths in two input graphs. Conversely, computing the match kernel between two sub-structure x_1 and x_1' may be difficult as it might require solving a subgraph isomorphism problem (Gärtner, 2003). All-sub-structure kernels may introduce sparseness in the feature representation of $\boldsymbol{x}$ in the sense that the parts of $\boldsymbol{x}$ significant for prediction are usually a small fraction of $R^{-1}(\boldsymbol{x})$. For large data structures the signal-to-noise ratio approaches zero and the mapping from inputs to labels cannot be learned. Therefore, it is important to carefully tune the decomposition R to different applications in order to characterize a suitable kernel for predictive tasks.

6.6 Weighted Decomposition Kernels

Weighted Decomposition Kernels (Menchetti et al., 2005) extend the definition of all-sub-structures kernels by weighting the match between identical parts (called here *selectors*) according to contextual information. A weighted decomposition kernel is characterized by a decomposition structure $\mathcal{R} = \langle \vec{\mathcal{X}}, R, \vec{K} \rangle$ where $\vec{\mathcal{X}} = \langle \mathcal{S}, \mathcal{X}_1, \ldots, \mathcal{X}_D \rangle$, $\vec{K} = \langle \delta, K_1, \ldots, K_D \rangle$, $R(s, x_1, \ldots, x_D, \boldsymbol{x})$ is true iff $s \in \mathcal{S}$ is a subgraph of $\boldsymbol{x}$ called the selector and $\vec{x} = \langle x_1, \ldots, x_D \rangle \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_D$ is a tuple of subgraphs of $\boldsymbol{x}$ called the *context* of occurrence of s in $\boldsymbol{x}$. This setting results in the following general form of the kernel:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\substack{(s, \vec{x}) \in R^{-1}(\boldsymbol{x}) \\ (s', \vec{x}') \in R^{-1}(\boldsymbol{x}')}} \delta(s, s') \prod_{d=1}^{D} K_d(x_d, x_d') \tag{6.11}$$

where δ is an exact matching kernel on $S \times S$ and K_d is a probability product kernel on $\mathcal{X}_d \times \mathcal{X}_d$. The direct sum between kernels over parts K_d can be replaced by the tensor product. Precise definitions of s and $\vec{x}$ are domain-dependent and will be described in the chapters devoted to experiments.

6.7 The Three-Dimensional Decomposition Kernel: A Novel Decomposition Kernel for Solid Structures

A three-dimensional structure is a special kind of relational data whose parts are related by their spatial distance. A graphical representation of a three-dimensional object is a completely connected graph with weighted edges. Typical methods for graph processing give more importance to graph topology than edges labels and would miss most of the information contained in the structure. Hence the need of a specialized method for processing data that represent solid objects. A novel decomposition kernel for solid structures is defined here: the structured object is decomposed into a set of overlapping solid parts with varied geometry, then two objects are compared by computing the similarity between their parts.

Let $\boldsymbol{x}$ be a solid structure with vertices V and edges E: each vertex $v_i = \langle t_i, \vec{z}_i \rangle \in V$ has a type $t_i \in T_V$ and a set of coordinates $\vec{z}_i \in \mathbb{R}^k$, while each edge $e_{jk} = \langle v_j, v_k, t_{jk} \rangle \in E$ has a type $t_{jk} \in T_E$ and express an additional relation (other than distance) between the vertices v_j and v_k. An R-decomposition of a solid structure is defined as a tuple $\mathcal{R} = \langle \mathcal{X}, R, K_s \rangle$, where $R(s, \boldsymbol{x})$ is true iff $s \in \mathcal{X}$ is a *shape* composed by vertices of $\boldsymbol{x}$ and K_s is a kernel between shapes. A shape $s = \langle V_s, E_s \rangle$ is itself a solid structure with $V_s \subseteq V$ and E_s being the complete set of edges such that $E_s = \{\forall v_j, v_k \in V_s : e_{jk} = \langle v_j, v_k, t^s_{jk}, l_{jk} \rangle\}$. Each edge $e_{jk} \in E_s$ has a associated length $l_{jk} = \|\vec{z}_j - \vec{z}_k\|$ and type $t^s_{j,k} = t_{jk}$ if $e_{jk} \in E$ or nil otherwise. The resulting decomposition kernel between two solid structures is:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\substack{s \in R^{-1}(\boldsymbol{x}) \\ s' \in R^{-1}(\boldsymbol{x}')}} K_s(s, s') \tag{6.12}$$

The kernel K_s between two shapes $s = \langle V_s, E_s \rangle$ and $s' = \langle V_{s'}, E_{s'} \rangle$ is defined as:

$$K_s(s, s') = \prod_{i=1}^{n} K_e(E^i_s, E^i_{s'}) \tag{6.13}$$

where $n = \max\{|V_s|, |V_{s'}|\}$, K_e is the kernel between edges, and E^i_s is the i-th ranked element of the edge set E_s if it exists, or a null element otherwise. The ranking of the edge set is performed using a suitable preference function:

let t_e^1 and t_e^2 the types of the two vertices connected by e with $t_e^1 \leq t_e^2$, and t_e be the type of the edge e,

$$e \prec e' \Leftrightarrow \quad (t_e^1 < t_{e'}^1) \vee (t_e^1 \equiv t_{e'}^1 \wedge t_e^2 < t_{e'}^2) \vee \\ \vee (t_e^1 \equiv t_{e'}^1 \wedge t_e^2 \equiv t_{e'}^2 \wedge t_e < t_{e'}) \tag{6.14}$$

Finally the kernel K_e between two edges is:

$$K_e(e, e') = \delta(t_e^1, t_{e'}^1)\ \delta(t_e^2, t_{e'}^2)\ \delta(t_e, t_{e'})\ e^{\sigma |l_e - l_{e'}|^2} \tag{6.15}$$

where l_e is the length of edge e and δ is the exact match kernel. The kernel K_e computed between a null edge and a valid edge is always 0, therefore the kernel between two shapes of different size is 0.

The kernel K_s between two shapes s and s' is a positive definite kernel, being a product of three exact match kernels and a Gaussian kernel. As mentioned in Sec. 6.5, the actual decomposition of a structured data in its parts must be tuned to the application. In this case, the set of shapes must be carefully chosen from the set 2^V of all possible combinations of vertices of the solid structure.

Part III

Prediction from Sequence

Chapter 7

Prediction of Protein Secondary Structure

Prediction of protein secondary structure is a classical problem in computational molecular biology and one of the first successful applications of machine learning to bioinformatics. In this task, given the amino-acid sequence of a protein of unknown three-dimensional structure the problem is to learn the local folding regularities formed and maintained by hydrogen bonds (Sec. 2.3.2). A supervised learning task is formulated as the association between an input sequence representing the protein primary structure and an output string that contains the secondary structure label at each residue.

Reliable methods to predict the secondary structure are fundamental for the estimation of protein conformation and function. Threading algorithms relies on accurate secondary structure predictions to find remote homologs and to align target sequences with templates (Sec. 2.5.2). Recognition of the folding family using the pattern of secondary structure is also a useful estimates of a protein functional category (Sec. 2.3.3).

A two-stages architecture for the prediction of secondary structure from amino-acid sequence is presented here[1]. A SVM is used to predict the secondary structure class of a single residue having as input a window of multiple alignment profiles (Sec. 2.3.1), while a BRNN is used to correlate and filter the predictions of the SVM classifier in consecutive positions of

[1]Results published in Ceroni et al. (2003a)

a protein sequence. In section 7.1 a brief overview of the most significant SS prediction methods is made. In section 7.2 the two-stages architecture is explained in details. In section 7.3 the preparation of the dataset is outlined, while the results of the experiments are discussed in section 7.4.

7.1 Prediction of Secondary Structure

The first attempt to apply machine learning techniques to the prediction of secondary structure (Qian and Sejnowski, 1988) employed a standard multilayer perceptron with a single hidden layer, and used as inputs a window of amino acids in one-hot code. The accuracy of this method, measured as the proportion of amino acids correctly assigned to one of the three secondary structure classes (three-state accuracy or Q_3), was well below 70%. Although the primary structure contains all the informations needed to define the native folding of a protein, the actual spatial configuration of every residue is influenced by the whole sequence of amino acids of the protein, therefore the same group of linked amino acids can appear in different conformations if its spatial context varies. Part of this problem can be overcome by looking at position conservation during evolution. The introduction of evolutionary information by multiple alignment profiles represented a major contribution to the field, allowing a significant improvement of the reported accuracy to about 72% (Rost and Sander, 1993b).

A major drawback of learning a SS classification from windows of profiles is the relative independence between the predictions of adjacent positions in the sequence, while the secondary structure of a protein is defined as a collection of segments composed by many consecutive amino acids. A measure of segment overlap (Zemla et al., 1999) is used to quantify the capability of a classifier to correctly predict entire segments of secondary structure. To improve both SOV and Q_3, Riis and Krogh (1996) and Jones (1999) used a structure-to-structure classifier to filter the predictions of the single-residue predictor. Feed-forward neural networks are used for both stages, and the filtering network is fed with a window of predictions output by the first stage. Thanks to this solution and to an increasing availability of training data, the architecture proposed by Jones (1999) achieves state-of-the-art performances with an accuracy of 78% and a SOV of 73.5%. A different approach has been presented by Baldi et al. (1999) and refined by Pollastri

et al. (2002c) which uses bidirectional recurrent neural networks trained with profiles as inputs. This architecture achieves results equivalent to the best feed-forward networks. Hua and Sun (2001) proposed the use of support vector machines instead of neural networks for secondary structure prediction from windows of multiple alignment profiles, claiming a higher value of SOV as compared to traditional single stage neural networks approaches, without a significant improvement in accuracy.

7.2 Architecture

Previous works have demonstrated that a classifier that is able to correlate outputs at different positions along the sequence is needed to increase the accuracy at predicting segments of secondary structure. Even if Hua and Sun (2001) claims differently, the experiments on a large dataset show (Tab. 7.1) that a standard single residue SVM classifier is not able by itself to reach a significantly different value of SOV than a classical NN approach.

Both approaches proposed by Jones (1999) and Pollastri et al. (2002c) have their drawbacks. In Jones (1999), a standard neural network is used as the first stage classifier, while in many classification task with high-dimensional input vectors SVMs have demonstrated superior performances. Moreover, NNs need a fixed input size, while the length of the sequence and the amount of information useful for predicting the class of a residue can vary in size. On the other side, large BRNNs with many weights are difficult to train and Pollastri et al. (2002c) was forced to adopt various tricks to improve the stability of the overall architecture such as: shortcuts connections between non-consecutive states, computation of outputs at each position in the sequence using a window of states, and ensembles of independent classifiers.

The architecture proposed here is a combination of the two approaches, where the local, single residue, first stage classifier is constituted by a multi-class SVM with multiple alignment profiles as input, while the filtering stage is made of a small BRNN with just three inputs trained on the SVM outputs.

7.2.1 Single Residue Classifier

Support vector machine learning (Cortes and Vapnik, 1995; Cristianini and Shawe-Taylor, 2000) is a mature approach for classification and regression

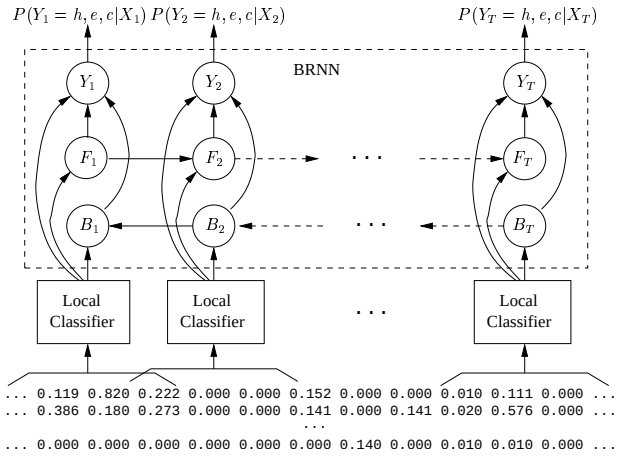


Figure 7.1. Two stages architecture. The local classifier is a support vector machine. The bidirectional recurrent neural network is unfolded over the chain.

and has been applied to several prediction problems in bioinformatics (Furey et al., 2000; Jaakkola et al., 2000; Hua and Sun, 2001; Leslie et al., 2002; Nair and Rost, 2005). While SVMs have been initially conceived for binary classification, different approaches exist in order to extend them to the multi-class case (Hsu and Lin, 2002). Here, a one-against-all approach is used. Each binary classifier is trained using windows of $w = 2k+1$ residues flanking the target amino acid, where each position in the window is represented by the corresponding profile of amino-acid frequencies. All the classifiers use Gaussian kernels to compute the dot product between instances. The optimal values for the coefficient of the Gaussian kernel γ, the regularization parameter C and the window size w are chosen by model selection. The results of the model search show a saturation in the performances for a window size $w = 15$. Even if SVMs are capable of dealing with high dimensional data augmenting the input window have the effect of increasing the quantity of noise more than the quantity of information.

The outputs of the three one-against-all classifiers are converted to conditional probabilities using an optimized soft-max to allow the application of the Viterbi decoder (Sec. 7.2.3). The method used here (Passerini et al., 2002) extends the algorithm presented by Platt (1999) for binary classifiers, where the mapping from margins to conditional probabilities is performed by a logistic function $P(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{Af(\boldsymbol{x})+B}}$ with offset B and slope

A. Parameters A and B are adjusted according to the maximum likelihood principle, assuming a Bernoulli model for the class variable. This is extended to the multi-class case by assuming a multinomial model and replacing the logistic function by a soft-max function (Bridle, 1989). More precisely, assuming Q classes, Q binary classifiers are trained according to the one-against-all output coding strategy. In this way, for each point $\boldsymbol{x}$ a vector $f_1(\boldsymbol{x}), \ldots, f_Q(\boldsymbol{x})$ of margins is obtained and then transformed into a vector of probabilities using the soft-max function:

$$P(y = q|\boldsymbol{x}) = \frac{e^{A_q f_q(\boldsymbol{x})+B_q}}{\sum_{r=1}^{Q} e^{A_r f_r(\boldsymbol{x})+B_r}}\,,\ q = 1, \ldots, Q. \tag{7.1}$$

The soft-max parameters $\theta_{sm} = \{\langle A_q, B_q\rangle\}_{q=1}^{Q}$ are determined as follows. Firstly, a new dataset $\mathcal{D}_{sm} = \{(f_1(\boldsymbol{x}_i), \ldots, f_Q(\boldsymbol{x}_i), \boldsymbol{z}_i), i = 1, \ldots, m\}$ of examples is defined whose input portion is a vector of Q margins and output portion is a vector $\boldsymbol{z}_i$ of indicator variables encoding (in one-hot form) one of the Q classes. As suggested by Platt for the two classes case, this dataset should be obtained either using a hold-out strategy, or a k-fold cross validation procedure. Secondly, the parameters θ_{sm} are optimized to maximize the log-likelihood function under a multinomial model:

$$\ell(\mathcal{D}_{sm}; \theta_{sm}) = \sum_i \sum_{q=1}^{Q} z_i^q \log P(y_i = q|\boldsymbol{x}_i) \tag{7.2}$$

where $z_i^q = 1$ if the i-th training example belongs to class q and $z_i^q = 0$ otherwise.

7.2.2 Filtering Classifier

BRNNs are recurrent neural networks with two set of states that are recursively copied forward and backward along the sequence (see Sec. 5.3). BRNNs can develop complex non-linear and non-causal dynamics that can be used to correct single-residue prediction by trying to capture valid segments of secondary structure. The output of a BRNN at each position of the sequence is a function of both the input at current position and the inputs at preceding and succeeding position through the content of the forward and backward states. However, the problem of vanishing gradients (Bengio et al., 1994) prevents from learning global dependencies, so it is impossible for the BRNN to model the whole conformation of the protein.

The filtering BRNN has three inputs for each position of the sequence, corresponding to the conditional probabilities calculated by the soft-max function for the first stage classifier (Figure 7.1). When cascading two learning machines it is necessary to feed the downstream machine (the BRNN in this case) with the predictions obtained from the upstream machine (the SVM), rather than using true values. Both the forward and backward recursive functions are implemented by neural networks with a single layer. The number of neurons in the state layers is set to 20 as chosen by model selection.

The BRNN cannot be trained using the margins calculated by the SVM on the training set otherwise the distribution of inputs would be biased since the margins are exactly -1 or +1 for all the bound support vectors. The problem is avoided by training the BRNN on data not in the SVM training set. In order to achieve this goal and at the same time not waste examples, the training set is split into three folds: the SVM is then trained on two folds and the predicted margins are obtained on the one left out. The operation is repeated three times so that all folds are used once for predicting the margins. BRNN weights are optimized by back-propagation with early stopping to control over-fitting.

7.2.3 Constraining Predictions using the Viterbi Decoder

In native protein structures, the segments of α helices and β strands are formed by specific patterns of hydrogen bonds between spatially close amino-acids. The labeling of secondary structure segments in folded structures is obtained by looking for such patterns. Therefore, a sequence of secondary structure labels extracted from a native conformation is forced by construction to follow certain rules:

- α helices must be at least 4 amino-acids long,
- β strands must be at least 2 amino-acids long.

Conversely, the sequences generated by the two-stages classifier frequently violates the previous rules. Even if the BRNN could learn such simple grammar from examples (Omlin and Giles, 1996), the error function used to train the network is defined only to maximize the multi-class accuracy while no penalty is used for violating these constraints.

Some other "propensities" of native structures observed in the dataset can be used to derive additional constraints that penalize interleaving of SS segments:

- a sequence must start and finish with a residue in coil conformation,
- between an α helix and a β strand (and vice-versa) there must be at least a residue in coil conformation.

All the previous facts can be expressed using a finite-state automaton which represents every possible allowed sequence in this minimal secondary structure grammar (Fig. 7.2). Now, the problem is how to output a sequence of predicted SS labels that can be expressed by this FSA.

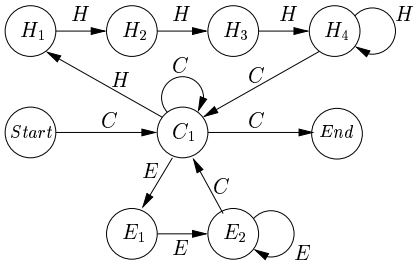


Figure 7.2. Finite-state automaton representing every possible allowed sequence of secondary structure.

The outputs of the SS classifier (both first and second stage) at each position t of the sequence can be interpreted as the probabilities $Pr(y_t = H|\boldsymbol{x}, t)$, $Pr(y_t = E|\boldsymbol{x}, t)$ and $Pr(y_t = C|\boldsymbol{x}, t)$ that amino acid in position t belongs to one of the three secondary structure classes given the input sequence $\boldsymbol{x}$ and the position t. A constraints satisfying method should output the most likely sequence of SS labels from the grammar defined by our FSA, using as likelihood its overall probability as estimated by the classifier predictions:

$$Pr(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{T} Pr(y_t|\boldsymbol{x}, t)\,,\ \boldsymbol{y} = \{y_1, \ldots, y_T\}\quad y_t \in \{H, E, C\}. \tag{7.3}$$

This task strictly resemble problem 2 of Hidden Markov Models (Rabiner, 1989): given the probabilities of observations and a state model of the data, the sequence of states with highest likelihood is requested. A finite-state automaton can be seen as a degenerated Hidden Markov Model where each state emits a single symbol with probability 1 and all the transitions have

the same probability (it can be set to 1 because there is no need the probabilities of all the transitions coming out of a state sum to 1). Therefore,

Algorithm 7.1 The Viterbi decoder.

Init the trellis:
for all $\langle s, t \rangle$ **do**
  $score(s, t) \leftarrow -\infty$
end for

Forward recursion:
$score(start, 0) \leftarrow 0$
for $t = 1$ to T **do**
  for all s_i **do**
    for all $s_j \in parents(s_i)$ **do**
      if $score(s_j, t-1) + \log Pr(symbol(s_i, s_j)|\boldsymbol{x}, t) > score(s_i, t)$ **then**
        $score(s_i, t) \leftarrow score(s_j, t-1) + \log Pr(symbol(s_i, s_j)|\boldsymbol{x}, t)$
        $last(s_i, t) \leftarrow s_j$
      end if
    end for
  end for
end for

Backward recursion:
$previous \leftarrow end$
for $t = T$ to 1 **do**
  $this \leftarrow previous$
  $previous \leftarrow last(this, t)$
  $y_t \leftarrow symbol(previous, this)$
end for
$\boldsymbol{y} \leftarrow \{y_1, \ldots, y_T\}$

the *Viterbi algorithm* can be used to align the model to the sequence using the probabilities of observations estimated by the classifier (Algorithm 7.1). The algorithm searches an optimal path on a trellis whose nodes are pairs $\langle s, t \rangle$, being s the corresponding state of the FSA and t the position in the sequence. Each node of the trellis has two attached variables: $score(s, t)$ is the score of the best sequence ending at this node, and $last(s, t)$ is the

preceding state in the best sequence ending at this node. Additionally, $symbol(s_i, s_j)$ is the symbol emitted during the transition from state s_i to s_j, and $parents(s)$ is the set of states which have a transition ending in s. Log-probabilities are used to avoid numerical problems. The score of the ending state of the sequence is the log-probability of the best sequence $\boldsymbol{y}$. This algorithm can be easily extended to a FSA with more than one starting state and more than one ending state. Moreover, the probabilities used to calculate the scores do not need to come from a particular type of classifier: either the predictions of the second stage BRNN or the predictions of the first stage SVM can be used.

7.3 Datasets

The experiments have been performed using a significant fraction of a representative set of non homologous chains extracted from Protein Data Bank (PDB Select, Hobohm and Sander (1994)). The sequences have been taken from the April 2002 release, listing 1779 chains with a percentage of homology lower than 25%. This set has been reduced by allowing only high quality structures determined by X-ray diffraction, without any physical chain breaks and resolution threshold lower than 2.5 Å. The final dataset contained 969 chains, about 180,000 amino acids, split in a training set of 490 chains, a validation set of 163 chains and a test set of 326 chains. Multiple alignments were generated running PSI-BLAST (Altschul et al., 1997) on each sequence using the Swiss-Prot+TrEMBL non-redundant database (Bairoch and Apweiler, 1999). Secondary structure labeling has been extracted from this set of native conformations using the DSSP program (Kabsch and Sander, 1983). DSSP labels H and E were used for the definition of α helix and β strand classes, while all the rest is in the coil class.

7.4 Results and Discussion

The performances of the two-stages architecture have been estimated using a 7-fold cross validation on the 969 proteins dataset. The values of Q_3 and SOV have been computed for each element of the architecture and are shown in Table 7.1. The results show that the two-stages architecture presented here is able to reach state-of-the-art values of accuracy and segment overlap.

The contribution of the BRNN to both measures of performance are decisive and demonstrates the efficacy of a non-local method to increase the segment overlap. Conversely, the SVM alone do not guarantee a high value of SOV differently to what has been claimed in Hua and Sun (2001).

Moreover, the experimental results demonstrates the Viterbi decoder is able to correct punctual errors in the predicted SS sequences and output longer segments to further increase the value of SOV. Richer finite-state automaton with constraints on secondary structure segments derived from known folds should be used to correct completely misclassified segments of secondary structure.

	Plain		VD	
Classifier	Q_3	SOV	Q_3	SOV
SVM	76.5%	68.9%	76.9%	73.5%
SVM+BRNN	77.9%	74.1%	78.0%	74.7%

Table 7.1. Results of the experiments for the various stages of the SS predictor

Chapter 8

Prediction of the Bonding State of Cysteines and Histidines

Accurate ab-initio methods for the prediction of metal-binding sites (see Sec. 3.2) would be of invaluable help to aid experimental structure determination and to estimate the function of a protein. The ultimate goal of structural genomics is to map the space of protein structures by providing an experimentally determined structure for each protein family, such that a resolved homolog is available for any known sequence. Interesting proteins have often little or no prior experimental knowledge. In this context, predicting the presence of a metal binding site in a target sequence could be crucial for determining the conditions under which that protein is to be expressed or purified (Sec. 2.4). Moreover, knowledge of the type of the bonded metal and of the residues forming the binding site could provide important clues about the protein function and could also be used to identify a specific functional family for which a representative structure is missing.

Predicting a metal binding site requires the identification of both the residues that bind a metal and the type of metal involved. This problem could be stated as a learning task in which for each residue a classification of the type of metal bonded (if any) is provided. Given the high number of metal and residue types possibly involved and the low number of bonded residues actually available in databases, this task is too difficult to solve

for a machine learning approach. Here, a restricted setting is adopted in which the bonding state is predicted only for cysteines and histidines and the types of metals are limited (see Sec. 8.1 for details). Disulfide bonding of cysteines is also taken into consideration and it is itself a relevant feature for the structural and functional characteristics of the protein. This work can be thought as an extension on previous works of cysteine bonding state prediction (Ceroni et al., 2003b; Passerini and Frasconi, 2004) by the addiction of histidine residues and a more complex architecture. A two-stages architecture (similar to the one presented in Chap. 7) is used to predict the bonding state as detailed in Sec. 8.2. In section 8.3 the preparation of the dataset is outlined, while the results of the experiments are discussed in section 8.4.

8.1 Prediction of Bonding State

Metal binding sites appear as very compact units in space and can be predicted with some success once the protein structure is known (Gregory et al., 1993; Sodhi et al., 2004; Schymkowitz et al., 2005). In contrast, predicting metal binding from sequence appear as a much more challenging task, because the residues involved in the ligand binding are usually spread all over the protein sequence with no clearly recognizable spacing rule (Harding, 2004). In practice, there is only a single published method so far that attempt to predict metal binding sites from sequence (Passerini and Frasconi, 2004).

In (Passerini and Frasconi, 2004) a SVM classifier is used to predict whether the cysteines in the target sequence are free, disulfide or metal bonded. This method is compared to a naive classifier that predict the metal binding sites for the same protein by searching for the presence of known metal-binding sequence motifs: PROSITE (Falquet et al., 2002; Hulo et al., 2004) patterns corresponding to recurrent motifs are extracted from the set of tagged proteins and compared to the target sequence. All the PROSITE patterns have very high specificity and very low coverage, therefore most of the metal-binding sites cannot be predicted by the naive method.

In this work, the general problem of identifying the binding sites in a protein is restricted by selecting a reduced set of metals and residue types for which the bonding state is predicted. The amino-acid composition of

binding sites for alkali and alkaline earth metal is the most diverse since the main chain carbonyls of all types of residues can be involved in the binding. In contrast, transition metals present a more limited spectrum of binding residues comprising, but for a few exceptions: cysteines, histidines, aspartic acids, asparagine, glutamic acids and glutamine. Most residues in this list are rarely found in metal binding sites, therefore learning to discriminate between bonded and free state for these amino-acids would be difficult. Following these considerations, the task has been reduced to the prediction of the bonding state of cysteines and histidines limited to the binding of transition metals, heme and iron sulfur clusters (this two iron complexes are pretty common and bind primarily to cysteines and histidines, Johnson et al. 2005; Paoli et al. 2002). The restrictions adopted here have theirs drawbacks, since the resulting predictor would not be able to identify all types of metal binding sites and, in cases for which the binding site included residues other than cysteines and histidines, it would be missing part of the information about the global architecture of the binding site. However, this is a reasonable approach when compared to the much greater difficulties that arise tackling metal binding site prediction in the most general case.

For cysteines residues the possibility of a disulfide bond is also taken into consideration. The knowledge of which cysteines in a sequence are actually bonded is a necessary step to the prediction of disulfide bridges. Disulfide bridges may link portions of a protein that are very distant in sequence, thus introducing long-ranged interactions and providing a very informative constraint on the conformational space (see Sec. 2.3.4). The prediction of disulfide bridges is therefore a first step towards a better understanding of structural properties of proteins and the solution of the folding problem. Prediction of cysteines disulfide bonding state has been tried by many different approaches (Fariselli et al., 1999; Fiser and Simon, 2000; Mucchielli-Giorgi et al., 2002; Martelli et al., 2002; Ceroni et al., 2003b; Passerini and Frasconi, 2004) even with high accuracy. However, the focus of this work is not on reaching top performances in this specific task, but on improving prediction of metal binding sites.

8.2 Architecture

In Passerini and Frasconi (2004) and Ceroni et al. (2003b) the individual SVM predictions are obtained *independently* for each target cysteine. While the independence assumption is reasonable for two residues that belong to different proteins, it is much less likely to hold for some of the residues of the same protein. Metal ions are typically coordinated by several residues, whose bonding states become inevitably correlated. In addition, the formation of disulfide bridges is often a global phenomenon associated with the whole chain, where two behaviors are dominant: either all or no cysteines are disulfide bonded (Fiser and Simon, 2000). In facts, if one cysteine is known to be free, then the probability that another cysteine in the same chain is also free would be increased. Ignoring these effects in the model can lead to loss of prediction accuracy.

The learning problem in presence of correlation between instances is often referred to as *collective classification* (Getoor et al., 2001) and has been tackled with some success using probabilistic graphical models. Presently, collective classification is not supported by SVM. An attempt to recover some of the data correlation is made by designing a hybrid solution based on the combination of SVM predictions for single residues and bidirectional recurrent neural networks for entire chains (Fig 8.1). This architecture is similar to the one described in Chap. 7 but, instead of producing an output for each residue of the sequence, the predictions are made only for cysteines and histidines.

8.2.1 Single Residue Classifier

The classifier used to predict the bonding state of single cysteines and histidines is a multi-class SVM that solves directly the multi-class optimization problem (Crammer and Singer, 2002). Input to the classifier is a window of size $2w + 1$ residues flanking the target amino acid, where each position is represented by the corresponding multiple alignment profile frequencies plus a flag indicating positions ranging out of the sequence limits. Overall, this accounted for $(2w + 1) * 21$ input features, to which the relative position of the residues with respect to the sequence length was added. Additionally, a global descriptor made of 33 numeric features is considered that takes into account the characteristics of the entire protein. The first 20 features de-

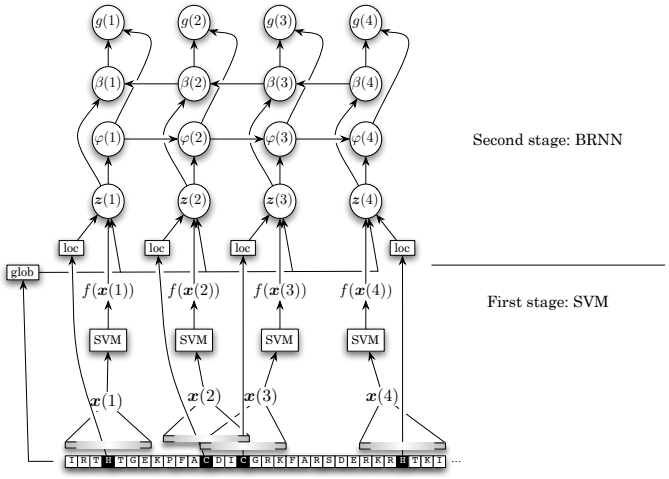


Figure 8.1. Architecture of the overall predictor. SVM predictions and other context information (boxes labeled "cont") form the input to the BRNN.

scribe the normalized amino acid composition of the chain, each entry being computed as $(N_i^j - \mu^j)/\sigma^j$ where N_i^j is the number of occurrences of the j-th amino acid type in the i-th chain, while μ^j and σ^j are the mean and the standard deviation of the number of occurrences of the j-th amino acid in the whole training set. More features included: sequence length relative to the average value in the training set, overall number of cysteines and overall number of histidines in the chain, both relative to their average values in the training set and to the sequence length itself. Finally, 10 features encoded for the average conservation of the cysteines and histidines in the chain. Conservation values in $[0, 1]$ were discretized into five equal length bins and a one hot encoding of the bin index was employed. A Gaussian kernel is used to compute the dot product between inputs.

8.2.2 Collective Classifier

In the sequence learning problem solved by BRNNs, data comes in the form of input-output sequence pairs. In the present application, sequences have as many elements as cysteines and histidines in the chain being considered. The input is formed by local predictions from the SVM classifier and additional information describing the local environment around the target residue and the environment that separates it from the successive cysteine or histidine, as detailed below.

The mapping between the input sequence and the output sequence in

a BRNN is *global* in the sense that the output at every position depends on the inputs at every other position through the content of forward and backward states. A soft-max function is used to guarantee that each output can be interpreted as a (conditional) probability that residue at that position is free, disulfide bonded or metal bonded (given the input sequence). The propagation scheme is graphically illustrated in the upper part of Figure 8.1.

While this scheme does not allow to model arbitrary correlations between the output targets at any two positions, it may still be useful in the present case since the bonding state at a given position is calculated not just using inputs for that position but also using inputs that may be separated in sequence and close in space. In this way, correlations between two outputs that are reflected into correlations between the corresponding inputs may be indirectly captured. Moreover, the number of cysteines and histidines in a chain is a order of magnitude lower than the total number of residues, therefore the problem of vanishing gradients (Bengio et al., 1994) should not prevent the BRNN from learning global relations as it happens in SS prediction (Sec. 7.2.2).

In this refinement stage, SVM predictions are enriched with additional features describing the target residue and its context (box `cont` in Figure 8.1). Residue features consisted of its conservation, discretized into the same five equal length bins used for the global descriptor (Sec. 8.2.1), its position relative to the sequence length, and a flag indicating the residue type (cysteine or histidine). Context features include gaps information and secondary structure predictions. Gaps information was intended to capture correlation between sequence separation and binding attitude. For each target residue, the sequence separation to the successive target is computed and then discretized into eight bins ([0], [1], [2], [3], $[4, 5]$, $[6, 19]$, $[20, 75]$, $[76, 200]$, $[201, \infty)$), where the last bin also includes the case when no successive target is present before then end of the chain. A one-hot encoding of the bins indices was employed as in Section 8.2.1. Nine real values encode the predicted secondary structure: the first three values represent predicted probabilities of the three SS classes for the target residue; the next three values encoded average SS predictions in a window of length five on both sides of the target residue; finally, average SS predictions for the context separating the residue from the successive target residue were included as

the last three values. Secondary structure predictions were obtained using the architecture detailed in Chap. 7.

The BRNN has been trained using the same strategy outlined in Sec. 7.2.2 to avoid a biased distribution of inputs in the training set for the examples corresponding to bound support vectors of the SVM classifier. During training the BRNN minimizes a standard cross-entropy cost function by using gradient descent with early stopping.

Ligand	CYS	HIS	ASP	GLU	MET	GLN	ASN
Zn	46 ($\frac{508}{1115}$)	24 ($\frac{374}{1562}$)	4 ($\frac{117}{3204}$)	2 ($\frac{89}{3705}$)	0 ($\frac{0}{1132}$)	0 ($\frac{9}{2047}$)	0 ($\frac{2}{2442}$)
Heme	50 ($\frac{115}{230}$)	34 ($\frac{151}{450}$)	1 ($\frac{5}{854}$)	0 ($\frac{2}{925}$)	6 ($\frac{23}{367}$)	1 ($\frac{7}{593}$)	0 ($\frac{1}{571}$)
FeS	63 ($\frac{205}{326}$)	3 ($\frac{10}{329}$)	0 ($\frac{0}{763}$)	0 ($\frac{1}{886}$)	0 ($\frac{0}{372}$)	0 ($\frac{0}{407}$)	0 ($\frac{0}{485}$)
Mg	0 ($\frac{0}{402}$)	1 ($\frac{10}{864}$)	4 ($\frac{97}{2209}$)	3 ($\frac{62}{2478}$)	0 ($\frac{0}{785}$)	1 ($\frac{10}{1451}$)	1 ($\frac{16}{1376}$)
Mn	1 ($\frac{1}{176}$)	10 ($\frac{36}{373}$)	7 ($\frac{60}{835}$)	4 ($\frac{39}{1056}$)	0 ($\frac{0}{296}$)	0 ($\frac{2}{459}$)	0 ($\frac{3}{604}$)
Cu	33 ($\frac{36}{108}$)	32 ($\frac{86}{269}$)	0 ($\frac{2}{513}$)	0 ($\frac{2}{455}$)	4 ($\frac{9}{228}$)	0 ($\frac{0}{251}$)	0 ($\frac{0}{391}$)
Cd	62 ($\frac{48}{77}$)	32 ($\frac{25}{79}$)	12 ($\frac{26}{216}$)	13 ($\frac{35}{262}$)	0 ($\frac{0}{44}$)	1 ($\frac{1}{158}$)	0 ($\frac{0}{176}$)
Fe	13 ($\frac{16}{122}$)	18 ($\frac{59}{325}$)	2 ($\frac{15}{610}$)	3 ($\frac{25}{745}$)	0 ($\frac{0}{189}$)	0 ($\frac{0}{364}$)	1 ($\frac{2}{381}$)
Ni	4 ($\frac{2}{46}$)	16 ($\frac{18}{112}$)	2 ($\frac{5}{250}$)	1 ($\frac{2}{271}$)	0 ($\frac{0}{100}$)	0 ($\frac{0}{132}$)	1 ($\frac{1}{153}$)
Co	0 ($\frac{0}{28}$)	17 ($\frac{7}{41}$)	7 ($\frac{6}{90}$)	5 ($\frac{4}{83}$)	0 ($\frac{0}{35}$)	0 ($\frac{0}{42}$)	0 ($\frac{0}{54}$)
Any	38 ($\frac{931}{2445}$)	19 ($\frac{776}{4045}$)	4 ($\frac{331}{8813}$)	3 ($\frac{261}{10093}$)	1 ($\frac{32}{3234}$)	1 ($\frac{29}{5504}$)	0 ($\frac{25}{6131}$)

Table 8.1. Percentage and fraction of times a given amino-acid type binds a specific metal ion (or complex) within chains containing a binding site for that ion. Metals are ordered by overall frequency of occurrence as binding cofactors.

8.3 Datasets

Protein sequences and structures are collected from the Protein Data Bank (Berman et al., 2002). A sequence-unique subset of proteins is obtained by running UniqueProt (Mika and Rost, 2003) on the list of primary structures arranged to have chains with metal and disulfide bonds at the beginning. This produced a total of 2,982 protein chains with no limitation to resolution quality and method. Multiple alignments are obtained by running one iteration of PSI-BLAST (Altschul et al., 1997) on each sequence using the non-redundant TrEMBL database (Bairoch and Apweiler, 1999) and an e-value cutoff of 0.005. Cysteines are labeled as disulfide bonded using the DSSP program (Kabsch and Sander, 1983) and considering both intra-chain

and inter-chain bonds. In an attempt at reducing noise in the data, a manually analysis is conducted for all cases in which two cysteines are found within a distance of 2.5 Å but were not labeled by DSSP as being in disulfide bridge. As a consequence, in 124 cases the DSSP labeling has been changed from free to disulfide.

Metal binding sites are detected by calculations performed directly on the protein 3D coordinate files. First, the files are parsed looking for the following metals and metal containing complexes: cadmium, cobalt, copper, heme, iron, iron/sulfur clusters, magnesium, manganese, nickel, zinc. Then, the ligands for each metal/complex are identified by considering a 3.0 Å cutoff as a maximal distance for contact between heavy atoms. For Fe/heme and Fe/S complexes, all ligands are considered, including those binding to the heme and sulfur atoms. In all cases, contacts with the protein backbone were not taken into consideration. Chains containing a single metal binding residue are discarded as outliers, under the arbitrary assumption that metals that bind to only one residue are unlikely to play any relevant structural or functional role in the protein. In fact, they could be artifacts resulting from the specific experimental protocol used for structure determination.

The list of all extracted metals and complexes along with the corresponding ligands is reported in Table 8.1. The only amino acids which occur with a reasonable frequency in metal binding sites are cysteines and histidines, therefore the other amino acids are ignored as candidate ligands as mentioned in Sec. 8.1. Magnesium, manganese and cobalt are further ignored as they too rarely bind cysteines or histidines. A total of 2,727 chains are left of which 1,648 contain only cysteines and histidines in their free state, 749 has a least one disulfide bridge and 383 had at least one metal binding site, with 53 chains having both a metal bindings site and a disulfide bridge (see Table 8.2).

	Free	DB	MBS
Cysteines	4593	3661	933
Histidines	12982	0	678
Chains	1648	749	383

Table 8.2. Number of cysteines, histidines and entire chains for the three different classes (MBS=Metal Binding Site, DB=Disulfide Bridge).

8.4 Results and Discussion

Performances were assessed by a stratified five fold cross validation procedure. The secondary structure predictor architecture was trained on the five folds in order to assure full independency of the test set with respect to the training data. Being the data set highly unbalanced (see table 8.2), accuracy (the fraction of correct predictions over the total number of predictions) can be a misleading parameter as it favors predictors biased towards the most populous class. In order to overcome such limitations, the area under the ROC curve (Bradley, 1997) has been used for both model selection and final performance measurement, with the simple multi-class extension proposed in Hand and Till (2001). Whenever a hard decision had to be done, as for confusion matrices and detailed recall values for binding site type and coordination number, each example has been assigned to the class achieving the maximum output. Model selection for binary and multi-class SVM hyper-parameters, namely Gaussian width and C regularization parameter, was conducted in a preliminary phase by running a three fold cross validation procedure on the training set of the first fold. However, the optimal values for hyper-parameters were very stable when varying the size of the input window. Therefore they have been kept fixed across folds and window sizes ($\gamma = 1e-2$,$C = 5e-1$ for binary SVM, $\gamma = 1e-2$,$C = 1e-1$ for multi-class SVM).

Predictor	SVM-BRNN					PROSITE				
Class	Pre	Rec	Free	DB	MBS	Pre	Rec	Free	DB	MBS
Free	94%	96%	16875	431	269	79%	99%	17440	67	68
DB	83%	83%	490	3036	135	83%	9%	3326	328	7
MBS	68%	55%	564	167	880	80%	19%	1309	0	302
Accuracy	91%					79%				

Table 8.3. Confusion matrices, recall precision for class and overall accuracies for the SVM-BRNN architecture and the PROSITE pattern based predictors. Confusion matrices have true class on rows, predicted class on columns.

The overall best results were obtained for a window of size $W = 15$, giving a multi-class AUC of 0.954 ± 0.003 where the confidence interval is the average standard error of the Wilcoxon-Mann-Whitney (Bradley, 1997) statistic. For sake of comparison, the baseline performance is obtained using

biological annotations in the form of PROSITE (Hulo et al., 2004) patterns (release 19.11 dated 27 September 2005), as described in Passerini and Frasconi (2004). A multi-class SVM predictor trained on PROSITE pattern matches achieves an AUC of merely 0.605 ± 0.007, thus proving the significance of the proposed approach. Table 8.3 reports confusion matrices and overall accuracies for the two methods. While precision/recall values are quite balanced for the two-stages architecture, the baseline predictor obtains high precision and very low recall for the DB and MBS classes, confirming the fact that PROSITE patterns are very specific but too sparse in order to produce a reasonable coverage of binding sites.

Figure 8.2 (left) reports recall-precision curves for disulfide bridge (DB) and metal binding site (MBS) predictions for the overall best predictor. While DB are much easier to predict with respect to MBS, the latter can still be predicted with 60% precision/recall at the break-even[1]. Figure 8.2 (right) reports recall-precision curves for MBS predictions for cysteines and histidines separately, showing how the former are far better predicted. As can be seen in Table 8.2, cysteines are much more common than histidines as binding residues, thus there are more positive examples to train on and the unbalancing with respect to negative examples is less critic.

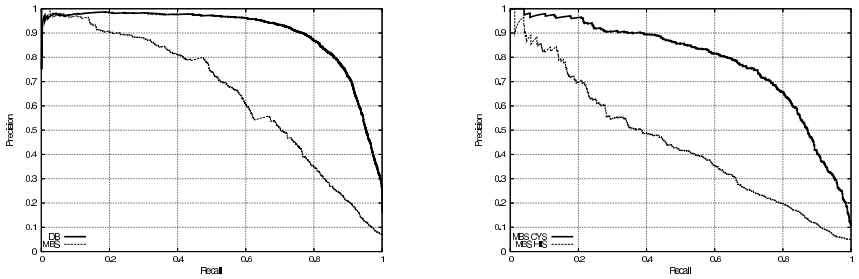


Figure 8.2. Recall precision curves for the best BRNN architecture ($W = 15$). Left: disulfide bridge (DB) and metal binding site (MBS) predictions. Right: metal binding site (MBS) predictions for CYS and HIS residues separately.

Table 8.4 reports recall values of MBS predictions separately for each ligand, ordered by frequency of occurrence. Iron sulphur and heme clusters are the easiest to predict, while single ions such as iron, cadmium and nickel

[1]The point in the recall precision curve where the two values coincide.

can be predicted with reasonable recall when binding cysteines, but are almost unpredictable whenever histidines are involved. Figure 8.3 (left) shows recall precision curves divided by ligand type, limited to the protein chains containing MBS of such type.

Ligand	CYS+HIS		CYS		HIS	
Zn_1^{2+}	0.53	(458/872)	0.69	(354/512)	0.29	(104/360)
HEME	0.67	(107/159)	0.93	(65/70)	0.47	(42/89)
Fe_4S_4	0.76	(99/130)	0.77	(98/128)	0.50	(1/2)
Cu_1^{2+}	0.26	(26/100)	0.35	(9/26)	0.23	(17/74)
HEME C	0.82	(66/80)	0.98	(43/44)	0.64	(23/36)
Cd_1^{2+}	0.35	(25/71)	0.52	(25/48)	0.00	(0/23)
Fe_2S_2	0.85	(44/52)	0.85	(39/46)	0.83	(5/6)
Fe_1^{3+}	0.40	(19/47)	0.94	(15/16)	0.13	(4/31)
Fe_1^{2+}	0.09	(2/22)	0.00	(0/0)	0.09	(2/22)
Ni_1^{2+}	0.06	(1/16)	0.50	(1/2)	0.00	(0/14)
Fe_3S_4	0.87	(13/15)	0.87	(13/15)	0.00	(0/0)
Cu_1^{1+}	0.58	(7/12)	0.75	(6/8)	0.25	(1/4)
$Fe_4Ni_1S_5$	0.33	(2/6)	0.40	(2/5)	0.00	(0/1)
Fe_8S_8	0.33	(2/6)	0.33	(2/6)	0.00	(0/0)
O_1Cu_2	0.33	(2/6)	0.00	(0/0)	0.33	(2/6)
$O_1Cl_1Fe_2$	0.00	(0/5)	0.00	(0/0)	0.00	(0/5)
$H_2O_4Fe_4S_6$	1.00	(4/4)	1.00	(4/4)	0.00	(0/0)
Cu_2^{3+}	0.75	(3/4)	1.00	(2/2)	0.50	(1/2)
$Fe_7Mo_1S_9$	0.00	(0/2)	0.00	(0/1)	0.00	(0/1)
SYROHEME	0.00	(0/1)	0.00	(0/0)	0.00	(0/1)
HEME D	0.00	(0/1)	0.00	(0/0)	0.00	(0/1)
Any	0.55	(880/1611)	0.73	(678/933)	0.30	(202/678)

Table 8.4. Recall values divided by ligand and ordered by ligand frequency, both overall and separate for CYS and HIS.

The prediction performance is also investigated as a function of the coordination number of binding ligands. Table 8.5 reports recall values of MBS predictions divided by coordination number. As expected, ligands which coordinate with a single residue are virtually always lost, and recall increases with growing coordination number, covering up to 68% and 71% of residues binding tetra and penta coordinated ligands respectively. Recall drops down

for the highest coordination numbers, as they include rare and anomalous cofactors such as O_1Cu_2, $Fe_4Ni_1S_5$ and $O_1Cl_1Fe_2$.

Coordination	1	2	3	4	5	6	7
Recall	0.03	0.19	0.31	0.68	0.71	0.44	0.27
	$\left(\frac{1}{33}\right)$	$\left(\frac{28}{150}\right)$	$\left(\frac{102}{333}\right)$	$\left(\frac{699}{1023}\right)$	$\left(\frac{24}{34}\right)$	$\left(\frac{12}{27}\right)$	$\left(\frac{3}{11}\right)$

Table 8.5. Recall values divided by coordination numbers of metal binding sites.

The performance of the proposed method in detecting metalloproteins is also measured. A single prediction is obtained for each protein by choosing the maximum MBS output between those of the residues contained in the chain. Figure 8.3 (left) reports the recall precision curve for such a task, showing a behavior very similar the one obtained at a residue level (see Figure 8.2 (left)).

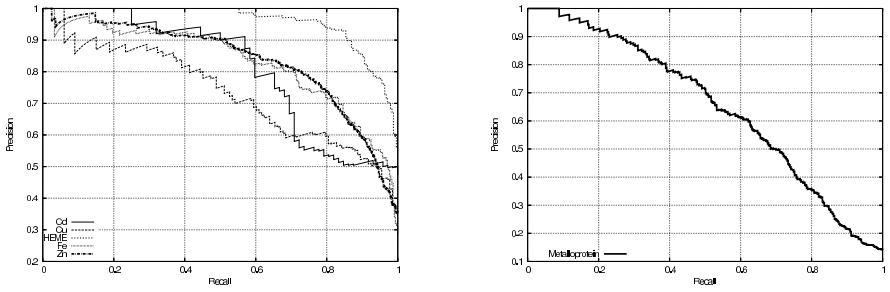


Figure 8.3. Left: recall precision curves divided by ligand type, limited to chains containing MBS of that type. Right: recall precision curve for metalloprotein prediction.

Finally, the ability of the method to generalize over metals is tested, that is the ability to predict a MBS for metal types it had not been trained on. To this extent a *leave-metal-out* procedure is used, which consists in dividing the cofactors by metal type, and for each of them, training the architecture on all the remaining types, and testing it on the left out type. Figure 8.4 (top-left) reports curves for different ligand types, while the other figures compare results for each ligand type with those obtained by training also on ligands of that type (see Figure 8.3 (left)). While it is clear that the method is able to generalize over ligand types, Cd, Cu and HEME suffer more for

missing explicit training data of their binding sites, while Fe and Zn seem more interchangeable.

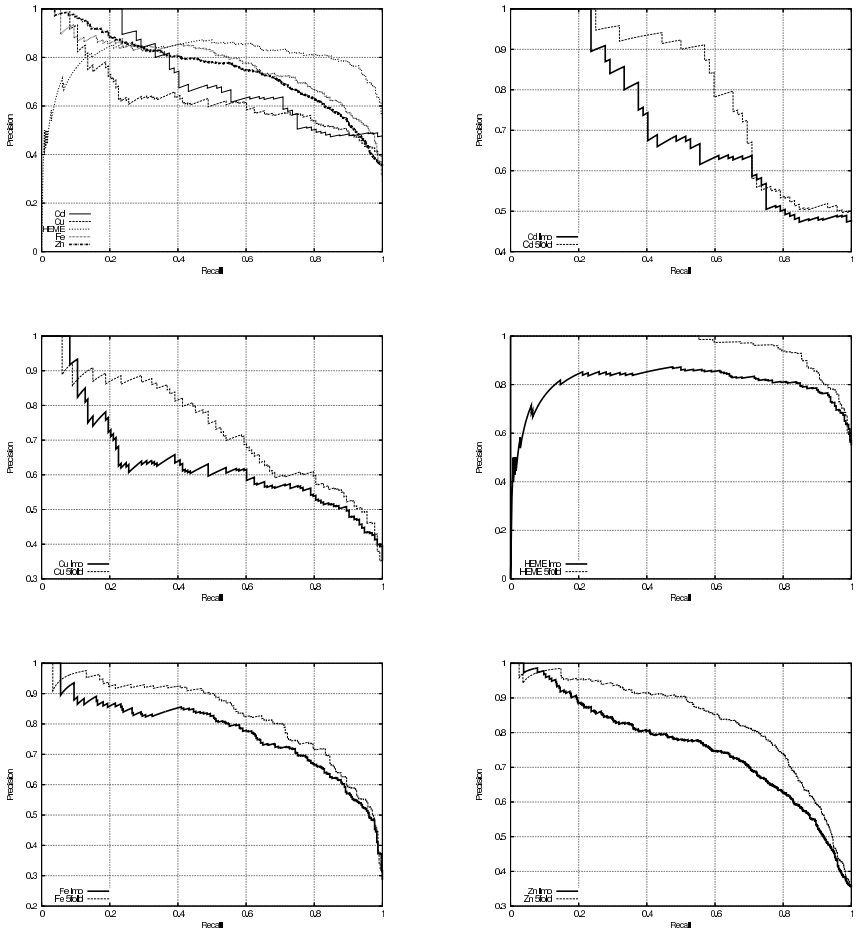


Figure 8.4. Leave-metal-out recall precision curves; each curve is computed by training on the binding sites of all metals except the target one, and testing on the target metal binding sites. The top left figure reports curves for different ligand types, while the other figures compare results for each ligand type with those obtained by training also on ligands of that type (the standard 5-fold cross validation procedure).

Part IV

Prediction from Contacts

Chapter 9

Learning Secondary Structure from Sequential and Relational Data

Long range interactions, i.e. interactions between residues that are distant in sequence but near in space, play an important role in the formation of secondary structures in folded proteins. For example, a β-sheet is composed of two or more strands that are held close by hydrogen bonds but that can be situated very far apart in the primary structure of the protein. Window-based classifier for SS prediction (Rost and Sander, 1993a; Riis and Krogh, 1996; Jones, 1999; Cuff and Barton, 2000) output predictions influenced only by the local context of the target residue and cannot take into consideration long range interactions. In Chap. 7 a two-stages architecture for the prediction of secondary structure has been presented that merge a window-based single-residue SVM classifier with a BRNN that correlate the SVM outputs at successive positions along the sequence. This architecture is inspired by works of Baldi et al. (1999) and Pollastri et al. (2002c) with BRNNs for SS prediction. The output at any position in a BRNN depends on the entire input sequence and thus a BRNN might in principle exploit long-range information, however well known problems of vanishing gradients (Bengio et al., 1994; Hochreiter et al., 2001) do not allow the model to *learn* these dependencies.

If the learner had explicit access to information about interacting posi-

tion pairs its task would be greatly simplified and it could possibly succeed. Interaction data can be extracted from contact maps (Sec. 9.1) and represented in the form of an *interaction graph* whose vertices are sequence positions and whose edges are interacting pairs. In order to operate in this extended setting, a learning algorithm that is able to exploit relational information other than the serial order is needed. Here [1], Interaction Enriched BRNNs (see Sec. 5.4) are applied to this extended SS prediction task. The use of this architecture may help towards improving the present state-of-the-art and elucidate in a quantitative way the importance of long-range information in the prediction of SS. Experiments are run using interaction graphs derived both from known protein structures and from predicted contact maps (Sec. 9.4). Although not useful in practice for constructing a prediction tool, the results from the first set of experiments show that IEBRNNs can effectively exploit relational information. These results also provide an empirical upper bound on the prediction accuracy that could be reached in this task by a learner that can access complete long-range information about the data. Moreover, the second set of experiments demonstrate that the proposed learning framework may be successfully exploited in a realistic setting.

9.1 Prediction of Secondary structure From Contact Maps

In the case of SS prediction, a reasonable source of information about long-range interaction can be obtained from contact maps, a graphical representation of the spatial neighborhood relation among amino-acids. An edge c_{ij} in a contact map indicates that the distance between the C_α atoms of the residues at positions i and j is lower than a predefined threshold (see Fig. 9.1). Contact maps at a coarser (e.g. secondary structure segment) or finer (e.g. atomic) level are also possible. In this work we choose residue contact maps because they provide an accurate enough picture to yield correct 3D structures (Vendruscolo et al., 1997), yet remain computationally tractable.

There are various reasons why using contact maps information in order to predict SS is worth investigation:

- Algorithms that reconstruct structures from contact maps are based on

[1] Results published in Ceroni et al. (2005b)

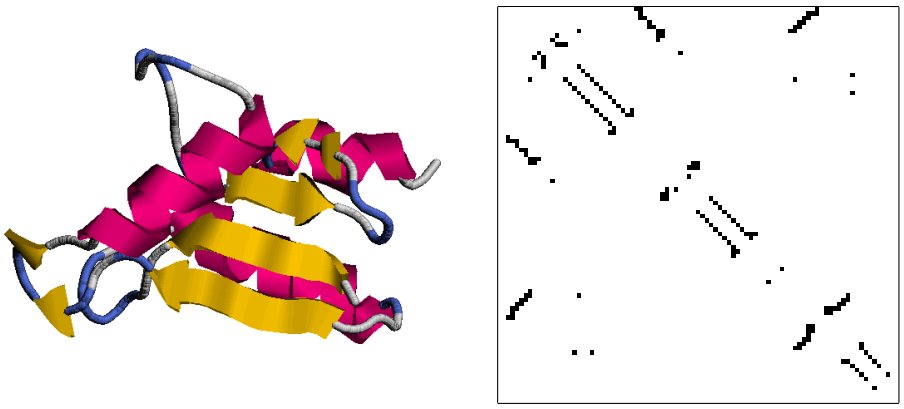


Figure 9.1. 3D structure and contact map at 6 Å of Glutaredoxin (PDB code 1ABA). Contacts between residues that are closer than 3 positions in sequences are omitted.

the definition of a potential energy function whose global minimization is not straightforward and requires stochastic optimization techniques to escape local minima (Vendruscolo et al., 1997). Thus it is not clear that a supervised learning algorithm can actually *learn* to recover SS from contact maps.

- Contact maps can be predicted from sequence (Fariselli et al., 2001; Pollastri and Baldi, 2002; Pollastri et al., 2003) or can be obtained from structures predicted by ab-initio methods (Bradley et al., 2003). Although accuracy of present methods is not yet sufficient to provide a satisfactory solution to the folding problem, predicted maps may still contain useful information to improve the prediction of lower order properties such as the SS.

- Even if contact maps are given, the design of a learning algorithm that can fully exploit their information content is not straightforward. For example, Meiler and Baker (2003) have shown that SS prediction can be improved by using information about inter-residue distances. Their architecture is a feed-forward network fed by *average* property profiles associated with amino-acids that are near in space to the target position. In this way, relative ordering among neighbors in the contact map is lost.

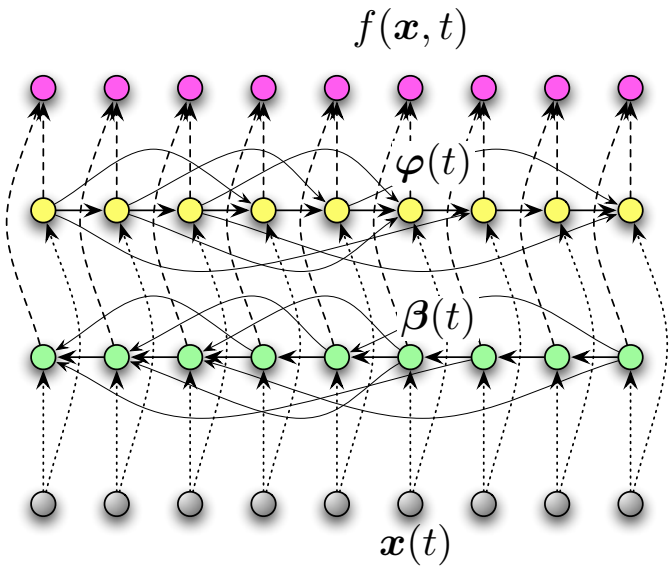


Figure 9.2. Graphical representation of the message passing in the IEBRNN architecture.

Information from contacts and sequence can be conveniently represented in the form of a graph whose vertices are sequence positions and whose edges are interacting pairs. The prediction of secondary structure from this structured data can be figured as a problem of learning a IO-isomorph map from a input graph $\boldsymbol{x}$ containing residues to a output graph $\boldsymbol{y}$ with the same topology and SS labels in each node. Since sequential interaction is a symmetric relation, we need methods that operate on *undirected* graphs to learn the mapping from inputs data to output labels. Here Interaction Enriched BRNN (see Sec. 5.4) are used. IEBRNN are a specialization of the class of Contextual RNNs built to process relational input data in the form of sequences with long range interactions. Similarly to BRNN, IEBRNN has two states vector for each node that contains information about preceding and succeeding positions along the sequence. On the other side, IEBRNN can exploit direct relations between non-adjacent sequence positions. In Fig. 9.2 is shown an example of information flow in an unfolded IEBRNN.

9.2 Prediction of Contact Maps

Interactions are extracted from predicted backbone coordinates, instead of resorting to contact maps predicted directly, for two reasons: maps extracted

from structures are generally more reliable, as they are physically realized; while direct predicted maps come with a fixed contact threshold (often 6 Å, 8 Å, 10 Å or 12Å), maps at any threshold can be extracted from structures. As ab-initio predictors are rarely publicly available, and when so they come with restrictions [2], Distill (Pollastri, 2004) is used here as an in-house fully automated prediction system. Distill has two main components: a set of predictors of protein features (secondary structure, relative solvent accessibility, residue contact maps, contact maps between secondary structure elements) and an optimization algorithm that searches the space of protein backbones.

Protein structural features are predicted by enhanced versions of the state-of-the-art SSpro (Pollastri et al., 2002c), ACCpro (Pollastri et al., 2002a), CMAPpro (Pollastri and Baldi, 2002) and CCMAPpro (Baldi and Pollastri, 2003) servers. The optimization is carried out by minimizing a simple potential function containing terms derived from the predicted features and terms representing geometrical constraints of the structure. Terms are present that penalize the violation of predicted residue contacts/non-contacts, predicted contacts/non-contacts between secondary structure elements, predicted strand locations; hard-core repulsion between amino-acids, and virtual C_α-C_α bond lengths. The actual minimization is performed in 3 stages: firstly, a set of initial structures is generated; secondly, a search is performed from each initial guess, giving rise to a number of refined structures; finally the resulting structures are ranked. In the initial guesses, predicted helices and strands are modeled: consecutive C_α atoms are set at a realistic distance (3.8 ± 0.2Å) and virtual C_α angles are restricted to the 90-180 interval. Each chain is grown from the N terminus to the C terminus by randomly selecting the next C_α with uniform distribution in the allowed space. From these initial guesses, a stochastic optimization is performed by introducing perturbations in the structure similar to "crankshaft" moves (Vendruscolo et al., 1997), except that helices are treated as rigid "rods" and their core C_αs are never moved on their own. The search is carried out by simulated annealing with a linear schedule for the temperature. 20,000 moves of every non-helical C_α and helical termini are attempted for each search. Ten searches are run for each protein structure. From these ten

[2] for instance Rosetta (Baker and Sali, 2001) explicitly disallows predictions of proteins of known structure, which would have made our experiments impossible

models, a consensus contact map is extracted that contains only those contacts that are present in all the reconstructions.

9.3 Datasets

The experiments have been performed using a representative set of non homologous chains from the Protein Data Bank (PDB Select (Hobohm and Sander, 1994)). The December 2002 release has been used, listing 1,950 chains with a percentage of homology lower than 25%. Only high quality proteins were retained on which the DSSP program does not crash, determined only by X-ray diffraction, without any physical chain breaks and resolution threshold lower than 2.5 Å. The final dataset contained 811 chains (137,926 residues) split in a training set R of 374 chains (68,074 residues), a validation set V of 197 chains (35,181 residues) and a test set T of 240 chains (34,671 residues).

In all the experiments, the input sequences consisted of profiles derived from multiple alignments generated using PSI-BLAST (Altschul et al., 1997) applied to the Swiss-Prot+TrEMBL non-redundant database (Bairoch and Apweiler, 1999). SS labels were assigned using the DSSP program (Kabsch and Sander, 1983): the DSSP eight classes were reduced into the three main classes by mapping H into α (helices), E into β (strands), and the rest (B, C, G, I, S, T) into γ (coils). Contacts were defined using a threshold of 6 Å, a value which comprises all the hydrogen bonded pairs and few side-chain contacts. Amino-acids spaced less than 3 positions in the sequence were not considered, because their C_α atoms are always at a distance lower than 6 Å.

9.4 Experiments

IEBRNN are used as a method for simplifying sequence learning tasks with neural networks by incorporating explicit knowledge about interactions between sequence elements. Empirical results show that this approach can greatly improve prediction of protein secondary structure when interaction knowledge is available. In addition, in this problem IEBRNNs are able to capture the serial order relation within interactions, a property that leads to better predictions than simply averaging the input at interacting positions in a sequence, as proposed in Meiler and Baker (2003).

The method described here does not yet advance the state-of-the-art in SS prediction but enlightens possible future directions of investigation. These findings provides further evidence that the knowledge of a relatively small number of reliable interactions (such as contacts between residues in helices and strands) is sufficient with the currently available data to obtain high quality predictions.

9.4.1 Prediction from Sequence alone

A first set of experiments was performed to obtain a baseline prediction accuracy for the SS problem on this dataset. In this work the main focus is on highlighting the contributions of long-range information to the prediction of SS rather than in obtaining state-of-the-art performances from the classifier. For this reason a "bare bones" BRNN classifier with multiple alignment profiles as input was used for this purpose. All common tricks that can be used to boost accuracy (in particular large training sets, shortcuts connections between non-consecutive states, cascaded predictors, computation of outputs at each position in the sequence using a window of states, and ensembles of independent classifiers, as in (Baldi et al., 1999; Pollastri et al., 2002c; Pollastri and McLysaght, 2005)) were omitted.

The network size, together with the parameters of the learning algorithm, were chosen using the validation set: an architecture with $d = 20$ neurons for each recursive state was then selected, the learning rate was set fixed to $1 \cdot 10^{-4}$ and an early stopping procedure was used to avoid over-fitting. The same parameters and sizes were used in all the experiments reported here and in the following. The validation set was also used for the early-stopping procedure.

Results of the baseline experiment are reported in the upper-left corner of Table 9.1. The corresponding confusion matrix is shown in the upper-left corner of Table 9.2. In this matrix, entry at row i and column j is the number of residues that are assigned to class i and belong to class j, divided by the number of residues assigned to class i. Precision for each class is marked in boldface and recall is reported separately in the last row.

9.4.2 Prediction from Contact Maps alone

These experiments are made to estimate the amount of information about the SS the IEBRNN architecture is capable to learn from contacts alone,

BRNN			IEBRNN		
Profiles only (baseline)	Q_3	74.6% ± 0.4%	Interactions only	Q_3	79.9% ± 0.4%
	SOV	66.7% ± 1.8%		SOV	73.3% ± 1.6%
Profiles + context	Q_3	82.5% ± 0.4%	Profiles + interactions	Q_3	84.6% ± 0.3%
	SOV	77.4% ± 1.5%		SOV	79.3% ± 1.5%
Profiles + context (no γ)	Q_3	95.9% ± 0.2%	Profiles + interactions (no γ)	Q_3	97.9% ± 0.1%
	SOV	94.6% ± 1.0%		SOV	95.5% ± 0.8%

Table 9.1. Prediction accuracy (Q_3) and segment overlap (SOV) along with 95% confidence intervals. Interaction graphs are obtained in this case from true protein structures.

therefore null inputs were used, i.e. $x(t) = 0$ for each position t. Information on contacts was inserted in the form of an interaction graph, as explained in Sec. 9.1. As reported in Table 9.1, the results are $Q_3 \approx 80\%$ and SOV $\approx 73\%$. These values are interesting considered that the classifier has no knowledge about the three-dimensional conformation of the hydrogen bonded atoms and the physics behind the formation of SS.

Comparing the predictions of this classifier and those of the classifier described above in Sec. 9.4.1, they overlap only for the 69% of the residues in the test set. Moreover, 92% of the times at least one of the two classifiers makes the correct prediction. It is then arguable that BRNNs trained on profile sequences and IEBRNNs trained on contact maps alone capture rather different regularities in the data. This suggests that training the IEBRNN with both inputs should allow us to obtain improved accuracy.

9.4.3 Prediction from Profiles and Contacts together

In this experiment, the IEBRNN is trained with both profiles and contacts as input. For sake of comparison with the results of Meiler and Baker (2003)

BRNN				IEBRNN			
Profiles only (baseline)				Interactions only			
	α	β	γ	α	β	γ
α	**77.8%**	5.8%	16.4%	**83.8%**	0.8%	15.5%
β	8.2%	**70.9%**	20.9%	1.1%	**76.5%**	22.4%
γ	11.6%	14.5%	**74.0%**	9.8%	11.4%	**78.8%**
Recall	*77.4%*	*58.9%*	*80.2%*	*85.2%*	*75.3%*	*78.4%*
Profiles + context				Profiles + interactions			
	α	β	γ	α	β	γ
α	**85.1%**	2.6%	12.2%	**88.6%**	0.7%	10.7%
β	4.5%	**76.0%**	19.5%	1.2%	**81.9%**	16.9%
γ	6.7%	9.6%	**83.7%**	8.0%	8.9%	**83.1%**
Recall	*87.6%*	*76.5%*	*81.8%*	*87.6%*	*80.1%*	*84.6%*
Profiles + context (no γ)				Profiles + inter. (no γ)			
	α	β	γ	α	β	γ
α	**93.4%**	6.5%	0.1%	**98.3%**	1.6%	0.1%
β	6.4%	**93.4%**	0.1%	2.4%	**97.5%**	0.1%
γ	0.2%	0.9%	**98.9%**	1.2%	0.9%	**97.9%**
Recall	*95.5%*	*88.3%*	*99.9%*	*96.6%*	*95.8%*	*99.9%*

Table 9.2. Confusion matrices for the different methods described in the paper. Interaction graphs are obtained from true protein structures.

a standard BRNN is trained with the same kind of inputs they used. In particular, the spatial *context* of each residue was computed by averaging the profile of the amino-acids in a sphere of 6 Å centered on the residue itself. This additional input was then given to the standard BRNN together with the usual profile. This method can be seen as a simplified version of the IEBRNN in which all contacts provide the same contribution to a given position. In particular, order among contacts cannot be distinguished.

As can be seen from Table 9.1, the information about contact ordering is efficiently exploited by the IEBRNN, which appreciably outperforms the solution based on the average context. Interestingly, prediction accuracy improves for helices and strands but not for coils (see confusion matrices in Table 9.2).

9.4.4 Effects of Interaction Robustness

The results of the above experiments show that IEBRNNs can effectively exploit the information contained in the contact map to improve prediction accuracy. However there is still about 15% residual error rate that would be interesting to explain. A possible explanation is that the reliability of the interactions that were injected as an additional input may play a significant role. In facts, edges in a contact map express spatial proximity but do not necessarily imply dependencies between the two close residues. This may be particularly true in the case of contacts that involve coil residues. Instead, contacts between residues that both belong to helices or strands can be expected to encode interactions in a more robust way since they are often maintained by hydrogen bonds.

In order to evaluate the effect of interaction robustness the experiments has been repeated using more sparse contact maps where edges only connect residues that belong to helices or strands. In so doing, about 60% of the edges have been removed from interaction graphs. Results are reported in the last row of Table 9.1 both for the standard BRNN (fed by profiles and context) and for the IEBRNN. The error reduction obtained in this way is dramatic. The residual error is comparable to the disagreement between different SS assignment programs (i.e. DSSP, STRIDE, and DEFINE). These experiments could indicate that part of the information contained in the contact map can be misleading, especially for the shorter segments.

9.4.5 Integration with Contact Map Predictor

A final set of experiments has been performed to estimate the accuracy of the SS predictor on a realistic prediction setting in which only the primary structure is known. In this case, the interaction graphs were derived from the contact map predicted by the system described in Sec. 9.2. The contact map predictor was fed with the secondary structure information output by the BRNN architecture described in Sec. 9.4.1. Therefore, the composition of BRNN, contact map predictor, and IEBRNN can be thought of as a single secondary structure predictor working on primary structure only as input. The data used to test the IEBRNN are independent from those used to train the BRNN and the contact map predictor, to avoid overestimation of the generalization performances of the IEBRNN.

IEBRNN

Profiles + predicted interactions	Q_3	76.0% ± 0.5%	Profiles + near predicted interactions	Q_3	75.7% ± 0.5%
	SOV	71.0% ± 1.8%		SOV	71.5% ± 1.8%
Profiles + true interactions	Q_3	77.4% ± 0.5%	Profiles + near true interactions	Q_3	79.4% ± 0.4%
	SOV	72.6% ± 1.7%		SOV	73.6% ± 1.6%

Table 9.3. Prediction accuracy (Q_3) and segment overlap (SOV) along with 95% confidence intervals. Networks are trained on interaction graphs obtained from predicted contact maps. Trained networks are tested both on predicted and true interaction graphs. Results in the right column were obtained by keeping only local interactions (distance between 3 and 7 positions).

Results are presented in Tables 9.3 and 9.4 and should be compared to the corresponding values in Tables 9.1 and 9.2. As expected, the accuracy obtained from sequences and predicted interaction graphs degrades compared to the results obtained when using true contact maps. Nonetheless, compared to prediction from sequence alone a statistically significant increase can be observed.

Tables 9.3 and 9.4 show also the value of accuracy of the same networks

trained on predicted interaction graphs when tested using true contact maps. Results demonstrate that the IEBRNN can effectively exploit even uncertain contacts information during training and could perform even better if the quality of predicted contact maps improved.

IEBRNN

Profiles + predicted interactions

	α	β	γ
α	**80.9%**	5.5%	13.6%
β	6.3%	**76.5%**	7.2%
γ	11.5%	15.6%	**72.9%**
Recall	*77.8%*	*57.6%*	*84.1%*

Profiles + near pred. inter.

α	β	γ
80.2%	6.2%	13.6%
7.5%	**74.3%**	18.2%
10.9%	15.7%	**73.4%**
78.2%	*57.0%*	*83.5%*

Profiles + true interactions

	α	β	γ
α	**80.5%**	5.5%	14.0%
β	4.6%	**78.7%**	16.7%
γ	9.9%	15.0%	**75.1%**
Recall	*81.7%*	*59.7%*	*83.5%*

Profiles + near true inter.

α	β	γ
85.8%	1.4%	12.8%
3.4%	**78.5%**	18.1%
7.7%	16.7%	**76.6%**
86.0%	*61.9%*	*83.7%*

Table 9.4. Confusion matrices obtained from networks trained on predicted interaction graphs. Trained networks are tested both on predicted and true interaction graphs. Results in the matrices on the right were obtained by keeping only local interactions (distance between 3 and 7 positions).

In a final set of experiments the distant contacts were removed. The right columns of Tables 9.3 and 9.4 report accuracies and confusion matrices obtained by keeping only predicted near interactions (distance in sequence between 3 and 7 positions). Unexpectedly, the performances increased, probably due to a higher confidence of the network on near contacts which usually are more accurately predicted.

Chapter 10

Protein Structure Assembly from Knowledge of Secondary Structure and β-sheet Motifs

Prediction of protein contact maps has been tried as a method for ab-initio determination of three-dimensional structure (see Sec. 4.3). Unfortunately, prediction of contact maps is still very unreliable and it is not clear whether the type of errors made by the predictor can be corrected by the reconstruction method. In an attempt to train more accurate predictors, a low-detail representation of protein conformation could be used to extract high level structural information. Prediction of coarse-grained contact maps with contacts defined among secondary structure segments has been recently tried (Vullo and Frasconi, 2003), but yet there exists no clear result concerning feasibility of reconstruction using only coarse information.

In this work [1], the analysis of the reconstruction method from coarse maps is restricted to a specific class of proteins, i.e. those that are are mainly characterized by residues in strand conformation. The quality of reconstruction for this kind of proteins can be enhanced by the knowledge of secondary structure and the indication of which strands are partners. The focus here is on contacts defined on β-partners, as the geometry and con-

[1] Results published in Ceroni and Frasconi (2004)

nectivity of β-strands imposes strong constraints on the overall structure of the protein. In Sec. 10.1 an efficient procedure is proposed to find a structure that matches the aforementioned characteristics of a given protein in its native conformation. In order to fully automate the structure prediction, in Sec. 10.2 it is also described an approach for predicting β-sheet motifs from sequence. Prediction of the arrangement of β-sheets is modelled as a learning task and solved by a Contextual Recursive Neural Network model (see Sec. 5.2). Finally, in Sec. 10.3 the outcome of experiments are reported and show encouraging results in both directions.

10.1 Backbone Reconstruction Algorithm

The reconstruction procedure performs the energy minimization of a reduced protein model, where knowledge about secondary structure and β-partners in the native conformation is enforced as a set of constraints on candidate solutions. The protein model comprises all backbone heavy atoms plus a single atom for the C_β to represent side chain occupation. Free parameters of this model are the dihedral ϕ and ψ angles formed by main chains atoms (see Fig. 10.1). The ω angle is set fixed to 180°, while bond lengths and angles are set to their average values calculated on the whole PDB dataset (Berman et al., 2002), the same for the coordinates of the C_β atom in the reference system defined by the N and C_α atoms.

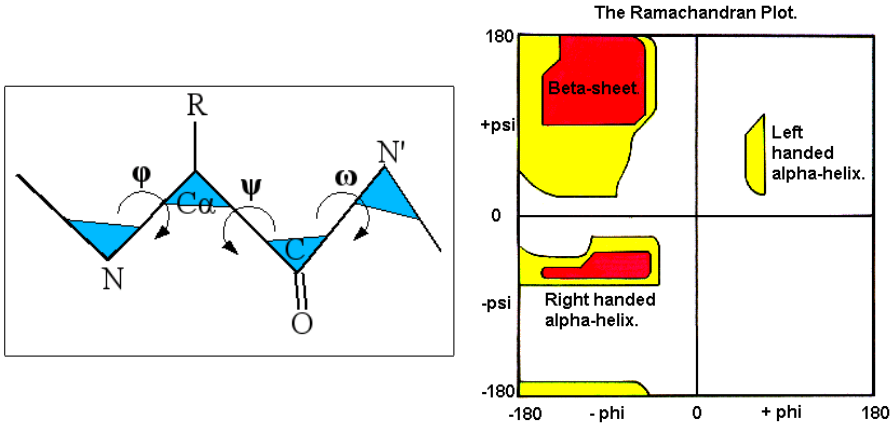


Figure 10.1. Left: definition of main-chain bond angles and dihedral angles; right: Ramachandran plot of dihedral angles with allowed regions as produced by atomic collisions

10.1.1 Constraints on Protein Structure

Secondary structure information is represented as a set of constraints imposed on the values of the dihedral angles: α-helices (H) and β-strands (E) correspond to two compact regions in the $\phi - \psi$ plot (see Fig. 10.1). Hence, for every residue in the H and E classes, the distance between its coordinates in the $\langle \phi, \psi \rangle$ space and the center of the corresponding region is forced to be lower than a specified threshold:

$$\| \langle \phi, \psi \rangle - \langle \phi_s, \psi_s \rangle \| \leq t_s \ , \quad s \in \{H, E\} \tag{10.1}$$

For each pair of β-strands, an indication if they are partners is provided, and in this case if they are either parallel or anti-parallel. The geometry of two β- partners forces the hydrogen-bonded residues to stay at a specific distance. Unfortunately, two partner strands can be of different dimensions, but here it has been chosen to not specify the partnership in terms of connectivity between single residues, because this would define an unrealistic setting for a predictor. Let I and J the sequences of indices of residues in two β-strands, with I^k and J^k two subsequences of size k of the residues indices. An alignment with parallel orientation is the set $\{\langle I_1^k, J_1^k \rangle, \ldots, \langle I_k^k, J_k^k \rangle\}$, while an alignment with anti-parallel orientation is the set $\{\langle I_1^k, J_k^k \rangle, \langle I_1^k, J_{k-1}^k \rangle, \ldots, \langle I_k^k, J_1^k \rangle\}$. The procedure tests all possible alignments for each pair of strands: (1) for partner strands, given a particular alignment, the distance between every pair of (supposedly) bonded atoms must be in a strict range of values

$$\forall_{i \in 1,k} : \ d_{min}^b \leq \| \ \vec{z}(I_i^k) - \vec{z}(J_i^k) \ \| \leq d_{max}^b \tag{10.2}$$

and the alignment that violates less constraints contributes to the solution; this enforces the existence of at least one good alignment between partners; (2) for non-partner strands both orientations are tested; given a particular alignment, the distances between paired atoms must be greater than a specified value

$$\forall_{i \in 1,k} : \| \ \vec{z}(I_i^k) - \vec{z}(J_i^k) \ \| > d_{min}^{nb}. \tag{10.3}$$

and the alignment that violates more constraints contributes to the solution; no good alignments must exist between non-partners.

Atomic forces impose a lower bound on the distance between two atoms, thus defining an excluded volume for each atom that prevents the protein

to collapse in a single point. These constraints are introduced in the procedure by forcing the distance between all pairs of atoms to be higher than a specified threshold:

$$\forall_{i,j\in 1,n} : \parallel \vec{z}(i) - \vec{z}(j) \parallel > d^{ex}_{min} \tag{10.4}$$

10.1.2 Optimization

In order to simplify the optimization task, all the constraints are expressed as quadratic penalty terms:

$$d_{min} \leq d \leq d_{max} \rightarrow \begin{cases} (d - d_{min})^2 & d < d_{min} \\ (d - d_{max})^2 & d > d_{max} \\ 0 & \text{otherwise} \end{cases} \tag{10.5}$$

Unfortunately, this gives rise to a highly non linear function of the model free parameters. Global optimization of non-linear cost functions is generally a difficult task. The approach followed in this work is constituted by a quasi-newton local optimization procedures (LBFGS, Liu and Nocedal 1989) coupled with a multi-start strategy. To mitigate the problem of local minima, a specific protocol is followed during optimization: firstly, the cost function is optimized only with secondary structure constraints, so that β-strands are formed in the backbone; secondly, the constraints for β-partners are added, relaxing the constraints on secondary structure so that a matching conformation is more easily found; finally, the constraints on atomic volumes are added. These are considered only at the end because they can prevent parts of the backbone to reach final positions that could be obstructed by other pieces of the chain.

10.2 Prediction of β-sheet Motifs

As shown in the previous section, the reconstruction procedure needs a bunch of important ingredients. Firstly, secondary structure hence the location of β-strands in the amino acid sequence must be known. Secondly, given the secondary structure, the information concerning the arrangement of β-sheets in the protein must be provided. All these information are obtainable either from the PDB files provided the structure is known, or from predictions. Clearly, the former case is of limited interest and it is considered here to analyze upper bound performance of reconstruction (see Sec.

10.3.1). In this work, the secondary structure is assumed to be available (for instance through the approaches proposed in Chaps. 7 and 9) and the focus is on the more difficult task of prediction of β-sheet configurations.

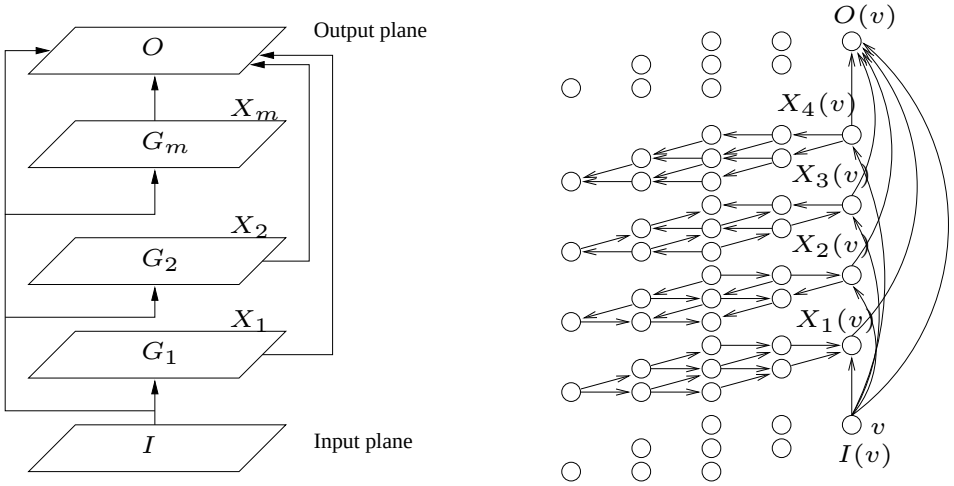


Figure 10.2. (left): Contextual RNNs, dependencies among input, state and output variables. (right): processing of grids with contextual RNNs (only a subset of connections are shown).

The β-sheets motifs inference problem is modelled as a classification task with multiple classes. Given a protein with a set $S = \{s_1, s_2, \ldots, s_n\}$ of strands, sheet configurations can be derived by mapping each pair of strands (s_i, s_j) defined on this set to one of three possible labels, say 0 (no hydrogen bonds between s_i and s_j), -1 (s_i and s_j are anti-parallel) and 1 (parallel strands). In this work, protein β-sheets motifs are modelled as global objects representing compact descriptions of a set of neighborhood relations, i.e. the connectivity matrix defined on S with elements $c_{ij} \in \{0, -1, 1\}$. Modelling β-sheets motifs in this way naturally gives rise to more complex structured, i.e. graphical, representations than simple fixed size attribute-value pairs. The main advantage of using structured data is the possibility to encode intrinsic dependencies among atomic entities allowing powerful learning algorithms to be employed. Here we model connectivity matrices as two-dimensional (square) undirected lattices, where nodes correspond to pairs of β-strands and edges connect adjacent pairs. Vertex output labels encodes for the spatial relation between the corresponding strands and are predicted with an approach resembling those of Pollastri and Baldi (2002)

and Pollastri et al. (2003), where Contextual Recursive Neural Networks are used to predict contact maps defined at the amino acid or segment level. Contextual architectures overcome the assumption of causality that is required with other related architectures. A detailed explanation of the CRNN model is given in Sec. 5.2. An example of information flow in a CRNN for prediction of matrices is given in Fig. 10.2.

10.3 Experiments

Experiments were performed using a representative set of non homologous chains from the Protein Data Bank (PDBSelect, december 2002). From this set only high quality proteins were retained without any physical chain breaks. The secondary structure class for the remaining chains was determined using the same procedure adopted for the CATH database (Orengo et al., 1997). The final dataset contained only *mainly-β* proteins, for a total of 154 chains whose sequence length is between 30 and 300 residues.

10.3.1 Reconstruction from True and Predicted Motifs

A first set of experiments is made to test whether the reconstruction procedure is able to reproduce β-sheet motifs, i.e. connectivity matrices of real protein structures. Accuracy is measured as the proportion of pairs of β-strands correctly assigned as partners or non-partners. The average value obtained has been 98.5%, with 74% of test proteins with all β-partners correctly assigned. Therefore, a second set of experiments is performed to test whether knowledge of β-sheet motifs is sufficient to reconstruct protein native conformations with good quality. Two measures of quality are used: the RMSD calculated on the C_α atoms for all the amino-acids in strands and the GDT_TS measure adopted in the CASP contest (Zemla et al., 2001). This results in an average RMSD value of 7.55 Å, and an average GDT_TS of 29.7. The distribution of those measures in the whole data-set is shown in Figs. 10.3 and 10.4. The same test is then performed on the β-sheet motifs as predicted from the recursive model. In this case, the average number of correctly assigned β-strands pairs dropped to 75%, since the predictor is likely to produce unrealistic structures. The average value of the RMSD became 16 Å, while the average value of GDT_TS was 24.

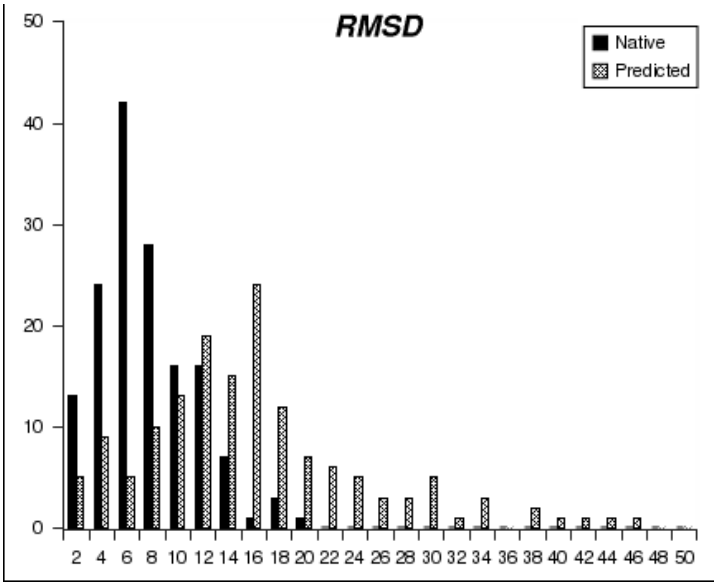


Figure 10.3. Histograms showing the distribution of RMSD on the set of reconstructed structures using native and predicted β-sheets topologies.

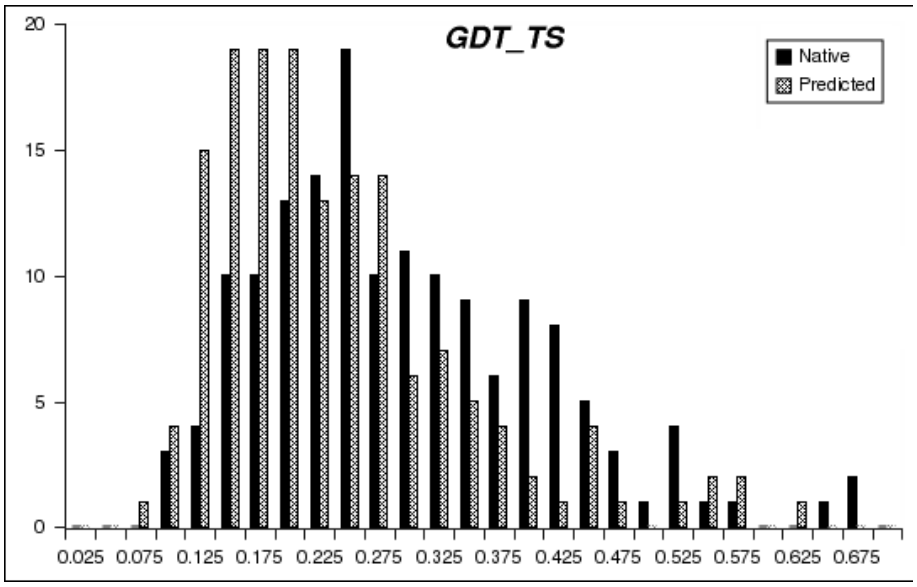


Figure 10.4. Histograms showing the distribution of GDT_TS (right) on the set of reconstructed structures using native and predicted β-sheets topologies.

10.3.2 Prediction of β-sheet Motifs

Together with contextual RNNs for grids, a multi-layered feed-forward neural networks (FF-NNs) is used to predict the class of contact for the $\langle s_i, s_j \rangle$ pair of β-strands, i.e. $P(c_{ij}|x_{ij})$, where c_{ij} can take 3 possible values (-1,0,1) and x_{ij} is the input associated to the pair $\langle s_i, s_j \rangle$. In either cases, input was represented by merging an attribute-value representation for the i-th and j-th strand in the sequence. Segment representations were derived by averaging multiple sequence alignment profiles, created with PSI-BLAST (Altschul et al., 1997), along the amino acid positions in the segment. Segments were finally described with 23-dimensional feature vectors by additionally including the relative index of one strand together with its normalized start and end amino acid positions with respect to N-terminus. In these experiments, a 5-fold cross validation procedure is used where, in order to control overfitting , for each fold part of the available training data is retained as validation set: learning is stopped when the minimum error over this set is achieved. The results of the cross-validation are shown in Table 10.1.

	Micro Avg.			
Method	Q_3	Q_{ap}	Q_{nb}	Q_p
Contextual RNNs	80.5±.9	39.1±3.0	89.1±.8	7.1±4.2
FF-NNs	85.1±.8	24.2±2.6	97.2±.4	0.0±.0

	Macro Avg.			
Method	Q_3	Q_{ap}	Q_{nb}	Q_p
Contextual RNNs	77.0±6.9	45.5±7.9	82.8±6.0	5.0±4.5
FF-NNs	79.3±6.3	29.2±7.2	94.0±3.8	0.0±.0

Table 10.1. Experimental comparison of contextual RNNs and feed-forward neural nets for the problem of predicting β-strands pairings (sec.10.2). Prediction indices as defined in sec.10.3.2.

Several indices are reported together with their 95% confidence intervals: micro- and macro-averaged global classification accuracy (Q_3) and accuracy of prediction of anti-parallel (Q_{ap}), parallel (Q_p) and unpaired strands (Q_{nb}). In either cases, the index Q_3 indicates consistently higher performance than a baseline approach: predicting the class randomly, but with the frequencies observed in the training sets would lead to an expected 72.6% correct class

prediction. A trivial predictor always assigning a pair of strands to the more numerous class (non-contact) would achieve 84.0% (clearly, no contact is predicted and Q_{ap} and Q_p would be 0). It clearly results that CRNNs predict anti-parallel and parallel strands consistently better than simple non-recursive nets. The latter model tends to predict the more numerous class in the majority of cases and then it shows better global results (Q_3). Therefore, the recursive model learned more than simple first-order statistics and it allows flexible integration of contextual information over ranges that exceed what can be achieved with fixed-input neural nets. Finally, the values of Q_{ap} and Q_p give a measure of the difficulty of the problem, the major obstacle being the unbalance among the classes: anti-parallel and parallel pairs represent 13.99% (resp. 1.96%) of the total set of pairs.

10.3.3 Discussion

Experimental results demonstrates how the proposed approach is able to build protein models matching the available characteristics of a native conformation. The proposed algorithm is inherently fast – reconstruction takes on average 20 minutes on standard workstations – because it is based on an efficient local optimization procedure combined with a simple multi-start strategy. Differently from other de-novo methods, this approach can then be tested over a large non-redundant set and statistically significant quality measures were obtained. Moreover, reconstruction of β-sheets topology does not require fine-grained information about the form of contacts between single residues. Therefore, the proposed algorithm can be used even if there are incomplete information about the native structure, e.g. during NMR modelling. Unfortunately, the algorithm did not prove to be sufficiently reliable to correct the errors of the β-motifs predictor. This led to a substantial decrease in the quality of reconstruction compared to the previous case. However, there are no similar experiments being conducted before, so this can be considered as a first step toward a complete reconstruction procedure based on coarse-grained information alone.

Part V

Prediction from Coordinates

Chapter 11

Supervised Scoring of Protein Models using Kernels on Statistical Potentials

State-of-the-art algorithms for the prediction of protein structures often produce multiple candidate three-dimensional models (see Sec. 2.5). These ensembles may contain very good structures and the primary difficulty is to discriminate correctly folded models (near-native structures) from incorrectly folded ones (decoys). Theoretically, the estimation of the free energy associated with a protein structure is a natural solution for ranking alternative candidate structures, since the native structure of a protein P is thought to correspond to the lower energy state accessible in equilibrium. In general, however, the real energy function cannot be evaluated and one needs to resort to an approximating scoring function $U(\boldsymbol{x}; M)$ defined on a conformation $\boldsymbol{x}$ according to an underlying statistical model M. In so doing, the predicted native structure can be retrieved as the $\arg\min_{\boldsymbol{x} \in \mathcal{X}_P} U(\boldsymbol{x}; M)$, being $\mathcal{X}_P$ the conformation space of P.

Over the past 20 years, a wide variety of scoring functions have been published in the literature. By far the most popular and largely successful approach is based on collecting statistics about pairwise distances between residues (Sippl, 1990; Simons et al., 1999; Miyazawa and Jernigan, 1999;

Domingues et al., 1999). Interestingly (see details in Sec. 11.1), the energy model based on statistical potentials can be naturally interpreted as a generative probabilistic model trained by maximum likelihood on the set of available native structures. Generative models are usually trained by unsupervised learning algorithms and do not make use of "negative" examples. In the present context this means that decoys generated by structure prediction algorithms are not taken into account.

In this paper a novel alternative technique is presented for deriving a suitable scoring function from data by means of a *discriminant* learning process (Sec. 11.2). Being informed about the bias due to the specific prediction algorithm in exploring the conformation space, supervised (discriminant) learning can be expected to produce more accurate solutions. The algorithm proposed in this work is based on support vector machines (SVM) trained on a preference learning task where decoys should receive lower preference scores when compared to good or native structures. To measure the similarity between structures, alternative functions are proposed for computing the kernel between probability distributions derived from pair potentials (Sec. 11.3). The kernels proposed here are based on the idea that a probability distribution can be naturally associated to the energy model of a protein structure according to Boltzmann law. In this way, the kernels will compare two conformations by measuring the contributions of interacting pairs to energy potentials. In Sec. 11.4 the results of empirical evaluation of this method on a data set of 266 proteins are finally reported.

11.1 Learning from Natives with Statistical Potentials

In this section the knowledge-based statistical potentials approach is briefly reviewed in terms of maximum likelihood unsupervised learning, a formulation that is suitable for subsequent derivation of the kernels to be used in conjunction with supervised learning algorithms. The statistical model proposed in Sippl (1990) is defined by a set of probability densities in which the range of distance values is split into S intervals $[h_{s-1}, h_s), s = 1, \ldots, S$, with $h_0 = 0$ and $h_S = \infty$. Specifically, the pair distribution $\Pr(s|a, b, k)$ is the conditional probability that a pair of residues has distance in $[h_{s-1}, h_s)$, given that the two amino acids have type a and b, and they are separated by

k positions along the sequence. The relationship between energy potentials $E_k^{ab}(s)$ and pair distributions is expressed by Boltzmann law:

$$\Pr(s|a,b,k) = \frac{e^{-\frac{E_k^{ab}(s)}{kT}}}{Z_k^{ab}(s)} \tag{11.1}$$

with normalization factor $Z_k^{ab}(s) = \sum_s e^{-E_k^{ab}(s)/kT}$. By assuming $Z_k^{ab}(s) \approx Z_k(s)$ for all a, b, the net potentials $\Delta E_k^{ab}(s) = E_k^{ab}(s) - E_k(s)$ can be approximated as

$$\Delta E_k^{ab}(s) \approx -kT \ln \left(\frac{\Pr(s|a,b,k)}{\Pr(s|k)} \right) \tag{11.2}$$

Given an amino acid sequence $a_1, \ldots, a_L$, a conformation $\boldsymbol{x}$ consists of a set of C_α atomic coordinates vectors $\vec{z}_1, \ldots, \vec{z}_L$. Let $d_{ij} = \|\vec{z}_i - \vec{z}_j\|$ the distance between residues i and j. According to pair potentials model M_{pp} the score of $\boldsymbol{x}$ is defined as its total energy:

$$U(\boldsymbol{x}; M_{pp}) = \Delta E(\boldsymbol{x}) = \sum_i \sum_{j>i} \Delta E_{|i-j|}^{a_i a_j}(d_{ij}) \tag{11.3}$$

where, as a notational convenience, $\Delta E_k^{ab}(d)$ stand for $\Delta E_k^{ab}(s)$ if $d \in [h_{s-1}, h_s)$.

For a given type $\langle a, b, k \rangle$ the distances are associated with a discrete set of intervals and therefore follow a multinomial distribution:

$$\Pr(d|a,b,k) = \prod_s \left(\theta_{ks}^{ab} \right)^{z(d,s)} \tag{11.4}$$

where $z(d, s) = 1$ iff $d \in [h_{s-1}, h_s)$ is an indicator variable mapping the distances to their intervals, and the model parameters simply coincide with the pairwise densities, i.e. $\theta_{ks}^{ab} = \Pr(s|a,b,k)$. The contribution of a structure $\boldsymbol{x}$ to the log-likelihood can then be calculated by combining Eqs 11.3 and 11.2:

$$\ell(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_i \sum_{j>i} \sum_s z(d_{ij}, s) \ln \left(\theta_{|i-j|,s}^{a_i b_j} \right) \tag{11.5}$$

Assuming a data set $\mathcal{D}$ of independent structures, the overall log likelihood function is simply $\ell(\mathcal{D}; \boldsymbol{\theta}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \ell(\boldsymbol{x}; \boldsymbol{\theta})$.

Sufficient statistics for the parameters θ_{ks}^{ab} are collected from a training set $\mathcal{D}$ of known native structures. Specifically, they consists of the counts m_{ks}^{ab} of the number of occurrences of pairs of type $\langle a, b, k \rangle$ at distance interval s in the training set:

$$m_{ks}^{ab} = \sum_{\boldsymbol{x} \in \mathcal{D}} m_{ks}^{ab}(\boldsymbol{x}) = \sum_{\boldsymbol{x} \in \mathcal{D}} \sum_i \sum_{j>i} \sum_s z(d_{ij}(\boldsymbol{x}), s) \tag{11.6}$$

As observed in Sippl (1990), in case of small data sets the normalized sufficient statistics may yield poor parameter's estimates because of data sparseness. One possible remedy in this case consists of estimating each density as a convex sum of a simplified model and the full model. In the simplified model, amino acid types are neglected and parameters reduce to θ_{ks}, which can be reliably estimated from sufficient statistics m_{ks}. The smoothed parameters are then computed as

$$\hat{\theta}^{ab}_{ks} = \frac{1}{1+m\sigma}\hat{\theta}_{ks} + \frac{m\sigma}{1+m\sigma}\frac{m^{ab}_{ks}}{\sum_s m^{ab}_{ks}} \tag{11.7}$$

where σ is a fixed smoothing factor (here the value $\frac{1}{50}$ is used as suggested by Sippl 1990) and m is the total number of observed pairs. When m is small, the mixing factor favors the simplified model which can be reliably estimated from a smaller number of observations.

11.2 Learning from Decoys with Ordinal Regression

Due to the impossibility of a sufficient sampling of the conformation space, it is not feasible to learn a scoring function that is guaranteed to take its minimum value on the native structure. However, a suitable scoring function can be specifically built to discriminate between alternative conformations generated by a particular threading or "de-novo" structure prediction algorithm. In this case, the space of decoys is reduced by the bias of the structure prediction algorithm, a bias that can be exploited by the learning process.

This task could be defined as a problem of learning to discriminate between "good" and "bad" structural models, where the quality of a model is decided by a measure provided by the user (e.g. a structural measure of similarity with the native conformation). This could be called the binary classification model M_{bc} in which the label $y^i_j = 1$ if the j-th conformation $\boldsymbol{x}^i_j$ is a good model for the i-th chain in the training set and $y^i_j = -1$ otherwise. A kernel machine would compute a real function of the conformation as a weighted average of the training set labels as follows:

$$U(\boldsymbol{x}; M_{bc}) = f(\boldsymbol{x}; \boldsymbol{\theta}_{bc}) = \sum_{i,j} \alpha^i_j K(\boldsymbol{x}^i_j, \boldsymbol{x}) y^i_j \tag{11.8}$$

where the weight of each example consists of a coefficient α_j^i measuring the importance of the example as calculated by some optimization procedure (e.g. SVM) and a kernel function $K(\boldsymbol{x}_j^i, \boldsymbol{x})$ measuring the similarity between $\boldsymbol{x}_j^i$ and $\boldsymbol{x}$.

Such a discriminative approach could find it difficult to estimate the decision function, because the resulting dataset would be highly unbalanced and contain only a small fraction of good models. Moreover, a continuous measure of quality should be unnaturally transformed in a discrete number of classes of structural models. Because of these considerations, in this work the search of a suitable scoring function is formulated as an ordinal regression task. In this formulation the training set consists of ordered pairs of alternative conformations for proteins of known structure. A preference relation $\prec$ is used to distinguish between two conformations. The preference model is then trained to learn either a partial or a total ordering between alternative conformations of a protein.

Herbrich et al. (1999) proposed a large margin classifier to learn preference relations in the above setting. When learning to score protein models the examples are tuples $\{\langle y_{jk}^i, \boldsymbol{x}_j^i, \boldsymbol{x}_k^i \rangle\}$, where $\boldsymbol{x}_j^i$ and $\boldsymbol{x}_k^i$ are two alternative conformations of protein i, and $y_{jk}^i = 1$ if $\boldsymbol{x}_j^i \prec \boldsymbol{x}_k^i$ or -1 otherwise. The labels y_{jk}^i can be determined for each training pair according to an evaluation measure that takes into account the known native structure. According to the preference model M_{pref}, $\boldsymbol{x} \prec \boldsymbol{x}' \Leftrightarrow f(\boldsymbol{x}; \boldsymbol{\theta}_{pref}) < f(\boldsymbol{x}'; \boldsymbol{\theta}_{pref})$ where

$$U(\boldsymbol{x}; M_{pref}) = f(\boldsymbol{x}; \boldsymbol{\theta}_{pref}) = \sum_{ijk} \alpha_{jk}^i \left(K(\boldsymbol{x}_j^i, \boldsymbol{x}) - K(\boldsymbol{x}_k^i, \boldsymbol{x}) \right) y_{jk}^i \qquad (11.9)$$

Since $f(\boldsymbol{x})$ can be written as the inner product $\vec{w} \cdot \phi(\boldsymbol{x})$, being $\phi(\boldsymbol{x})$ the feature mapping induced by the kernel, and since the kernel is a inner product in feature space and therefore a bilinear operator, the preference model can be seen as a binary classifier on *pairs* of conformations with

$$f_p(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle; \boldsymbol{\theta}_{pref}) = \sum \alpha_{jk}^i K_p \left(\langle \boldsymbol{x}_j^i, \boldsymbol{x}_k^i \rangle, \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \right) y_{jk}^i \qquad (11.10)$$

being

$$K_p \left(\langle \boldsymbol{x}_j^i, \boldsymbol{x}_k^i \rangle, \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \right) = K(\boldsymbol{x}_j^i, \boldsymbol{x}) - K(\boldsymbol{x}_j^i, \boldsymbol{x}') - K(\boldsymbol{x}_k^i, \boldsymbol{x}) + K(\boldsymbol{x}_k^i, \boldsymbol{x}'). \qquad (11.11)$$

In this way one can still apply the standard binary classification SVM algorithm to a data set of conformation pairs in order to obtain the coefficients α_{jk}^i.

11.3 Computing Similarity between Structures

In order to train the preference model, a similarity measure between two conformations must be provided in the form of a kernel function K between them. In principle, a hyper-graph could be associated to the conformation $\boldsymbol{x}$, whose nodes are residues (or atoms) and whose hyper-edges are r-wise interactions with an associated measure of interaction strength (e.g. the distance in a pairwise interaction). In such an approach, the computation of the kernel between two conformations would be performed by computing the similarity between the corresponding hyper-graphs, thus effectively exploiting higher order relations among atoms as features. Many statistical potentials have terms for bond angles (involving three atoms) and dihedral angles (involving four atoms). In principle, statistics could be made for the properties of small polyhedra of any given size extracted from the protein structure. Moreover, any sort of graph similarity measure could be used to calculate the kernel between two conformation hyper-graphs. Unfortunately, exploitation of r-order relations with r greater than 2 poses computational and sparseness problems because of the exponential growth of the number of features. In the following we introduce two kernel functions that are inspired from pairwise interactions, with the aim of defining preference models that can be compared directly to the approach based on knowledge-based potentials.

11.3.1 Product Kernel

In Sec. 6.3 the discrete form of a probability product kernel has been shown. Inspired by this definition, a kernel based on a representation consisting of distance counts is introduced here, where a conformation $\boldsymbol{x}$ is represented by the sufficient statistics $m_{ks}^{ab}(\boldsymbol{x})$ defined in Sec. 11.1 for knowledge-based potentials. The distance counts can be seen as the (unnormalized) discrete distributions associated with $\boldsymbol{x}$ and the kernel can be written as

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{a,b,k,s} m_{ks}^{ab}(\boldsymbol{x})^{\rho} \cdot m_{ks}^{ab}(\boldsymbol{x}')^{\rho} \tag{11.12}$$

Absolute counts are used instead of normalized frequencies because the resulting features retain informations also on the dimensions of the three-dimensional model. In all subsequent work ρ is simply set to 1 and the

preceding kernel can also be interpreted as a inner product between fixed-size vector representations of conformations. In this case it is immediate to see that the feature space consists of S distance bins for each triplet $\langle a, b, k \rangle$.

11.3.2 Intersection Kernel

A histogram intersection kernel (HIK) is similar to a probability product kernel in the case of discrete distributions, but uses the minimum operator instead of product (Sec. 6.3):

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{a,b,k,s} \min \left[m_{ks}^{ab}(\boldsymbol{x}), m_{ks}^{ab}(\boldsymbol{x}') \right] \tag{11.13}$$

In this context, HIKs have an interesting interpretation in term of similarity between the interaction graph associated with conformations. In particular, if each edge is labeled with the tuple $\langle a, b, k, s \rangle$, the HIK between two conformations measures similarity by counting how many edges with the same label are in common between the graphs derived from the two conformations.

11.3.3 Fisher Kernel

Fisher kernels (see Sec. 6.4) can be applied to the present problem in order to combine the advantages of discriminant learning and knowledge-based potentials. The Fisher vector $\phi(\boldsymbol{x})$ associated with a conformation $\boldsymbol{x}$ is defined as the gradient of the log-likelihood of $\boldsymbol{x}$ with respect to the model's parameters $\boldsymbol{\theta}$:

$$\phi(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}; \boldsymbol{\theta}) \tag{11.14}$$

From Eqs 11.4 and 11.5 follows:

$$\frac{\partial \ell(\boldsymbol{x}; \boldsymbol{\theta})}{\partial \theta_{ks}^{ab}} = \sum_i \sum_{j>i} \sum_s z(d_{ij}, s) \frac{\partial \ln \left(\theta_{|i-j|,s}^{a_i b_j} \right)}{\partial \theta_{ks}^{ab}} = \frac{m_{ks}^{ab}(\boldsymbol{x})}{\theta_{ks}^{ab}} \tag{11.15}$$

The Fisher kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ between two conformations $\boldsymbol{x}$ and $\boldsymbol{x}'$ is then defined as the dot-product of the corresponding Fisher vectors:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}') = \sum_{a,b,k,s} \frac{m_{ks}^{ab}(\boldsymbol{x}) m_{ks}^{ab}(\boldsymbol{x}')}{\left(\theta_{ks}^{ab}\right)^2} \tag{11.16}$$

Compared to the kernels of Eq. 11.12 and Eq. 11.13, the Fisher kernel takes into account information from the entire data set used to compute

the potentials when comparing two conformations. This information is used to scale each feature seen by those kernels by a value that indicates the significance of that particular contact.

11.4 Experiments and Discussion

A representative non-redundant subset of the Protein Data Bank was taken from PISCES (Wang and Dunbrack Jr., 2003), with resolution better than 1.8A and R-factor less than 0.3. Standard pair potential approaches have been shown to be affected by protein length (Melo et al., 2002) such that improved performance is achieved when the size range of the proteins used for deriving a potential is the same as the size range of the proteins being assessed. This analysis has been restricted here to a particular size range: small protein chains less than 100 amino acids in length. Filtering the PISCES-derived set based on this criteria resulted in 266 protein chains. Because of the high computational cost of producing large decoy sets for each of the 266 protein chains using a "de-novo" protocol, here the decoys have been generated using a threading algorithm. Each of the sequences was scanned against a fold library using a profile-profile fold recognition algorithm known as Phyre[1]. For each sequence, 3D models were constructed based on the fold recognition alignments to the top 20 scoring matches in the fold library. Model quality was assessed with sequence-dependent structural alignment using the TM-score (Zhang and Skolnick, 2004) of the model to the known experimental structure of the query sequence. Each protein chain was assigned a fold according to the SCOP hierarchy (Murzin et al., 1995). For each protein the models labeled as *good* were those with TM-score greater than 0.2 while the *bad* models were the rest.

Sufficient statistics $m_{ks}^{ab}(\boldsymbol{x})$ were estimated for each structure $\boldsymbol{x}$ by extracting all the contacts with lengths between 0 and *MaxDist*. Sequence separation values between $k = 0$ and $k = 8$ were taken independently, while all the residues spaced more than 8 positions were assumed to be equivalent and conventionally assigned $k = 9$. The histograms of distances were constructed by dividing the interval $[0, MaxDist]$ into S bins of equal size.

Using the 266 proteins a set of experiments is performed to compare pair potentials (PP) to the alternative kernel based algorithms introduced

[1]manuscript in preparation http://www.sbg.bio.ic.ac.uk/phyre

in this work, namely the product kernel between histograms of Eq. 11.12, the histogram intersection kernel of Eq. 11.13 and the Fisher kernel of Eq. 11.16. For each method, performance was estimated using a five-fold cross validation procedure. The partition in five subsets was created by ensuring that chains in the same SCOP fold are all in the same subset. The scoring functions produced by the learning algorithms were then used to predict the ranking of unseen models. Particular care has been taken in eliminating any homology between train and test proteins to ensure that experimental results are indicative of the performances for "de-novo" prediction.

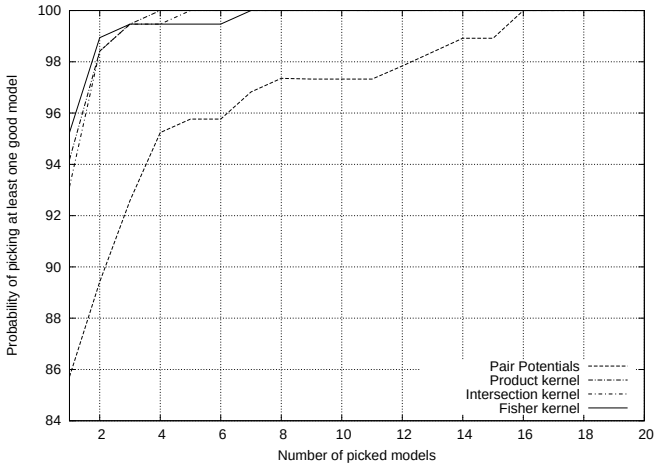
Method	*Accuracy*	*RankCoeff*	*MinRank*	*BestPos*
PP	74.7%	0.327	1.65	3.64
Product	84.0%	0.495	1.08	1.98
Intersection	85.2%	0.513	1.10	2.07
Fisher	85.1%	0.491	1.08	2.10

Table 11.1. Experimental results. Histograms were created by setting MaxDist = 20Å and $S = 20$ (bin size = 1Å).

The results of the experiments are outlined in Table 11.4. Four different measures of performance are reported: *Accuracy* refers to the accuracy for the binary classification task associated with the comparison of two structures, i.e. the percentage of times the result of comparing two models is the same when using the TM-score and the scoring function generated by the learning algorithm; *RankCoeff* is the Kendall correlation coefficient computed between the rankings induced by the TM-score and by the estimated scoring function; *MinRank* is the average minimum rank of the good models; *BestPos* is the average rank of the model having highest TM-score. In Figure 11.1 is shown the percentage of times at least one good model was ranked within the first n positions. The results consistently indicate the superiority of discriminant training. When measuring performance using accuracy, all the kernels significantly outperform the pair potentials approach, with the null hypothesis of identical performance rejected at 99% confidence level in all cases. The comparison between the various kernel methods demonstrate a slight advantage of the Fisher on the Linear kernel, with a quite low confidence of 66%, while there is almost no difference between Fisher and Intersection kernel. Looking at the curves plotted in figure 11.1, it is worth

noticing that all the discriminative approaches have a probability greater than 98% of getting a near native model within the best two choices, while PP would only have about 90% probability.

Figure 11.1. Percentage of times at least one good model was ranked within the first n positions.



In a subsequent experiment the data set used to train pair potentials is extended in order to compare the accuracy of kernel based methods trained on small data sets to the accuracy of pair potentials trained on a large data set of non redundant chains. The five fold cross validation procedure has been repeated adding a second set of chains obtained as follows. A precompiled CulledPDB set of 512 native structures taken from PISCES, with chain's length shorter than 100 residues, resolution better than 2.5Å, R-factor lower than 0.25 and homology lower than 90%. To enforce non-redundancy, for each of the five subsets in the cross validation procedure, the set of 512 native structures used to build the potentials was further reduced by taking out those proteins belonging to any SCOP fold that occurs in the test set. In this setting the pair potential method obtained $\mathit{Accuracy} = 76.9\%$, $\mathit{RankCoeff} = 0.351$, $\mathit{MinRank} = 1.65$ and $\mathit{BestPos} = 3.20$. Kernels trained on small data sets outperformed pair potentials even when the latter were trained on larger data sets, maintaining a confidence level of 99% for this result.

The results presented here indicate that proposed kernel methods clearly outperform the standard pair potential approach for scoring decoys. The

discriminative approach can exploit the bias of the structure prediction algorithm and lead to improved discriminatory power and thus more accurate predictions of protein structure.

Chapter 12

Prediction of Biological Activity of Molecules

Biologically active substances interact in most cases with proteins, thus triggering specific molecular mechanisms that finally lead to a biological response. Therefore, each chemical that enter an organism can alter its internal state and possibly lead to disease or have a positive effect and cure a preexisting illness. Discovering a new compounds that can be used to cure a specific disease (without causing another) is the main scope of pharmaceutic research. However, drug discovery is an expensive process due to the high costs of screening thousands of molecules for the desired effects.

The ability of a chemical compound to bind a target protein with sufficient strength and to affect its functions lies in the compound three-dimensional structure, that must fit perfectly into the protein binding site (*receptor*). Compounds that binds to the same receptor would present similar geometrical and physico-chemical characteristics. If the structure of the receptor is not known, as in the case of drug screenings, one can still infer that compounds with similar sub-structures would present similar activities. This hypothesis is the basis of QSAR methods, that propose different approaches to compare molecules structures to find features that correlates with biological activity. QSAR methods have become increasingly popular due to their possibility of virtually screening huge databases of chemical compounds.

A review of the various existing QSAR approaches is made in Sec. 12.1.

In Sec. 12.2 a new QSAR method based on the three-dimensional decomposition kernel is proposed. This kernel function, combined with a specific instantiation of the Weighted Decomposition Kernels class described in Sec. 12.3, is then experimented on two relevant QSAR datasets: the NCI Cancer Screening (Sec. 12.4) and the NCI anti-HIV Screening (Sec. 12.5). The two combined kernel functions prove to outperform the previously published kernel approaches that are reputedly the state-of-the-art on these dataset.

12.1 Introduction to QSAR analysis

In the early 1960s, the pioneering work of Hansch et al. (1962) demonstrated that the biological activity of a chemical compound is a function of its physico-chemical properties. From these findings, many different techniques have been proposed to correlate biological response with molecular properties of the target compound through quantitative structure activity/property relationships (QSAR/QSPR) analysis. In classical QSAR methods, each chemical structure is transformed into a vector of numerical values (*descriptors*) corresponding to relevant properties of the molecule. After the transformation, any statistical method capable of handling numerical features can be used to correlate the vector values with the biological activity. QSAR descriptors can be built from chemical and three-dimensional structure of the compounds using steric and hydrophobic properties (Hansch et al., 1995), topological properties (Devillers and Balaban, 1999), connectivity indices (Kier and Hall, 1986) or quantum-chemical effects (Karelson et al., 1996).

Alternatively, 3D-QSAR (Kubinyi et al., 2002) methods have been proposed to generate the descriptors exclusively from the three-dimensional structure of the molecule. Among the most pre-eminent 3D-QSAR methods figure CoMFA (Comparative Molecular Field Analysis, Cramer III et al. 1988) and CoMSIA (Comparative Molecular Similarity Indices Analysis, Klebe and Mietzner 1994). In both approaches, the molecules structures are first aligned and then placed into a three-dimensional grid to calculate their interaction energy with an atomic probe. The molecular fields obtained in this way are then correlated with the biological activity by partial least squares analysis.

The performance of descriptors-based approaches relies, to a large ex-

tent, on the a priori identification of the relevant properties that capture the structure-activity relationships for the particular problem. Identifying this relevant set of properties requires considerable domain expertise and extensive experimentation. To overcome this problem, a different set of techniques have been developed that operate directly on the structure of the chemical compound and try to automatically identify sub-structures that can be used to discriminate between the different behaviors. One of the earlier approaches (King et al., 1996) that follow this paradigm is based on inductive logic programming (ILP). In this approach the chemical compound is expressed with first order logic and an ILP system is used to discover rules that are good for discriminating between the different activity values. Since these rules consist of predicates describing atoms and bonds, they essentially correspond to sub-structures that are present in the chemical compounds. A decisive advantage of these techniques is that the discovered rules can be easily understood by experts and could be used to generate new compounds with the desired behaviors.

Though ILP-based approaches are quite powerful, the high computational complexity of the underlying rule-induction system limits the size of the dataset to which they can be applied. One of the fundamental reasons limiting the scalability of ILP-based approaches is their use of first order logic-based representation. For this reason a number of researchers have explored the much simpler graph-based representation of the compound chemical structure and transformed the problem of finding chemical sub-structures to that of finding subgraphs in this graph-based representation (Berthold and Borgelt, 2002; Deshpande et al., 2003). An even simpler representation is used in Kramer et al. (2001b): in this scheme each chemical compound is represented as a SMILES string (Weininger, 1988, 1989), and is thought of as a sequence of SMILES objects. SMILES is a string representation of a chemical structure encoding atomic content and connectivity. A single compound can be represented by different SMILES strings, unless a canonical ordering is defined. This representation simplifies the problem of discovering frequently occurring sub-structures at the cost of losing some information on the represented molecules.

Both recursive neural networks (Micheli et al., 2001) and kernel methods for structured data (Horváth et al., 2004; Swamidass et al., 2005; Menchetti et al., 2005) have been successfully applied to QSAR problems. In Micheli

et al. (2001), the chemical structures are first transformed into trees (this is possible thanks to the particular class of compounds investigated) and then a standard RNN is used for learning the structure/activity relation. On the other hand, kernels do not require the input structure to be ordered and have proved to be very effective in QSAR tasks. In Horváth et al. (2004) a decomposition kernel has been proposed that compare the sets of tree and cyclic patterns present in the input molecules. This kernel has been applied to the HIV dataset and will be used as a comparison in this work (Sec. 12.5). In Swamidass et al. (2005) various kernels for one-, two- and three-dimensional representations of the input molecules are tested on the Cancer dataset (Sec. 12.4). Finally, Menchetti et al. (2005) introduced the Weighted Decomposition Kernel for graphs and demonstrated its high level of performance on various datasets. This kernel is used here in conjunction with the three-dimensional decomposition kernel to reach top performances on the proposed QSAR problems.

12.2 Three-Dimensional Decomposition Kernel

Descriptor-based QSAR methods use a fixed set of features whose biological relevance have been identified by expert knowledge and previous experimentation. This approach is thus limited by the a-priori identification of the feature set, that can't be comprehensive of all the aspects of the molecule structure. On the contrary, the various methods for mining the graphical structure formed by atoms and bonds are not limited to pre-determined set of features but also have no direct information about the geometrical conformation of the input compound. In Deshpande et al. (2003) and Swamidass et al. (2005) there are two different proposal to handle the geometrical information, both of which are limited to first-order spatial relations (distances between atoms). On the other hand, 3D-QSAR methods try to estimate the binding energy of the compound by computing an interaction potential with an atomic probe. The resulting energy function is thus computed using both geometrical and chemical properties of the compound. However, these grid-based approaches are strongly influenced by the alignment of compounds structure, that is the critical step of these methods.

In Sec. 6.7 a decomposition kernel (3DK) for three-dimensional structures has been proposed. This kernel is able to compare molecule struc-

tures by using chemical (atoms and bonds types) and geometrical (atoms distances) informations. 3DK performs a decomposition of a molecule structure into shapes defined by variable sized groups of atoms with their relative distances. The shapes are invariant of the coordinate system (no alignment is needed) and represent r-wise relations between groups of r (≥ 2) atoms. The conformations of two shapes with identical composition (atoms and bonds types) are compared by looking at the variations between atoms distances. The similarity between two shapes is thus a smooth function of the similarity between their dimensions.

The actual decomposition of the molecule structure into shapes defines the feature set associated to the kernel and influences its accuracy and time complexity. All the experiments performed in this work are made using a decomposition strategy that follows the chemical structure of the molecule: for each atom a_i all the atoms a_{ij} that are less than k bonds apart from a_i are considered; for each a_{ij} the atoms bonded to a_{ij}, the atom a_{ij} itself and the atom a_i forms a shape. In this way the number of shapes is limited to the essential (only bonded atoms are considered) and the kernel can be efficiently computed. Atom and bond types are considered when forming the shapes.

12.3 Weighted Decomposition Kernel

The Weighted Decomposition Kernels (see Sec. 6.6 for details) are a class of decomposition kernels for undirected graphs that are both general and computationally efficient. The basic idea behind WDK is that each substructure in which a graph is decomposed is enriched with its graphical context. All the vertices and edges of the graphical structure are annotated with tuple of attributes. The exact match kernel between two sub-structures is thus weighted by the kernel between the distribution of attribute values in their contexts.

Attributes are organized into classes and can be instantiated for each vertex or edge. In the following, η denotes a generic attribute class, $\eta(v)$ its value at vertex v and $\eta(u, v)$ its value at edge $\langle u, v\rangle$. For a given structure $\boldsymbol{x} = \langle V, E\rangle$, $\eta(\boldsymbol{x})$ denotes the value set associated with attribute η: in case of vertex attributes $\eta(\boldsymbol{x}) = \{\eta(v) : v \in V\}$, while in case of edge attributes $\eta(\boldsymbol{x}) = \{\eta(u, v) : \langle u, v\rangle \in E\}$. Given a graph $\boldsymbol{x}$ and an attribute η_i, let $p_i(j)$

be the observed frequency of value j in $\eta_i(\boldsymbol{x})$. A kernel function K_a that compares the distribution of attributes in two graphs can then be chosen between one of the kernel for distribution of properties described in Sec. 6.3.

A molecule is naturally represented by an undirected graph $\boldsymbol{x} = \langle V, E \rangle$ where vertices are atoms and edges are covalent bonds. In this work, vertex attributes are the atom type and the atom charge, while edge attributes are the bond type and a triplet encoding the bond type and the two bonded atoms types. Given a vertex $v \in V$ and an integer $l \geq 0$, let $\boldsymbol{x}(v, l)$ the subgraph of $\boldsymbol{x}$ induced by the set of vertices reachable from v by a path of length at most l and by the set of all edges that have at least one end in the vertex set of $\boldsymbol{x}(v, l)$. Two different versions of the WDK are used in this work. The first WDK is obtained by choosing $D = 1$ (see Sec. 6.5 on decomposition kernels for more details) and a relation R that depends on the context radius l defined as $R = \{\langle s, z, \boldsymbol{x} \rangle : \boldsymbol{x} \in \mathcal{X}, s = v, z = \boldsymbol{x}(v, l), v \in V(\boldsymbol{x})\}$, where s is the selector and z is the context for vertex v. The kernel is then defined as

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\substack{\langle s, z \rangle \in R^{-1}(\boldsymbol{x}) \\ \langle s', z' \rangle \in R^{-1}(\boldsymbol{x}')}} \delta(s, s') K_a(z, z') \tag{12.1}$$

In the second WDK, $D = 2$ and two types of contexts are used, $z = \boldsymbol{x}(v, l)$ and its graph complement denoted by z_c. Probability kernels over contexts can then be combined under direct sum:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_{\substack{\langle s, z, z_c \rangle \in R^{-1}(\boldsymbol{x}) \\ \langle s', z', z'_c \rangle \in R^{-1}(\boldsymbol{x}')}} \delta(s, s') \left(K_a(z, z') + K_a(z_c, z'_c) \right) \tag{12.2}$$

12.4 Experiments on NCI Cancer Dataset

The Cancer dataset has been made publicly available by the National Cancer Institute (NCI), and provides screening results for the ability of more than 70,000 compounds to suppress or inhibit the growth of a panel of 60 human tumor cell lines. The dataset corresponding to the concentration parameter GI50, essentially the concentration that causes 50% growth inhibition, is used here. For each cell line, ≈ 3500 compounds are provided together with

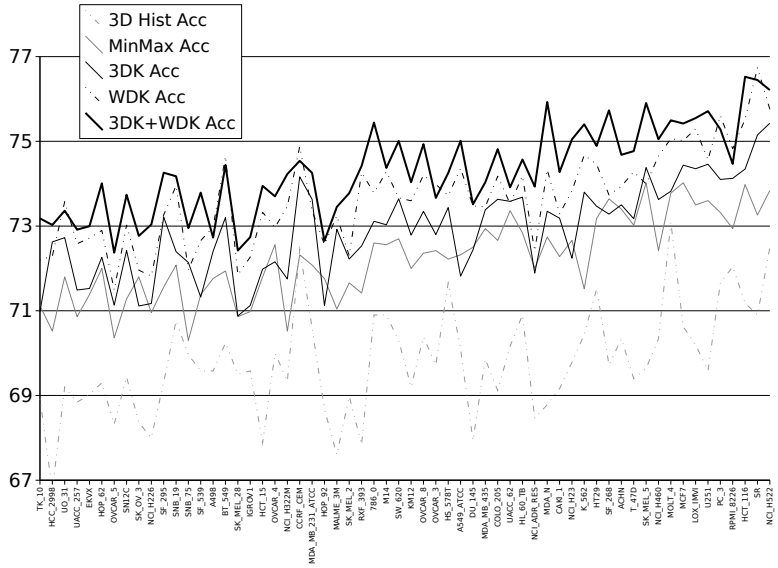


Figure 12.1. Accuracy values for all 60 cell lines of the NCI Cancer screening dataset. The 3DK, the WDK and their sum are compared to the MinMax and 3D-Hist kernels.

information on their cancer-inhibiting action. A two-classes classification problem is thus defined, with roughly 50% of examples in each class. For all the compounds 3D coordinates have been generated by running CORINA (Gasteiger et al., 1990; Sadowski et al., 2003) on the 2D structure.

The results of the experiments on the 60 cell lines are reported in Figs. 12.1 and 12.2. Classification performance has been measured both by the value of accuracy an by the mean of the ROC area (AUC) on a ten folds cross validation. The performances of WDK and 3DK are compared to the results obtained by the top performing kernel (MinMax) proposed by Swamidass et al. (2005). The values of accuracy of the 3D Histogram kernel from the same work are also reported, in order to compare 3DK with the only other method that make use of atom coordinates on this problem. The first version of WDK is used here, with no graph complement and context radius $l = 3$. For the 3DK, the maximum radius k has been set to 3. A combination of the two kernels has also been tested (WDK+3DK). The three kernels are then composed with a polynomial kernel with degree equal to 2. The C value is optimized by model selection.

The results of the experiments demonstrate the superiority of the kernels

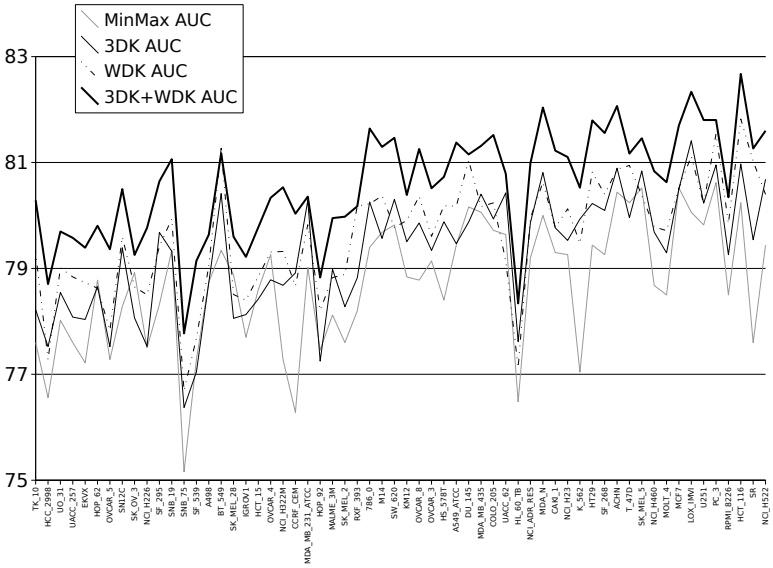


Figure 12.2. AUC values for all 60 cell lines of the NCI Cancer screening dataset. The 3DK, the WDK and their sum are compared to the MinMax kernel.

proposed in this work on this dataset. Comparing the accuracy values, the 3DK performs better than MinMax on 50 over 60 cell lines, the WDK on 59 over 60 cell lines and the two combined kernels perform always better than MinMax. Almost the same conclusions can be drawn by comparing the AUC values: the 3DK performs better than MinMax on 47 over 60 cell lines, the 3DK on 53 over 60 cell lines, and their combination always perform better than MinMax. Moreover, the 3DK outperform the 3D Histogram kernel on all the cell lines, thus showing the importance of using higher-order spatial relation when comparing three-dimensional structures.

12.5 Experiments on NCI HIV Dataset

The HIV dataset contains 42,687 compounds evaluated for evidence of anti HIV activity from the DTP AIDS Anti-viral Screen program of the National Cancer Institute[1]. The compounds are divided in three classes: 422 compounds are confirmed active (CA), 1081 are moderately active (CM) and 41184 are confirmed inactive (CI). A compound is considered inactive if a

[1]The dataset is available at http://dtp.nci.nih.gov/docs/aids/aids_data.html

Method	CA vs CM	CA+CM vs CI	CA vs CI
FSG	78.6%	78.6%	91.4%
FSG+3D	81.1%	81.9%	94.0%
γCPK	84.0% ± 1.0%	83.7% ± 1.2%	94.7% ± 0.8%
γWDK	85.4% ± 1.9%	84.1% ± 0.6%	94.5% ± 0.9%
γ3DK	85.3% ± 4.0%	84.4% ± 0.7%	95.1% ± 0.6%
γ(WDK+3DK)	86.1% ± 2.8%	84.8% ± 0.9%	95.1% ± 0.7%

Table 12.1. Results of the experiments on the NCI Anti-HIV screening dataset. The 3DK and WDK are compared to the Frequent Sub-graphs approach and to the Cyclic Pattern Kernel. The table reports the value of AUC for the various methods.

test showed less than 50% protection of human CEM cells. All other compounds were re-tested. Compounds showing less than 50% protection (in the second test) are also classified inactive. The other compounds are classified active, if they provided 50% protection in both tests, and moderately active, otherwise. For all the compounds 3D coordinates have been generated by running CORINA (Gasteiger et al., 1990; Sadowski et al., 2003) on the 2D structure.

Following the experimental setup of Deshpande et al. (2003), three classification problems are formulated on this dataset: in the first problem (CA vs CM), positive examples are confirmed active compounds, while moderately active compounds forms the negative class; for the second problem (CA+CM vs CI), the positive class is formed by the combination of moderately active and confirmed active compounds; in the last problem (CA vs CI) the positive examples are confirmed active compounds and the negative class is formed by confirmed inactive compounds.

The results of the experiments are reported in Tab. 12.1. Classification performance has been measured by the mean and standard deviation of the ROC area on a five folds cross validation in which the original class distribution was preserved in each fold. The performances of WDK and 3DK are compared to the best results obtained with the Frequent Subgraphs method proposed by Deshpande et al. (2003) and the Cyclic Pattern Kernel described in Horváth et al. (2004) on the same dataset. For the WDK, graph complement and context radius $l = 4$ have been used as in Menchetti et al. (2005). For the 3DK, the maximum radius k has been set

to 3. A combination of the two kernels has also been tested (WDK+3DK). The three kernels are then composed with a Gaussian kernel with $\gamma = 1$. The C value is optimized by model selection.

The results of the experiments show that both the 3DK alone and in combination with WDK visibly outperform the CPK on all three problems. The significance of these results is difficult to estimate but the number of experiments (comprising those on the Cancer dataset) in which the mean performance of the 3DK+WDK approach is higher than state-of-the-art should prove these conclusions. The experiments also shown that 3DK perform better than WDK when higher amount of data is available. This can be a consequence of the fact that 3DK feature space is bigger than WDK, thus resulting in more orthogonal pairs of examples.

Part VI

Conclusions

Chapter 13

Conclusions

During this work several aspects of the prediction of protein structure and function have been faced. Suitable representations of the protein molecules have been used regarding of the amount of information available. Ad-hoc neural and kernel architectures have been developed and applied to the learning of the map from input graphical structures to output properties.

In Chap. 7 a two-stages architecture for the prediction of secondary structure from amino-acid sequence has been presented. Experimental results show this architecture, composed by a local SVM classifier and a filtering BRNN, can reach state-of-the-art values of accuracy and segment overlap on this task. The predictor outputs have been subsequently refined using a Viterbi decoder to enforce constraints derived from prior knowledge into secondary structure predictions. Results demonstrate the Viterbi decoder is able to correct single-residue errors and to output sequences that contain longer SS segments, thus achieving higher values of SOV. However, the Viterbi decoder cannot correct completely misclassified segments of secondary structure. Improvements to this idea would involve the creation of richer finite-state automata with constraints on whole secondary structure segments derived from libraries of known folds.

In Chap. 9 a modification of the standard BRNN architecture have been proposed to simplify the sequence learning tasks by incorporating explicit knowledge about spatial interactions between residues. Empirical results show this approach can greatly improve prediction of protein secondary structure when reliable interaction knowledge is available. IEBRNNs are able to capture the serial order relation within interactions, a property that

leads to better predictions than simply averaging the input at interacting positions, as proposed in previous works. The IEBRNN approach does not yet advance the state-of-the-art in SS prediction but enlightens possible future directions of investigation. The experiments provide further evidence that knowledge of a relatively small number of reliable interactions is sufficient with the currently available data to obtain high quality predictions. Improvements to the IEBRNN architecture are needed in order to match the characteristics and performances of the latest more complex SS predictors. A closed loop procedure which iterates the SS/Contact-Map prediction process would also be an interesting development.

In Chap. 10 a procedure to reconstruct a protein three-dimensional structure from predicted contacts between β-strands has been introduced. The proposed approach does not require fine-grained information about contacts between single residues and can be used to reconstruct native structure when incomplete information is available. Experimental results demonstrate the proposed approach is able to build protein models matching the available characteristics of the native conformation. The algorithm is inherently fast (reconstruction takes on average 20 minutes on standard workstations) and could be tested over a large non-redundant set of proteins. Unfortunately, the reconstructed structures are quite distant from corresponding native conformations. However, the quality of reconstruction could possible improve if different types of contacts between secondary structure elements are added. Reconstruction has also been tested using the strands topology derived from a predictor of β-strands partnership. The algorithm have not proven to be sufficiently reliable to correct the unavoidable errors of the predictor, however this can be considered as a first step toward a complete reconstruction procedure based on coarse-grained information alone.

In Chap. 11 a kernel method for learning preference relations have been used to build an energy function for scoring protein models generated by structure prediction algorithms. Energy function derivation is normally based on statistical potentials, a technique that is widely-used by the protein structure prediction community. Such scoring functions invariably focus only on positive examples (known experimental structures), while ignoring the potentially rich source of information available from negative examples (computer-generated decoys). The principle of computational learning based on positive and negative examples is a well-established research domain. The

results presented here indicate the information contained in protein models can be successfully harnessed using kernel methods, and this can lead to improved discriminatory power and more accurate predictions of protein structures.

All these results can be considered as a succession of steps that starts from the amino-acid sequence and arrive at the predicted protein three-dimensional structure. From sequence to secondary structure segments, from SS segments to coarse-grained contacts, from contact maps to three-dimensional structures, from tentative protein models to high-quality structures. All the machinery needed for the construction of an ab-initio protein structure predictor have been described here. The logical consequence of this work would then be the integration of these pieces, a unified experimentation and its final assessment in a CASP contest (Zemla et al., 2001).

Prediction of protein function has also been considered in this work. In Chap. 8 the two-stages architecture presented before have been applied to the problem of predicting metal binding sites (MBS) from amino-acid sequence. MBS prediction is an important task for the functional study of a protein that has not been previously approached by machine learning techniques. When homology modeling cannot be applied, MBS are commonly identified using known patterns of functionally relevant residues. By construction, this approach has very high precision but covers only a small percentage of the existing MBS. The discriminant approach proposed in this work noticeably improve the coverage of the method based on pattern search, while maintaining high-accuracy in predicting the bonding state of cysteines and histidines residues. Moreover, this approach is able to accurately predict the presence of disulfide-bonded cysteines, a valuable information for the study of protein structure and function.

Finally, prediction of the function of small molecules has been addressed. In Chap. 12 the three-dimensional decomposition kernel (3DK) and the weighted decomposition kernel (WDK) have been applied to the prediction of biological activity of chemical compounds. The 3DK is a decomposition kernel that can compute the similarity between three-dimensional structures. 3DK uses r-wise spatial relations between molecule atoms, differently to other approaches that makes use only of pair-wise interactions (in the form of histograms of distances or linear combination of distance values). 3DK and WDK have been tested on two significant datasets composed by thousands of

molecules. The results show the combination of the two kernels outperforms the current state-of-the-art on all the proposed tasks. The combination of 3DK and WDK is thus proved to be an accurate method for QSAR analysis, a fundamental step in the drug discovery process.

Abbreviations

Acyclic Graph:	AG
Area Under ROC Curve:	AUC
Bi-directional Recurrent Neural Network:	BRNN
Contextual Recursive Neural Network	CRNN
Direct Acyclic Graph:	DAG
Direct Ordered Acyclic Graph:	DOAG
Hidden Markov Models:	HMM
Inductive Logic Programming:	ILP
Multi-Layer Perceptron:	MLP
Neural Network:	NN
Recursive Neural Network:	RNN
Root Mean Square Deviation:	RMSD
Quantitative Structure Activity Relationship:	QSAR
Quantitative Structure Property Relationship:	QSPR
Support Vector Machine:	SVM
Segment Overlap:	SOV
Viterbi Decoder:	VD

Bibliography

Abagyan, R. A. and Batalov, S. (1997). Do aligned sequences share the same fold? *Journal of Molecular Biology*, 273(1):355–368.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI–BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402.

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

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68.

Bairoch, A. and Apweiler, R. (1999). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Res*, 27:49–54.

Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, 294:93–96.

Baldi, P. and Brunak, S. (2001). *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA, 2nd edition.

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

Baldi, P. and Pollastri, G. (2003). The principled design of large-scale recursive neural network architectures – dag-rnns and the protein structure

prediction problem. *Journal of Machine Learning Research*, 4(Sep):575–602.

Barla, A., Odone, F., and Verri, A. (2003). Histogram intersection kernel for image classification. In *Proceedings of International Conference on Image Processing*, volume 3, pages 513–516.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Berg, J., Tymoczko, J., Stryer, L., and Clarke, N. (2002). *Biochemistry*. W.H. Freeman & Co., 5th edition.

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J. D., and Zardecki, C. (2002). The protein data bank. *Acta Cryst.*, D58:899–907.

Berthold, M. and Borgelt, C. (2002). Mining molecular fragments: Finding relevant substructures of molecules. In *Proceedings of the Second IEEE International Conference on Data Mining*.

Bower, M. J., Cohen, F. E., and Dunbrack Jr., R. L. (1997). Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. *Journal Molecular Biology*, 267(5):1268–1282.

Bradley, A. P. (1997). The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Bradley, P., Chivian, D., Meiler, J., Misura, K. M. S., Rohl, C. A., Schief, W. R., Wedemeyer, W. J., Schueler-Furman, O., Murphy, P., Schonbrun, J., Strauss, C. E. M., and Baker, D. (2003). Rosetta predictions in CASP5: Successes, failures, and prospects for complete automation. *Proteins*, 53(S6):457–68.

Bridle, J. (1989). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In

Fogelman-Soulie, F. and Hérault, J., editors, *Neuro-computing: Algorithms, Architectures, and Applications*. Springer-Verlag.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., , and Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4:187–217.

Bru, C., Courcelle, E., Carrère, S., Beausse, Y., Dalmar, S., and Kahn, D. (2005). The ProDom database of protein domain families: more emphasis on 3d. *Nucleic Acids Research*, 33:212–215.

Ceroni, A. and Frasconi, P. (2004). Protein structure assembly from knowledge of beta-sheet motifs and secondary structure. In Apolloni, B., Marinaro, M., and Tagliaferri, R., editors, *Proceedings of WIRN04*. Kluwer.

Ceroni, A., Frasconi, P., and Kelley, L. (2005a). Supervised scoring of protein models using kernels on statistical potentials. Technical Report RT 11/2005, DSI Università di Firenze.

Ceroni, A., Frasconi, P., Passerini, A., and Vullo, A. (2003a). A combination of support vector machines and bidirectional recurrent neural networks for protein secondary structure prediction. In *Proceedings of AI*IA 2003*, volume 2829 of *Lecture Notes in Computer Science*, pages 142–153. Springer.

Ceroni, A., Frasconi, P., Passerini, A., and Vullo, A. (2003b). Predicting the disulfide bonding state of cysteines with combinations of kernel machines. *Journal of VLSI Signal Processing*, 35:287–295.

Ceroni, A., Frasconi, P., and Pollastri, G. (2005b). Learning protein secondary structure from sequential and relational data. *Journal of Neural Networks*, 18(8):1029 – 1039.

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:1–25.

Cramer III, R., Patterson, D., and Bunce, J. (1988). Comparative molecular field analysis (CoMFA): Effect of shape on binding of steroids to carrier proteins. *Journal of the American Chemical Society*, 110:5959–5967.

Crammer, K. and Singer, Y. (2002). On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2–3):201–233.

Crippen, G. and Havel, T. (1988). *Distance geometry and molecular conformation*. Taunton : Research Studies Press.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.

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

Date, C. (1995). *An Introduction to Database Systems*. Addison-Wesley.

Degtyarenko, K. (2000). Bioinorganic motifs: towards functional classification of metalloproteins. *Bioinformatics*, 16:851–864.

Deshpande, M., Kuramochi, M., and Karypis, G. (2003). Frequent sub structure based approaches for classifying chemical compounds. In *Proceedings of ICDM 03*, pages 35–42.

Devillers, J. and Balaban, A. T., editors (1999). *Topological indices and related descriptors in QSAR and QSPR*. Gordon & Breach.

Dietterich, T. G. (2002). Machine learning for sequential data: A review. *Lecture Notes in Computer Science*, 2396:15–31.

Domingues, F. S., Koppensteiner, W. A., Jaritz, M., Prlic, A., and et al., C. W. (1999). Sustained performance of knowledge-based potentials in fold recognition. *Proteins*, 37:112–120.

Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C., Hofmann, K., and Bairoch, A. (2002). The PROSITE database, its status in 2002. *Nucleic Acids Research*, 30(1):235–238.

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

Fariselli, P., Riccobelli, P., and Casadio, R. (1999). Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. *Proteins*, 36.

Fiser, A. and Simon, I. (2000). Predicting the oxidation state of cysteines by multiple sequence alignment. *Bioinformatics*, 16(3):251–256.

Frasconi, P., Gori, M., and Sperduti, A. (1998). A general framework for adaptive processing of data structures. *IEEE Transactions on Neural Networks*, 9(5):768–786.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissu samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Exploration Newsletter*, 5(1):49–58.

Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. (2002). Multi-instance kernels. In Sammut, C. and Hoffmann, A., editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann.

Gärtner, T., Flach, P. A., and Wrobel, S. (2003). On graph kernels : Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop*, pages 129–143.

Gärtner, T., Lloyd, J., and Flach, P. (2004). Kernels and distances for structured data. *Machine Learning*, 57(3):205–232.

Gasteiger, J., Rudolph, C., and Sadowski, J. (1990). Automatic generation of 3d-atomic coordinates for organic molecules. *Tetrahedron Computational Methods*, 3:537–547.

Gers, F., Schraudolph, N., and Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 3:115–143.

Getoor, L., Friedman, N., Koller, D., and Taskar, B. (2001). Learning probabilistic models of relational structure. In *Proc. 18th International Conf. on Machine Learning*, pages 170–177. Morgan Kaufmann, San Francisco, CA.

Goller, C. and Kuechler, A. (1996). Learning task-dependent distributed structure-representations by backpropagation through structure. In *IEEE International Conference on Neural Networks*, pages 347–352.

Goodsell, D. S. and Olson, A. J. (1990). Automated docking of substrates to proteins by simulated annealing. *Proteins: Structure, Function, and Genetics*, 8(3):195–202.

Gregory, D. S., Martin, A. C., Cheetham, J. C., and Rees, A. R. (1993). The prediction and characterization of metal binding sites in proteins. *Protein Engineering*, 6(1):29–35.

Hand, D. J. and Till, R. J. (2001). A simple generalization of the area under the ROC curve to multiple class classification problems. *Machine Learning*, 45(2):171–186.

Hansch, C., Leo, A., and Hoekman, D. H., editors (1995). *Exploring QSAR : hydrophobic, electronic, and steric constants*. American Chemical Society.

Hansch, C., Maolney, P. P., Fujita, T., and Mui, R. M. (1962). Correlation of biological activity of phenoxyacetic acids with hammett substituent constants and partition coefficients. *Nature*, 194:178–180.

Harding, M. M. (2004). The architecture of metal coordination groups in proteins. *Acta Crystallographica*, D60(5):849–859.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, UC Santa Cruz.

Hawkins, J. and Bodén, M. (2005). The applicability of recurrent neural networks for biological sequence analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. In press.

Herbrich, R., Graepel, T., and Obermayer, K. (1999). Support vector learning for ordinal regression. In *Proceedings of the Ninth International Conference on Artificial Neural Networks, 1999.*, volume 1, pages 97 – 102.

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

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1681–1726.

Hochreiter, S., Schmidhuber, J., Frasconi, P., and Bengio, Y. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kolen, J. and Kremer, S. C., editors, *A Field Guide to Dynamical Recurrent Networks*. Wiley-IEEE Press.

Horváth, T., Gärtner, T., and Wrobel, S. (2004). Cyclic pattern kernels for predictive graph mining. In *Proceedings of KDD 04*, pages 158–167. ACM Press.

Hsu, C. W. and Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Neworks*, 13(2):415–425.

Hua, S. and Sun, Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *Journal of Molecular Biology*, 308(2):397–407.

Hulo, N., Sigrist, C. J. A., Saux, V. L., Langendijk-Genevaux, P. S., Bordoli, L., Gattiker, A., Castro, E. D., Bucher, P., and Bairoch, A. (2004). Recent improvements to the PROSITE database. *Nucleic Acids Research*, 32(Database-Issue):134–137.

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1–2):95–114.

Jaakkola, T. S. and Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 487 – 493.

Jebara, T., Kondor, R., and Howard, A. (2004). Probability product kernels. *J. Mach. Learn. Res.*, 5:819–844.

Johnson, D. C., Dean, D. R., Smith, A. D., and Johnson, M. K. (2005). Structure, function, and formation of biological iron-sulfur clusters. *Annual Review of Biochemistry*, 74:247–281.

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202.

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

Karelson, M., Lobanov, V., and Katritzky, A. (1996). Quantum-chemical descriptors in qsar/qspr studies. *Chemical Review*, 96(3):1027–1044.

Karplus, K., Barrett, C., and Hughey, R. (1998). Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14:846–56.

Kelley, L. A., MacCallum, R. M., and Sternberg, M. J. E. (2000). Enhanced genome annotation using structural profiles in the program 3d-pssm. *Journal of Molecular Biology*, 299(2):499–520.

Kier, L. B. and Hall, L. H. (1986). *Molecular Connectivity in Structure-Activity Analysis*. John Wiley.

King, R., Muggleton, S., Srinivasan, A., and Sternberg, M. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenecity by inductive logic programming. *Proceedings of National Academy of Science*, 93:438–442.

Klebe, G. and Mietzner, U. A. T. (1994). Molecular similarity indices in a comparative analysis (CoMSIA) of drug molecules to correlate and predict their biological activity. *Journal of Medicinal Chemistry*, 37(24):4130–4146.

Kramer, S., Lavrač, N., and Flach, P. (2001a). Propositionalization approaches to relational data mining. In *Relational Data Mining*, pages 262–286. Springer-Verlag.

Kramer, S., Raedt, L. D., and Helma, C. (2001b). Molecular feature mining in hiv data. In Provost, F. and Srikant, R., editors, *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining*, pages 136–143. ACM Press.

Kubinyi, H., Folkers, G., and Martin, Y. C., editors (2002). *3D QSAR in Drug Design*. Kluwer.

Lavrač, N. and Dĕzeroski, S. (2001). *Relational Data Mining*. Springer-Verlag.

Leslie, C., Eskin, E., and Noble, W. (2002). The spectrum kernel: A string kernel for SVM protein classification. In *Proc. Pacific Symposium on Biocomputing*, pages 564–575.

Liang, J., Edelsbrunner, H., and Woodward, C. (1998). Anatomy of protein pockets and cavities: Measurement of binding site geometry and implications for ligand design. *Protein Science*, 7:1884–1897.

Liu, D. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528.

Martelli, P., Fariselli, P., Malaguti, L., and Casadio, R. (2002). Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. *Protein Science*, 11(11):2735–2739.

Meiler, J. and Baker, D. (2003). Coupled prediction of protein secondary and tertiary structure. *Proc Natl Acad Sci U.S.A.*, 100(2):12105–12110.

Melo, F., Sanchez, R., and Sali, A. (2002). Statistical potentials for fold assessment. *Prot. Sci.*, 11:430–448.

Menchetti, S., Costa, F., and Frasconi, P. (2005). Weighted decomposition kernels. In *Proceedings of the 22nd International Conference on Machine Learning*.

Micheli, A., Sperduti, A., Starita, A., and Bianucci, A. (2001). Analysis of the internal representations developed by neural networks for structures applied to quantitative structure-activity relationship studies of benzodiazepines. *Journal Chemical Information Computer Science*, 41:202–218.

Mika, S. and Rost, B. (2003). Uniqueprot: creating sequence-unique protein data sets. *Nucleic Acids Res.*, 31(13):3789–3791.

Miyazawa, S. and Jernigan, R. L. (1999). An empirical energy potential with a reference state for protein fold and sequence recognition. *Proteins*, 36:357–369.

Morris, G. M., Halliday, D. S. G. R. S., Huey, R., Hart, W. E., Belew, R. K., J., A., and Olson (1998). Automated docking using a lamarckian genetic algorithm and and empirical binding free energy function. *Journal of Computational Chemistry*, 19:1639–1662.

Mucchielli-Giorgi, M., Hazout, S., and Tuffèry, P. (2002). Predicting the disulfide bonding state of cysteines using protein descriptors. *Proteins*, 46:243–249.

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

Nair, R. and Rost, B. (2005). Mimicking cellular sorting improves prediction of subcellular localization. *J Mol Biol*, 348(1):85–100.

Odone, F., Barla, A., and Verri, A. (2005). Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, 14(2):169–180.

Omlin, C. W. and Giles, C. L. (1996). Constructing deterministic finite-state automata in recurrent neural networks. *Journal of the ACM*, 43(6):937–972.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. (1997). CATH - a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1108.

Paoli, M., Marles-Wright, J., and Smith, A. (2002). Structure-function relationships in heme-proteins. *DNA and Cell Bioliogy*, 21(4):271–280.

Passerini, A. and Frasconi, P. (2004). Learning to discriminate between ligand-bound and disulfide-bound cysteines. *Protein Eng Des Sel*, 17(4):367–373.

Passerini, A., Pontil, M., and Frasconi, P. (2002). From margins to probabilities in multiclass learning problems. In van Harmelen, F., editor, *Proc. 15th European Conf. on Artificial Intelligence*.

Pearlman, D., Case, D., Caldwell, J., Ross, W., Cheatham III, T., DeBolt, S., Ferguson, D., Seibel, G., and Kollman, P. (1995). Amber, a computer program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Computer Physics Communications*, 91:1–41.

Peters, K. P., Fauck, J., and Frömmel, C. (1996). The automatic search for ligand binding sites in proteins of known three-dimensional structure using only geometric criteria. *Journal of Molecular Biology*, 256:201–213.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Smola, A., Bartlett, P., Schölkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*. MIT Press.

Pollastri, G. (2004). Distill: fast, automated predictions of protein residue contacts and backbone coordinates by machine learning. *Proceedings of the CASP6 conference*, in press.

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

Pollastri, G., Baldi, P., Fariselli, P., and Casadio, R. (2002a). Improved prediction of solvent accessibility and number of residue contacts in proteins. *Proteins: Structure, Function and Genetics*, 47(2):142–53.

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

Pollastri, G., Baldi, P., Vullo, A., and Frasconi, P. (2003). Prediction of protein topologies using GIOHMMs and GRNNs. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15*. MIT Press.

Pollastri, G. and McLysaght, A. (2005). Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics*, in Press.

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

Ponder, J. and Case, D. (2003). Force fields for protein simulations. *Advances in Protein Chemistry*, 66:27–85.

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

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Riis, S. K. and Krogh, A. (1996). Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *Journal of Computational Biology*, 3.

Roberts, G. (1999). Nmr spectroscopy in structure-based drug design. *Current Opinions Biotechnology*, 10(1):42–47.

Rosenblat, F. (1958). The perceptron: a probabilistic model for information storage and organisation in the brain. *Psychological Review*, 65:386–408.

Rost, B. (1998). Protein structure prediction in 1d, 2d and 3d. In von Rague-Schleyer, P., Allinger, N. L., T. Clark, J. G., Kollman, P. A., and Schaefer, H. F., editors, *Encyclopedia of Computational Chemistry*, pages 2242–2255. John Wiley, Chisester, Sussex.

Rost, B. and Sander, C. (1993a). Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proc. Natl. Acad. Sci. USA*, 90(16):7558–7562.

Rost, B. and Sander, C. (1993b). Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, 232:584–599.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing*, volume 1. MIT Press.

Sadowski, J., Schwab, C., and Gasteiger, J. (2003). 3d structure generation and conformational searching. In *Computational Medicinal Chemistry and Drug Discovery*, pages 151–212. Dekker Inc.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Schymkowitz, J., Rousseau, F., Martins, I., Ferkinghoff-Borg, J., Stricher, F., and Serrano, L. (2005). Prediction of water and metal binding sites and their affinities by using the fold-x force field. *PNAS*, 102:10147–10152.

Simons, K. T., Ruczinski, I., Kooperberg, C., Fox, B. A., Bystroff, C., and Baker, D. (1999). Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. *Proteins*, 34:82–95.

Sippl, M. J. (1990). Calculation of conformational ensembles from potentials of mean force. *J. Mol. Biol.*, 213:859–883.

Sippl, M. J. (1995). Knowledge-based potentials for proteins. *Curr. Opin. Struct. Biol.*, 5:229–235.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197.

Sodhi, J. S., Bryson, K., McGuffin, L. J., Ward, J. J., Wernisch, L., and Jones, D. T. (2004). Predicting metal-binding site residues in low-resolution structural models. *Journal of Molecular Biology*, 342(1):307–320.

Swamidass, S. J., Chen, J., Bruand, J., Phung, P., Ralaivola, L., and *, P. B. (2005). Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(suppl1):359–368.

Thomson, A. J. and Gray, H. B. (1998). Bio-inorganic chemistry. *Current Opinion Chemical Biology*, 2(2):155–158.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proc. International Conference on Machine Learning*.

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley, New York.

Vendruscolo, M., Kussel, E., and Domany, E. (1997). Recovery of protein structure from contact maps. *Fold. Des.*, 2:295–306.

Vullo, A. and Frasconi, P. (2003). Prediction of protein coarse contact maps. *Journal Bioinformatics and Computational Biology*, 1(2):411–431.

Vullo, A. and Frasconi, P. (2004). Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20(5):653–9.

Wang, G. and Dunbrack Jr., R. (2003). Pisces: a protein sequence culling server. *Bioinformatics*, 19:1589–1591.

Weininger, D. (1988). SMILES, a chemical language and information system: 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36.

Weininger, D. (1989). SMILES: 2. algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, 29(2):97–101.

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

Zemla, A., Venclovas, C., Moult, J., and Fidelis, K. (2001). Processing and evaluation of predictons in CASP4. *Proteins*, 5:13–21.

Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins*, 57:702–710.