UNIVERSITY OF CALIFORNIA,
IRVINE

Machine Learning Algorithms for Protein Structure Prediction

DISSERTATION

submitted in partial satisfication of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Jianlin Cheng

Dissertation Committee:
Professor Pierre Baldi, Chair
Professor Eric Mjolsness
Professor G. Wesley Hatfield

2006

The dissertation of Jianlin Cheng
is approved and is acceptable in quality
and form for publication on microfilm:

Committee Chair

University of California, Irvine
2006

To my wife Liwen Wu and my son Fei Cheng.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES x

ACKNOWLEDGMENTS xii

CURRICULUM VITAE xv

ABSTRACT OF THE DISSERTATION xviii

1 Introduction 1

1.1 Protein Sequence, Structure, and Function 1

1.2 1D, 2D, and 3D Protein Structure Prediction 3

1.3 Machine Learning Algorithms for Protein Structure Prediction 6

1.4 Other Projects 11

2 Prediction of Protein Disordered Regions Using Recursive Neural Networks and Evolutionary Information 13

2.1 Introduction 13

2.2 Methods 15

2.2.1 Data 15

2.2.2 Input and Output of Neural Networks 18

2.2.3 The Architecture of 1-D-Recursive Neural Networks (1D-RNNs) 19

2.3 Results 22

2.4 Conclusion 24

3 Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks 26

3.1 Introduction 26

3.2 Methods 29

3.2.1 Data 29

3.2.2 Inputs and Outputs 31

3.2.3 Post-Processing of the 1D-RNN Output 32

3.3 Results 34

3.4 Conclusions 36

4 Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines 39

4.1 Introduction 39

4.2 Materials and Methods 43

4.2.1 Data 43

4.2.2 Inputs and Encoding Schemes 46

4.2.3 Prediction of Stability Changes Using Support Vector Machines 47

4.3 Results and Discussion 51

4.4 Conclusions 57

5 Large-Scale Prediction of Disulphide Bridges Using Kernel Methods, Two-Dimensional Recursive Neural Networks, and Weighted Graph Matching 58

5.1 Introduction 58
5.1.1 Disulphide Connectivity 58
5.1.2 Overview of Disulphide Connectivity Prediction 60
5.2 Methods 63
5.2.1 Data Preparation 63
5.2.2 Kernel Methods for Chain Classification 66
5.2.3 Recursive Neural Networks to Predict Cysteine Pairing Probabilities 69
5.2.4 Input Specifications 73
5.2.5 Graph Matching to Derive Connectivity from Pairing Probabilities 73
5.3 Results 75
5.3.1 Statistical Analysis 75
5.3.2 Protein Chain Classification 78
5.3.3 Disulphide Bridge Prediction Assuming Knowledge of Bonded Cysteines 80
5.3.4 Disulphide Connectivity Prediction from Scratch 81
5.4 Conclusion 87

6 Three-Stage Prediction of Protein Beta-Sheets by Neural Networks, Alignments, and Graph Algorithms 90

6.1 Introduction 90
6.2 Materials and Methods 93
6.2.1 Data 93
6.2.2 Prediction of β-residue pairs using 2D-RNNs 96
6.2.3 Pseudo-energy for β-strand alignment 99

6.2.4 Prediction of β-sheet topology using graph algorithms 101

6.2.5 Strand pairing prediction benchmark 104

6.3 Results and Discussion 105

6.4 Conclusion 110

7 A Machine Learning Information Retrieval Approach to Protein Fold Recognition 113

7.1 Introduction 113

7.1.1 Classical Approaches to Fold Recognition 114

7.1.2 A Machine Learning Approach to Fold Recognition 116

7.2 Methods 118

7.2.1 Feature Extraction 118

7.2.2 Fold Recognition with Support Vector Machines (SVMs) . . . 124

7.2.3 Training and Benchmarking 125

7.3 Results 125

7.4 Discussion 131

8 Protein Bioinformatics Software and Servers 135

8.1 Introduction 135

8.2 Protein Bioinformatics Software and Servers 136

8.3 Inputs, Outputs, and Performance 139

8.4 Software Structure and System Training 143

Bibliography 144

LIST OF FIGURES

2.1 Frequency of Lengths of Disordered Regions 17

2.2 1D-RNN associated with variables 20

2.3 A 1D-RNN architecture 21

2.4 ROC curve for DISpro on the set of 723 protein chains 23

3.1 Frequency of single and multi-domain chains 30

3.2 Distributions of the lengths of single and multi-domain chains 31

3.3 Example of smoothing applied to the raw output from the 1D-RNN . 33

3.4 Histograms of length distributions of domain boundary regions 34

3.5 Sensitivity vs specificity in CAFASP-4 37

4.1 Classification with SVM 48

4.2 Regression with SVM 49

4.3 The plot of the predicted energy changes for single-site mutations . . 56

5.1 Disulphide bridge connectivity of intestinal toxin 1 59

5.2 General layout of a DAG for two-dimensional objects 70

5.3 Length distribution of sequences containing disulphide bridges 76

5.4 Distribution of the number of disulfide bridges per sequence in the SPX dataset 77

5.5 Distribution of sequential density of disulfide bridges 78

5.6 Distribution of disulphide bridge lengths in the SPX dataset 79

5.7 Predicted bond number versus true bond number 83

6.1 Illustration of inter-strand β-residue pairs and hydrogen-bonding pattern 91

6.2 Statistics plots of β-residue pairs and β-strand pairs 95

6.3 The structure of protein 1VJG 97

6.4 Inter-strand β-residue pairing map of protein 1VJG 98

6.5 Predicted β-residue pairing map of 1VJG 99

6.6 Predicted pseudo-energy matrix and β-sheet assembly process 103

6.7 Schematic diagram of β-sheet topology of protein 1VJG 104

6.8 Strand pairing graph of protein 1VJG 105

6.9 Histogram of pairing distances between linearly adjacent strands . . . 106

6.10 ROC curve of prediction of inter-strand β-residue pairs 108

6.11 Histogram of alignment offsets of antiparallel strand pairs 110

6.12 Histogram of alignment offsets of parallel strand pairs 111

7.1 Specificity-sensitivity plot at the family level 131

7.2 Specificity-sensitivity plot at the superfamily level 132

7.3 Specificity-sensitivity plot at the fold level 133

LIST OF TABLES

2.1 Results for DISpro on 723 non-homologous protein chains 23

2.2 Summary of comparison results for six predictors 24

3.1 CAFASP-4 evaluation results 36

4.1 Results on the SR1496 dataset 52

4.2 Results on the S388 dataset 53

4.3 Results on the SR1135 dataset 53

4.4 Specificity and sensitivity of the SO scheme for helix, strand, and coil 54

4.5 Results on the SR1023 dataset using the SO scheme 55

4.6 Results on the SR1539 dataset using SVM regression 55

5.1 Statistics of bonded and non-bonded cysteines 80

5.2 Protein classification results using kernel methods on the SP41 dataset 81

5.3 Protein classification results using kernel methods on the SPX dataset 82

5.4 Disulphide connectivity prediction on the SP39 dataset 84

5.5 Cysteine bonding state sensitivity and specificity on the SPX dataset 84

5.6 Prediction accuracy for the number of disulphide bridges on the SPX dataset 85

5.7 Results of the disulphide bridge classification on the SPX dataset using different combinations of inputs 86

5.8 Prediction of disulphide bridges with 2D DAG-RNN on the SP41 dataset 87

5.9 The prediction accuracy of disulphide bridge connectivity on the SPX dataset 87

5.10 Performance comparison of the disulphide prediction pipeline 88

7.1 Features used in fold recognition 123

7.2 Twenty top-ranked features using information gain 127

7.3 The sensitivity of 12 methods on the Lindahl's benchmark dataset . . 129

Acknowledgments

I would like to take the opportunity to thank the following people who have supported me through my PhD experience.

First of all, I thank my advisor Pierre Baldi for his several years of patience and guidance. His tireless pursuit of excellence in research, teaching, and scientific writing and presentation is truely inspirational. His remarkable trust and support are essential for the success of my PhD research.

I thank my committee members and collaborators, Eric Mjolsness and G. Wesley Hatfield, for having followed my work and given constructive suggestions to my thesis and for generously supporting my academic career. I also thank Eric Mjolsness for the opportunity of working together on the Sigmoid project.

I thank my coworkers Gianluca Pollastri, Michael Sweredoski, and Arlo Randall. Particularly, I thank Gianluca Pollastri for helping me learn protein structure prediction. I thank Mike and Arlo for the wonderful, cooperative experience.

I thank my coauthors of a number of papers, Alessandro Vullo, Hiroto Saigo, Lucas Scharenbroich, Sam Sanziger, Liza Larsen, Suzanne Sandmeyer, and Richard Lathrop for the fruitful collaborations.

I thank my statistics mentor Hal Stern and my data mining mentor Dennis Decoste. The methods I learned from them are essential tools to conduct my research. I am also indebted to them for their generous support of my academic career.

I also would like to thank Wiley, Springer Science and Business Media, and Oxford University Press for kind permission to include the following materials.

Chapter 2 is mainly based on an article published in "Data Mining and Knowledge Discovery" as:

- J. Cheng, M. Sweredoski, and P. Baldi. Accurate Prediction of Protein Disordered Regions by Mining Protein Structure Data. Data Mining and Knowledge

Discovery, vol. 11, no. 3, pp. 213-222, 2005.

Chapter 3 is mainly based on an article published in "Data Mining and Knowledge Discovery" as:

- J. Cheng, M. Sweredoski, and P. Baldi. DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks. Data Mining and Knowledge Discovery, vol. 13, no. 1, pp. 1-10, 2006.

Chapter 4 is mainly based on an article published in "Proteins" as:

- J. Cheng, A. Randall, and P. Baldi. Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines. Proteins: Structure, Function, Bioinformatics, vol. 62, no. 4, pp. 1125-1132, 2006.

Chapter 5 is mainly based on an article published in "Proteins" as:

- J. Cheng, H. Saigo, and P. Baldi. Large-Scale Prediction of Disulphide Bridges Using Kernel Methods, Two-Dimensional Recursive Neural Networks, and Weighted Graph Matching. Proteins: Structure, Function, Bioinformatics, vol 62, no. 3, pp. 617-629, 2006.

Chapter 6 is mainly based on an article published in "Bioinformatics" as:

- J. Cheng and P. Baldi. Three-Stage Prediction of Protein Beta-Sheets by Neural Networks, Alignments, and Graph Algorithms. Bioinformatics, vol. 21 (suppl 1), pp. i75-84, 2005.

Chapter 7 is mainly based on an article published in "Bioinformatics" as:

- J. Cheng and P. Baldi. A Machine Learning Information Retrieval Approach to Protein Fold Recognition. Bioinformatics, vol. 22, no. 12, pp. 1456-1463, 2006.

Chapter 8 is partially based on an article published in “Nucleic Acids Research” as:

- J. Cheng, A. Randall, M. Sweredoski, and P. Baldi. SCRATCH: a Protein Structure and Structural Feature Prediction Server. Nucleic Acids Research, vol. 33 (web server issue), w72-76, 2005.

Curriculum Vitae

Personal Information

Name:	Jianlin Cheng
Address:	Institute for Genomics and Bioinformatics School of Information and Computer Sciences University of California, Irvine
E-mail:	jianlinc@ics.uci.edu
Web page:	http://www.ics.uci.edu/˜jianlinc/

Education

2002-2006 PhD in Information and Computer Science, University of California, Irvine.

1999-2001 MS in Computer Science, Utah State University, Logan.

1990-1994 BS in Computer Science, Huazhong University of Science and Technology, China

Honors and Awards

- PhD Dissertation Fellowship, School of Information and Computer Sciences, University of California Irvine, 2006.

- Campus Prize for Second Place in the Time Series Competition during the 2005 UC Data Mining Contest. (Out of over 90 teams representing eight University of California campuses)

- Campus Prize for Third Place in the Classification Competition during the 2005 UC Data Mining Contest. (Out of over 90 teams representing eight University of California campuses)

- Guanghua Fellowship, Huazhong University of Science and Technology, China, 1992.

- Outstanding Undergraduate Student Scholarship, Huazhong University of Science and Technology, China, 1990-1994.

Publications

- P. Baldi, J. Cheng, and A. Vullo. Large-Scale Prediction of Disulphide Bond Connectivity. Advances in Neural Information Processing Systems 17 (NIPS 2004), L. Saul,Y. Weiss, and L. Bottou editors, MIT press, pp.97-104, Cambridge, MA, 2005.

- J. Cheng and P. Baldi. Three-Stage Prediction of Protein Beta-Sheets by Neural Networks, Alignments, and Graph Algorithms. Bioinformatics, vol. 21(suppl 1), pp. i75-84, 2005.

- J. Cheng, A. Randall, M. Sweredoski, and P. Baldi. SCRATCH: a Protein Structure and Structural Feature Prediction Server. Nucleic Acids Research, vol. 33 (web server issue), pp. w72-76, 2005.

- J. Cheng, L. Scharenbroich, P. Baldi, and E. Mjolsness. Sigmoid: Towards a Generative, Scalable, Software Infrastructure for Pathway Bioinformatics and Systems Biology. IEEE Intelligent Systems, vol. 20, no. 3, pp. 68-75, 2005.

- J. Cheng, M. Sweredoski, and P. Baldi. Accurate Prediction of Protein Disordered Regions by Mining Protein Structure Data. Data Mining and Knowledge Discovery, vol. 11, no. 3, pp. 213-222, 2005.

- J. Cheng, H. Saigo, and P. Baldi. Large-Scale Prediction of Disulphide Bridges Using Kernel Methods, Two-Dimensional Recursive Neural Networks, and Weighted Graph Matching. Proteins: Structure, Function, Bioinformatics, vol. 62, no. 3, pp. 617-629, 2006.

- J. Cheng, A. Randall, and P. Baldi. Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines. Proteins: Structure, Function, Bioinformatics, vol. 62, no. 4, pp. 1125-1132, 2006.

- S. A. Danziger, S. J. Swamidass, J. Zeng, L. R. Dearth, Q. Lu, J. H. Chen, J. Cheng, V. P. Hoang, H. Saigo, R. Luo, P. Baldi, R. K. Brachmann, and R. H. Lathrop. Functional Census of Mutation Sequence Spaces: The Example of p53 Cancer Rescue Mutants. IEEE Transactions on Computational Biology and Bioinformatics, vol. 3, no. 2, pp. 114-215, 2006.

- J. Cheng, M. Sweredoski, and P. Baldi. DOMpro: Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks. Data Mining and Knowledge Discovery, vol. 13, no. 1, pp. 1-10, 2006.

- J. Cheng and P. Baldi. A Machine Learning Information Retrieval Approach to Protein Fold Recognition. Bioinformatics, vol. 22, no. 12, pp. 1456-1463, 2006

Abstract of the Dissertation

Machine Learning Algorithms for Protein Structure Prediction

By

Jianlin Cheng

Doctor of Philosophy in Information and Computer Science

University of California, Irvine, 2006

Professor Pierre Baldi, Chair

The amino acid sequence of a protein dictates its three dimensional (3D) structure, which determines its function. With the exponential growth of protein sequences without determined 3D structures in the post-genomic era, the prediction of protein structure from sequence has become one of the most fundamental problems in structural and functional bioinformatics.

Protein structure prediction can be classified into three different levels: 1D - predict structural features along one dimensional sequences; 2D - predict relationship between residues; and 3D - predict protein tertiary structure. In this thesis, we present several statistical machine learning methods for predicting protein structure at the three levels.

At the 1D level, we use 1D-Recursive Neural Networks and sequence profiles to predict protein disordered regions and domain boundaries. We also apply support vector machines to predict protein stability changes for single-site mutations using sequence, structure, or both.

At the 2D level, we use 2D-Recursive Neural Networks, alignments, and graph algorithms to predict beta-sheet architecture and disulfide bond connectivity.

At the 3D level, we present a novel machine learning information retrieval framework for protein fold recognition - the key step for template-based 3D structure prediction.

At last, we describe the protein structure prediction software and web servers built on these algorithms. These software and servers are freely available to the scientific community.

Chapter 1

Introduction

1.1 Protein Sequence, Structure, and Function

Protein is a macromolecule composed of 20 different amino acids which are linked by peptide bonds in the linear order (Sanger and Thompson, 1953a,b). The linear polypeptide chain is called the primary structure of the protein. The primary structure can be represented as a sequence of 20 different letters, where each letter denotes an amino acid.

In the native state, the amino acids (or residues) of a protein fold into local secondary structures including alpha helix, beta sheet, and non-regular coil (Pauling and Corey, 1951; Pauling et al., 1951). The secondary structure elements are further packed to form tertiary structure due to hydrophobic forces and side chain interactions between amino acids (Kendrew et al., 1960; Perutz et al., 1960; Dill, 1990). The tertiary structures of several related proteins can bind together to form protein complex called quaternary structure.

In a cell, proteins with native tertiary structures interact to carry out all kinds of biological functions including enzymatic catalysis, transport and storage, coordinated motion, mechanical support, immune protection, generation and transmission of nerve impulses, and control of growth and differentiation (Laskowski et al., 2003). Extensive biochemical experiments (Kendrew et al., 1960; Perutz et al., 1960; Travers, 1989; Bjorkman and Parham, 1990) have shown that a protein's function is determined by its structure. Thus, elucidating a protein's structure is the key to understand its function, which has fundamental significance in biological and medical sciences.

Experimental approaches such as X-ray Crystallography (Bragg, 1975; Blundell and Johnson, 1976) and Nuclear Magnetic Resonance (NMR) spectroscopy (Wuthrich, 1986; Baldwin et al., 1991) are the main techniques to determine protein structure. Since the determination of the first two protein structures–myoglobin and haemoglobin– using X-ray crystallography (Kendrew et al., 1960; Perutz et al., 1960), the number of proteins with solved structures has increased rapidly. Currently, there are about 30,000 proteins with determined structures deposited in the Protein Data Bank (PDB) (Berman et al., 2000). These diverse and abundant structures provide invaluable data for us to understand how a protein folds into its unique 3D structure and to predict the structure from its sequence (Chandonia and Brenner, 2006).

Since the pioneering experiments (Sanger and Thompson, 1953a,b; Kendrew et al., 1960; Perutz et al., 1960; Anfinsen, 1973) showed that a protein's structure is dictated by its sequence, predicting protein structure from its sequence has become one of the most fundamental problems in structural biology. In the post-genomic era, with the application of high-throughput DNA and protein sequencing technologies, the number of protein sequences has increased exponentially, whereas the experimental determination of protein structure is still very expensive, time-consuming, labor-intensive, and sometime impossible. Currently, only about 1.5%

of protein sequences (about 30,000 out of 2 million) have solved structures and the gap between proteins with known structures and with unknown structures is still increasing. Thus, predicting protein structure from its sequence is increasingly imperative and useful. Protein structure prediction software is becoming a vital tool in understanding phenomena in modern molecular and cell biology (Petrey and Honig, 2005) and has important applications in medical sciences such as drug design (Jacobson and Sali, 2004).

1.2 1D, 2D, and 3D Protein Structure Prediction

Protein structure prediction is a very challenging problem. We tackle the problem from different aspects and at different levels. Protein structure prediction is often classified into three levels: 1D, 2D, and 3D (Rost et al., 2003). 1D prediction is to predict the structural features such as secondary structure (Rost and Sander, 1993a,b; Jones, 1999b; Pollastri et al., 2002b) and solvent accessibility (Rost and Sander, 1994; Pollastri et al., 2002a) of each residues along one dimensional protein sequence. 2D prediction is to predict the relationship between residues (e.g., contact map prediction (Fariselli et al., 2001; Pollastri and Baldi, 2002) and disulfide bond prediction (Fariselli and Casadio, 2004; Vullo and Frasconi, 2003; Baldi et al., 2005; Cheng et al., 2006b)). 3D prediction is to predict the 3D coordinates for all residues or all atoms in a protein. Although the ultimate goal is to predict 3D structure, 1D and 2D prediction is of great interest to biologists and is also an important step toward 3D structure prediction.

Two main methodologies of protein structure prediction are *ab-initio* method (Levinthal, 1968; Qian and Sejnowski, 1988; Skolnick and Kolinski, 1991; Prevost et al., 1991; Rost and Sander, 1993b,a; Sali et al., 1994; Baldwin, 1995; Elofsson and et al., 1995; Skolnick and et al., 1997.; Simons et al., 1997; Ortiz et al., 1999; Honig, 1999; Lazaridis and Karplus, 2000; Simons et al., 2001; Baker and Sali, 2001; Zhang et al., 2003) and template-based method (Browne et al., 1969; Blundell et al., 1987; Greer, 1990; Bowie et al., 1991; Taylor, 1991; Jones et al., 1992; Godzik et al., 1992; Levitt, 1992; Sali and Blundell, 1993; Bryant and Lawrence, 1993; Fisher and Eisenberg, 1996; Rost et al., 1997; Karplus et al., 1998; Xu et al., 1998; Marti-Renom et al., 2000; Kelley et al., 2000; Shi et al., 2001; Xu et al., 2003a; Zhou and Zhou, 2004). *Ab-initio* approaches simulate protein folding or structure using physicochemical principles or statistical machinery. The common feature of *ab-initio* methods is to try to predict protein structure without referring to a specific template protein with known structure. *Ab-initio* approaches have been successfully used in 1D structure prediction such as secondary structure prediction (Rost and Sander, 1993a,b; Jones, 1999b; Pollastri et al., 2002b). For 3D structure prediction, the accuracy of *ab-initio* approaches is still very low. So this *de novo* approach is usually applied to proteins without a good structure template in the Protein Data Bank (Berman et al., 2000)

The existing, rather practical, method for 3D structure prediction is the template-based approach including comparative modeling (or homology modeling) and fold recognition (or threading). This approach is based on the observation that nature tends to reuse existing structures/folds to accommodate new protein sequences and functions during the evolution (Chothia, 1992). Thus, although protein sequence space (number of protein sequences) is very large, the protein structure space (the number of unique protein folds) is relatively small and expected to be limited (Grant et al., 2004). Currently, we have collected millions of protein sequences, but the number of unique structures (folds) is only about 1000 (out of about 30000 protein structures) in the protein classification databases such as SCOP (Murzin et al.,

1995) and CATH (Orengo et al., 2002). Moreover, among the newly determined protein structures in the structural genomics projects, the novel folds only account for a small portion (about 10%) and the overall fraction of new folds continues to decrease during the past 15 years (Chandonia and Brenner, 2006).

Thus, most protein sequences, particularly similar protein sequences within the same family and superfamily evolving from the common ancestors, have the similar structures with other proteins. So given a query protein without known structure, template-based prediction is first to identify a template protein–if exists–that has the solved, similar structure with the query protein. And then it uses the structure of the template protein to model the structure of the query protein based on the alignment between the query sequence and the template structure (Browne et al., 1969; Blundell et al., 1987; Greer, 1990; Levitt, 1992; Sali and Blundell, 1993; Koehl and Delarue, 1994; Bates et al., 2001; Petrey et al., 2003; Schwede et al., 2004).

Despite the common feature of using template, comparative modeling traditionally refers to easy modeling case when the query protein has high sequence similarity (>= 40% identity) with the template protein, which can be identified by PSI-BLAST (Altschul et al., 1997); fold recognition refers to hard modeling case when the query protein has less sequence similarity with the template protein and the structural relevance between query and template proteins is harder to recognize. However, with the development of more sensitive fold recognition methods such as sequence-profile alignment (Bailey and Gribskov, 1997; Baldi et al., 1994; Krogh et al., 1994; Hughey and Krogh, 1996; Altschul et al., 1997; Bailey and Gribskov, 1997; Karplus et al., 1998; Eddy, 1998; Park et al., 1998; Koretke et al., 2001; Gough et al., 2001) and profile-profile alignment (Thompson et al., 1994; Rychlewski et al., 2000; Notredame et al., 2000; Yona and Levitt, 2002; Madera and Gough, 2002; Mitelman et al., 2003; Ginalski et al., 2003b; Sadreyev and Grishin, 2003; Edgar and Sjolander, 2003, 2004;

Ohlson et al., 2004; Wallner et al., 2004; Wang and Dunbrack, 2004; Marti-Renom et al., 2004; Söding, 2005), the separation between comparative modeling and fold recognition is blurred (Ginalski, 2006). Now we usually use template-based method to refer to both of them and redefine the term "fold recognition" as a process of identifying a template protein for a query protein no matter how similar their sequences are.

Using both template-based and *ab-initio* methodologies, this thesis presents several machine learning algorithms to tackle the protein structure prediction problems at 1D, 2D, and 3D levels, respectively.

1.3 Machine Learning Algorithms for Protein Structure Prediction

Statistical machine learning approaches provide powerful means to recognize patterns from the vast amount of noisy data and have been widely used in Bioinformatics. This thesis presents a number of novel machine learning methods for 1D structure prediction (protein disordered region, domain, and the stability change of a single point mutation), 2D structure prediction (disulfide bond connectivity and beta-sheet structure), and 3D structure prediction (protein fold recognition). The thesis is organized as follows.

Chapter 2 describes the algorithms of using 1D-Recursive Neural Networks (1D-RNN) to predict protein disordered regions. Intrinsically disordered regions in proteins are relatively frequent and important for our understanding of molecular recognition and assembly, and protein structure and function. From an algorithmic standpoint, flagging large disordered regions is also important for *ab initio* protein structure prediction methods. We first extract a curated, non-redundant, data set of protein disordered regions from the Protein Data Bank and compute relevant statistics on the length and location of these regions. We then develop an *ab initio* predictor of disordered regions called DISpro which uses evolutionary information in the form of profiles, predicted secondary structure and relative solvent accessibility, and ensembles of 1D-recursive neural networks. DISpro is trained and cross validated using the curated data set. The experimental results show that DISpro achieves an accuracy of 92.8% with a false positive rate of 5%.

Chapter 3 uses the similar 1D-RNN architecture to predict protein domain boundary. Protein domains are the structural and functional units of proteins. The ability to parse protein chains into different domains is important for protein classification and for understanding protein structure, function, and evolution. We use machine learning algorithms, in the form of recursive neural networks, to develop a protein domain predictor called DOMpro. DOMpro predicts protein domains using a combination of evolutionary information in the form of profiles, predicted secondary structure, and predicted relative solvent accessibility. DOMpro is trained and tested on a curated dataset derived from the CATH database. DOMpro correctly predicts the number of domains for 69% of the combined dataset of single and multi-domain chains. DOMpro achieves a sensitivity of 76% and specificity of 85% with respect to the single-domain proteins and sensitivity of 59% and specificity of 38% with respect to the two-domain proteins. DOMpro also achieved a sensitivity and specificity of 71% and 71% respectively in the Critical Assessment of Fully Automated Structure Prediction 4 (CAFASP-4) (Fischer et al., 1999; Saini and Fischer, 2005) and was ranked among the top *ab initio* domain predictors.

Chapter 4 uses Support Vector Machine algorithm to predict the stability change of a single site mutation. Accurate prediction of protein stability changes for single amino acid mutations is important for understanding protein structures and designing new proteins. We use support vector machines to predict protein stability changes for single amino acid mutations leveraging both sequence and structural information. We evaluate our approach using cross-validation methods on a large dataset of single amino acid mutations. When only the sign of stability changes is considered, the predictive method achieves 84% accuracy - a significant improvement over previously published results. Moreover, the experimental results show that the prediction accuracy obtained using sequence alone is close to the accuracy using tertiary structure information. Because our method can accurately predict protein stability changes using primary sequence information only, it is applicable to many situations where tertiary structure is unknown, overcoming a major limitation of previous methods which require tertiary information.

Chapter 5 uses kernel methods, 2D-Recursive Neural Networks (2D-RNN), and graph matching algorithms to predict protein disulfide bonds. The formation of disulphide bridges between cysteines plays an important role in protein folding, structure, function, and evolution. We develop new methods for predicting disulphide bridges in proteins. We first build a large curated data set of proteins containing disulphide bridges to extract relevant statistics. We then use kernel methods to predict whether a given protein chain contains intra-chain disulphide bridges or not, and recursive neural networks to predict the bonding probabilities of each pair of cysteines in the chain. These probabilities in turn lead to an accurate estimation of the total number of disulphide bridges and to a weighted graph matching problem that can be addressed efficiently to infer the global disulphide bridge connectivity pattern. This approach can be applied both in situations where the bonded state of each cysteine is known, or in *ab initio* mode where the state is unknown. Furthermore, it can easily cope with chains containing an arbitrary number of disulphide bridges, overcoming one of the major limitations of previous approaches. It can classify individual cysteine residues as bonded or non-bonded with 87% specificity and 89% sensitivity. The estimate for the total number of bridges in each chain is correct 71% of the times, and within one from the true value over 94% of the times. The prediction of the overall disulphide connectivity pattern is exact in about 51% of the chains. In addition to using profiles in the input to leverage evolutionary information, including true (but not predicted) secondary structure and solvent accessibility information yields small but noticeable improvements. Finally, once the system is trained, predictions can be computed rapidly on a proteomic or protein-engineering scale.

Chapter 6 uses alignment, 2D-RNN, and graph algorithms to predict protein beta-residue pairings, beta-strand pairings, and beta-sheet architecture. Protein β-sheets play a fundamental role in protein structure, function, evolution, and bioengineering. Accurate prediction and assembly of protein β-sheets, however, remains challenging because protein β-sheets require formation of hydrogen bonds between linearly distant residues. Previous approaches for predicting β-sheet topological features, such as β-strand alignments, in general have not exploited the global covariation and constraints characteristic of β-sheet architectures. We propose a modular approach to the problem of predicting/assembling protein β-sheets in a chain by integrating both local and global constraints in three steps. The first step uses recursive neural networks to predict pairing probabilities for all pairs of inter-strand β-residues from profile, secondary structure, and solvent accessibility information. The second step applies dynamic programming techniques to these probabilities to derive binding pseudo-energies and optimal alignments between all pairs of β-strands. Finally, the third step, uses graph matching algorithms to predict the β-sheet architecture of the protein by optimizing the global pseudo-energy while enforcing strong global β-strand pairing constraints. The approach is evaluated using cross-validation methods on a large non-homologous dataset and yields significant improvements over previous methods.

Chapter 7 presents a machine learning information retrieval approach to protein fold recognition. Recognizing proteins that have similar tertiary structure is the key step of template-based protein structure prediction methods. Traditionally, a variety of alignment methods are used to identify similar folds, based on sequence similarity and sequence-structure compatibility. Although these methods are complementary, their integration has not been thoroughly exploited. Statistical machine learning methods provide tools for integrating multiple features, but so far these methods have been used primarily for protein and fold *classification*, rather than addressing the *retrieval* problem of fold recognition–finding a proper template for a given query protein. We present a two-stage machine learning, information retrieval, approach to fold recognition. First, we use alignment methods to derive pairwise similarity features for query-template protein pairs. We also use global profile-profile alignments in combination with predicted secondary structure, relative solvent accessibility, contact map, and beta-strand pairing to extract pairwise structural compatibility features. Second, we apply support vector machines to these features to predict the structural relevance (i.e. in the same fold or not) of the query-template pairs. For each query, the continuous relevance scores are used to rank the templates. The FOLDpro approach is modular, scalable, and effective. Compared to 11 other fold recognition methods, FOLDpro yields the best results in almost all standard categories on a comprehensive benchmark dataset. Using predictions of the top-ranked template, the sensitivity is about 85%, 56%, and 27% at the family, superfamily, and fold levels respectively. Using the 5 top-ranked templates, the sensitivity increases to 90%, 70%, and 48%.

Chapter 8 describes a number of protein Bioinformatics software and servers built on the algorithms above. These software and servers are freely available to the scientific community.

1.4 Other Projects

During my PhD I also carried out the following projects.

Sigmoid: a generative, scalable software infrastructure for pathway bioinformatics and systems biology

Sigmoid (Cheng et al., 2005b) is a collaborative project with Professor Eric Mjolsness's group. In this project, we build the computational infrastructure to automate modeling and simulation of biological networks (pathways). We have created an online simulation system called Sigmoid using service oriented architecture (SOA). Sigmoid has four major parts: biological network database, Mathematica simulation engine, web service logic, and GUI, organized in three tiers. SIGMOID is the first system having all three key functions for pathway bioinformatics: storage, simulation, and visualization. In addition to designing the architecture of Sigmoid, I implemented the first version of object-oriented biological network database, the web services for the remote database access and model simulation, and the automated translation of biological networks into Mathematica/Cellerator commands (Shapiro et al., 2002).

Model the structure of ty3 retrovirus

This is a collaborative project with Professor Suzanne Sandmeyer's group. Like HIV-1 virus, ty3 virus is a retrovirus that reproduces DNA from its RNA after infecting the host. The goal of the project is to investigate the structure and function of the ty3 capsid protein that plays a key role in the assembly of ty3 retrovirus envelope. My task is to predict and analyze the structure of the ty3 capsid protein. I predict the secondary structure, solvent accessibility, domains, mutation stability changes, tertiary structures, and quaternary structure for ty3 capsid protein, using the tools developed in my research such as SSpro 4.0 (Cheng et al., 2005a), ACCpro 4.0 (Cheng et al., 2005a), DOMpro (Cheng et al., 2005d), MUpro (Cheng et al., 2006a), BETApro (Cheng and Baldi, 2005) and FOLDpro (Cheng and Baldi, 2006) as well as publicly available tools. The predictions generate several interesting hypotheses for the following site-directed mutagenesis experiments and suggest that ty3 has the similar tertiary structure and assembly mechanism as other retroviruses such as HIV-1. The predicted structure can explain some interesting observations in the experiments conducted by Professor Suzanne Sandmeyer's group (paper in preparation).

Chapter 2

Prediction of Protein Disordered Regions Using Recursive Neural Networks and Evolutionary Information

2.1 Introduction

Proteins are fundamental organic macromolecules consisting of linear chains of amino acids bonded together by polypeptide bonds and folded into complex three-dimensional structures. The biochemical function of a protein depends on its three-dimensional structure, thus solving protein structures is a fundamental goal of structural biology. Tremendous efforts have been made to determine the three-dimensional structures of proteins in the past several decades by experimental and computational methods. Experimental methods such as X-ray diffraction and NMR (Nuclear Magnetic Resonance) spectroscopy are used to determine the coordinates of all the atoms in a protein and thus its three-dimensional structure. While most regions in a protein assume stable structures, some regions are partially or wholly unstructured and do not fold into a stable state. These regions are labeled as disordered regions by structural biologists.

Intrinsically disordered proteins (IDPs) play important roles in many vital cell functions including molecular recognition, molecular assembly, protein modification, and entropic chain activities (Dunker et al., 2002). One of the evolutionary advantages of proteins with disordered regions may be their ability to have multiple binding partners and potentially partake in multiple reactions and pathways. Since the disordered regions may be determined only when the IDPs are in a bound state, IDPs have prompted scientists to reevaluate the structure-implies-function paradigm (Wright and Dyson, 1999). Disordered regions have also been associated with low sequence complexity and an early survey of protein sequences based on sequence complexity predicted that a substantial fraction of proteins contain disordered regions (Wootton, 1994). This prediction has been confirmed to some extent in recent years by the growth of IDPs in the Protein Data Bank (PDB) (Berman et al., 2000), which currently contains about 30,000 proteins and 18,800,000 residues . Thus the relatively frequent occurrence of IDPs and their importance for understanding protein structure/function relationships and cellular processes makes it worthwhile to develop predictors of protein disordered regions. Flagging large disordered regions may also be important for *ab initio* protein structure prediction methods. Furthermore, since disordered regions often hampers crystallization, the prediction of disordered regions could provide useful information for structural biologists and help guide experimental designs. In addition, disordered regions can cause the poor expression of a protein in bacteria, thus making it difficult to manufacture the protein for crystallization or other purposes. Hence, disordered region predictions could provide biologists with important information that would allow them to improve the expression of the protein. For example, if the N or C termini regions were disordered, they could be omitted from the gene.

Comparing disorder predictors can be difficult due to the lack of a precise definition of disorder. Several definitions exist in the literature including loop/coil regions where the carbon alpha (C_α) on the protein backbone has a high temperature factor and residues in the PDB where coordinates are missing as noted in a REMARK465 PDB record (Linding et al., 2003a). Here, consistent with Ward et al. (2004), we define a disordered residue as any residue for which no coordinates exist in the corresponding PDB file.

Previous attempts at predicting disordered regions have used sequence complexity, support vector machines, and neural networks (Wootton, 1994; Dunker et al., 2002; Linding et al., 2003a; Ward et al., 2004). Our method for predicting disordered regions, called DISpro, involves the use of evolutionary information in the form of profiles, predicted secondary structure and relative solvent accessibility, and 1D-recursive neural networks (1D-RNN). These networks are well suited for predicting protein properties and have been previously used in our SCRATCH suite of predictors, including our secondary structure and relative solvent accessibility predictions (Pollastri et al., 2002a,b; Baldi and Pollastri, 2003; Cheng et al., 2005a).

2.2 Methods

2.2.1 Data

The proteins used for the training and testing of DISpro were obtained from the PDB in May 2004. At that time, 7.6% (3,587) of the protein chains in the PDB obtained by X-ray crystallography contained at least one region of disorder at least three residues in length. Most of these disordered regions were short segments near the two ends of protein chains (N- and C- termini).

We first filtered out any proteins that were not solved by X-ray diffraction methods, were less than 30 amino acids in length, or had resolution coarser than 2.5Å. Next, the proteins were broken down into their individual chains. For the creation of our training and testing sets, we selected only protein chains that had sections of disordered regions strictly greater than three residues in length. The determination of residues as being ordered or disordered is based on the existence of an ATOM field (coordinate) for C_α atom of a given residue in the PDB file. If no ATOM records exist for a residue listed in the SEQRES record, the residue is classified as disordered.

We then filtered out homologous protein chains using UniqueProt (Mika and Rost, 2003) with a threshold HSSP value of 10. The HSSP value between two sequences is a measure of their similarity taking into account both sequence identity and sequence length. An HSSP value of 10 corresponds roughly to 30% sequence identity for a global alignment of length 250 amino acids.

Secondary structure and relative solvent accessibility were then predicted for all the remaining chains by SSpro and ACCpro (Pollastri et al., 2002a,b; Baldi and Pollastri, 2003; Cheng et al., 2005a). Using predicted, rather than true secondary structure and solvent accessibility, which are easily-obtainable by the DSSP program (Kabsch and Sander, 1983), introduces additional robustness in the predictor, especially when it is applied to sequences with little or no homology to sequences in the PDB. The filtering procedures resulted in a set of 723 non-redundant disordered chains. To leverage evolutionary information, PSI-BLAST (Altschul et al., 1997) is used to generate profiles by aligning all chains against the Non-Redundant (NR) database, as in (Jones, 1999b; Przybylski and Rost, 2002; Pollastri et al., 2002b). As in the case of secondary structure prediction, profiles rather than primary sequences are used in the input, as explained in next section. Finally, these chains were randomly split into ten subsets of approximately equal size for tenfold cross-validated training and testing. The final dataset is available through: http://www.ics.uci.edu/ baldig/scratch/.

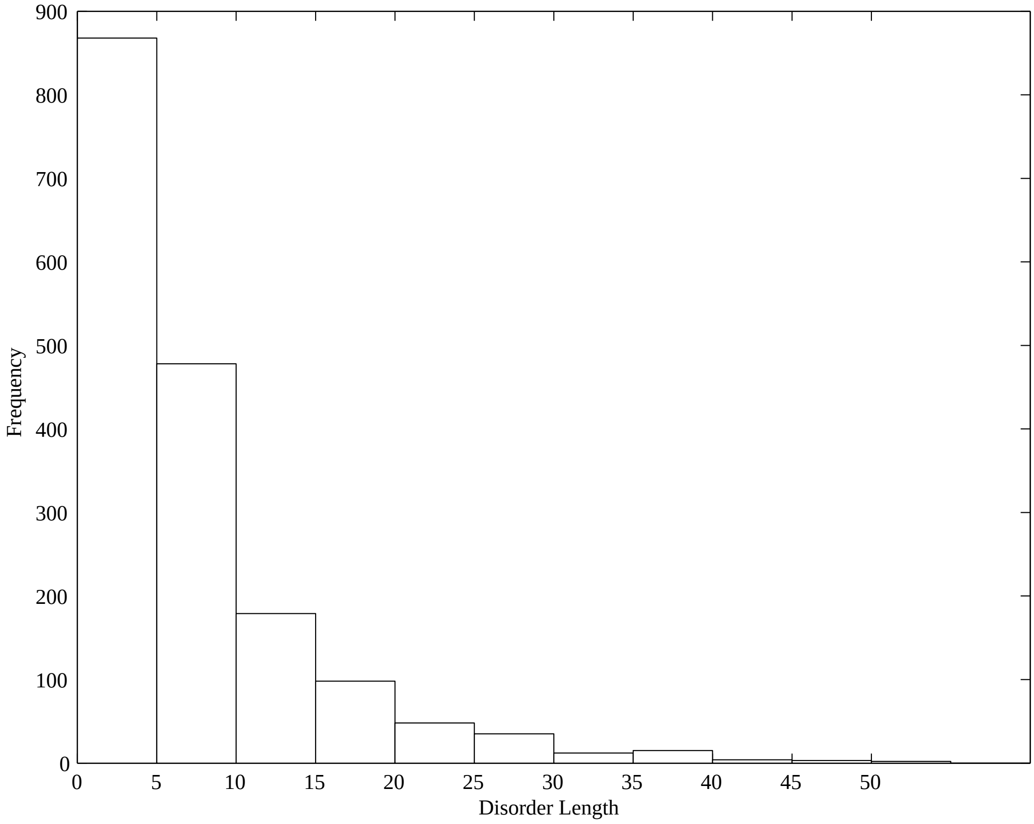


Figure 2.1: Frequency of Lengths of Disordered Regions

The final dataset has 215,612 residues, 6.4% (13,909) of which are classified as disordered. Of the 13,909 disordered residues, 13.8% (1,924) are part of long regions of disorder ($\geq$ 30 AA). Figure 2.1 shows a histogram of the frequency of disordered region lengths in our dataset.

2.2.2 Input and Output of Neural Networks

The problem of predicting disordered regions can be viewed as a binary classification problem for each residue along a one dimensional (1-D) protein chain. The residue at position i is labeled as ordered or disordered. A variety of machine learning methods can be applied to this problem, such as probabilistic graphical models, kernel methods, and neural networks. DISpro employs 1-D recursive neural networks (1D-RNN)(Baldi and Pollastri, 2003). For each chain, our input is the 1-D array I, where the size of I is equal to the number of residues in the chain and each entry I_i is a vector of dimension 25 encoding the profile as well as secondary structure and relative solvent accessibility at position i. Specifically, twenty of the values are real numbers which correspond to the amino acid frequencies in the corresponding column of the profile. The other five values are binary. Three of the values correspond to the predicted secondary structure class (Helix, Strand, or Coil) of the residue and the other two correspond to the predicted relative solvent accessibility of the residue (i.e., under or over 25% exposed).

The training target for each chain is the 1-D binary array T, whereby each T_i equals 0 or 1 depending on whether residue at position i is ordered or disordered. Neural networks (or other machine learning methods) can be trained on the data set to learn a mapping from the input array I onto an output array O, whereby O_i is the predicted probability that residue at position i is disordered. The goal is to make the output O as close as possible to the target T.

2.2.3 The Architecture of 1-D-Recursive Neural Networks (1D-RNNs)

The architecture of the 1D-RNNs used in this study is derived from the theory of probabilistic graphical models, but use a neural network parameterization to speed up belief propagation and learning (Baldi and Pollastri, 2003). 1D-RNNs combine the flexibility of Bayesian networks with the fast, convenient, parameterization of artificial neural networks without the drawbacks of standard feedforward neural networks with fixed input size. Under this architecture, the output O_i depends on the entire input I instead of a local fixed-width window centered at position i. Thus, 1D-RNNs can handle inputs with variable length and allow classification decisions to be made based on contextual long-ranged information outside of the traditional local input window. Since 1D-RNNs use weight sharing in both forward and backward recursive networks, only a fixed number of weights are required to handle propagation of long-ranged information. This is in contrast to local window approaches, where the number of weights (parameters) typically grows linearly with the size of the window, increasing the danger of overfitting. Nevertheless, it is important to recognize that since the problem of disordered region prediction can be formulated as a standard classification problem, other machine learning or data mining algorithms such as feed forward neural networks or support vector machines can in principle be applied to this problem effectively, provided great care is given to the problem of overfitting.

The architecture of the 1D-RNN is described in Figures 2.2 and 2.3 and is associated with a set of input variables I_i, a forward H_i^F and backward H_i^B chain of hidden variables, and a set O_i of output variables. In terms of probabilistic graphical models (Bayesian networks), this architecture has the connectivity pattern of an input-output HMM (Bengio and Frasconi, 1996), augmented with a backward chain of hidden states. The backward chain is of course optional and used here to capture the spatial, rather than temporal, properties of biological sequences.

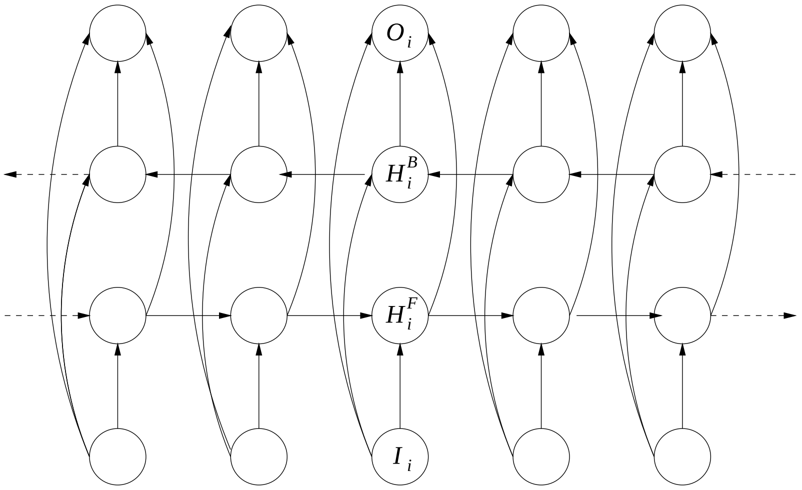


Figure 2.2: 1D-RNN associated with input variables, output variables, and both forward and backward chains of hidden variables.

The relationship between the variables can be modeled using three separate neural networks to compute the output, forward, and backward variables respectively. These neural networks are replicated at each position i; (i.e., weight sharing). One fairly general form of weight sharing is to assume stationarity for the forward, backward, and output networks, which leads to a 1D-RNN architecture, previously named a bidirectional RNN architecture (BRNN), and is implemented using three neural networks $\mathcal{N}_O$, $\mathcal{N}_F$, and $\mathcal{N}_B$ in the form

$$\begin{aligned} O_i &= \mathcal{N}_O(I_i, H_i^F, H_i^B) \\ H_i^F &= \mathcal{N}_F(I_i, H_{i-1}^F) \\ H_i^B &= \mathcal{N}_B(I_i, H_{i+1}^B) \end{aligned} \tag{2.1}$$

as depicted in Figure 2.3. In this form, the output depends on the local input I_i

at position i, the forward (upstream) hidden context $H_i^F \in \mathbb{R}^n$ and the backward (downstream) hidden context $H_i^B \in \mathbb{R}^m$, with usually $m = n$. The boundary conditions for H_i^F and H_i^B can be set to 0, i.e. $H_0^F = H_{N+1}^B = 0$ where N is the length of the sequence being processed. Alternatively these boundaries can also be treated as a learnable parameter. Intuitively, we can think of $\mathcal{N}_F$ and $\mathcal{N}_B$ in terms of two "wheels" that can be rolled along the sequence. For the prediction at position i, we roll the wheels in opposite directions starting from the N- and C- terminus and up to position i. We then combine the wheel outputs at position i together with the input I_i to compute the output prediction O_i using $\mathcal{N}_O$.

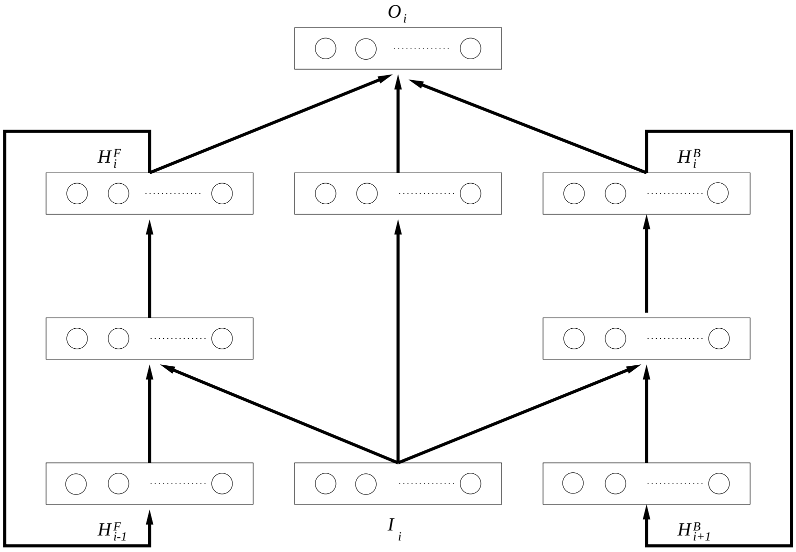


Figure 2.3: A 1D-RNN architecture with a left (forward) and right (backward) context associated with two recurrent networks (wheels).

The output O_i for each residue position i is computed by two normalized-exponential units, which is equivalent to one logistic output unit. The error function is the relative entropy between the true distribution and the predicted distribution.

All the weights of the 1D-RNN architecture, including the weights in the recurrent wheels, are trained in supervised fashion using a generalized form of gradient descent on the error function, derived by unfolding the wheels in space. To improve the statistical accuracy, we average over an ensemble of five trained models to make prediction.

2.3 Results

We evaluate DISpro using ten-fold cross validation on the curated dataset of 723 non-redundant protein chains. The resulting statistics for DISpro are given in Table 2.1, including a separate report for the special subgroup of long disordered regions($>$ 30 residues), which have been shown to have different sequence patterns than N- and C- termini disordered regions (Li et al., 1999). Performance is assessed using a variety of standard measures including correlation coefficients, area under the ROC curves, Accuracy at 5% FPR (False Positive Rate), Precision [TP/(TP+FP)], and Recall [TP/(TP+FN)]. The accuracy at 5% FPR is defined as [(TP+TN)/(TP+FP+TN+FN)] when the decision threshold is set so that 5% of the negative cases are above the decision threshold. Here, TP, FP, TN, and FN refer to the number of true positives, false positives, true negatives, and false negatives respectively.

The area under the ROC curve of DISpro computed on all regions is .878. An ROC area of .90 is generally considered a very accurate predictor. An area of 1.00 would correspond to a perfect predictor and an area of .50 would correspond to a random predictor. At 5% FPR, the TPR is 92.8% for all disordered regions. DISpro achieves a precision and recall rate of 75.4% and 38.8% respectively, when the decision threshold is set at .5. Figure 2.4 shows the ROC curves of DISpro corresponding to all disordered regions and to disordered regions 30 residues or more in length. It shows that the long disordered regions are harder to predict than

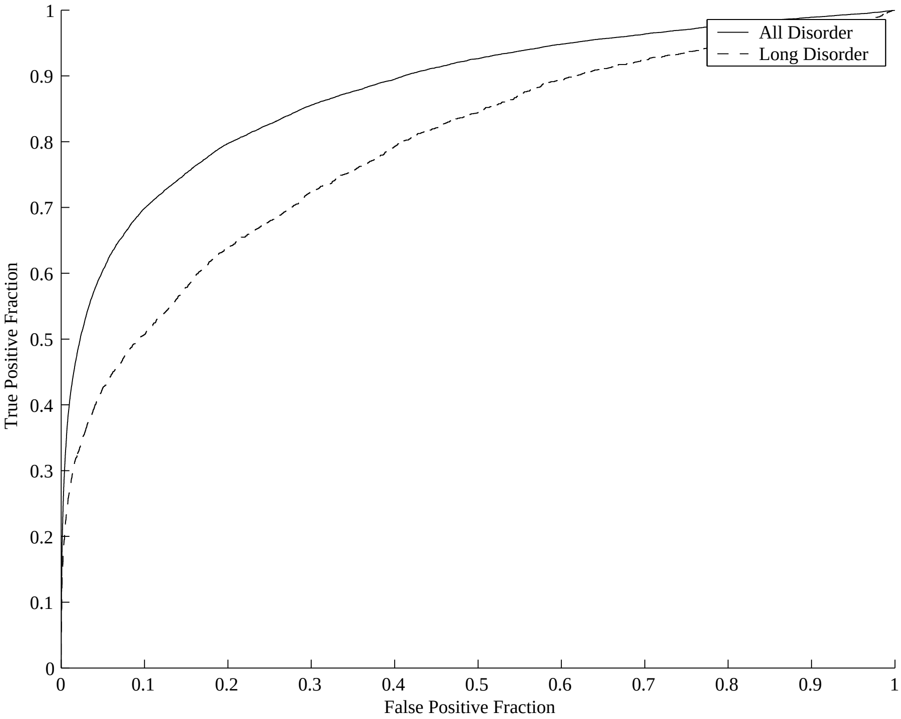


Figure 2.4: ROC curve for DISpro on the set of 723 protein chains

the shorter disordered regions.

We have also compared our results to those of other predictors from CASP5 (Ward et al., 2004) (Critical Assessment of Structure Prediction). The set of proteins from CASP5 should be considered a fair test since each chain had a low HSSP score (< 7) in comparison to our training set. Table 2.2 shows our results in comparison to other predictors. DISpro achieves an ROC area of 0.935, better than all the

Table 2.1: Results for DISpro on 723 non-homologous protein chains

Dataset	Corr. Coef.	ROC area	Accuracy (5% FPR)	Precision	Recall
All disorder	0.589	0.878	92.8%	75.4%	38.8%
Long disorder ($\geq$ 30 AA)	0.255	0.789	94.5%	22.1%	25.9%

Table 2.2: Summary of comparison results for six predictors using the proteins from CASP5. Results for predictors other than DISpro were reported by Ward et al., 2004.

Predictor	Corr. Coef.	ROC area	Accuracy
			(5% FPR)
DISpro	0.51	0.935	93.2%
DISOPRED2	0.52	0.900	93.1%
Dunker VLXT	0.31	0.809	91.4%
Dunker VL2	0.36	0.786	91.8%
Obradovic VL3	0.38	0.801	92.1%
FoldIndex	0.26	0.738	91.0%

other predictors. The correlation coefficient of DISpro is 0.51, roughly the same as DISOPRED2. The accuracy of DISpro at a 5% FPR is 93.2% on the CASP5 protein set. Thus, on the CASP5 protein set, DISpro is roughly equal or slightly better than all the other predictors on all three performance measures. DISOPRED2 and DISpro performance appear to be similar and significantly above all other predictors.

2.4 Conclusion

DISpro is a predictor of protein disordered regions which relies on machine learning methods and leverages evolutionary information as well as predicted secondary structure and relative solvent accessibility. Our results show that DISpro achieves an accuracy of 92.8% with a false positive rate of 5% on large cross-validated tests. Likewise, DISpro achieves an ROC area of 0.88.

There are several directions for possible improvement of DISpro and disordered region predictors in general that are currently under investigation. To train better models, larger training sets of proteins with disordered regions can be created as new proteins are deposited in the PDB. In addition, protein sequences containing no disordered regions, which are excluded from our current experiment, may also be included in the training set to decrease false positive rate.

Our results confirm that short and long disordered region behave differently and therefore it may be worth training two separate predictors. In addition, it is also possible to train a separate predictor to detect whether a give protein chain contains any disordered regions or not using another machine learning technique, such as kernel methods, for classification, as is done for proteins with or without disulphide bridges (Frasconi et al., 2002). Results derived from contact map predictors (Baldi and Pollastri, 2003) may also be used to try to further boost the prediction performance. It is reasonable to hypothesize that disordered regions ought to have poorly defined contacts. We are also in the process of adding to DISpro the ability to directly incorporate disorder information from homologous proteins. Currently, such information is only used indirectly by the 1D-RNNs. Prediction of disordered regions in proteins that have a high degree of homology to proteins in the PDB should not proceed entirely from scratch but leverage the readily available information about disordered regions in the homologous proteins. Large disordered regions may be flagged and be removed or treated differently in *ab initio* tertiary structure prediction methods. Thus it might be useful to incorporate disordered region predictions into the full pipeline of protein tertiary structure prediction. Finally, beyond disorder prediction, bioinformatics integration of information from different sources may shed further light on the nature and role of disordered regions. In particular, if disordered regions act like reconfigurable switches allowing certain proteins to partake in multiple interactions and pathways, one might be able to cross-relate information from pathway and/or protein-protein interaction databases with protein structure databases.

Chapter 3

Protein Domain Prediction Using Profiles, Secondary Structure, Relative Solvent Accessibility, and Recursive Neural Networks

3.1 Introduction

Domains are considered the structural and functional units of proteins. They can be defined using multiple criteria, or combinations of criteria, including evolutionary conservation, discrete functionality, and the ability to fold independently (Holm and Sander, 1994). A domain can span an entire polypeptide chain or be a subunit of a polypeptide chain that can fold into a stable tertiary structure independently of any other domain (Levitt and Chothia, 1976). While typical domains consist of a single continuous polypeptide segment, some domains may be comprised of several discontinuous segments.

The identification of domains is an important step for protein classification and for the study and prediction of protein structure, function, and evolution. The topology of secondary structure elements in a domain is used by human experts or automated systems in structural classification databases such as FSSP-Dali Domain Dictionary (Holm and Sander, 1998a,b), SCOP (Murzin et al., 1995), and CATH (Orengo et al., 2002). The prediction of protein tertiary structure, especially *ab initio* prediction, can be improved by segmenting the protein using the putative domain boundaries and predicting each domain independently (Chivian et al., 2003). However, the identification of protein domains based on sequence alone remains a challenging problem.

A number of methods have been developed to identify protein domains starting from their primary sequence. These methods can be roughly classified into three categories: template based methods (Chivian et al., 2003; Heger and Holm, 2003; Marsden et al., 2002; von Ohsen et al., 2004; Zdobnov and Apweiler, 2001; Gewehr and Zimmer, 2005), non-template based (*ab initio*) methods (Bryson et al., 2005; George and Heringa, 2002; Lexa and Valle, 2003; Linding et al., 2003b; Liu and Rost, 2004; Nagarajan and Yona, 2004; Wheelan et al., 2000), and meta domain prediction methods (Saini and Fischer, 2005). Some template-based methods use a sequence alignment approach where domains are identified by aligning the target sequence against sequences in a domain classification database (Marchler-Bauer et al., 2003). Other methods use alignments of secondary structures (Marsden et al., 2002). In these methods, domains are assigned by aligning the predicted secondary structure of a target sequence against the secondary structure of chains in CATH, which have known domain boundaries.

Some *ab initio* methods, such as tertiary structure folding approaches, average several hundred predictions obtained from coarse *ab initio* simulations of protein folding to assign domain boundaries to a given sequence (George and Heringa, 2002). One drawback of these approaches is that they are computationally intensive. Other *ab initio* use a statistical approach, such as Domain Guess by Size (Wheelan et al., 2000), to predict the likelihood of domain boundaries within a given sequence based on the distributions of chain and domain lengths.

The *ab initio* prediction of domains using machine learning techniques is aided by the availability of large, high quality, domain classification databases such as CATH, SCOP and FSSP-Dali Domain Dictionary. Two recently published algorithms attempt to predict domain boundaries using neural networks (Nagarajan and Yona, 2004; Liu and Rost, 2004). The networks used by Nagarajan and Yona (2004) incorporate the position specific physio-chemical properties of amino acid and predicted secondary structure. Liu and Rost (2004) use neural networks with amino acid composition, positional evolutionary conservation, as well as predicted secondary structure and solvent accessibility.

Here we describe DOMpro, an *ab initio* machine learning approach for predicting domains, which uses profiles along with predicted secondary structure and solvent accessibility in a 1D-recursive neural network (1D-RNN). These networks are also used for the prediction of secondary structure and solvent accessibility (Pollastri et al., 2002b,a) in the SCRATCH suite of servers (Baldi and Pollastri, 2003; Cheng et al., 2005a). Unlike previous neural network-based approaches (Liu and Rost, 2004; Nagarajan and Yona, 2004), the direct use of profiles in DOMpro is based on the assumption that sequence motifs and their level of conservation in the boundary regions are different from those found in the rest of the protein. The final assignment of protein domains is the result of post-processing and statistical inference on the output of the 1D-RNN.

3.2 Methods

3.2.1 Data

DOMpro is trained and tested on a curated dataset derived from the annotated domains in the CATH domain database, version 2.5.1. Because the CATH database contains only the sequences of domain regions, sequences from the Protein Data Bank (PDB)(Berman et al., 2000) must be incorporated to reconstruct entire chains. Once the chains are reconstructed, short sequences (< 40 residues) are filtered out.

UniqueProt (Mika and Rost, 2003) is then used to reduce sequence redundancy in the dataset by ensuring that no pair of sequences have a HSSP value greater than 5. The HSSP value between two sequences is a measure of their similarity and takes into account both sequence identity and sequence length. A HSSP value of 5 corresponds roughly to a sequence identity of 25% in a global alignment of length 250.

Finally, the secondary structure and relative solvent accessibility are predicted for each chain using SSpro and ACCpro (Baldi and Pollastri, 2003; Pollastri et al., 2002b,a). Using predicted secondary structure and solvent accessibility values rather than the true values, which can be easily obtained using the DSSP program (Kabsch and Sander, 1983), gives us a more realistic and objective evaluation since the actual secondary structure and solvent accessibility are not known during the prediction phase. To leverage evolutionary information, PSI-BLAST (Altschul et al., 1997) is used to generate profiles by aligning all chains against the Non-Redundant (NR) database, as in other methods (Jones, 1999b; Przybylski and Rost, 2002; Pollastri et al., 2002b).

After redundancy reduction, our curated dataset contained 354 multi-domain chains and 963 single-domain chains. The ratio of single to multi-domain chains

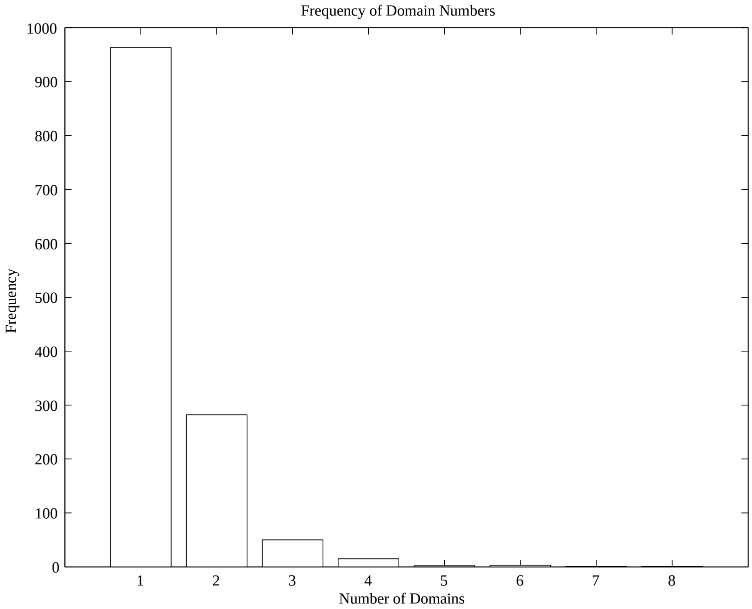


Figure 3.1: Frequency of single and multi-domain chains in the redundancy-reduced dataset.

reflects the skewed distribution of single-domain chains in the PDB. Figure 3.1 shows the frequency of single and multi-domain chains in the redundancy-reduced dataset. Figure 3.2 shows the distribution of chain lengths among single and multi-domain chains.

Because the recursive neural networks are trained to recognize domain boundaries, only multi-domain proteins are used during the training process. During the training and testing of the neural networks on multi-domain proteins, ten fold cross-validation is used. Additional testing is performed on single-domain proteins using models trained with multi-domain proteins.

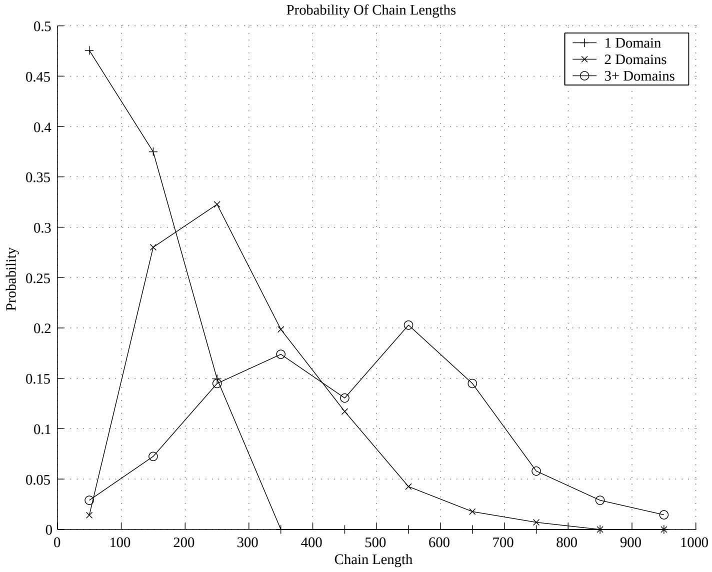


Figure 3.2: Distributions of the lengths of single and multi-domain chains in the redundancy-reduced dataset.

3.2.2 The Inputs and Outputs of the One Dimensional Recursive Neural Network

The problem of predicting domain boundaries can be viewed as a binary classification problem for each residue along a one-dimensional protein chain. Each residue is labeled as being either a domain boundary residue or not.

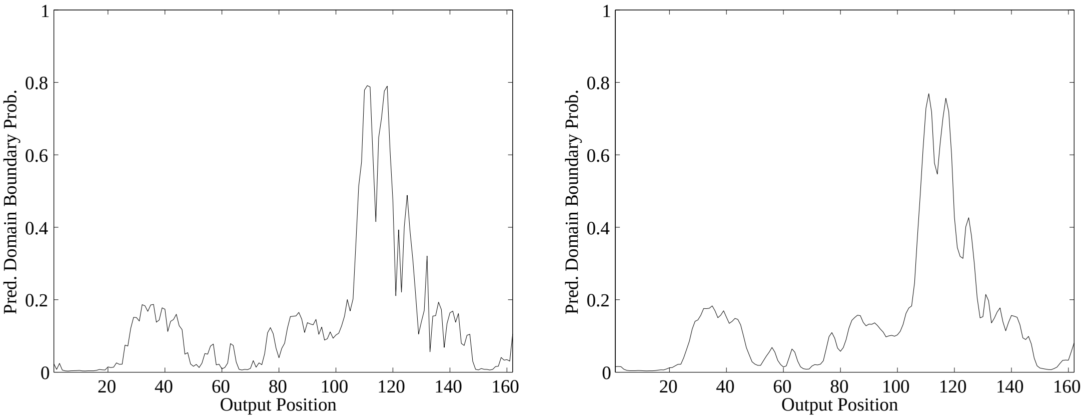
Specifically, the target class for each residue is defined as follows. Following the conventions used in prior domain boundary prediction papers (Liu and Rost, 2004; Marsden et al., 2002), residues within 20 amino acids of a domain boundary are considered domain boundary residues and all other residues are considered non-boundary residues. A variety of machine learning methods can be applied to this classification problem, such as probabilistic graphical models, kernel methods, and neural networks. DOMpro employs 1-D recursive neural networks (1D-RNNs) (Baldi and Pollastri, 2003), which have been applied successfully in the prediction of secondary structure, solvent accessibility, and disordered regions (Cheng et al., 2005c; Pollastri et al., 2002b,a). For each chain, the input is the array I, where the length of I is equal to the number of residues in the chain. Each element I_i is a vector with 25 components, which encodes the profile as well as secondary structure and relative solvent accessibility at position i. Twenty components of the vector I_i are real numbers corresponding to the amino acid profile probabilities. The other five components are binary: three correspond to the predicted secondary structure class of the residue (Helix, Strand, or Coil) and two correspond to the predicted relative solvent accessibility of the residue (i.e., under or over 25% exposed).

The training target for each chain is the 1-D binary array T, where each T_i equals 1 or 0 depending on whether or not the residue at position i is within a boundary region. Neural networks (and most other machine learning methods) can be trained on the dataset to learn a mapping from the input array I onto an output array O, where O_i is the predicted probability that the residue at position i is within a domain boundary region. The goal is to make the output O as close as possible to the target T.

3.2.3 Post-Processing of the 1D-RNN Output

The raw output from the 1D-RNN is quite noisy (see Figure 3.3). DOMpro uses smoothing to help correct for the random noise that is the result of false positive hits. The smoothing is accomplished by averaging over a window of length three around each position. Figure 3.3 shows how this smoothing technique helps to reduce the noise found in the raw output of the 1D-RNN. After smoothing, a domain state (boundary/not boundary) is assigned to each residue by thresholding the network's output at 0.5.

While smoothing the neural network's output helps correct for random spikes, it



Raw output from 1D-RNN

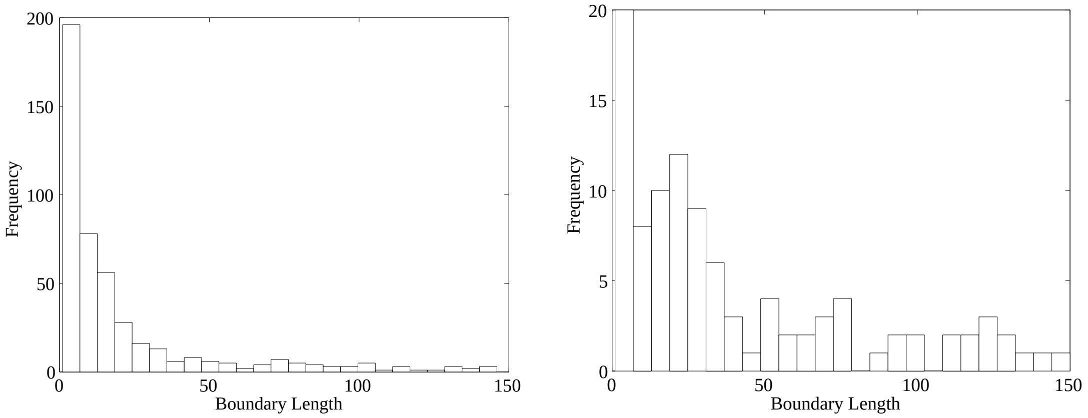
Smoothed output from 1D-RNN
with window of width 3

Figure 3.3: Example of smoothing applied to the raw output from the 1D-RNN

does not necessarily create the long, continuous segments of boundary residues that are required for domain assignment. Therefore, further inference on the output is required.

DOMpro infers the domain boundary regions from the residues predicted as domain boundaries by pattern matching on the discretized output. Any section of the output that matches the regular expression pattern $((B+N\{0,m\})+B+)$ is considered a domain boundary region, where B is a predicted boundary residue, N is a predicted non-boundary residue and m is the maximum separation between two boundary residues that should be merged into one region.

Once DOMpro has inferred all possible domain boundary regions, it needs to identify false positive domain boundary regions. DOMpro considers the boundary region's length a measure of its signal strength. Figure 3.4 shows that there is a clear difference between the length distributions of true domain boundary regions and false domain boundary regions. Based on these statistics, domain boundary regions shorter than three residues are considered false positive hits and are ignored. The target sequence is then cut into domain segments at the middle residue of



False positive boundaries

True boundaries

Figure 3.4: Histograms of length distributions for false positive and true positive boundary regions

each boundary region. A target sequence with no predicted domain boundaries is classified as a single-domain chain. The final step of DOMpro is to assign domain numbers to each predicted domain segment. Our method simply assigns each domain segment to a separate domain, ignoring at this time the relatively rare problem of non-contiguous domains.

3.3 Results

The evaluation and comparison of domain predictors is complicated by the existence of several domain datasets/databases that sometimes conflict with each other (Liu and Rost, 2004). Thus, the performance of a predictor on a dataset other than its training dataset is limited by the percentage of agreement between the training and testing datasets. With this caveat in mind, we observe that DOMpro correctly predicts the number of domains for 69% of the combined dataset of single and multi-domain proteins. DOMpro achieves a sensitivity of 76% and specificity of 85% with respect to the single-domain proteins and sensitivity of 59% and specificity of 38%

with respect to the two-domain proteins.

The precise prediction of domain boundaries for multi-domain proteins is more difficult than the prediction of the number of domains (domain number). DOMpro is able to correctly predict the domain number and boundary for 25% of the two-domain proteins in our dataset derived from CATH. Additionally, DOMpro is able to correctly predict both the domain number and domain boundary location for 20% of the multi-domain chains. For the evaluation of multi-domain chains, we consider that a domain boundary has been correctly identified if the predicted domain boundary is within 20 residues of the true domain boundary as annotated in the CATH database. This definition is consistent with previous work (Marsden et al., 2002).

DOMpro was independently evaluated along with 12 other predictors in the Critical Assessment of Fully Automated Structure Prediction 4 (CAFASP-4)(Fischer et al., 1999; Saini and Fischer, 2005). The results, kindly provided by Dr. Saini, are available at *http://cafasp4.bioinformatics.buffalo.edu/dp/update.html.* The evaluation set consisted of 41 single-domain CASP6 targets and 17 two-domain CASP6 targets (58 targets in total). Since this evaluation set contains only comparative modeling and fold recognition targets (no new fold targets), predictors based on templates have an advantage in this evaluation. DOMpro achieved a higher sensitivity and specificity than one method that uses homologous information and all other *ab initio* predictors including Armadillo, Biozon (Nagarajan and Yona, 2004), Dompred-DPS (Bryson et al., 2005), Globplot (Linding et al., 2003b), and Mateo (Lexa and Valle, 2003), averaged over all of the targets (See Table 3.1 and Figure 3.5). However, the performance of the top three *ab initio* predictors (DOMpro, Globplot, and Dompred-DPS) is close. The specificity and sensitivity of DOMpro is 4-5% higher than the template-based method ADDA (Heger and Holm, 2003),

Table 3.1: CAFASP-4 evaluation results

Predictor	1-D Sen.	1-D Spec.	2-D Sen.	2-D Spec.	All Sen.	All Spec.
DOMpro	0.85	0.76	0.35	0.50	0.71	0.71
ADDA †‡	0.85	0.73	0.18	0.33	0.66	0.67
Armadillo †	0.10	1.00	0.24	0.18	0.14	0.31
Biozon †	0.10	1.00	0.35	0.19	0.17	0.29
Domssea ‡	0.80	0.75	0.29	0.63	0.66	0.73
Dompred-DPS †	0.68	0.78	0.47	0.50	0.62	0.69
Dopro ‡	0.85	0.88	0.53	0.64	0.76	0.81
Globplot †	0.83	0.71	0.18	0.60	0.64	0.70
InterProScan ‡	0.93	0.75	0.24	0.67	0.72	0.74
Mateo †	0.51	0.78	0.12	0.15	0.40	0.58
SSEP-Domain ‡	0.93	0.84	0.47	0.73	0.79	0.82
RobettaGinzu ‡	0.80	0.92	0.53	0.69	0.72	0.86
Rosettadom ‡	0.83	0.94	0.71	0.75	0.79	0.88

† had lower sensitivity and specificity averaged over all targets compared to DOMpro

‡ template based methods

similar to Domssea (Marsden et al., 2002), and lower than other template-based methods such as Dopro (von Ohsen et al., 2004), InterProScan (Zdobnov and Apweiler, 2001), SSEP-Domain (Gewehr and Zimmer, 2005), and RobettaGinzu (Chivian et al., 2003).

3.4 Conclusions

We have created DOMpro, an *ab initio* predictor of protein domains using a recursive neural network that leverages evolutionary information in the form of profiles and predicted secondary structure and relative solvent accessibility. The raw output of the 1D-RNN in DOMpro goes through a post-processing procedure to produce the final domain segmentation and assignment. In the CAFASP-4 evaluation, DOMpro was ranked among the top *ab initio* domain predictors.

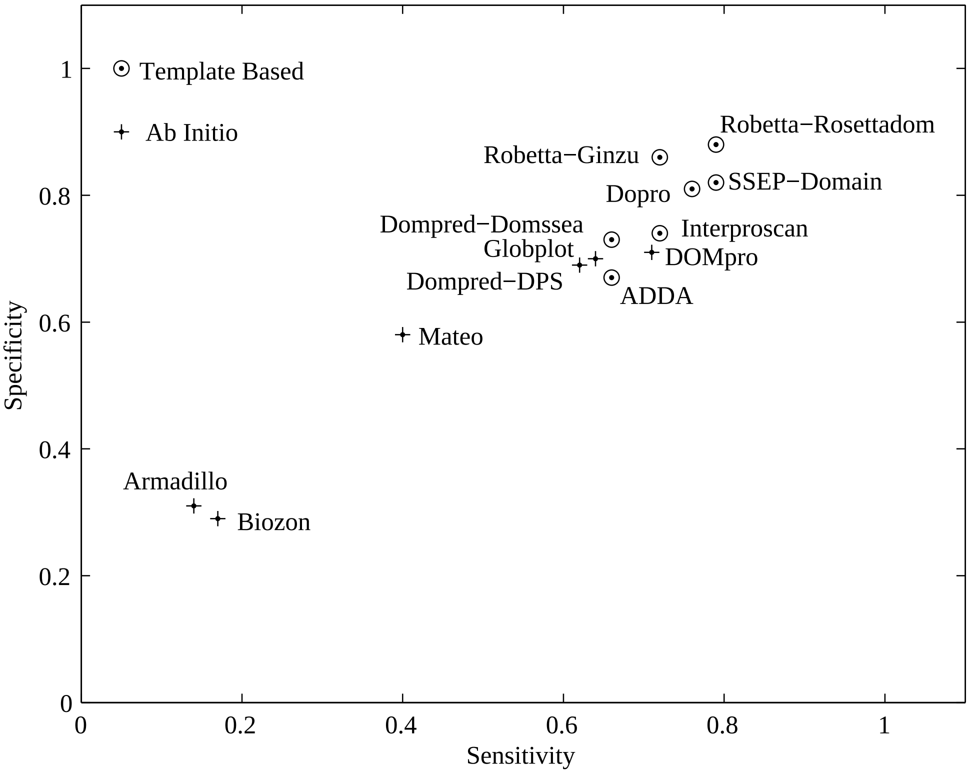


Figure 3.5: Sensitivity vs specificity in CAFASP-4

Despite recent advances, domain prediction remains a challenge. A 25% accuracy on the prediction of two-domain proteins is encouraging but not sufficient for most applications and clearly there is room for improvement. We are currently adding a module to DOMpro which would incorporate known domain assignments for proteins that are homologous to structures in the PDB and CATH databases. We are also training ensembles of predictors, although preliminary experiments so far have not lead to significant improvements. In addition, we are focusing on the prediction/classification of discontinuous domains. To overcome the current limitations of DOMpro and the naive assignment of domain numbers, we are experimenting with the use of predicted contact maps, as well as domain length statistics, in the assignment of domain boundaries and the creation of domains consisting of multiple non-adjacent domain segments. The contact maps are predicted using 2D-RNNs (Baldi and Pollastri, 2003; Pollastri and Baldi, 2002). The basic idea is that domains should be associated with a relatively higher density of contacts. Following this logic, two discontinuous segments having the proper length statistics and a sufficient number of inter-segment residue-residue contacts would be predicted to be in the same domain.

Chapter 4

Prediction of Protein Stability Changes for Single-Site Mutations Using Support Vector Machines

4.1 Introduction

Single amino acid mutations can significantly change the stability of a protein structure. Thus, biologists and protein designers need accurate predictions of how single amino acid mutations will affect the stability of a structure (Dahiyat, 1999; DeGrado, 1999; Street and Mayo, 1999; Saven, 2002; Mendes et al., 2002; Bolon et al., 2003; Looger et al., 2003). The energetics of mutants has been studied extensively both through theoretical and experimental approaches. The methods for predicting protein stability changes resulting from single amino acid mutations can be classified into four general categories: (1) physical potential approach; (2) statistical potential approach; (3) empirical energy function approach; and (4) machine learning approach (Capriotti et al., 2004). The first three categories are similar in that they all rely on energy functions (Guerois et al., 2002). Physical potential approaches (Bash et al., 1987; Dang et al., 1989; Prevost et al., 1991; Tidor and Karplus, 1991; Lee and Levitt, 1991; Miyazawa and Jernigan, 1994; Lee, 1995; Pitera and Kollman, 2000) directly simulate the energy of the atomic force fields present in a given structure and, as such, remain too computationally intensive to be applied on large datasets. (Guerois et al., 2002) Statistical potential approaches (Lee, 1995; Sippl, 1995; Gilis and Rooman, 1997; Topham et al., 1997; Gilis and Rooman, 1999; Carter et al., 2001; Kwasigroch et al., 2002; Zhou and Zhou, 2002, 2004) derive potential functions using statistical analysis of environmental propensities, substitution frequencies, and correlations of contacting residues in solved tertiary structures. Statistical potential approaches achieve predictive accuracy comparable to physical potential approaches. (Lazaridis and Karplus, 2000)

The empirical energy function approach (Guerois et al., 2002; Villegas et al., 1996; Munoz and Serrano, 1997; Lacroix et al., 1998; Takano et al., 1999; Domingues et al., 2000; Taddei et al., 2000; Funahashi et al., 2001; Bordner and Abagyan, 2004) derives an energy function of weighted combinations of physical energy terms, statistical energy terms, and structural descriptors, and by fitting the function to the experimental energy data. From a data fitting perspective, both machine learning methods (Capriotti et al., 2004; Casadio et al., 1995; Frenz, 2005) and empirical potential methods learn a function for predicting energy changes from an experimental energy dataset. However, instead of fitting a linear combination of energy terms, machine learning approaches can learn more complex nonlinear functions of input mutation, protein sequence, and structure information. This is desirable for capturing complex local and non-local interactions that affect protein stability. Machine learning approaches such as support vector machines (SVM) and neural networks are more robust in their handling of outliers than linear methods, thus, explicit outlier detection used by empirical energy function approaches (Guerois et al., 2002) is unnecessary. Furthermore, machine learning approaches are not limited to using energy terms; they can readily leverage all kinds of information relevant to protein stability. With suitable architectures and careful parameter optimization, neural networks can achieve performance similar to SVMs, We choose to use SVMs in this study because they are not susceptible to local minima and a general high-quality implementation of SVMs (SVM-light (Joachims, 1999, 2002)) is publicly available.

Most previous methods use structure-dependent information to predict the stability changes, and therefore can not be applied when tertiary structure information is not available. Although non-local interactions are the principal determinant of protein stability (Gilis and Rooman, 1997), previous research (Gilis and Rooman, 1997; Bordner and Abagyan, 2004; Casadio et al., 1995) shows that local interactions and sequence information can play important roles in stability prediction. Casadio et al. (Casadio et al., 1995) uses sequence composition and radial basis neural networks to predict the energy changes caused by mutations. Gillis and Rooman (Gilis and Rooman, 1997; Gillis and Rooman, 1996) shows that statistical torsion potentials of local interactions along the chain based on propensities of amino acids associated with backbone torsion angles is important for energy prediction, especially for the partial buried or solvent-accessible residues. The AGADIR algorithm (Munoz and Serrano, 1997; Lacroix et al., 1998), which uses only local interactions, has been used to design the mutations that increase the thermostability of protein structures. Bordner and Abagyan (Bordner and Abagyan, 2004) show that the empirical energy terms based on sequence information can be used to predict the energy change effectively, even though accuracy is still significantly lower than when using structural information. Frenz (Frenz, 2005) uses neural networks with sequence-based similarity scores for mutated positions to predict protein stability changes in Staphylococcal nuclease at 20 residue positions.

Here we develop a new machine learning approach based on support vector machines to predict the stability changes for single site mutations in two contexts taking into account structure-dependent and sequence-dependent information, respectively. In the first classification context, we predict whether a mutation will increase or decrease the stability of protein structure as in Capriotti et al. (2004). In this framework, we focus on predicting the sign of the relative stability change ($\triangle\triangle G$). In many cases, the correct prediction of the direction of the stability change is more relevant than its magnitude. (Capriotti et al., 2004) In the second regression context, we use an SVM-based regression method to predict directly the $\triangle\triangle G$ resulting from single site mutations, as most previous methods do. A direct prediction of the value of relative stability changes can be used to infer the directions of mutations by taking the sign of $\triangle\triangle G$.

There are a variety of ways in which sequence information can be used for protein stability prediction. Previous methods use residue composition (Casadio et al., 1995) or local interactions derived from a sequence. Our method directly leverages sequence information by using it as an input to the SVM. We use a local window centered around the mutated residue as input. This approach has been applied successfully to the prediction of other protein structural features, such as secondary structure and solvent accessibility (Rost and Sander, 1993b,a; Jones, 1999b; Pollastri et al., 2002b,a). The direct use of sequence information as inputs can help machine learning methods extract the sequence motifs which are shown to be important for protein stability (Lacroix et al., 1998). Like the neural network approach (Capriotti et al., 2004), we take advantage of the large amount of experimental mutation data deposited in the ProTherm (Gromiha et al., 2000) database to train and test our method. On the same dataset compiled in (Capriotti et al., 2004), our method yields a significant improvement over previous energy-based and neural network-based methods using 20-fold cross-validation.

An important methodological caveat results from the dataset containing a significant number of identical mutations applied to the same sites of the same proteins. We find that it is important to remove the site-specific redundancy to accurately evaluate the prediction performance for mutations at different sites. On the redundancy-reduced dataset, the prediction accuracy obtained using only sequence information alone is close to the accuracy obtained using structure-dependent information. Thus, our method can make accurate predictions in the absence of tertiary structure information. Furthermore, to estimate the performance on unseen and nonhomologous proteins, we remove the mutations associated with the homologous proteins and split the remaining mutations by individual proteins. We use the mutations of all proteins except for one to train the system and use the remaining one for testing (leave-one-out cross validation). Thus we empirically estimate how well the method can be generalized to unseen and nonhomologous proteins.

4.2 Materials and Methods

4.2.1 Data

We use the dataset S1615 compiled by Capriotti et al. (Capriotti et al., 2004). S1615 is extracted from the ProTherm (Gromiha et al., 2000) database for proteins and mutants. The dataset includes 1615 single site mutations obtained from 42 different proteins. Each mutation in the dataset has six attributes: PDB code, mutation, solvent accessibility, pH, temperature, and energy change ($\triangle\triangle G$). To make values of solvent accessibility, pH, and temperature near the same scale as other attributes, they are divided by 100, 10, and 100 respectively. If the energy change $\triangle\triangle G$ is positive, the mutation increases stability and is classified as a positive example. If $\triangle\triangle G$ is negative, the mutation is destabilizing and is classified as a negative example. For the classification task, there are 119 redundant examples having the exactly same values as some other example for all six attributes, provided only the sign of the energy changes is considered. These examples correspond to identical mutations at the same sites of the same proteins with the same temperature and pH, only the magnitudes of the energy changes are slightly different. To avoid any redundancy bias, we remove these examples for classification task. We refer to this redundancy-reduced dataset as SR1496. To leverage both sequence and structure information, we extract full protein sequences and tertiary structures from the Protein Data Bank (Berman et al., 2000) for all mutants according to their PDB codes.

We test three different encoding schemes (SO: sequence only, TO: structure only, ST: sequence and structure). (See section 2.2, below) Since solvent accessibility contains structure information, to compare SO with TO and ST fairly, we test SO scheme without using solvent accessibility on the SR1496 dataset. All schemes are evaluated using 20-fold cross validation. Under this procedure, the dataset is split randomly and evenly into 20 folds. 19 folds are used as the training dataset and the remaining fold is used as the test dataset. This process is repeated 20 times where each fold is used as the test dataset once. Performance results are averaged across the 20 experiments. The cross-validation results are compared with similar results in the literature obtained using a neural network approach (Capriotti et al., 2004). Using the same experimental settings as in Capriotti et al. (2004), the subset S388 of S1615 dataset is also used to compare our predictor with other predictors based on potential functions and available over the web. The S388 dataset includes 388 unique mutations derived under physiological conditions. We gather the cross validation predictions restricted to the data points in the S388 dataset, and then compute the accuracy and compare it with the three energy function based methods (Guerois et al., 2002; Kwasigroch et al., 2002; Zhou and Zhou, 2002; Gillis and Rooman, 1996) available over the web.

There is an additional subset of 361 redundant mutations that are identical to other mutations in the S1615 dataset, except for difference in temperature or pH. The energy changes of these mutations are highly correlated; and the signs of the energy changes are always same with a few exceptions. This is in contrast to the S388 subset, which contains no repeats of the same mutations at the same site. We find that it is important to remove this redundancy for comparing the performance of structure-dependent and sequence-dependent encoding schemes. Thus we derive a dataset without using solvent accessibility, pH, and temperature information and remove all the mutations–with the same or different temperature and pH–at the same site of the same proteins. The stringent dataset includes 1135 mutations in total. We refer to this dataset as SR1135.

In order to estimate the performance of mutation stability prediction on unseen and nonhomologous proteins, we use also UniqueProt (Mika and Rost, 2003) to remove homologous proteins by setting HSSP threshold to 0, so that the pairwise similarity between any two proteins is below 25%. Because the proteins in S1615 dataset are very diverse, only six proteins (1RN1, 1HFY, 1ONC, 4LYZ, 1C9O, and 1ANK) are removed. We remove 154 mutations associated with these proteins. Then we split the mutation data into 36 folds according to the remaining 36 proteins. For each fold, we further remove all the identical mutations at the same sites. There are 1023 mutations left in total. We refer to this dataset as SR1023. We apply an encoding scheme using only sequence information to this dataset without using solvent accessibility, pH, and temperature. We use 36-fold cross validation to evaluate the scheme by training SVMs on 35 proteins and testing them on the remaining one. Thus, we empirically estimate how well the method can be generalized to unseen and nonhomologous proteins.

For the regression task, we use sequence or structure information without considering solvent accessibility, temperature, and pH. We remove the identical mutations at the same sites and with identical energy changes. The final dataset has 1539 data points. We refer to this dataset as SR1539.

4.2.2 Inputs and Encoding Schemes

Most previous methods, including the neural network approach (Capriotti et al., 2004), use tertiary structure information for the prediction of stability changes and in general do not use the local sequence context directly. To investigate the effectiveness of sequence-dependent and structure-dependent information, we use three encoding schemes: Sequence-Only (SO), Structure-Only (TO) and the combinations of sequence and structure (ST). All the schemes include the mutation information consisting of 20 inputs, which code for the 20 different amino acids. We set to -1 the input corresponding to the deleted residue and to 1 the new introduced residue; all other inputs are set to 0. (Capriotti et al., 2004)

The SO scheme encodes the residues in a window centered on the target residue. We investigate how window size affects prediction performance. A range of window sizes work well for this task, however, we chose to use window of size 7 because it is the smallest size which produces accurate results. As more data becomes available, we imagine a larger window may become helpful. Since the target residue is already encoded in the mutation information, the SO scheme only needs to encode three neighboring residues on each side of the target residue. 20 inputs are used to encode the residue type at each position. So the total input size of the SO scheme is

140 (6*20+20). The TO scheme uses 20 inputs to encode the three-dimensional environment of the target residue. Each input corresponds to the frequency of each type of residue within a sphere of 9 Å radius around the target mutated residue. The cut-off distance threshold of 9 Å between C_a atoms worked best in the previous study (Capriotti et al., 2004). So the TO encoding scheme has 40 (20+20) inputs. The ST scheme containing both sequence and structure information in SO and TO scheme has 160 inputs (6*20+20+20).

On the SR1496 dataset, two extra inputs (temperature, and pH) are used with SO scheme; three extra inputs (solvent accessibility, temperature, and pH) are used with the TO and ST schemes. These additional inputs are not used for all other experiments on the SR1135, SR1023, and SR1539 datasets.

4.2.3 Prediction of Stability Changes Using Support Vector Machines

From a classification standpoint, the objective is to predict whether a mutation increases or decreases the stability of a protein, without concern for the magnitude of the energy change, as in Capriotti et al. (2004). From a regression perspective, the objective is to predict the actual value of $\triangle\triangle G$. Here we apply SVMs (Vapnik, 1998) (see (Burges, 1998; Smola and Scholkopf, 1998) for tutorials on SVMs) to the stability classification and regression problems.

The SVM provides nonlinear function approximation by nonlinearly mapping the input vectors into high dimensional feature space and using linear methods for regression or classification in feature space (Vapnik, 1998, 1995; Drucker et al., 1997; Schölkopf and Smola, 2002). Thus SVMs, and more generally kernel methods, combine the advantages of linear and nonlinear methods by first embedding

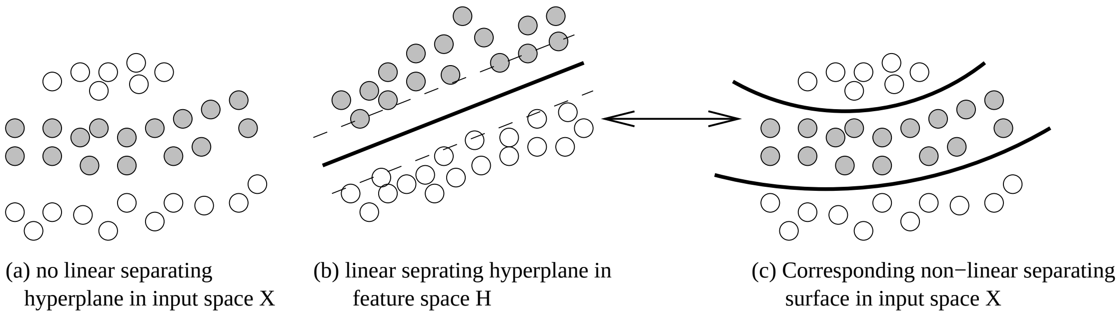

(a) no linear separating hyperplane in input space X
(b) linear seprating hyperplane in feature space H
(c) Corresponding non−linear separating surface in input space X

Figure 4.1: Classification with SVM. (a) The negative and positive examples (white and gray circles) can not be linearly separated in input space $\mathcal{X}$. (b) Instead of looking for a separating hyperplane (thick line) directly in the input space, SVM maps training data points implicitly into a *feature space* $\mathcal{H}$ through a function ϕ in which a linear hyperplane is computed. (c) This hyperplane corresponds to a nonlinear complex surface in the original input space. The virtual lines in the feature space are the boundary of positive and negative examples respectively and the distance between them is the margin of SVM.

the data into a feature space equipped with a dot product and then using linear methods in feature space to perform classification or regression tasks based on the dot product between data points. One important feature of SVMs is that computational complexity is reduced because data points do not have to be explicitly mapped to the high dimensional feature space. Instead SVMs use a kernel function, $K(x, y) = \phi(x) \cdot \phi(y)$ to calculate the dot product of $\phi(x)$ and $\phi(y)$ implicitly, where x and y are input data points, $\phi(x)$ and $\phi(y)$ are the corresponding data vectors of x and y in feature space, and ϕ is the map from input space to feature space. This significantly reduces the computation cost. Figure 4.1 illustrates how SVM maps nonlinearly separatable task into feature space and then finds a linear separating hyperplane with the maximum margin in the feature space. Figure 4.2 shows how SVM constructs a linear regression line in feature space for nonlinearly related data points in input space.

Given a set of data points S (S^+ denotes the subset of positive training data points ($\triangle\triangle G > 0$) and S^- denotes the subset of negative training data points

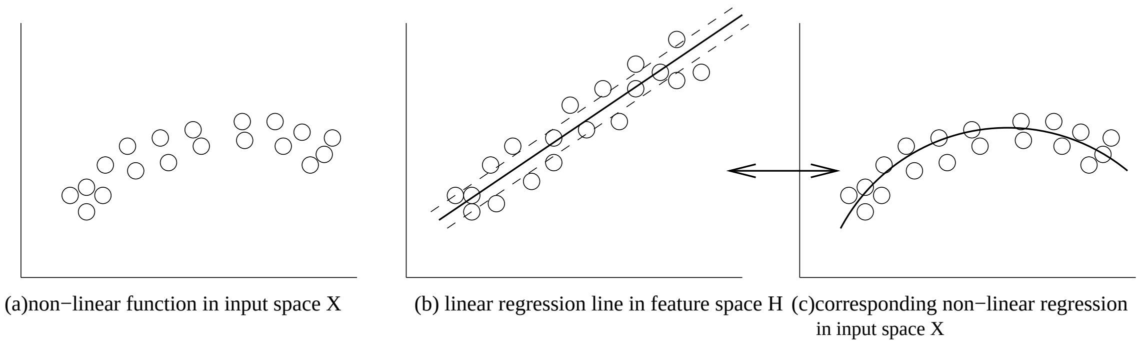

(a)non−linear function in input space X
(b) linear regression line in feature space H
(c)corresponding non−linear regression in input space X

Figure 4.2: Regression with SVM. (a) The data points can not be fitted by a linear regression line in input space $\mathcal{X}$. (b) SVM maps data points implicitly into a *feature space* $\mathcal{H}$ through a function ϕ in which a linear regression line is computed. (c) This linear regression line corresponds to a nonlinear regression curve in the original input space. The two virtual lines centered around regression line in feature space form a regression tube with width 2ϵ.

($\triangle\triangle G < 0$)), based on structure risk minimization theory (Vapnik, 1998, 1995; Drucker et al., 1997; Schölkopf and Smola, 2002), SVMs learn a classification function $f(x)$ in the form of

$$f(x) = \sum_{x_i \in S^+} \alpha_i K(x, x_i) - \sum_{x_i \in S^-} \alpha_i K(x, x_i) \quad + b$$

or a regression function in the form of

$$f(x) = \sum_{x_i \in S} (\alpha_i - \alpha_i^*) K(x, x_i) \quad + b$$

where α_i or α_i^* are non-negative weights assigned to the training data point x_i during training by minimizing a quadratic objective function and b is the bias term. K is the kernel function, which can be viewed as a function for computing the similarity between two data points. Thus the function $f(x)$ can be viewed as a weighted linear combination of similarities between training data points x_i and target data point x. Only data points with positive weight α in the training dataset affect the final solution - these are called support vectors. For classification problems, a new data point x is predicted to be positive ($\triangle\triangle G > 0$) or negative ($\triangle\triangle G < 0$) by taking the sign of $f(x)$. For regression, $f(x)$ is the predicted value of $\triangle\triangle G$.

We use SVM-light (http://svmlight.joachims.org) (Joachims, 1999, 2002) to train and test our methods. We experimented with several common kernels including linear kernel, Gaussian radial basis kernel (RBF), polynomial kernel, and sigmoid kernel. In our experience, the RBF ($e^{-\gamma||x-y||^2}$ or $e^{-\frac{||x-y||^2}{\sigma^2}}$) works best for mutation stability prediction. Using the RBF kernel, $f(x)$ is actually a weighted sum of Gaussians centered on the support vectors. Almost any separating boundary or regression function can be obtained with this kernel (Vert et al., 2004), thus it is important to tune the parameters of SVMs to achieve good generalization performance and void overfitting. We adjust three critical parameters in both classification and regression. For both tasks, we adjust the width parameter γ of the RBF kernel and regularization parameter C. γ is the inverse of the variance (σ^2) of the RBF and controls how peaked are the Gaussians centered on the support vectors. The bigger is γ, the more peaked are the Gaussians, and the more complex are the resulting decision boundaries (Vert et al., 2004). C is the maximum value that weights (α) can have. C controls the trade-off between training errors and the smoothness of $f(x)$ (particularly, the margin for classification) (Vapnik, 1998, 1995; Drucker et al., 1997; Schölkopf and Smola, 2002). A larger C corresponds to less training errors and a more complex (less smooth) function $f(x)$ which can overfit training data.

For classification, the ratio of penalty of training error between positive examples and negative examples, is another parameter that we tune. A penalty ratio $>$ 1 penalizes the training errors of positive examples more than that of negative examples. For regression, the width of the regression tube (ϵ) which controls the sensitivity of the cost associated with training errors ($f(x) - \triangle\triangle G$), needs to be tuned as well. The training error within range $[-\epsilon, +\epsilon]$ does not affect the regression function.

The three parameters for each task (penalty ratio, γ, and C for classification; tube width, γ, and C for regression) are optimized on the training data. For each cross-validation fold, we optimize these parameters using the LOOCV (leave one out cross validation) procedure. Under the LOOCV procedure, for a training dataset with N data points, in each round, one data point is held out and the model is trained on the remaining $N - 1$ data points. Then the model is tested on the held-out data point. This process is repeated N times until all data points are tested once and the overall accuracy is computed for the training dataset.

For all the parameter sets we tested, we choose a set of parameters with the best accuracy to build the model on the training dataset; and then it is blindly tested on the testing dataset. A set of good parameters for classification on the SR1496 dataset is (γ=0.1, penalty ratio=1, C=5) for SO scheme, (γ=0.1, penalty ratio=2, C=5) for TO schemes, and (γ=0.1, penalty ratio=2, C=5) for ST scheme. A set of good parameters on the SR1135 dataset is (γ=0.05, penalty ratio=1, C=2) for SO scheme, (γ=0.05, penalty ratio=1, C=4) for TO scheme, and (γ=0.06, penalty ratio=1, C=0.5) for ST scheme. For regression task, a set of good parameters for all schemes is (γ=0.1, tube width=0.1, C=5).

4.3 Results and Discussion

For classification, we use a variety of standard measures to evaluate the prediction performance of our method and compare it with previous methods. In the following equations, TP, FP, TN, and FN refer to the number of true positives, false positives, true negatives, and false negatives respectively. The measures we use include corre-

Table 4.1: Results(correlation coefficient, accuracy, specificity, sensitivity of both positive and negative examples) on the SR1496 dataset. The last row (NeuralNet*) is the current best results reported in (Capriotti et al., 2004).

Method	Corr. Coef.	Accuracy	Sens.(+)	Spec.(+)	Sens.(-)	Spec.(-)
SO	0.59	0.841	0.711	0.693	0.897	0.888
TO	0.60	0.845	0.711	0.712	0.895	0.895
ST	0.60	0.847	0.671	0.733	0.910	0.883
NeuralNet*	0.49	0.810	0.520	0.710	0.910	0.830

lation coefficient [(TP*TN-FP*FN)/((TP+FN)*(TP+FP)*(TN+FN) *(TN+FP)) $^{1/2}$], accuracy [(TN+TP) / (TN+TP+FN+FP)], specificity [TP/(TP+FP)] and sensitivity [TP/(TP+FN)] of positive examples, and specificity [TN/(TN+FN)] and sensitivity [TN / (TN+FP)] of negative examples.

Table 4.1 reports the classification performance of three schemes on the SR1496 dataset. The results show that the performance of all three schemes is improved over neural network approach (Capriotti et al., 2004) using most measures, even though we use redundancy reduced dataset instead of the S1615 dataset. (On the original S1615 dataset, the accuracy is about 85-86% for all three schemes). For instance, on average, accuracy is improved by 3% to about 84%, and correlation coefficient is increased by 0.1. The sensitivity of positive examples is improved by more than 10% using these three schemes, while the specificity of positive examples is very similar. The sensitivity of negative examples using the SO and TO schemes is slightly worse than neural network approach, but the specificity of negative examples is improved by more than 5% over the neural network approach. The accuracy of the SO scheme is slightly lower than that of the TO and ST schemes.

Following the same comparison scheme, we compare our methods with energy-based methods (Guerois et al., 2002; Kwasigroch et al., 2002; Zhou and Zhou, 2002; Gillis and Rooman, 1996) available on the web and with the neural network method

Table 4.2: Results on the S388 dataset

Method	Corr. Coef.	Accuracy	Sens.(+)	Spec.(+)	Sens.(-)	Spec.(-)
FOLDX	0.25	0.75	0.56	0.26	0.78	0.93
DFIRE	0.11	0.68	0.44	0.18	0.71	0.90
PoPMuSic	0.20	0.85	0.25	0.33	0.93	0.90
NeuralNet	0.25	0.87	0.21	0.44	0.96	0.90
SO	0.26	0.86	0.30	0.40	0.94	0.90
TO	0.28	0.86	0.31	0.42	0.94	0.91
ST	0.27	0.86	0.31	0.40	0.93	0.91

Table 4.3: Results on the SR1135 dataset

Method	Corr. Coef.	Accuracy	Sens.(+)	Spec.(+)	Sens.(-)	Spec.(-)
SO	0.31	0.78	0.28	0.64	0.95	0.80
TO	0.39	0.79	0.46	0.60	0.90	0.83
ST	0.34	0.79	0.29	0.71	0.97	0.80

(Capriotti et al., 2004) in the classification context on the S388 dataset. We compare the predictions of the following methods: FOLDX (Guerois et al., 2002), DFIRE (Zhou and Zhou, 2002) and PoPMuSiC (Kwasigroch et al., 2002; Gillis and Rooman, 1996), and NeuralNet (Capriotti et al., 2004). In Table 4.2, we show the results obtained with the three schemes (SO, TO, ST) and the four external predictors on the S388 dataset, where results for the energy function based methods are taken from (Capriotti et al., 2004). The results show that our method, using the three encoding schemes for this specific task, performs similarly to or better than, all other methods using most evaluation measures. For instance, the correlation coefficient of our method is better than all other methods, while the accuracy is better than DFIRE, FOLDX, and PoPMuSic, but slightly worse than NeuralNet. FOLDX and DFIRE have relatively higher sensitivity but lower specificity on positive examples than other methods.

Table 4.3 reports the results on the SR1135 dataset without any site-specific

Table 4.4: Specificity and sensitivity of the SO scheme for helix, strand, and coil on the SR1135 dataset.

Secondary structure	Sens.(+)	Spec.(+)	Sens.(-)	Spec.(-)
Helix	0.31	0.67	0.94	0.79
Strand	0.16	0.48	0.97	0.84
Coil	0.30	0.68	0.95	0.79

redundancy. All the schemes do not use solvent accessibility, pH, and temperature. The results show that the accuracy of the structure-dependent schemes (TO and ST) are about 1% higher than the sequence-dependent scheme (SO). Specifically, the correlation coefficient of the TO scheme is significantly higher than the SO scheme. But the accuracy of the SO scheme is still very close to the accuracy derived using tertiary structure information . This is probably due to two reasons. First, the sequence window contains significant amount of information related to the prediction of mutation stability. Second, the method for encoding structural information in the TO and ST schemes is not optimal for the task and does not capture all structure information that is relevant to protein stability. On this redundancy reduced dataset, we also compare the accuracy according to the type of secondary structure encountered at mutation sites. The secondary structure is assigned by the DSSP program (Kabsch and Sander, 1983). Table 4.4 reports the specificity and sensitivity for both positive and negative examples according to three types of secondary structure (helix, strand, and coil) using the SO scheme. The SO scheme achieves the similar performance on helix and coil mutations. The sensitivity and specificity for positive examples on β-strands, however, is significantly lower. This is probably due to the long-range interactions between β-strands.

Table 4.5 reports the results of the SO scheme on the SR1023 dataset after removing both the homologous proteins and site-specific redundancy. The overall

Table 4.5: Results on the SR1023 dataset using the SO scheme

Method	Corr. Coef.	Accuracy	Sens.(+)	Spec.(+)	Sens.(-)	Spec.(-)
SO	0.13	0.74	0.15	0.42	0.93	0.77

Table 4.6: Results(correlation between predicted energy and experimental energy, and standard error) on the SR1539 dataset using SVM regression.

Scheme	SO	TO	ST
correlation	0.75	0.76	0.75
std	1.10	1.09	1.09

accuracy is 74%. Not surprisingly, the accuracy is lower than the accuracy obtained when mutations on the homologous and identical proteins are included in the training and test dataset. The sensitivity and specificity of the positive examples drop significantly. This indicates that the accuracy of the method depends on having seen mutations on the similar or identical proteins in the training dataset. The results show that the prediction of mutation stability on unseen and nonhomologous proteins remains very challenging.

The performance of SVM regression is evaluated using the correlation between the predicted energy and experimental energy, and the standard error (std or root mean square error) of the predictions. Table 4.6 shows the performance of the direct prediction of $\triangle\triangle G$ using SVM regression with three encoding schemes. The three schemes have similar performance, where the TO scheme performs slightly better with a correlation of 0.76, and std of 1.09. Figure 4.3 shows the scatter plots of predicted energy versus experimental energy using the SO and TO schemes. Overall, the results show that our method effectively uses sequence information to predict energy changes associated with single amino acid mutations both in regression and classification tasks.

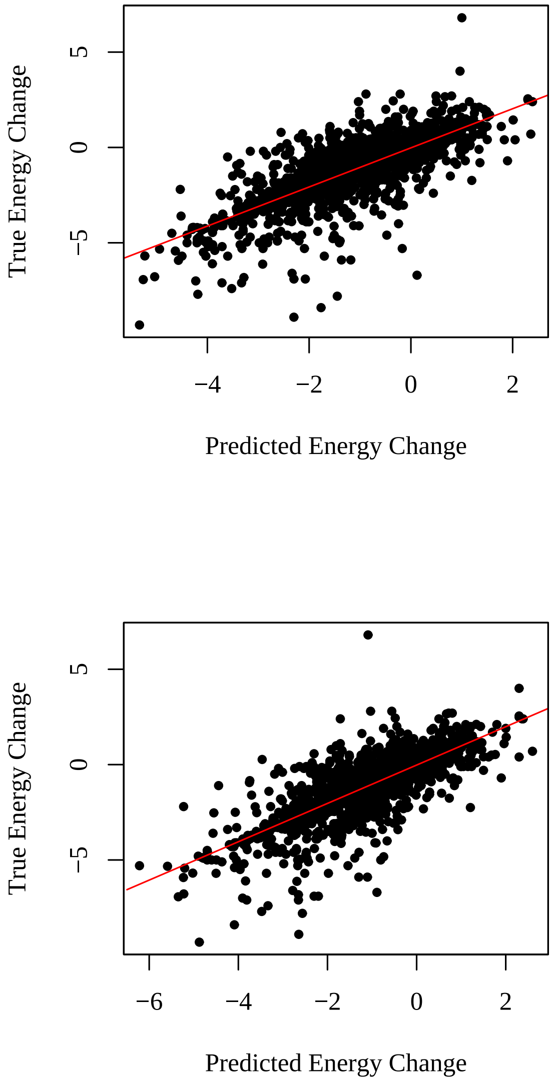


Figure 4.3: **(a)** The experimentally measured energy changes versus the predicted energy changes using SVM regression with the SO scheme on the SR1539 dataset. The correlation is 0.75. The std is 1.10. The slope of the red regression line is 1.03. **(b)** The experimentally measured energy changes versus the predicted energy changes using SVM regression with the TO scheme on the SR1539 dataset. The correlation is 0.76. The std is 1.09. The slope of the red regression line is 1.01.

4.4 Conclusions

In this study, we have used support vector machines to predict protein stability changes for single-site mutations. Our method consistently shows better performance than previous methods evaluated on the same datasets. We demonstrate that sequence information can be used to effectively predict protein stability changes for single site mutations. Our experimental results show that the prediction accuracy based on sequence information alone is close to the accuracy of methods that depend on tertiary structure information. This overcomes one shortcoming of previous approaches that require tertiary structures to make accurate predictions. Thus, our approach can be used on a genomic scale to predict the stability changes for large numbers of proteins with unknown tertiary structures.

Chapter 5

Large-Scale Prediction of Disulphide Bridges Using Kernel Methods, Two-Dimensional Recursive Neural Networks, and Weighted Graph Matching

5.1 Introduction

5.1.1 Disulphide Connectivity

The formation of covalent links between cysteine (Cys) residues by disulphide bridges is an important and unique feature of protein folding and structure. Simulations (Abkevich and Shankhnovich, 2000), experiments in protein engineering (Matsumura et al., 1989; Clarke and Fersht, 1993; Klink et al., 2000), theoretical studies (Betz, 1993; Doig and Sternberg, 1995; Wedemeyer et al., 2000), and even evolutionary models (Demetrius, 2000) stress the importance and selective advantage of disulphide bridges in stabilizing the native state of proteins. This stabilizing role of disulphide bridges derives from a reduction of the number of configurational states, thus of the entropic cost of folding a polypeptide chain into its native state (Matsumura et al., 1989). Moreover, disulphide bridges not only contribute to the energetics of folding but, depending on their number and location, they can also contribute to catalytic activity (Klink et al., 2000). Thus, knowledge or prediction of disulphide bridges in a protein is important: it can provide essential insights into its structure, function, and evolution, as well as valuable long-ranged structural constraints (Harrison and Sternberg, 1994) that can be incorporated into a protein structure prediction pipeline. However, it is precisely because disulphide bridges link linearly distant portions of a protein that their prediction has remained a considerable challenge. To address this challenge, here we develop and test new methods that significantly improve the prediction of disulphide bridges.

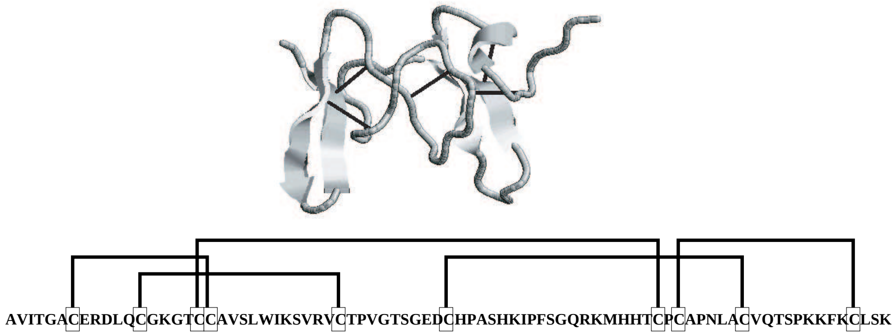

AVITGACERDLQCGKGTCCAVSLWIKSVRVCTPVGTSGEDCHPASHKIPFSGQRKMHHTCPCAPNLACVQTSPKKFKCLSK

Figure 5.1: Structure (top) and disulphide bridge connectivity pattern (bottom) of intestinal toxin 1, PDB code 1IMT. There are five disulphide bridges shown as thick lines.

5.1.2 Overview of Disulphide Connectivity Prediction

Only in recent years has the problem of predicting disulphide bridges in a systematic manner received sustained attention. (Fariselli and Casadio, 2001; Fariselli et al., 2002; Klepeis and Floudas, 2003; Vullo and Frasconi, 2004). The prediction of disulphide bridges can in fact be subdivided into four related prediction subproblems (Figure 5.1). First, only a minority of protein chains contain disulphide bridges. Thus it is desirable to be able to classify protein chains into those containing disulphide bridges and those that are entirely devoid of disulphide bridges (chain classification). Second, even in a chain that contains disulphide bridges, not all the cysteines may be bonded. Thus the second problem is the classification of cysteine residues into bonded and non-bonded (residue classification). Third, given a pair of cysteines, one can ask whether they are linked or not by a disulphide bridge (bridge classification). And fourth, the most important and challenging problem is to determine all the pairs of cysteines that are bonded to each other by a disulphide bridge (connectivity prediction). Although these problems can be tackled separately, it is clear that they are not independent and that "mixed" solutions can also be considered. Furthermore, tackling them sequentially and independently of each other may not be always optimal. For instance, deciding in isolation whether a given cysteine is bonded or not, may fail to take into consideration information about the bonding state of other cysteines in the same sequence and the obvious *global* constraint that the total number of intra-chain disulphide-bonded cysteines must be even. Finally, it is worth noting that inter-chain disulphide bridges associated with quaternary structure do occur also. However, they are considerably less frequent and very few such examples can be found in the Protein Data Bank (PDB) (Berman et al., 2000). Thus, in the current state of affairs, and consistently with all existing literature, it is not unreasonable to focus exclusively on intra-chain disulphide bridges. Here we address all four problems, with a particular emphasis on the most challenging problem of predicting intra-chain disulphide connectivity, directly or in combination with the second and third problems.

None of the approaches published so far in the literature address all four problems. Published approaches to disulphide connectivity prediction use stochastic global optimization (Fariselli and Casadio, 2001), combinatorial optimization (Klepeis and Floudas, 2003) and machine learning techniques (Fariselli et al., 2002; Vullo and Frasconi, 2004). The early work in Fariselli and Casadio (2001) provides a first, fairly comprehensive, treatment of disulphide connectivity prediction by reducing it to a matching problem in a complete weighted graph, where the vertices represent oxidized cysteines. Edge weights correspond to interaction strengths or contact potentials between the corresponding pairs of cysteines. The weights are learned using a simulated annealing approach. A candidate set of bridges is then derived by finding the maximum weight perfect matching[1]. The prediction of which cysteines are oxidized (residue classification) is not addressed in this work. In a subsequent improvement (Fariselli et al., 2002), neural network predictions are used for labeling edges with contact potentials, increasing the predictive power and reducing training time. This method achieves good results in the simplest cases of chains containing only 2 or 3 bridges.

The method in Vullo and Frasconi (2004) attempts to solve the connectivity prediction problem using a different machine learning approach by modeling candidate connectivity patterns as undirected graphs (see Fig.5.1, bottom). A recursive neural network architecture (Frasconi et al., 1998) is trained to score candidate graphs by their similarity with respect to the correct graph. The vertices of the graphs are

[1] A perfect matching of a graph (V, E) is a subset $E' \subseteq E$ such that each vertex $v \in V$ is met by only one edge in E'.

labeled by fixed-size vectors corresponding to multiple alignment profiles in a local window around each cysteine. During prediction, the score computed by the network is used to exhaustively search the space of candidate graphs. This method yields slight improvements over Fariselli et al. (2002) when tested on the same dataset. Unfortunately, for computational reasons, the applicability of this method remains limited because it too can only deal with sequences containing a small number of bridges, in practice up to five.

A different approach to predicting disulphide bridge connectivity is reported in Klepeis and Floudas (2003), where finding disulphide bridges is part of a more general protocol aimed at predicting the topology of β-sheets in proteins. The approach assumes hydrophobic rather than hydrogen interactions as the main driving force of β-sheet formation. Residue-to-residue contacts (including Cys-Cys bridges) are predicted by solving a series of integer linear programming problems in which customized hydrophobic contact energies must be maximized. Model constraints define allowable sheets and disulphide connectivity configurations. The most interesting aspect of this approach is its ability to predict cysteine-cysteine contacts, without assuming prior knowledge of the bonding state of the cysteines. This method, however, cannot be compared with the other approaches because the authors report validation results on only two relatively short sequences with few bonds (2 and 3). In contrast, Fariselli and Casadio (2001) and Vullo and Frasconi (2004) assess their methods on a broad spectrum of sequences.

The simpler problem of predicting whether a given cysteine is bonded or not has also been addressed using a variety of machine learning methods including neural networks (NNs), hidden Markov models (HMMS), and support vector machines (SVMs) (Fariselli et al., 1999; Fiser and Simon, 2000; Martelli et al., 2002; Frasconi et al., 2002; Ceroni et al., 2003). For instance, SVMs and kernels methods are used in Ceroni et al. (2003) to predict in two stages whether a given protein contains oxidized cysteines – in fact, whether all, none or a mixture of its cysteines are oxidized – and subsequently to predict the oxidation state of each cysteine. The best accuracies reported in the literature are around 85%.

We present an integrated, modular, approach to address all four problems. We leverage evolutionary information in the form of profiles and curated training sets in combination with kernel methods to address the chain classification problem. We use two-dimensional graphical models and recursive neural networks to predict the bonding probability of each pair of cysteines, leveraging in addition secondary structure and relative solvent accessibility information. These predictions can be derived for *all* the cysteines in a given chain, or only for the subset of disulphide-bonded cysteines, when pre-existing information about residue classification is available. Finally, we use graph matching methods to infer the disulphide bridge connectivity of each protein chain, which in turn yields a solution for both the bridge and residue classification problems, even in the case where the bonding state of individual cysteines is not known. Thus, the approach works for both situations where the bonded state of each cysteine is known or unknown and, after training, produces predictions that are rapid enough for genome-scale projects.

5.2 Methods

5.2.1 Data Preparation

In order to assess our methods, we used two existing data sets (SP 39 and SP41, courtesy of Dr. A. Vullo) to compare our results with previously published results. We also curated a third, larger, data set (SPX), to take advantage of recent growth in the PDB (Protein Data Bank) (Berman et al., 2000).

Previous Data Sets (SP39 and SP41)

SP39 is the set described and used in Fariselli et al. (2002); Vullo and Frasconi (2004) compiled from the Swiss-Prot database (Bairoch and Apweiler, 2000) release no. 39 (October 2000). SP41 is the updated version, compiled with the same filtering procedures as SP39, using the Swiss-Prot version 41.19 (August 2003). Specifically, only chains whose structure is deposited in the PDB are retained. Protein chains with disulphide bonds assigned tentatively or inferred by similarity are filtered out yielding a data set comprising 966 chains, containing at least one, and up to 24, disulphide bridges. Because our method is not limited by the number of disulphide bonds, the entire set of chains is retained. This set contains a subset of 712 sequences containing at least two disulphide bridges ($K \geq 2$)–the case $K = 1$ being trivial when the cysteine bonding state is known. By comparison, SP39 contains 446 chains only, with no chain having more than 5 bridges. Thus SP41 contains 266 additional sequences, and 112 of these have more than 10 oxidized cysteines.

In order to avoid biases during the assessment procedure and to perform k-fold cross validation, SP41 is partitioned into ten different subsets, with the constraint that sequence similarity between two different subsets be less or equal to 30%. This is comparable to the criteria adopted in Vullo and Frasconi (2004); Fariselli and Casadio (2001), where SP39 was split into four subsets. Sequence similarity is derived by a procedure analogous to the one adopted for building the PDB non-redundant selection of chains (Hobhom et al., 1992) by running an all-against-all rigorous Smith-Waterman local pairwise alignment (Smith and Waterman, 1981), using the BLOSUM65 scoring matrix with gap penalty 12 and gap extension 4. Pairs of chains with a negative distance (Abagyan and Batalov, 1997), or with an alignment length shorter than 30 residues, are considered unrelated. To address the chain classification problem, we augment the 966 positive sequences in SP41 with a set of 506 negative sequences, containing no disulphide bridges, taken form PDB Select (Hobohm and Sander, 1994).

New Dataset (SPX)

We downloaded all the proteins from the PDB on May 17, 2004. Some (26.8%) of these proteins contain at least one disulphide bridge (6,827 out of 25,465). Among these 6,827 proteins, 89% (6,058) contain disulphide bridges that are exclusively intra-chain and these are retained for further processing (Fariselli et al., 1999; Fariselli and Casadio, 2001; Vullo and Frasconi, 2004). These proteins containing exclusively intra-chain disulfide bonds yield a total of 10,793 chains, after removal of short sequences containing less than 12 amino acids. Among these 10,793 chains, 96% (10,378) have at least one intra-chain disulfide bond. The disulfide bond information is extracted from the PDB files by analyzing SSBOND records (Westbrook and Fitzgerald, 2003). To reduce over-representation of particular protein families, we use UniqueProt (Mika and Rost, 2003), a protein redundancy reduction tool based on the HSSP (Sander and Schneider, 1991) distance, to choose 1,018 representative chains by setting the HSSP cut-off distance to 10. The HSSP distance is a similarity measure which takes into account sequence length. An HSSP-distance of 10 between two sequences of length 250 is roughly equivalent to 30% sequence identity. To leverage and assess the role of secondary structure and solvent accessibility information during prediction, we use DSSP (Kabsch and Sander, 1983) to annotate secondary structure and solvent accessibility for all selected protein chains. These sequences contain 5983 cysteines in total, 85% (5,082) of which are involved in disulphide bridges. These sequences are randomly split into ten subsets of roughly the same size. During each ten-fold cross validation experiment, nine subsets are used for training and the remaining subset is used for validation. Final results are averaged across the ten cross validation experiments. To address the chain classification problem, we augment a subset of 897 positive sequences selected with an even lower HSSP cutoff distance of 5 (roughly below 20% similarity) with a set of 1,650 negative sequences, containing no disulphide bridges, extracted from PDB and redundancy-reduced using UniqueProt with a stringent HSSP cutoff distance of 0 (no similarity).

5.2.2 Kernel Methods for Chain Classification

Kernels methods (Cristianini and Shawe-Taylor, 2000; Müller et al., 2001; Schölkopf and Smola, 2002) are an important class of flexible machine learning methods that have proven useful for several problems in bioinformatics (Jaakkola et al., 2000; Leslie et al., 2004; Liao and Noble, 2002; Ceroni et al., 2003; Schölkopf et al., 2004; Lanckriet et al., 2004). The basic idea behind these methods is to try to retain the elegance and simplicity of linear methods when dealing with nonlinear data, by embedding the original data into a feature space, equipped with a dot product, where linear methods can be applied to perform classification, regression, and other computational tasks. In practice, a kernel approach really depends on two independent modules: (1) a module for computing the kernel and the Gram matrix; and (2) a module for computing the optimal manifold (usually a hyperplane in classification problems) in feature space, typically using techniques from quadratic convex optimization. Since relatively standard packages exist to cover the second module, here for conciseness we focus on the description of the kernels and refer the readers to Schölkopf and Smola (2002) for additional details about kernel methods. We use six different kernels (Spectrum, Mismatch, Profile, Smith Waterman, Local Alignment, and Fisher) to classify protein chains, according to whether they contain at least one disulphide bridge or not.

Spectrum Kernel

Spectrum kernels for sequences are derived by constructing, for each sequence x, the feature vector $\phi_k(x)$ counting the occurrences of all possible substrings of length k (Leslie et al., 2002). Similarity between spectral vectors can be computed by simple scalar product, or further processed using Gaussian exponentials or other positive convex functions (Schölkopf and Smola, 2002).

Mismatch Kernel

Mismatch kernels (Kuang et al., 2005) are a variation of spectrum kernels which allows inexact matching of substrings. Specifically, mismatch kernels count co-occurrences of substrings of length k, allowing up to $m \leq k$ mismatches, between the two input sequences. Like spectrum kernels, mismatch kernels can be computed efficiently by using trie data structures (suffix trees).

Profile Kernel

Profile kernels (Kuang et al., 2005) are another variation on mismatch kernels which use a position-dependent mutation neighborhood for inexact matching of subsequences of length k. The local neighborhood is defined using probabilistic profiles, such as those produced by the PSI-BLAST algorithm, by aligning the sequences to the NR database. A given substring is in the local neighborhood if its negative log probability, according to the profile, is smaller than some threshold σ. Once the profiles have been derived, profile kernels can also be computed efficiently using a trie data structure. In the simulations, for each sequence in the SP41 or SPX dataset, profiles are derived using two iterations of PSI-BLAST

(Altschul et al., 1997) against the NR database with default parameter settings. Software for computing spectrum, mismatch, and profile kernels is available from http://www1.cs.columbia.edu/compbio/string-kernels/.

Smith-Waterman Kernel

The SW kernel is an empirical kernel technique (Schölkopf and Smola, 2002) which uses the E-values of a Smith-Waterman alignment score as a measure of similarity and maps each sequence into a feature vector consisting of its SW scores with all the input training sequences.

Local Alignment Kernel

The local alignment kernel is a variation on the SW kernel constructed to ensure positivity conditions (Mercer kernel conditions Schölkopf and Smola (2002)) using dynamic programming. It is described in detail in Saigo et al. (2004).

Fisher Kernel

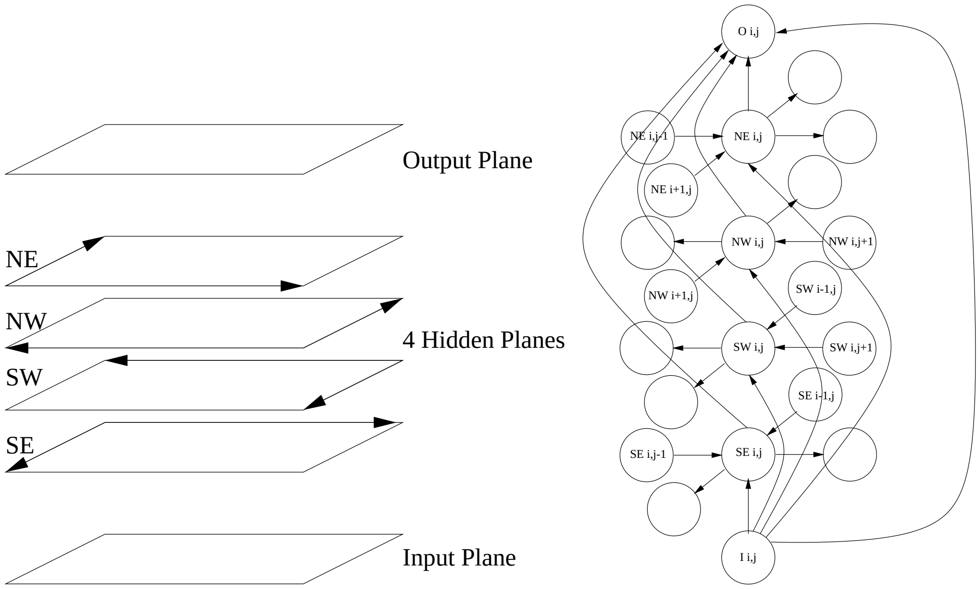
Fisher kernels are derived from probabilistic generative models of the data. Here we train a protein HMM (Baldi and Brunak, 2001) on the positive examples using the SAM software package available at www.cse.ucsc.edu/research/compbio/sam.html (with local alignment option "SW 2" and Dirichlet prior option "recode3.20comp") The feature vector (Fisher score) is obtained from the derivatives of the likelihood of a sequence with respect to the HMM parameters (see Jaakkola et al. (2000) for details). It is obtained using the "get_fisher_scores" function of SAM with the "fisher_feature_match" option which ensures that only matching states are considered for scoring. The Fisher scores are then fed into a Gaussian kernel.

Kernel Combinations

Basic algebraic operations such as addition, multiplication, and exponentiation preserve the positive properties of a kernel matrix and provide a mechanism for combining information from different kernels. Here we experimented with convex linear combinations of kernels, in particular with simple additivity where a new kernel matrix $K(x, y) = K_1(x, y) + K_2(x, y)$ is constructed using two normalized kernel matrices K_1 and K_2 with the corresponding feature vector defined by $\phi(x) = (\phi_1(x), \phi_2(x))$. These kernel combinations did not seem to lead to significant improvements and therefore are not discussed in detail.

5.2.3 Recursive Neural Networks to Predict Cysteine Pairing Probabilities

To predict disulphide connectivity patterns, we use the 2D DAG-RNN (Directed Acyclic Graph-Recursive Neural Network) approach described in (Baldi and Pollastri, 2003), whereby a suitable Bayesian network is recast, for computational effectiveness, in terms of recursive neural networks. Local conditional probability tables in the underlying Bayesian network are replaced by deterministic relationships between a variable and its parent node variables. These functions are parameterized by neural networks using appropriate weight sharing, as described below. Here the underlying DAG for disulphide connectivity has 6 2D-layers: input, output, and four hidden layers (Figure 5.2(a)). Vertical connections, within an (i, j) column, run from input to hidden and output layers, and from hidden layers to output (Figure 5.2(b)). In each one of the four hidden planes, square lattice connections are oriented towards one of the four cardinal corners. Detailed motivation for these architectures can be found in Baldi and Pollastri (2003) and a mathematical analysis of their



(a) (b)

Figure 5.2: (a) General layout of a DAG for processing two-dimensional objects such as disulphide contacts, with nodes regularly arranged in one input plane, one output plane, and four hidden planes. In each plane, nodes are arranged on a square lattice. The hidden planes contain directed edges associated with the square lattices. All the edges of the square lattice in each hidden plane are oriented towards one of the four possible cardinal corners: NE, NW, SW, SE. Additional directed edges run vertically within a vertical column from the input plane to each hidden plane, and from each hidden plane to the output plane. (b) Connections within a vertical column (i, j) of the DAG. I_{ij} represents the input, O_{ij} the output, and NE_{ij} represents the hidden variable in the North-East hidden plane. Similarly for the other hidden variables.

relationships to Bayesian networks in Baldi and Rosen-Zvi (2005). The essential point is that they combine the flexibility of graphical models with the deterministic propagation and learning speed of artificial neural networks. Unlike traditional neural networks with fixed-size input, these architectures can process inputs of variable structure and length, and allow lateral propagation of contextual information over considerable length scales.

In a disulphide contact map prediction, the (i,j) output represents the probability of whether the i-th and j-th cysteines in the sequence are linked by a disulphide bridge or not. This prediction depends directly on the (i,j) input and the four-hidden units in the same column, associated with omni-directional contextual propagation in the hidden planes. Hence, using weight sharing across different columns, the model can be summarized by 5 distinct neural networks in the form

$$\begin{cases} O_{ij} = \mathcal{N}_O(I_{ij}, H^{NW}_{i,j}, H^{NE}_{i,j}, H^{SW}_{i,j}, H^{SE}_{i,j}) \\ H^{NE}_{i,j} = \mathcal{N}_{NE}(I_{i,j}, H^{NE}_{i-1,j}, H^{NE}_{i,j-1}) \\ H^{NW}_{i,j} = \mathcal{N}_{NW}(I_{i,j}, H^{NW}_{i+1,j}, H^{NW}_{i,j-1}) \\ H^{SW}_{i,j} = \mathcal{N}_{SW}(I_{i,j}, H^{SW}_{i+1,j}, H^{SW}_{i,j+1}) \\ H^{SE}_{i,j} = \mathcal{N}_{SE}(I_{i,j}, H^{SE}_{i-1,j}, H^{SE}_{i,j+1}) \end{cases} \tag{5.1}$$

where $\mathcal{N}$ denotes NN parameterization. In the simulations, these 5 NNs have a single hidden layer containing 9 hidden units. The number of output units in each of the four NNs associated with the four cardinal corners is also 9. Weights are initialized randomly using a uniform distribution over the [-0.1,0.1] interval. Because of the acyclic nature of the underlying graph, learning can proceed by gradient descent (backpropagation). We use a stochastic form of gradient in the sense that training examples are used online and in randomized order after each training epoch. The learning rate is set to 0.008.

The input information is based on the sequence itself or rather the corresponding profile derived by multiple alignment methods to leverage evolutionary information, possibly augmented by secondary structure and solvent accessibility information derived from the PDB files with DSSP and/or the SCRATCH suite of predictors available at: www.igb.uci.edu/servers/psss.html and described in Pollastri et al. (2002b); Baldi and Pollastri (2003); Cheng et al. (2005a). For a sequence of length N and containing M cysteines, the output layer contains $M \times M$ units. The input and hidden layer can scale like $N \times N$ if the full sequence is used, or like $M \times M$ if only fixed-size windows around each cysteine are used, as in the experiments reported here. It is also possible to use 1D DAG-RNN to locally encode the input, as described in Baldi and Pollastri (2003).

It is essential to remark that the same DAG-RNN approach can be trained and applied in two different modes. In the first mode, we can assume that the bonded state of the individual cysteines is known, for instance through the use of a specialized predictor for residue classification. Then if the sequence contains M cysteines, $2K$ ($2K \leq M$) of which are intra-chain disulphide bonded, the prediction of the connectivity can focus on the $2K$ bonded cysteines exclusively and ignore the remaining $M - 2K$ cysteines that are not bonded. In the second mode, we can try to solve both prediction problems–residue and bridge classification–at the same time by focusing on all cysteines in a given sequence. In both cases, the output is an array of pairwise probabilities from which the overall disulphide connectivity graph must be inferred. In the first case, the total number of bonds or edges in the connectivity graph is known (K). In the second case, the total number of edges must be inferred. In section 5.3, we show that sum of all probabilities across the output array can be used to effectively estimate the number of disulphide contacts.

5.2.4 Input Specifications

The results reported here are obtained using local windows of size 5 around each cysteine, as in Vullo and Frasconi (2004). To improve prediction by exploiting evolutionary information and conserved sequence patterns encoded in homologous protein sequences, we derive position-specific profiles (also called Position Specific Scoring Matrix) from multiple sequence alignments by aligning all proteins against the NR database using PSI-BLAST (Altschul et al., 1997) according to the same protocol for creating profiles described in Pollastri et al. (2002b). Gaps are treated as if "-" corresponded to one additional amino acid. Thus the position-specific profile for each position in a sequence is a real vector of length 21, representing the probability of the 20 amino acids plus gap. For a window of 5 amino acids centered around two cysteines, the profile-component of the input consists of 210 ($21 \times 5 \times 2$) numbers. One extra input encodes the linear sequence separation between the two cysteines. To study how secondary structure (SS) and solvent accessibility (SA) information affect prediction accuracy, we also add SS and SA information to the input in all four possible combinations.

5.2.5 Graph Matching to Derive Connectivity from Pairing Probabilities

In the case where the bonded state of the cysteines is known, one has a graph with $2K$ nodes, one for each cysteine. The weight associated with each edge is the probability that the corresponding bridge exists, as computed by the predictor. The problem is then to find a connectivity pattern with K edges, where each cysteine is paired uniquely with another cysteine. This can be solved using Edmond's maximum weight matching algorithm (Edmonds, 1965), which has $O(V^4)$ time complexity on a graph with V edges, or rather the faster $O(V^3)$ implementation derived by Gabow (Gabow, 1976), with linear $O(V) = O(K)$ space complexity beyond the storage of the graph. Note that because the number of bonded cysteines in general is not very large, it is also possible in many cases to use an exhaustive search of all possible combinations. Indeed, the number of possible combinations is $1 \times 3 \times 5 \times \ldots \times (2K-1)$, which in the case of 10 cysteines with 5 disulphide bridges result in only 945 possible connectivity patterns.

The case where the bonded state of the cysteines is not known is slightly more involved and the Gabow algorithm cannot be applied directly since the graph has M nodes but only a subset of $2K < M$ nodes may participate in the final maximum weighted matching. However, we can still use Gabow's algorithm as follows. Assume first that we can get a good estimate of the total number K of bonds. In general, it is still not possible to try all $\binom{M}{2K}$ possible subsets and run Gabow's algorithm on each one of them, but one can use a good heuristic approximation. If M is even ($M = 2R$) we apply Gabow algorithm to the $2R$ nodes and then prune down the final result by removing, from the final set of R edges, the $R - K$ edges with lowest probabilities. If M is odd, $M = 2R + 1$ we apply the same strategy as above $2R + 1$ times, each time removing one of the cysteines. We then select the matching with K edges that has the highest probability. In practice this procedure gives very good results although it is not guaranteed to find the global optimum and, furthermore, it relies on a good estimate of the total number K of bonds. In the results section, we show that the total number K of bonds can be estimated from the sum of all the probabilities produced by the predictor using a simple regression approach. Although this may seem surprising, we have observed similar effects in contact map prediction, where the sum of the probabilities along a diagonal band is closely related to the total number of contacts in that band.

Alternatively, it is also possible to use a slightly different greedy algorithm to derive the connectivity pattern using the estimate of the total number of bonds. First, we order the edges in decreasing order of probabilities. Then we pick the edge with the highest probability, followed by the edge with the next highest probability that is not incident to the first edge, and so forth, until K edges have been selected. Because this greedy procedure is not guaranteed to find the global optimum, it is useful to repeat it L times. In each run $i = 1, \ldots, L$, the first edge selected is the i-th most probable edge. This is based on the observation that in practice the optimal solution always contain one of the top L edges and, for L reasonably large, the optimal connectivity pattern is usually found. We have compared this method with Gabow's algorithm in the case where the bonding state is known and observed that when $L = 6$, this greedy heuristic yields results that are as good as those obtained by Gabow's algorithm which, in this case, is guaranteed to find a global optimum. Thus the simulation results we report are derived using the greedy procedure with $L = 6$. The advantage of the greedy algorithm is its low $O(M^2 \log M + LKM)$ time complexity. This is because it takes $O(M^2 \log M)$ steps to sort all the pairing probabilities, and at most $O(KM)$ steps to derive a matching, starting from one of the L most promising edges.

5.3 Results

5.3.1 Statistical Analysis

Basic statistics extracted from the larger SPX dataset are shown in Figures 5.3, 5.4, 5.5, and 5.6 and Table 5.1. Figure 5.3 provides the distribution of sequence lengths. As the number of disulphide bridges in a protein chain increases, the number of possible disulphide connectivity patterns increases exponentially. Thus,

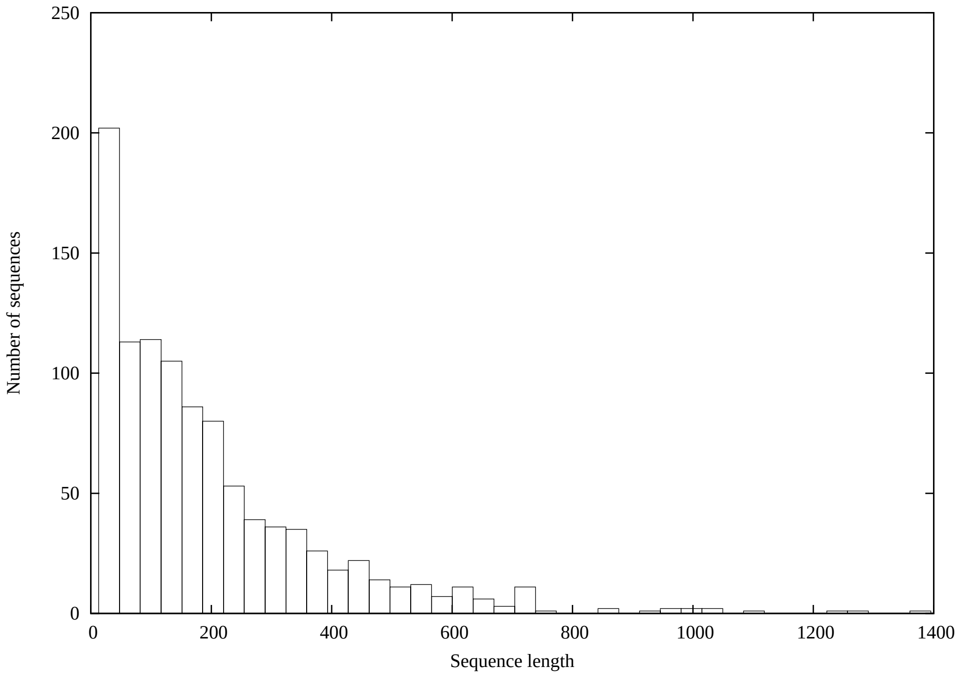


Figure 5.3: Distribution of sequence lengths in redundancy-reduced dataset (SPX) of sequences containing disulphide bridges.

it is important to study the distribution of the number of bridges per protein and investigate how connectivity prediction deteriorates with the number of bridges. Figure 5.4 shows that most sequences have less than 5 disulphide bridges, but there are exceptions, and a fraction of the sequences contains over 10 disulphide bridges. In SPX, the average number of disulphide bridges per chain is 2.5, with a standard deviation of 2.14. Figure 5.5 illustrates the distribution of disulphide bridge densities measured by the number of bridges divided by the sequence length. Figure 5.6 shows the distribution of disulphide bridge lengths measured in terms of the number of intervening amino acids. A very significant fraction of bridges is long-ranged with lengths above 30, far exceeding the scale of local secondary structure. This is the dual signature of the important stabilizing role of disulphide bridges and the challenge they pose for prediction methods.

To analyze the relationship between disulphide bridges and secondary struc-

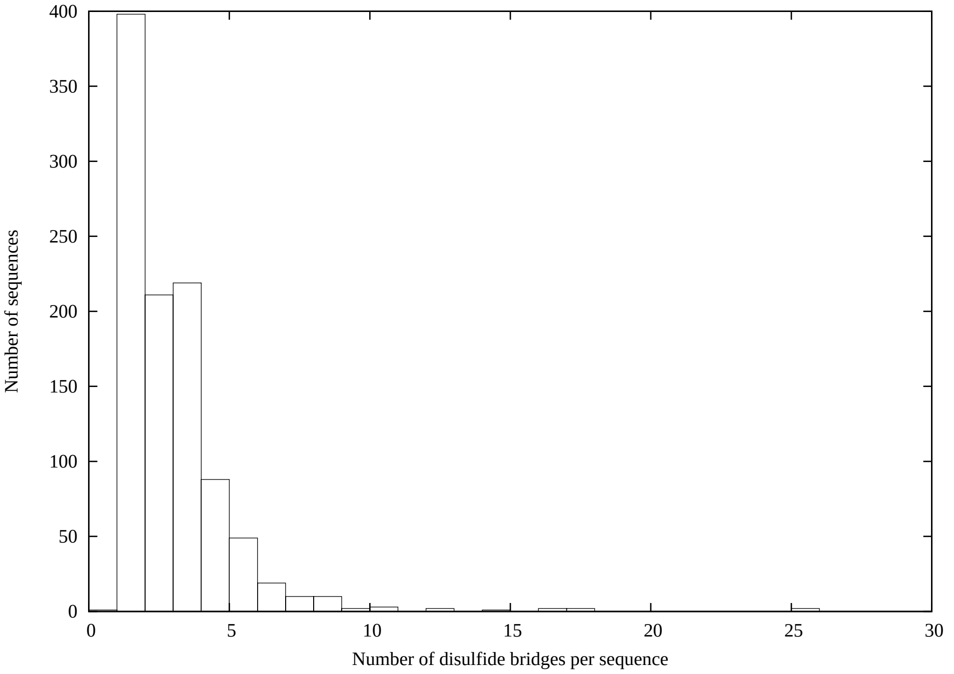


Figure 5.4: Distribution of the number of disulfide bridges per sequence in the SPX dataset

ture and relative solvent accessibility, we compute the empirical distribution of secondary structure classes (Helix, Beta Strand, or Coil) and relative solvent accessibility classes (Exposed or Buried with respect to a 25% cutoff) for both bonded and non-bonded cysteines (Table 5.1). We observe several statistical relationships between the oxidized state of cysteines and their secondary structure and solvent accessibility. For example, 30% of non-bonded cysteines are found in helices, versus only 19% of bonded cysteines. About half of bonded cysteines (49%) are found in coils, versus only 39% for non-bonded cysteines. Most cysteines tend to be buried, however bonded cysteines have a slight tendency towards solvent exposure, compared to non-bonded cysteines. The last two rows of Table 5.1 show slight pairing biases. For instance, 13% of disulphide bridges are established between two beta strands (EE) versus 10% if the secondary structure of the pairs were selected at random (0.32×0.32). Taken together, these statistics suggest that secondary structure

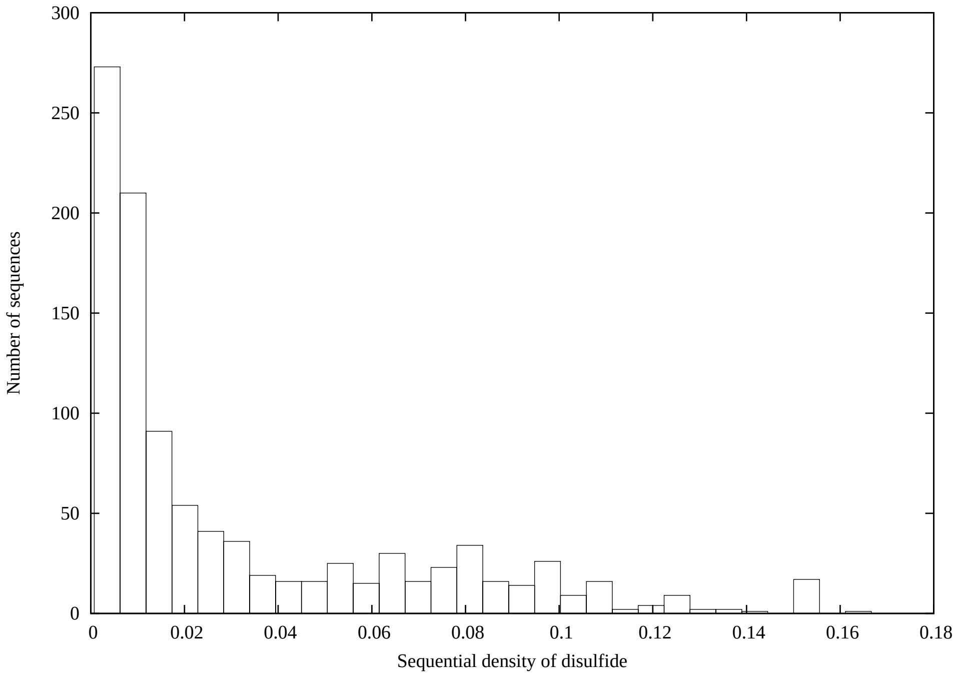


Figure 5.5: Distribution of sequential density of disulfide bridges (number of bridges divided by sequence length) in SPX.

information, and to a lesser extent solvent accessibility information, may be useful for predicting disulphide bridges and worth incorporating in the inputs.

5.3.2 Protein Chain Classification

Results obtained on the problem of separating protein chains containing disulphide bridges from those that do not contain any bridges using kernel methods are shown in Tables 5.2 and 5.3 for the SP41 and the SPX datasets respectively. Each kernel is assessed in terms of sensitivity, specificity, accuracy, and ROC score. The ROC score is the normalized area under the curve relating true positives as a function of false positives. Each performance metric is averaged across the ten folds. Overall the results are consistent across both datasets and confirm expected trends. In general, the more complex and flexible kernels (e.g. profile, mismatch, Fisher) tend to perform better than the simpler kernels (spectrum of fixed length) and combina-

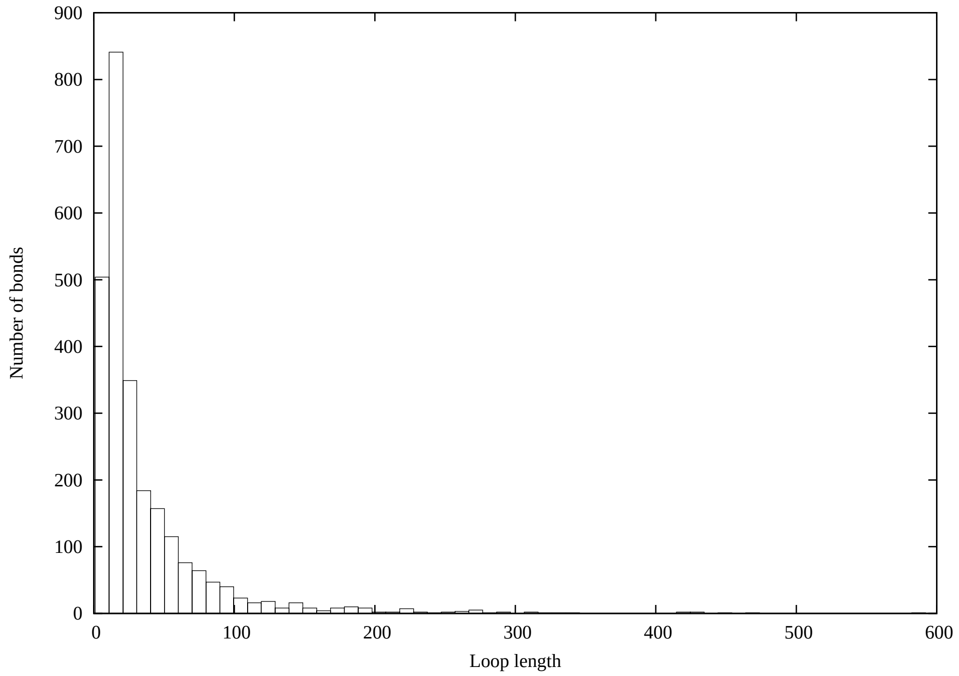


Figure 5.6: Distribution of disulphide bridge lengths in the SPX dataset

tions of spectrum (resp. mismatch) kernels outperform individual spectrum (resp. mismatch) kernels. On the SP41 dataset, for instance, the profile kernel achieves the best overall performance with accuracy of 85% and ROC score of 0.9. The superiority of more complex kernels is less pronounced on the SPX datasets and a few other differences are observed between the two datasets, probably resulting from the fact that SP41 has 966 positive examples and 506 negative examples, and SPX has 897 positive examples and 1,650 negative examples, with much greater variability in the negative examples. In general, the results are weaker on the SPX datasets and higher sensitivity is observed on SP41 versus higher specificity on SPX. On the SPX dataset, the profile kernel is still among the best but is slightly outperformed by the Fisher and combined-mismatch kernels. The combined mismatch kernel, for instance, achieves 75% accuracy and 0.75 ROC score.

Table 5.1: Statistics relating proportion of bonded and non-bonded cysteines in the SPX dataset to secondary structure (SS) and relative solvent accessibility. H = helix, E= strand, C =coil. The first two rows correspond to percentages of individual cysteines and the last two rows to percentages of pairs of cysteines. Random values correspond to the product of the individual frequencies.

Bonding state	Num	Helix	Strand	Coil	Exposed	Buried
Non-bonded Cys	901	0.30	0.31	0.39	0.15	0.85
Bonded Cys	5082	0.19	0.32	0.49	0.21	0.79
SS Pairs	HH	HE	HC	EE	EC	CC
Bonded Pairs	0.07	0.10	0.15	0.13	0.28	0.28
Random Pairs	0.04	0.12	0.19	0.10	0.31	0.24

5.3.3 Disulphide Bridge Classification and Connectivity Prediction Assuming Knowledge of Bonded Cysteines

To compare with previous methods, most of which assume that the bonding state of each cysteine is known, we first train and test 2D DAG-RNN architectures using the SP39 dataset under the same assumption. Thus the output pairing probabilities are predicted only for the cysteines known to participate in a disulphide bridge. The precision percentages at the level of both individual pairs and entire connectivity patterns are reported in Table 5.4 as a function of the number K of disulphide bridges in the chain. In all but one case, the results are better than those previously reported in the literature (Vullo and Frasconi, 2004; Fariselli et al., 2002). In some cases, the results are substantially better. For instance, for 3 disulphide bridges ($K = 3$), the precision reaches 0.61 and 0.51 at the pair and pattern levels respectively, whereas the best results reported in the literature on the same dataset are 0.51 and 0.41. Note that SP39 contains only sequences with 5 bridges or less and thus only results for $K <= 5$ are reported here. The observed improvement in performance is likely to result from the architectural differences between that approach described in Vullo and Frasconi (2004) and the one introduced here.

Table 5.2: Protein classification results using kernel methods on the SP41 dataset. Top two accuracy and mean ROC scores are in bold face. Spectrum k=2,3,4,5) corresponds to the sum of the four spectrum kernels from k=2 to k=5. Mismatch(k=3,4,5,6, m=1) corresponds to the sum of the four mismatch kernels from k=3 to k=6 while m=1 is kept unchanged. For the LA and SW kernels, alignments are derived using the BLOSUM 62 matrix with gap open and extension penalties of 12 and 2 respectively. The scaling parameter β of the LA kernel is set to $\beta = 0.5$.

Kernel	Sensitivity	Specificity	Accuracy	meanROC
Spectrum(k=2)	0.78	0.60	0.72	0.76
Spectrum(k=3)	0.82	0.51	0.71	0.77
Spectrum(k=4)	0.88	0.36	0.70	0.77
Spectrum(k=5)	0.88	0.25	0.66	0.72
Spectrum(k=2,3,4,5)	0.85	0.69	0.80	0.85
Mismatch(k=3, m=1)	0.82	0.58	0.74	0.79
Mismatch(k=4, m=1)	0.86	0.58	0.76	0.83
Mismatch(k=5, m=1)	0.90	0.49	0.76	0.82
Mismatch(k=6, m=1)	0.93	0.08	0.64	0.76
Mismatch(k=3,4,5,6, m=1)	0.86	0.74	0.82	0.87
Fisher kernel	0.75	0.87	0.79	**0.88**
SW kernel	0.83	0.81	0.82	**0.88**
LA kernel	0.89	0.76	**0.84**	0.87
Profile kernel(k=6, σ=9.0)	0.87	0.82	**0.85**	**0.90**

5.3.4 Disulphide Connectivity Prediction from Scratch

In this set of experiments, we do not assume any knowledge regarding whether individual cysteines are disulphide bonded or not and apply the 2D DAG-RNN approach to predict pairing probabilities for *all* pairs of cysteines in each sequence. Thus, for each chain we predict the number of disulphide bridges, and address the residue and bridge classification problems, as well as the global connectivity problem.

Prediction of Cysteine Bonding States (Residue Classification)

Prediction of the bonding state of individual cysteines is assessed in Table 5.5 using the larger SPX dataset. Specificity and sensitivity of bonding state predictions

Table 5.3: Protein classification results using kernel methods on the SPX dataset. Top two accuracy and mean ROC scores are in bold face. Spectrum k=2,3,4,5) corresponds to the sum of the four spectrum kernels from k=2 to k=5. Mismatch(k=3,4,5,6, m=1) corresponds to the sum of the four mismatch kernels from k=3 to k=6 while m=1 is kept unchanged. For the LA and SW kernels, alignments are derived using the BLOSUM 62 matrix with gap open and extension penalties of 12 and 2 respectively. The scaling parameter β of the LA kernel is set to $\beta = 0.5$.

Kernel	Sensitivity	Specificity	Accuracy	meanROC
Spectrum(k=2)	0.63	0.68	0.66	0.71
Spectrum(k=3)	0.56	0.72	0.66	0.67
Spectrum(k=4)	0.39	0.86	0.70	0.66
Spectrum(k=5)	0.25	0.91	0.68	0.62
Spectrum(k=2,3,4,5)	0.54	0.83	**0.73**	0.74
Mismatch(k=3, m=1)	0.57	0.66	0.63	0.67
Mismatch(k=4, m=1)	0.57	0.77	0.70	0.71
Mismatch(k=5, m=1)	0.49	0.87	**0.73**	0.71
Mismatch(k=6, m=1)	0.25	0.94	0.70	0.66
Mismatch(k=3,4,5,6, m=1)	0.56	0.83	**0.74**	**0.75**
Fisher kernel	0.55	0.82	0.72	**0.76**
SW kernel	0.46	0.80	0.68	0.66
LA kernel	0.45	0.88	**0.73**	0.72
Profile kernel(k=6, σ=9.0)	0.49	0.86	**0.73**	0.71

are close to 87% and 89% in the absence of additional secondary structure or relative solvent accessibility information, with at best a small improvement when this information is added.

Prediction of the Number of Disulphide Bridges

Analysis of the prediction results shows that there is a relationship between the sum $S(p)$ of all the probabilities in the graph (or the output layer of the 2D DAG-RNN) and the total number of bonded cysteines. Using both SS and SA as inputs, the correlation coefficient between $2K$ and $S(p)$ is 0.89, the correlation coefficient between $2K$ and M is 0.87, and the correlation coefficient between $2K$ and $\sqrt{(S(p))}\log M$

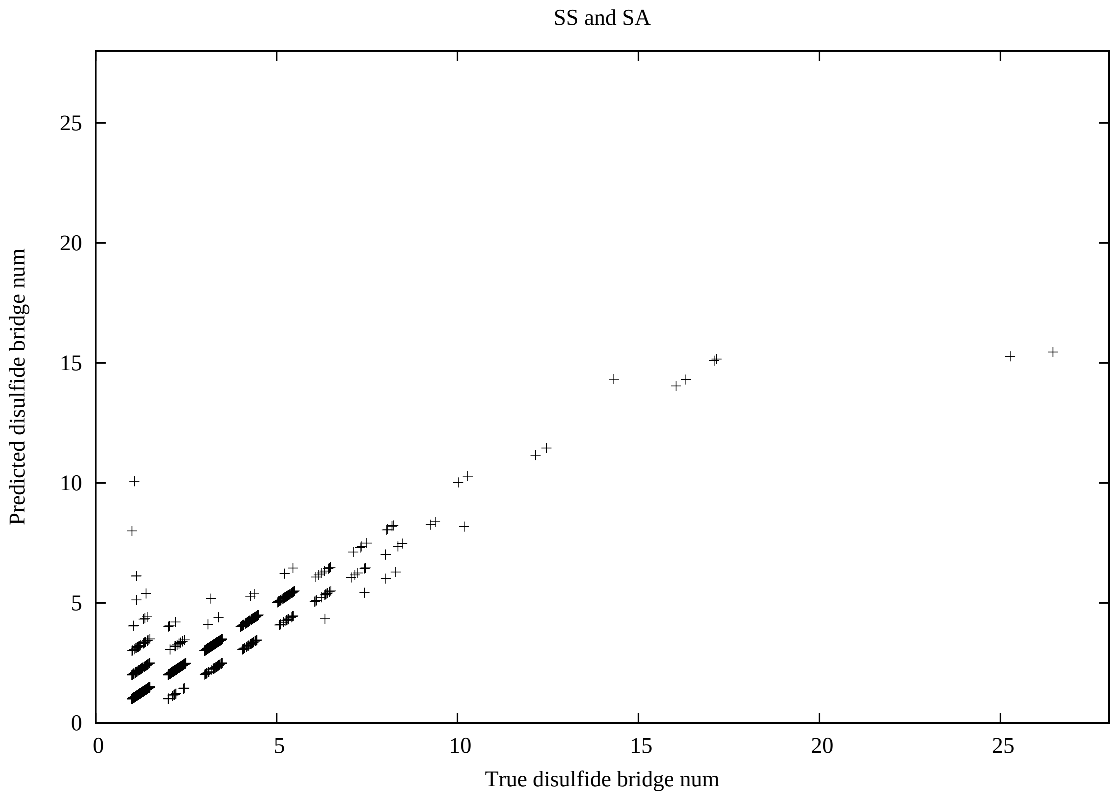


Figure 5.7: Predicted bond number is plotted against the true bond number using both profiles, SS, and SA as inputs. With a total 1,018 protein chains in the SPX dataset, the number of disulphide bridges of 71% of these sequences are predicted correctly using 10-fold cross validation. Uniform random noise in the range of [0,0.5] is added to both the true bond number and the predicted bond number to improve readability.

Table 5.4: Disulphide connectivity prediction with 2D DAG-RNN assuming the cysteine bonding state is known derived on the SP39 dataset for comparison purposes. Last row reports performance on all test chains. Asterisque indicates level of precision exceeding best previously reported results given in parentheses (Vullo and Frasconi, 2004).

K	Pair Precision	Pattern Precision
2	0.74* (0.73)	0.74* (0.73)
3	0.61* (0.51)	0.51* (0.41)
4	0.44* (0.37)	0.27* (0.24)
5	0.41* (0.30)	0.11 (0.13)
$2 \dots 5$	0.56* (0.49)	0.49* (0.44)

Table 5.5: Cysteine bonding state sensitivity and specificity with different combinations of secondary structure and solvent accessibility information on the SPX dataset. SS = secondary structure; SA = solvent accessibility; PSS = predicted secondary structure; PSA = predicted solvent accessibility.

	no SS no SA	SS	SA	SS and SA	PSS and PSA
Bond. State Sens.	0.886	0.883	0.884	0.894	0.889
Bond. State Spec.	0.876	0.878	0.872	0.878	0.879

is 0.94, where M is the total number of cysteines in the sequence being considered. Thus, we estimate the total number of bonded cysteines using this linear regression approach and rounding off the result, making sure that the total number of bonded cysteines is even and does not exceed the total number of cysteines in the sequence. Figure 5.7 represents the plot of predicted bond numbers against true bond numbers on the SPX dataset. As shown in the plot, the bond number prediction is rather accurate for most $K > 1$ cases, with few exceptions for very large K ($K > 20$). For $K = 1$, the method tends to over-predict the number of bridges. Table 5.6 reports the accuracy for predicting the number of bridges. The total number of disulphide bridges in 68% chains is correctly predicted with no additional inputs, with a standard error (mean square root of residuals) of 1.06. With true SS and SA input information the performance reaches 71% of correct predictions, with a

Table 5.6: Prediction accuracy for the number of disulphide bridges on the SPX dataset

	no SS no SA	SS	SA	SS and SA	PSS and PSA
Accuracy (num. of bridges)	0.68	0.68	0.67	0.71	0.68
Mean square root of residual	1.06	1.05	1.09	1.04	1.05

standard error of 1.04. In more than 94% of the cases, the predicted number of bridges is within one from the correct value. With predicted SS and SA there is no noticeable improvement (68% accuracy).

Prediction of Disulphide Bridges (Bridge Classification)

Table 5.7 reports the specificity and sensitivity for the prediction of individual bridges. The sensitivity for chains with one disulphide bridge is around 71 %, while the specificity is around 47%. Both specificity and sensitivity for chains with two or three disulphide bridges using true SA and SS information in the inputs fall in the range of 62-67%. The specificity and sensitivity for chains with four disulphide bridges using true SA and SS information are 55% and 50% respectively.

When the number of disulphide bridges increases in chains, the performance decreases in general. The overall specificity and sensitivity using four different input schemes are around 51-55%. The variation of the performance for chains with many disulphide bridges ($K > 6$) is large because there are very few such examples in the dataset. Thus, we should take the accuracy for large disulfide bridge number beyond six cautiously. The results also show that secondary structure information improve prediction accuracy of disulphide bridges by two percentage points on average. Solvent accessibility alone does not help much, but when used in combination with secondary structure the best results are achieved in most cases. Predicted SS

Table 5.7: Specificity and sensitivity for the disulphide bridge classification problem derived on the SPX dataset, as a function of the number K of bridges in the chain from 1 to 26, and with different combinations of input information.

	no SS no SA		SS		SA		SS and SA		PSS and PSA	
K	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
1	0.71	0.47	0.71	0.47	0.70	0.46	0.71	0.48	0.71	0.48
2	0.59	0.59	0.63	0.63	0.59	0.59	0.63	0.63	0.59	0.60
3	0.59	0.65	0.61	0.67	0.58	0.64	0.62	0.67	0.55	0.61
4	0.44	0.49	0.48	0.53	0.46	0.52	0.50	0.55	0.44	0.48
5	0.33	0.37	0.33	0.37	0.31	0.35	0.37	0.41	0.32	0.35
6	0.24	0.28	0.32	0.37	0.29	0.34	0.29	0.33	0.32	0.36
7	0.26	0.31	0.30	0.36	0.21	0.25	0.31	0.36	0.29	0.32
8	0.16	0.18	0.21	0.25	0.26	0.30	0.30	0.32	0.20	0.22
9	0.50	0.59	0.56	0.64	0.55	0.63	0.61	0.71	0.44	0.52
10	0.4	0.43	0.27	0.29	0.40	0.43	0.37	0.40	0.33	0.36
12	0.38	0.44	0.54	0.61	0.46	0.58	0.50	0.55	0.38	0.39
14	0.71	0.83	0.50	0.54	0.42	0.50	0.57	0.62	0.79	0.85
16	0.19	0.20	0.19	0.20	0.31	0.33	0.22	0.23	0.13	0.13
17	0.35	0.40	0.41	0.47	0.38	0.43	0.35	0.40	0.53	0.60
25	0.08	0.13	0.24	0.40	0.28	0.47	0.24	0.40	0.32	0.53
26	0.38	0.67	0.31	0.53	0.23	0.40	0.42	0.73	0.31	051
Overall	0.52	0.51	0.54	0.53	0.52	0.51	0.55	0.54	0.52	0.51

and SA do not seem to help.

Table 5.8 reports the results of disulphide bridge classification on the SP41 dataset. On this dataset, only sequence information and profiles are used in the RNN input. While the accuracy on the SP41 dataset is lower than that on the SPX dataset, it follows the same pattern and in general deteriorates with the number of bridges.

Prediction of Disulphide Bridge Connectivity Patterns

It is very difficult to correctly predict the entire disulphide connectivity pattern because the number of connectivity patterns increases exponentially with K. Not

Table 5.8: Prediction of disulphide bridges with 2D DAG-RNN on all the cysteines, without assuming knowledge of the bonding state on the SP41 dataset.

K	1	2	3	4	5	6	7	8	9	10	11	12	15	16	17	18	19
Sens.	.74	.61	.54	.52	.33	.27	.36	.27	.23	.30	.34	.17	.27	.11	.22	.06	.11
Spec.	.39	.51	.45	.59	.42	.34	.55	.41	.35	.45	.47	.23	.50	.13	.33	.09	.20

Table 5.9: The prediction accuracy of disulphide bridge connectivity on the SPX dataset

Bridge Num	no SS no SA	SS	SA	SS and SA	PSS and PSA
1	0.58	0.60	0.58	0.59	0.59
2	0.55	0.58	0.55	0.59	0.56
3	0.50	0.53	0.50	0.54	0.47
4	0.27	0.33	0.28	0.34	0.22
Overall(1-26)	0.48	0.50	0.48	0.51	0.48

knowing the bonding states of individual cysteines makes the prediction even harder. Table 5.9 reports the pattern prediction accuracy for K between 1 and 4, and the overall accuracy for all the chains in the SPX dataset. The overall accuracy with no additional input information is 48% and reaches 51% using true SS and SA as inputs. Thus for about half of all the chains, we can predict the entire pattern of disulphide bridges correctly. Consistently with our other experiments and with recent results in (Ferre and Clote, 2005), using true SS and SA information slightly improves performance, however predicted SS or SA information seems too noisy at this stage to be helpful.

5.4 Conclusion

We have presented a framework for disulphide bridge predictions that addresses all four sub-problems in this area: chain classification, residue classification, bridge

Table 5.10: Performance comparison of the disulphide prediction pipeline (residue and bridge classification) with and without the initial chain classification step. The F measure here is defined by the harmonic mean of sensitivity and specificity, F= 2 × sens. × spec. /(sens. + spec.).

	Residue			Bridge		
	Sens.	Spec.	F	Sens.	Spec.	F
Without Chain Classification	0.89	0.39	0.54	0.55	0.24	0.33
With Chain Classification	0.52	0.77	0.62	0.32	0.48	0.38

classification, and connectivity prediction. Table 5.10 summarizes the motivation for the initial chain classification step, by comparing overall results on residue and bridge classification derived with and without the chain classification step. In all cases, as expected, the chain classification step increases the specificity but reduces the sensitivity. The tradeoff, assessed by the F measure of information retrieval, is in favor of having the chain classification step (F=0.62 versus F = 0.54 for residue classification, and F= 0.38 versus F= 0.33 for bridge classification). Thus retaining the chain classification step in the overall pipeline is justified both in terms of overall performance, and because the classification can be of biological interest too. Furthermore, our Web server reports the results of each prediction stage separately.

Beyond the chain classification step, the prediction pipeline we have described presents several advantages over other approaches. First, assuming knowledge of cysteine bonding states, the method outperforms existing approaches on the same validation data. Second, the method can easily cope with chains containing an arbitrary number of bonded cysteines, overcoming the limitation of previous approaches which restrict predictions to chains containing at most 10 oxidized cysteines ($K = 5$). As an added bonus, larger training and testing sets can now be used. Third, the method proposed can deal with *ab initio* predictions and, in particular, it does not require preexisting knowledge or prediction of cysteine bonding states. Good specificity and sensitivity on connectivity predictions are achieved even when the bonding state of individual cysteines is not known. Equally important, for previous methods that rely on predicting the cysteine bonding state first, false predictions are fatal. Once a false prediction has been made at the residue level, the corresponding disulphide bridges cannot be recovered (false negative) or eliminated (false positive) during subsequent bridge classification, or connectivity prediction. When used in *ab initio* mode, the method presented here delays the prediction of the bonding state by first predicting the total number of disulphide bridges in a cooperative, robust fashion, and then globally predicting the overall connectivity from which cysteine bonding states are trivially inferred. The same fundamental idea of combining pairing probabilities with graph matching algorithms to enforce global constraints has now been expanded and applied to the problem of beta-sheet topology prediction (Cheng and Baldi, 2005). Fourth, the method can leverage true secondary structure and relative solvent accessibility information. Results demonstrate the role secondary structure and solvent accessibility can play in disulphide bridge prediction. Overall, inclusion of SS and SA information leads to small, but noticeable improvements in the range of one percent. This is demonstrated by appropriately encoding the corresponding information in the input layer of the architecture. Predicted SS or SA information, however, is currently not accurate enough to improve performance and therefore is not retained in our implementation. Finally, while training can take days, once trained predictions can be carried on a proteomic or protein engineering scale to sift through large numbers of proteins. The resulting disulphide bridge prediction server DIpro is available through: http://www.igb.uci.edu/servers/psss.html.

Chapter 6

Three-Stage Prediction of Protein Beta-Sheets by Neural Networks, Alignments, and Graph Algorithms

6.1 Introduction

Beta-sheets are a fundamental component of protein architectures–more than 75% of all protein domains in the Protein Data Bank (Berman et al., 2000) contain β-sheets (Zhang and Kim, 2000). β-sheets are formed by the pairing of multiple β-strands held together by characteristic patterns of hydrogen bonds running in parallel or anti-parallel fashion (Figure 6.1). These patterns, which are essential for β-sheet and protein stability (Smith and Regan, 1997), involve interactions between residues that are often separated by large distances along the primary sequence.

The β-sheet topology or architecture of a protein, i.e. the pairing organization of all the β-strands contained in a given protein, is essential for understanding its structure (Zhang and Kim, 2000). Prediction of β-sheet topology from amino acid sequence is very useful not only for predicting tertiary structure (Zaremba and Gregoret, 1999; Steward and Thornton, 2002; Ruczinski et al., 2002; Rost et al., 2003), but also for elucidating folding pathways (Merkel and Regan, 2000; Mandel-Gutfreund et al., 2001), and designing new proteins (Smith and Regan, 1995, 1997; Kortemme et al., 1998; Kuhlman et al., 2003). Many experimental and theoretical studies have been conducted to better understand the formation and stability of β-sheets. For instance, Minor and Kim (1994) report that intrinsic β-sheet propensities of different amino acids contribute to the local structure and stability of β-sheets and that the magnitude and order of β-sheet propensities depend on the local sequence and structural context. Statistical studies (Lifson and Sander, 1980; Wouters and Curmi, 1995) reveal nonrandom distribution and pairing preferences

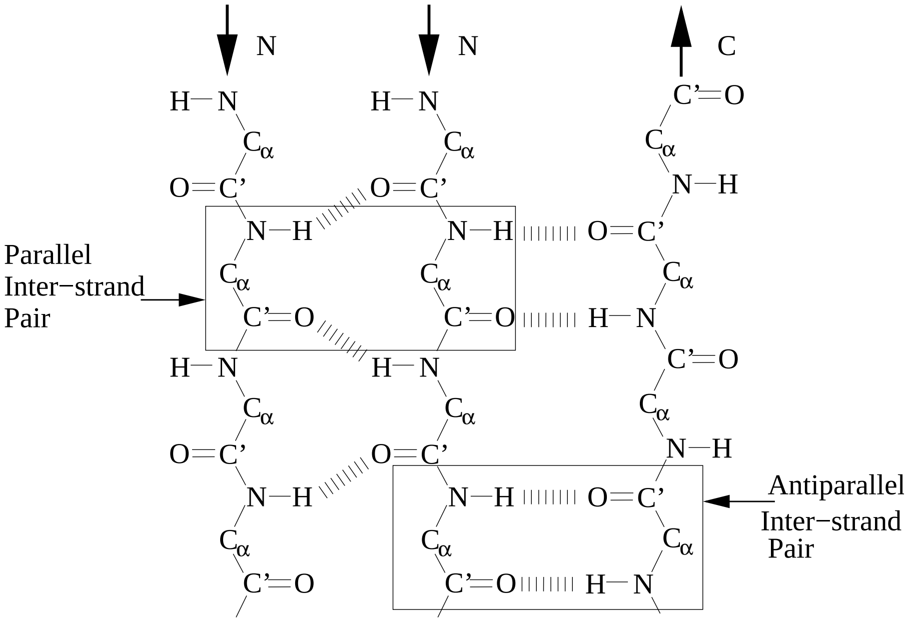


Figure 6.1: Illustration of inter-strand β-residue pairs and hydrogen-bonding pattern in parallel and antiparallel β-strands. Arrows show the amide (N) to carbonyl (C) direction of β-strands. Hydrogen bonds are represented by hatched blocks.

of residue pairs in aligned β-strands while evolutionary conservation of β-residue interactions suggests also that pairing preferences depend on structural context such as solvent accessibility (Zaremba and Gregoret, 1999). Clearly, favorable side-chain interactions between residue pairs contribute to β-sheet stability (Smith and Regan, 1995; Hutchinson et al., 1998). However, the evolutionary pressure to maintain complementarity between pairs on neighboring strands appear to be weak (Mandel-Gutfreund et al., 2001) and the overall pairing preferences are not very strong and appear to be modulated by the local environment to a high-degree.

Several methods, mostly statistical data-driven approaches, have been proposed to predict topological features of β-sheets with moderate accuracy (Rost et al., 2003). An early method (Hubbard, 1994) uses a statistical potential approach to predict β-strand alignments with an accuracy level of about 35-45%. Asogawa (1997) proposes to use pairwise statistical potentials of β-residue pairs to improve β-sheet secondary structure prediction by considering clusters of β-residue contacts. Pairwise statistical potentials are used also in Zhu and Braun (1999) to identify up to 35% of native strand alignments from alternative strand alignments. In Baldi et al. (2000), elaborate neural networks are used to improve the prediction accuracy of inter-strand β-residue contacts, but the method is not extended to the prediction of strand pairings, strand alignments, and β-sheet topologies. Using an information theoretic approach, Steward and Thornton (2002) report an accuracy of 45-48% for strand alignments in β-triplets, and 31-37% for any native strand alignments. While encouraging, all these approaches seem to leave room for major improvements.

In particular, these approaches fail to exploit systematically the global covariation and constraints characteristic of β-sheet architectures. Instead of treating each pair of β-residues or β-strands independently of each other, as previous methods do, one ought to leverage β-sheet constraints such as the fact that each β residue has at most two partners, that neighboring β-residues in a strand are paired sequentially in parallel or anti-parallel fashion with another strand, and that each β-strand has at least one partner strand and rarely more than two or three partner strands.

Here we develop a novel modular approach for predicting inter-strand β-residue pairings, β-strand pairings, β-strand alignments, and β-sheet topology altogether from scratch by integrating both local and global constraints in three steps. First, 2D-Recursive Neural Networks(2D-RNN)(Baldi and Pollastri, 2003) are trained to predict pairing probabilities of inter-strand β-residue pairs using profile, secondary structure, and relative solvent accessibility information. Second, dynamic programming techniques are applied to these probabilities to derive pairing pseudo-energies and alignments between all pairs of β strands. Third, weighted graph matching algorithms are used to optimize the global β-sheet architecture of the protein satisfying the β-strand pairing constraints. While inter-chain β-sheets play an important role in protein-protein interactions and complex formation (Dou et al., 2004), it is worth noting that here, consistently with the available literature, we focus exclusively on the already-challenging prediction of intra-chain β-sheets. However, we believe that the methods developed here can be adapted to the problem of predicting both intra- and inter-chain β-sheets and training datasets for the latter are available through the ICBS database (Dou et al., 2004).

6.2 Materials and Methods

6.2.1 Data

The dataset is extracted from the Protein Data Bank of May 2004. Only structures determined by X-ray diffraction and having resolution better than 2.5 Å are retained. Chains containing unknown or non-standard amino acids, backbone interruptions, or whose length is less than 50 amino acids are excluded. DSSP (Kabsch and Sander, 1983) is used to assign secondary structure and relative solvent accessibility values to each residue. Residues with secondary structure E (extended strand) and B (isolated β-bridge) are considered β-residues. Each β-residue may have 0, 1 or 2 partners according to DSSP. A consistency check is used to remove chains containing non-consistent β-residue pair assignments (e_i, e_j), whereby e_i pairs with e_j, but e_j does not pairs with e_i according to DSSP. A filtering procedure is used to select the chains that contain 10-100 β-residues, of which 90% must have at least one partner. The redundancy in the dataset is reduced by the UniqueProt (Mika and Rost, 2003) with a HSSP threshold of 0, which corresponds to sequence identity of roughly 15-20%.

The final dataset contains 916 chains corresponding to 187,516 residues. Of these, 26% (48,996) are β-residues participating in 31,638 inter-strand residue pairs. The dataset has 10,745 β-strands with an average length of 4.6 residues and 8,172 β-strand pairs, including 4,519 antiparallel pairs, 2,214 parallel pairs, and 1,439 pairs involving isolated β-bridges. These strand pairs form 2,533 β-sheets. The average sequence separation between residue pairs and strand pairs is 43 and 40 respectively. Sequence separation histograms are displayed in Figures 6.2a and 6.2b. Figures 6.2c and 6.2d show that the number of inter-strand residue pairs or strand pairs has a strong correlation with the number of β-residues or strands in the chain, as expected.

To leverage evolutionary information, PSI-BLAST (Altschul et al., 1997) is used to generate profiles by aligning all chains against the Non-Redundant (NR) database, as in Pollastri et al. (2002b). Finally the dataset is evenly and randomly split into 10 folds to perform 10-fold cross-validation studies. The final dataset (BetaSheet916) and the splitted folds are available through http://www.igb.uci.edu/servers/psss.html.

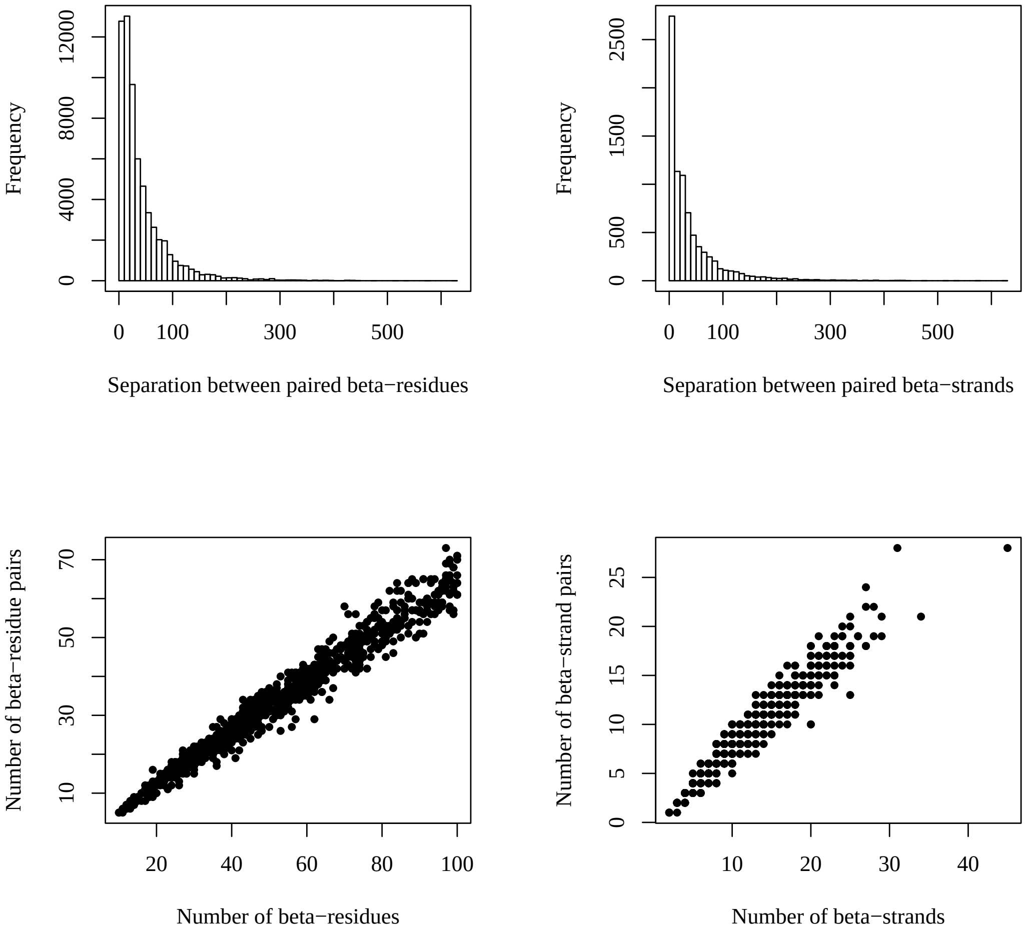


Figure 6.2: **(a)** Amino acid separation between β-residue pairs (Mean = 43, Minimum = 3, Maximum = 626, Standard deviation = 49). **(b)** Amino acid separation between β-strand pairs (Mean = 40, Minimum = 2, Maximum = 626, Standard deviation = 54). **(c)** Scatterplot of number of β-residue pairs (y) versus number of β-residues (x) per chain. The correlation coefficient is 0.98. Linear regression given by: $y = 0.66x - 0.65$. **(d)** Scatterplot of number of β-strand pairs(y) versus number of β-strands (x) per chain. The correlation coefficient is 0.97. Linear regression given by: $y = 0.74x + 0.27$.

6.2.2 Prediction of β-residue pairs using 2D-RNNs

Like contact map prediction (Fariselli et al., 2001; Pollastri and Baldi, 2002; Shao and Bystroff, 2003; MacCallum, 2004; Punta and Rost, 2005), we treat prediction of inter-strand residue pairing as a binary classification problem on a 2D grid. For each chain, our input is a 2D square matrix **I**, where the size of **I** is equal to the number of β-residues in the chain and each entry $I_{i,j}$ is a vector of dimension 251 encoding the local context information of β-residues (e_i,e_j), as well as their separation. Specifically, we use a local window of size 5 around e_i and e_j. Each position in the window corresponds to a vector of length 25 with 20 positions for the amino acid profile, 3 positions for the secondary structure (Helix, Sheet, Coil), and 2 positions for the relative solvent accessibility (buried or exposed at 25% threshold). The two windows correspond to 250=25×5×2 entries. One additional entry represents the sequence separation between e_i and e_j.

The training target is a binary matrix **T**, whereby each $T_{i,j}$ equals 1 or 0 depending on whether β-residue e_i and e_j are paired or not. Figure 6.3 and 6.4 show protein 1VJG in the PDB and its corresponding target matrix which nicely displays the constraints and directions (parallel or antiparallel) of strand pairing. Neural networks or other machine learning methods can be trained on the data set to learn a mapping from the input matrix **I** onto an output matrix **O**, whereby $O_{i,j}$ is the predicted probability that e_i and e_j are paired. The goal is to make the output matrix **O** as close as possible to the target matrix **T**. The standard approach with feed-forward neural networks is to treat each pair (e_i,e_j) independently and to learn a mapping from a series of independent $(I_{i,j}, T_{i,j})$ examples (Baldi et al., 2000). This simplified approach, however, does not explicitly leverage covariations and interactions between β-residue pairs and might not effectively enforce the constraints

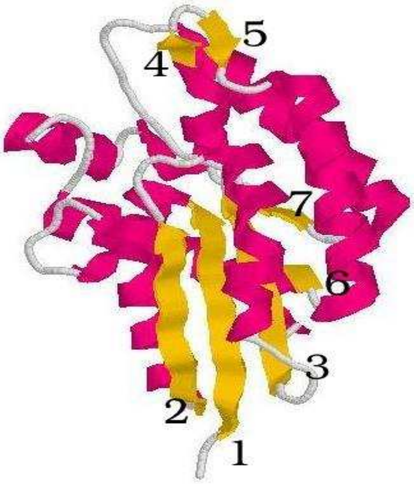


Figure 6.3: Protein 1VJG is an α/β protein with 7 strands. Strands 1, 2, 3, 6, and 7 form a parallel β-sheet. Strands 4 and 5 form an antiparallel β-sheet. The parallel β-sheet forms the hydrophobic core and is surrounded by tightly packed α-helices.

of β-residue and strand pairings. Here we use a two-dimensional recursive neural network architecture to exploit covariations and constraints between β-residue pairs globally. This 2D-RNN architecture, previously used in contact map prediction, is described in detail in Baldi and Pollastri (2003) and in chapter 5. Under this architecture, the output $O_{i,j}$ depends on the entire input matrix $\mathbf{I}$ instead of $I_{i,j}$ only. As for feed-forward neural networks, learning in a 2D-RNN is implemented using gradient descent. In the simulations, the outputs of five models are averaged in an ensemble to produce the predicted probability matrix $\mathbf{O}$. Finally, it is important to notice that because our approach is modular, it is not constrained in any way to the use of recursive or even feedforward neural networks–the output of any algorithm that produces an estimate of the pairing probabilities O_{ij} can be used as input for the second and third steps described below.

Because $\mathbf{I}$ and $\mathbf{T}$ are presented to the 2D-RNN as a whole during training, the network can identify pairing constraints encoded in these matrices beyond the local environment of each residue. As a result, by thresholding the values of the output

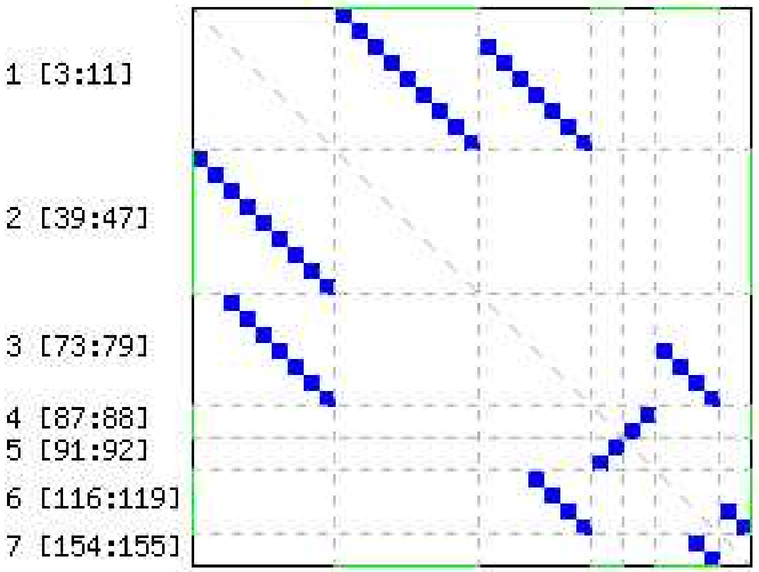


Figure 6.4: Inter-strand β-residue pairing map of protein 1VJG. The seven strands are ordered along the vertical and horizontal axis. Alternating colors (black and green) are used to distinguish adjacent strands in sequence order. The three numbers associated with each strand on the left are strand number and its starting and ending position along the chain. The map is symmetric. Each blue square represents a native β-residue pairing. A line segment parallel to the main diagonal corresponds to the alignment of a parallel strand pair. A line segment perpendicular to the main diagonal corresponds to the alignment of an antiparallel strand pair. Each row or column has at most two blue squares reflecting the constraint that one residue has at most two partners.

O, the predicted inter-strand residue pairs tend to form line segments parallel or perpendicular to the main diagonal, which correspond to parallel or antiparallel strand pairs. This suggests that aggregate prediction of β-residue pairings can be used to predict β-strand pairings, pairing directions, and alignments. Figure 6.5 shows the predicted inter-strand residue pairs of 1VJG with a 0.15 threshold. The predicted map recalls most β-residue pairs and satisfies pairing constraints with few violations. It is worth noting that post-prediction inferences can be used to further enforce some constraints and retrieve some of the missing residue pairs. The predicted inter-strand β-residue map can be used directly to infer β-strand pairs.

One simple approach we tested is to consider two strand paired if any two of their residues are predicted to be paired. In isolation, however, such an approach cannot be optimal since it disregards global constraints on the number of partners a strand can have (see Section 2.4).

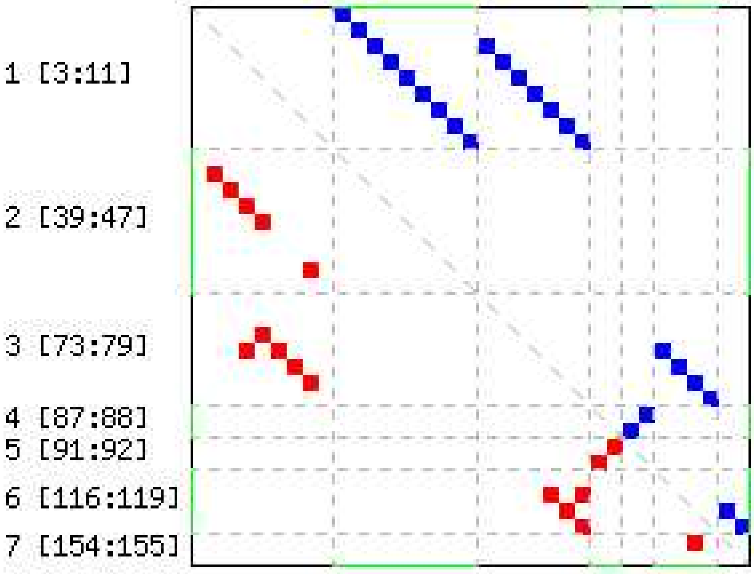


Figure 6.5: Predicted β-residue pairing map of 1VJG. Upper triangle (blue) is the true map and lower triangle (red) is the predicted map. The predicted pairs form three segments parallel to the main diagonal corresponding to the true parallel strand pair (1,2), (1,3),and (3,6). Two residue pairs in the true antiparallel strand pair (4,5) are also recalled. One out of two residue pairs in the parallel strand (6,7) is correctly predicted. There are two false positives in strand pair (1,3) and (3,6). For instance, one residue in strand 3 is wrongly predicted as having two partners in strand 1. This error can be detected by checking pairing constraints: a residue can have up to two partners in total, and at most one partner in any single strand. A few residue pairs between strand 1 and 2, which are missing in the predicted map, can be inferred once strands 1 and 2 are predicted to pair.

6.2.3 Pseudo-energy for β-strand alignment

For each pair of strands, we can define an optimal alignment and an overall alignment score using dynamic programming techniques in parallel and anti-parallel directions with local scores or penalties derived from the matrix $\mathbf{O}$ of residue-pairing probabilities. Additional intra-strand gap penalties corresponding to β-bulges, as well as penalties for gaps at the end of the strands, can be introduced. The penalty for the bulges can be derived from their frequency. Because β-bulges tend to be isolated and rare (only 14% of paired strands contain a bulge, and 90% of these contain only a single bulge), to a first-order-approximation here we do not allow bulges in the alignments by setting the bulge penalty to infinity. This is also consistent with previous studies (Hubbard, 1994; Zhu and Braun, 1999; Steward and Thornton, 2002). Gaps at the edges of the strands are allowed but are not penalized (penalty =0). Under these assumptions, we can simply search exhaustively through all possible alignments by "sliding" one strand along the other, in both parallel and anti-parallel fashion. Assuming in addition that two paired strands must have at least one residue pairing, two strands with length $m \geq 2$ and $n \geq 2$ have $2(m+n-1)$ possible alignments, counting parallel and antiparallel directions. If one strand is an isolated bridge ($m = 1$ or $n = 1$), then there are $max(m, n)$ possible alignments. Without considering β-bulges, one alignment can be uniquely specified by its direction (parallel, antiparallel, or isolated bridge) and by one inter-strand residue pair.

To discriminate native alignments from alternative ones, the binding pseudo-energy $W(\mathcal{A}[E_r, E_s])$ of each alignment $\mathcal{A}$ of each pair of strands E_r and E_s can be computed by adding the pseudo-energies of each pair of residues i and j in the alignment, derived from the pairing probabilities O_{ij}, or their logarithm $\log O_{ij}$. The binding pseudo-energy W_{rs} of a pair of strands can then be defined by taking the maximum over all their possible alignments: $W_{rs} = \max_{\mathcal{A}} W(\mathcal{A}[E_r, E_s])$. For any pair of strands r and s in a given protein chain, the pseudo-energy is used to identify the best putative alignment, i.e. the one with maximal pseudo-energy W_{rs}, between these two strands. Figure 6.6a shows the resulting pseudo-energy matrix

$\mathbf{W} = (W_{rs})$ for the best alignments between all strand pairs of protein 1VJG. Note how the native strand pairs tend to have higher energy scores suggesting that the pseudo-energy can be used effectively to score and rank strand pairs.

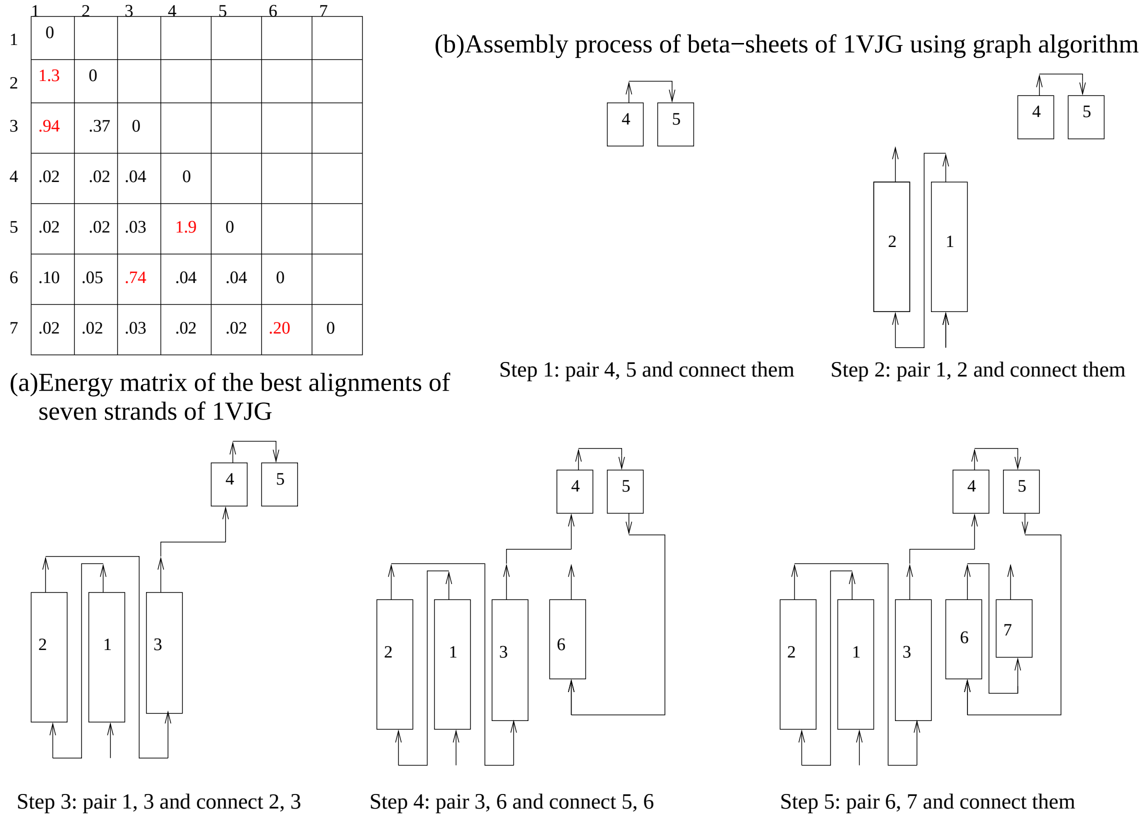
6.2.4 Prediction of β-strand pairs and β-sheet topology using graph algorithms

Unlike previous methods (Hubbard, 1994; Zhu and Braun, 1999; Steward and Thornton, 2002) which treat strand pairs independently of each other, here prediction of strand pairing and alignment takes into account additional physical constrains characteristic of β-sheet architectures. To illustrate β-sheet topology and its constraints, we use schematic diagrams (similar to Branden and Tooze (1999)) where β-strands are represented by rectangles of length proportional to the length of the strand. Figure 6.7 shows the diagram of 1VJG. Lines with arrows connect adjacent strands in sequence order from the N to the C terminus. Such schematic diagrams readily reveal several pairing constraints for β-sheet architectures. First, each strand has two edges available for pairing with other strands and, as a result, a β-residue can have at most two partners. It is important to note that this does *not* imply that a strand can pair at most with two other strands, since a long strand may pair with several short strands on either side. Second, one strand can only pair with one side of another strand sequentially in parallel or antiparallel fashion. If two strands pair with the same side of another strand, no overlap is allowed. Third, all strands must have at least one strand partner (ignoring inter-chain pairings) and we impose the additional condition that they should have at most three strand partners. This condition is not absolute but it is very reasonable since 98.6% of strands have 1, 2, or 3 partners in the large non-redundant dataset. We let $\mathcal{C}$ denote all these constraints.

With these constraints in mind, we develop graph matching algorithms to infer strand pairings and overall β-sheet architecture from the matrix $\mathbf{W}$ of pseudo-energies of the best alignments of all strands pairs in a given chain. This pseudo-energy matrix defines a completely connected and weighted Strand Pairing Graph (SPG), where vertices represent strands, edges represent possible pairing relations, and weights optimal pairing energies. The fully connected SPG of course does not satisfy the set of constraints $\mathcal{C}$. To predict the β-sheet topology, the goal is to prune the complete SPG to derive the true SPG (Figure 6.8), where β-sheets appear as maximal connected components. These components are to be derived by maximizing the global pseudo-energy while satisfying all the strand pairing constraints above, that is by maximizing $\sum_S W_{rs}$ taken over all subsets S of edges that satisfy $\mathcal{C}$. The global pseudo-energy of an architecture is the sum of the pseudo-energies of each of its β-sheets, and the pseudo-energy of a β-sheet is the sum of the pseudo-energies of all the strand pairs it comprises. To address this constrained optimization problem, we first use a greedy heuristic approach (see box).

```
Start with a complete SPG with weight matrix $\mathbf{W}$. Order all the edges according to the weights into a list $L$.
$\emptyset \rightarrow S$. $S$ is the set of chosen edges.
Repeat
    Remove one edge $e$ with maximum weight from $L$.
    If both vertices of $e$ are not in $S$, add $e$ into $S$.
    If both vertices of $e$ are in $S$, discard $e$.
    If one vertex of $e$ is in $S$, align the strand of the vertex
        with the strand of another vertex not in $S$ using
        their best alignment. If the pair and its alignment
        satisfy the strand pairing constraints $\mathcal{C}$, add $e$ into $S$.
        Otherwise discard $e$.
Until all vertices in $G$ appear in $S$ once.
```

The greedy algorithm has time complexity $O(N^2 \log N)$, where N is the number of strands. After converging, the edges and vertices in S constitute a spanning sub-



	1	2	3	4	5	6	7
1	0						
2	1.3	0					
3	.94	.37	0				
4	.02	.02	.04	0			
5	.02	.02	.03	1.9	0		
6	.10	.05	.74	.04	.04	0	
7	.02	.02	.03	.02	.02	.20	0

(a)Energy matrix of the best alignments of seven strands of 1VJG

(b)Assembly process of beta−sheets of 1VJG using graph algorithm

Step 1: pair 4, 5 and connect them

Step 2: pair 1, 2 and connect them

Step 3: pair 1, 3 and connect 2, 3

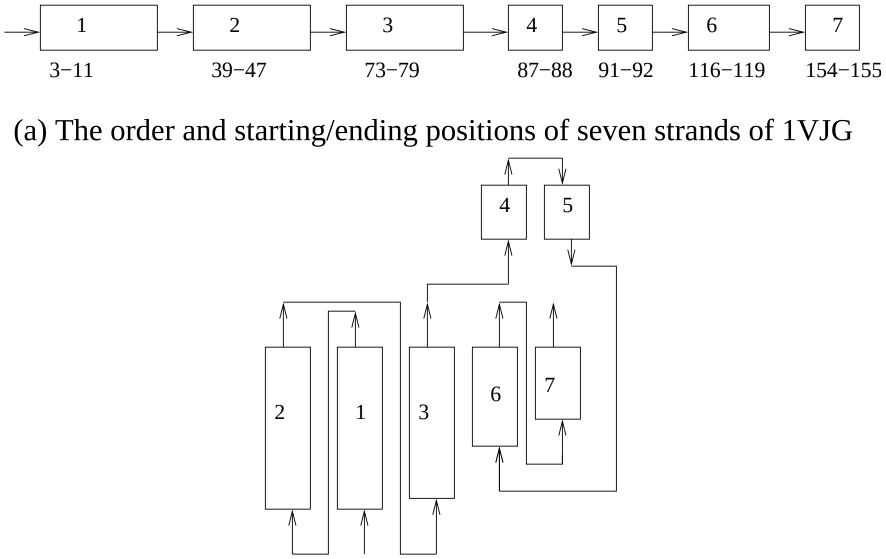
Step 4: pair 3, 6 and connect 5, 6

Step 5: pair 6, 7 and connect them

Figure 6.6: **(a)** Predicted pseudo-energy matrix $\mathbf{W}$ of the best alignments of all strand pairs of protein 1VJG. Red numbers denote the pseudo-energy of the alignments of true strand pairs. **(b)** β-sheet assembly process. It takes five steps to assemble seven strands into two β-sheets using the energy matrix in (a). In step 1-4, the strand pair with maximum energy is added. In step 5, pair(2,3) has higher energy than pair(6,7). But it is not chosen because strand 2 and 3 have already been selected in previous steps.

graph G^* of G. Connected components in G^* are in 1:1 correspondence with the protein β-sheets and provide the global predicted β-sheet architecture. Figure 6.6 illustrates how the algorithm assembles the strands of protein 1VJG.

By treating β-sheets as spanning trees of complete SPGs, a variant of the well-known algorithm for finding Minimum/Maximum Spanning Tree (MST) (Even, 1979), Kruskal's algorithm (Kruskal, 1956), is also used to predict β-sheets (trees) with maximum pseudo-energy. The only difference between this constrained-MST algorithm with the previous greedy algorithm is that it does not always discard edge



(a) The order and starting/ending positions of seven strands of 1VJG

(b) Strand pairing topology of beta−sheets

Figure 6.7: Schematic diagram of β-sheet topology of protein 1VJG. **(a)** Unpaired strands in sequence order. **(b)** Paired strands in each β-sheet are aligned side by side. This diagram includes two β-sheets consisting of strand 1,2,3,6,7 and strand 4,5 respectively.

e when its adjacent vertices are already in the set S. Instead, it adds e into L if its two vertices belong to two disconnected components and the alignment satisfies the strand pairing constraints. Not surprisingly this algorithm tends to choose more strand pairs (edges) than the greedy graph algorithm. It is worth noting that both the greedy and constrained-MST algorithms as described do not allow for cycles and all the components they produce are trees. This approximation is not entirely correct in the case of circular β-sheets, such as β-barrels. To handle β-barrels, we are currently modifying these algorithms to allow up to one cycle in each component.

6.2.5 Strand pairing prediction benchmark

Pairs of sequentially adjacent strands account for about 50% of total strand pairs. A naive algorithm that pairs all adjacent strands can be used as a baseline for the evaluation of more sophisticated strand pairing algorithms. More generally, to

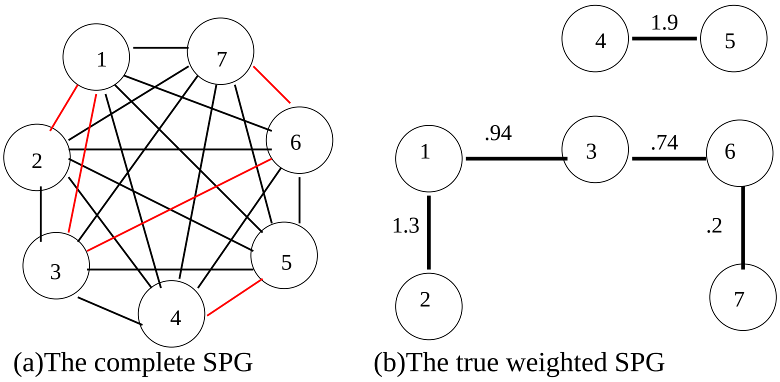


Figure 6.8: Strand pairing graph of protein 1VJG. **(a)** The complete SPG. Red edges denote true strand pairs. **(b)** The true SPG. Two components (1,2,3,6,7) and (4,5) correspond to two β-sheets. The weights are the pseudo-energy of the best alignments of strand pairs.

investigate the question of how linearly adjacent strands are connected in β-sheets we define the pairing distance between two strands as the length of the shortest path joining them in the true SPG. A distance of 1 means that the two strands are paired. A distance of 2 means that the two strands pair with a common third strand, and so forth. The absence of a path between two strands (denoted here by distance = -1) indicates that the two strands belong to two different β-sheets. Figure 6.9 provides the histogram of pairing distances between sequentially adjacent strands, obtained using Dijkstra's shortest path algorithm (Dijkstra, 1959). The probabilities of adjacent strands being paired (distance = 1) and of adjacent strands being in different β-sheets (distance = -1) are comparable and close to 41%.

6.3 Results and Discussion

The performance of β-residue pairing prediction is assessed using a variety of standard measures including: area under ROC curve, True Positive Rate [TPR = TP/(TP+FN)] at 5% False Positive Rate [FPR = FP/(FP+TN)], specificity [TP/(TP+FP)], sensitivity [TP/(TP+FN)], and correlation coefficient [(TP×TN-FP×FN) / ((TP+FN)

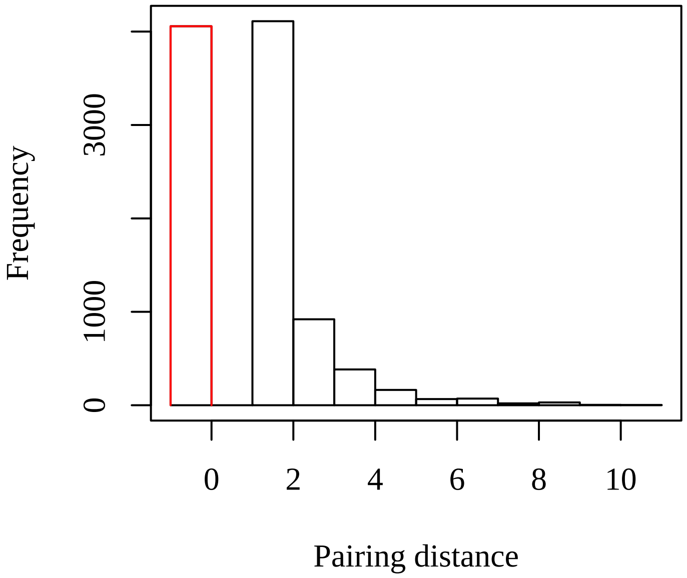


Figure 6.9: Histogram of pairing distances between linearly adjacent strands. 41% of them (red bar) are not connected (distance = -1); 42% of them are paired (distance = 1); 17% of them have pairing distance greater than 1.

(TP+FP) (TN+FN) (TN+FP)) $^{1/2}$], and compared with predictions associated with the base-line and with a a general-purpose contact map predictor. At the break-even point where the total number of predicted β-residue pairs is equal to the true number of β-residue pairs, the specificity and sensitivity of inter-strand β-residue pairings are equal to 41% with a correlation coefficient of 0.4. The accuracy of the base-line predictor (the number of true β-residue pairs / total number of inter-strand β-residue pairs) is 2.3%. Thus the improvement factor, i.e. the ratio between the accuracy (specificity or sensitivity) of our method over the base-line (Fariselli et al., 2001), is 17.8. To the best of our knowledge, only one method in the literature (Baldi et al., 2000) reports quantitive evaluation of β-residue pairing prediction. However, it only reports specificity without mentioning the corresponding sensitivity, thus a direct comparison cannot be made. However, we can compare the β-residue pairing predictor with a general-purpose contact map predictor (Pollastri and Baldi, 2002) focusing exclusively on β-residue pairings. We use a pre-trained 8Å contact map predictor (CMAPpro) to predict contacts for all chains in the same dataset. To make the comparison even more stringent, we do not take into consideration any homology between the current dataset and the dataset used to train CMAPpro. We then extract the contact probabilities for β-residue pairings from the full predicted contact map and evaluate them using the same measures. At the break-even point, the specificity and sensitivity of CMAPpro are equal to 27% and the correlation coefficient is 0.26. Thus our method improves the specificity and sensitivity of CMAPpro restricted to β-residues by 14%. The area under the ROC curve for the beta-pairing predictor is 0.86 versus 0.80 for CMAPpro (Figure 6.10). At 5% FPR, TPR for the beta-pairing predictor is 58% versus 42% for CMAPpro. Thus the specialized β-residue pairing predictor significantly improves the predictions of our general-purpose contact map predictor restricted to β-strands, consistently with previous expectations (e.g. Rost et al. (2003)).

The correlation coefficients of strand pairing by the greedy and constrained-MST graph algorithms are virtually identical (0.502 and 0.503 respectively). The specificity and sensitivity of strand pairing using the greedy graph algorithm are 59% and 54% respectively. In contrast, the specificity and sensitivity of the naive algorithm that always pairs sequentially adjacent strands are 42% and 50% respectively. Thus, around similar operating regimes, the greedy graph algorithm yields improvements of 17% in specificity and 4% in sensitivity over the naive algorithm. The smaller improvement in sensitivity is still very significant because 16% of correctly predicted strand pairs are non-adjacent strand pairs. The constrained-MST graph algorithm has specificity and sensitivity of 53% and 59% respectively. Its sensitivity is 9%

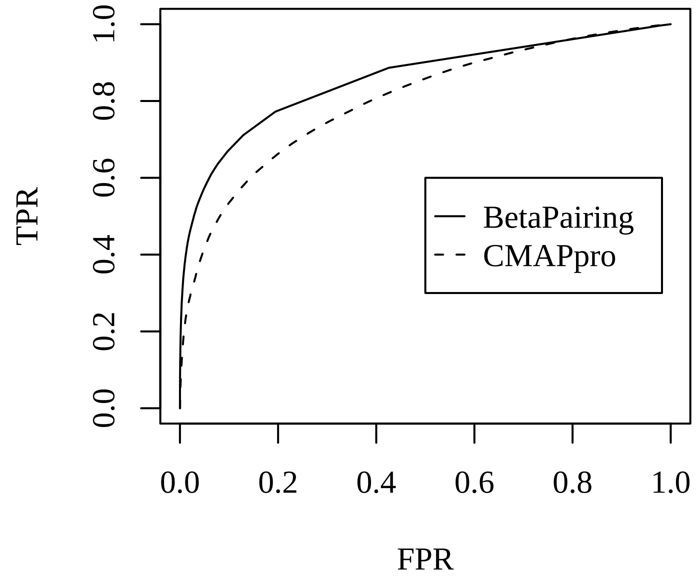


Figure 6.10: ROC curve of prediction of inter-strand β-residue pairs using the β-residue pairing predictor and CMAPpro.

higher than the naive algorithm and 20% of correctly predicted strand pairs are non-adjacent strand pairs.

Using the pseudo-energy to align strand predicted to be paired by the greedy graph algorithm, pairing directions (parallel, antiparallel, or isolated bridge) of 93% of the correctly predicted strand pairs are correctly identified, 72 % of which are correctly aligned (71% of parallel pairs, 69% of antiparallel pairs, 88% of strand pairs involving isolated bridges). The constrained-MST graph algorithm yields similar results.

To further evaluate the ability of the pseudo-energy to discriminate true alignments from false alignments, we use it to align all native strand pairs. Pairing directions of 84% native pairs are correctly predicted. Considering only parallel and antiparallel pairs, the pairing directions of 82% of these pairs are predicted correctly, which yields a 15% improvement over the 67% precision achieved by the trivial algorithm which labels all pairs as being antiparallel. Among all strand pairs with correctly predicted directions, 66% of them are aligned correctly (66% of parallel pairs, 63% of antiparallel pairs, 72% of isolated bridges). In comparison, on different datasets, the statistical potential approach in Hubbard (1994) aligns 35-45% of strand pairs correctly, when pairing directions are correctly predicted. If we assume all pairing directions are known, as some previous methods do (Zhu and Braun, 1999; Steward and Thornton, 2002), then 61% of all native parallel pairs and 60% of all native antiparallel pairs are aligned correctly. The pseudo-energy approach based on pairwise potentials in Zhu and Braun (1999) discriminates 35% of native alignments from alternative alignments, assuming pairing directions are known. Thus on a larger albeit different dataset, the accuracy of the method presented here is significantly higher than previous approaches. Assuming that pairing direction and position of one strand is known, the information theoretic approach of Steward and Thornton (2002), which aligns the known strand with all sub-sequences in a ±10 offset around another strand to identify the best alignment, achieves precisions of 48% and 45% for parallel and antiparallel pairs in strand triplets, and 37% and 31% for arbitrary parallel and antiparallel pairs respectively. Since our methods assume that the position of the two β-strands under consideration is known–in a purely *ab initio* setting, this would have to be predicted (Rost and Sander, 1993b,a; Jones, 1999b; Pollastri et al., 2002b)– the alignment accuracy of our methods can not be compared directly with the information theoretic approach. However, our results show that it is easier to align parallel strand pairs than antiparallel ones, which agrees with the observations derived using the information theoretic approach. Figure 6.11 and 6.12 show the histograms of alignment offsets of all parallel and antiparallel pairs where paring directions are correctly predicted. No simple metric is yet avail-

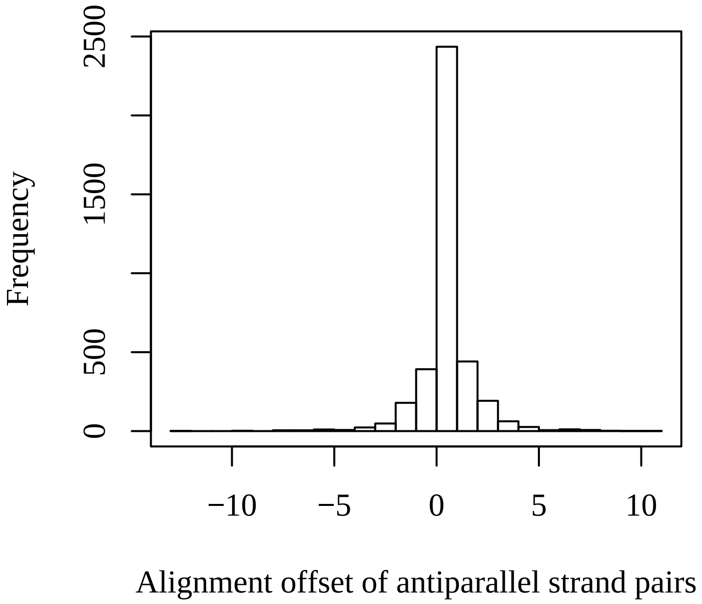


Figure 6.11: Histogram of alignment offsets of antiparallel strand pairs. A perfect alignment corresponds to a 0 offset.

able for evaluating the prediction of β-sheet topologies. Here we report the strand pairing precision of predicted β-sheets, i.e. the proportion of correctly predicted strand pairs in each β-sheet. Using the greedy graph algorithm, for instance, 51% of predicted β-sheets have β-strand pairing precision greater than 60%.

6.4 Conclusion

We have proposed a new *ab initio* modular approach to the problem of predicting and assembling β-sheets. The method is modular in the sense that alternative algorithms can be "plugged in" for each one of its stages, for instance in order to predict residue pairing probabilities. Starting from β-residue pairing probabilities, the method provides an integrated prediction of β-sheet architectures by predict-

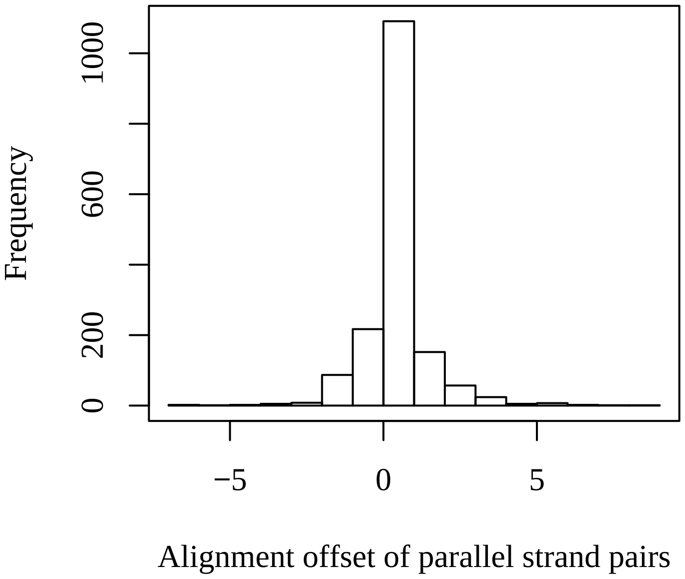


Figure 6.12: Histogram of alignment offsets of parallel strand pairs. A perfect alignment corresponds to a 0 offset.

ing β-strand pairs, β-strand alignments, and β-sheets assembly. The pseudo-energy derived from pairing probabilities of β-residue pairs can rather accurately predict β-strand alignments and score β-strand pairs. The greedy and constrained-MST graph algorithms are able to predict strand pair and β-sheet topology from pseudo-energy matrices by globally optimizing the pseudo-energy of β-sheets. While the performance of, for instance, β-strand alignment appears significantly improved over previous statistical data-driven approaches, it is clear that even further improvements should be possible in each one of the three stages. For instance, in the first step, more information about the inter-strand sequence can be included (Punta and Rost, 2005). In the second step, gap penalties for β-bulges can be taken into account. In the third step, graph algorithms that allow cycles ought to recover cyclic β-sheets. Furthermore, constrained optimization of the binding pseudo-energy derived here is at best an approximation that will need to be refined to include other packing constraints associated with other secondary structure elements.

β-sheets have remained one of the main stumbling blocks of protein structure prediction over the years. Thus new methods for the accurate prediction of β-sheets may lead to noticeable improvements in the study of protein structure and folding, and in protein design. Our results suggests that the methods presented here can be combined with contact map prediction to generate more accurate contact maps, which in turn can be used in fold recognition and 3D reconstruction. Accurate β-residue and β-strand pairings may also provide strong constraints for improving *ab initio* sampling of tertiary structures and derive energy terms to help select near native structures from decoys.

Chapter 7

A Machine Learning Information Retrieval Approach to Protein Fold Recognition

7.1 Introduction

The key step of template-based protein structure prediction approaches (comparative modeling and fold recognition) is to recognize proteins that have similar tertiary structures. This task becomes very challenging when there is little sequence similarity between the query and the template protein. Several alignment methods have been used to try to identify fold similarity, using sequence information, structural information, or both. Instead of developing a new specialized alignment method for fold recognition (Shi et al., 2001; Xu et al., 2003b; Zhou and Zhou, 2004), or integrating existing fold recognition servers (Lundstrm et al., 2001; Fischer, 2003; Ginalski et al., 2003a), here we propose a machine learning information retrieval approach that leverages features extracted using existing, general-purpose, alignment tools as well as protein structure prediction program and combines them using support vector machines to rank all the templates.

7.1.1 Classical Approaches to Fold Recognition

Alignment methods for fold recognition include sequence-sequence, sequence-profile (or profile-sequence), profile-profile, and sequence-structure methods.

Sequence-sequence alignment methods (Needleman and Wunsch, 1970; Smith and Waterman, 1981; Dayhoff et al., 1983; Pearson and Lipman, 1988; Altschul et al., 1990; Henikoff and Henikoff, 1992; Vingron and Waterman, 1994) are effective at detecting homologs with significant sequence identity (>40%).

Sequence-profile (or profile-sequence) alignment methods (Baldi et al., 1994; Krogh et al., 1994; Hughey and Krogh, 1996; Altschul et al., 1997; Bailey and Gribskov, 1997; Karplus et al., 1998; Eddy, 1998; Park et al., 1998; Koretke et al., 2001; Gough et al., 2001) are more sensitive at detecting distant homologs with lower sequence identity (> 20%). Profiles can correspond to simple multiple alignments, to position specific scoring matrices (PSSMs), or to hidden Markov models (HMMs).

Profile-profile alignment approaches (Thompson et al., 1994; Rychlewski et al., 2000; Notredame et al., 2000; Yona and Levitt, 2002; Madera and Gough, 2002; Mitelman et al., 2003; Ginalski et al., 2003b; Sadreyev and Grishin, 2003; Edgar and Sjolander, 2003, 2004; Ohlson et al., 2004; Wallner et al., 2004; Wang and Dunbrack, 2004; Marti-Renom et al., 2004; Söding, 2005) are even more sensitive at detecting distant homologs and compatible structures, and often achieve even better performance than sequence-structure alignment methods that leverage template structural information (Rychlewski et al., 2000).

Sequence-structure alignment methods (or threading) (Bowie et al., 1991; Jones et al., 1992; Godzik et al., 1992; Bryant and Lawrence, 1993; Abagyan et al., 1994;

Murzin and Bateman, 1997; Xu et al., 1998; Jones, 1999a; Panchenko et al., 2000; David et al., 2000; Shi et al., 2001; Skolnick and Kihara, 2001; Xu et al., 2003b; Kim et al., 2003) align query sequences with template structures and compute compatibility scores according to structural environment fitness and contact potentials. These methods are particularly useful for detecting proteins with similar folds but no recognizable evolutionary relationship.

The separation between sequence-based and structure-based methods, however, is becoming blurred as new methods are developed that combine both kinds of information together. Combining both sequence and structure information has been shown to improve both fold recognition (Elofsson et al., 1996; Jaroszewski et al., 1998; Al-Lazikani et al., 1998; Fischer, 2000; Kelley et al., 2000; Panchenko et al., 2000; Shan et al., 2001; Tang et al., 2003; Pettitt et al., 2005) and alignment quality (Thompson et al., 1994; Al-Lazikani et al., 1998; Domingues et al., 2000; Notredame et al., 2000; Griffiths-Jones and Bateman, 2002; Tang et al., 2003; O'Sullivan et al., 2004). Even the sequence-derived predicted secondary structure can be used to increase the sensitivity of fold recognition (Rost and O'Donoghue, 1997; Jones, 1999a; Ginalski et al., 2003b; Xu et al., 2003b; Zhou and Zhou, 2004).

In fold recognition, different alignment tools are often used independently to search protein databases for similar structures. Previous research (Jaroszewski et al., 1998; Lindahl and Elofsson, 2000; Shan et al., 2001; Ohsen et al., 2003; Wallner et al., 2004) has shown that these alignment methods are complementary and can find different correct templates. But combining these methods is difficult (Lindahl and Elofsson, 2000). Meta or jury approaches (Lundstrm et al., 2001; Fischer, 2003; Ginalski et al., 2003a; Juan et al., 2003; Wallner et al., 2004) collect the predicted models from external fold recognition predictors and derive predictions based on a small set of returned candidates. This popular, hierarchical approach increases the reach of fold recognition. However, it relies on the availability of external predictors and cannot recover true positive templates discarded prematurely by individual predictors.

7.1.2 A Machine Learning Information Retrieval Approach to Fold Recognition

Statistical machine learning methods provide powerful means for integrating disparate features in pattern recognition. So far, however, machine learning integration of features has been used in this area primarily for coarse homology detection, such as protein structure/fold *classification* (Jaakkola et al., 2000; Leslie et al., 2002). Classifying proteins into a few categories or even dozens of families, superfamilies, and folds, however, does not provide the specific templates required for template-based structure modeling. Furthermore, current classification methods are not likely to scale up to the thousands of families, superfamilies, and folds already present in current protein classification databases, such as SCOP (Murzin et al., 1995). Fold recognition is different from protein classification–it is fundamentally a *retrieval* problem, very much like finding a document or a Web page (Rocchio, 1966; Page et al., 1998). Given a query protein, the objective of fold recognition is rather to rank all possible templates according to their structural relevance, like Google and other search engines rank Web pages associated with a user's query.

Machine learning methods (such as binary classifiers) have been used also in threading approaches (Jones, 1999a; Xu et al., 2003b) to combine multiple scores produced by threading into a single scores to rank the templates. Here we generalize this idea and derive a broad machine learning framework for the fold recognition/retrieval problem. The framework integrates a variety of similarity features and feature extraction tools, including standard alignment tools. However, unlike meta approaches, it does not require any preexisting fold recognition programs or servers.

Consistently with the major trend in machine learning towards kernel methods (Schölkopf and Smola, 2002), we first focus on the computation of a variety of similarity measures between query-template pairs. Instead of extracting features and analyzing individual sequences, we focus exclusively on pairs of sequences and use a variety of complementary alignment tools to align the query protein with the template proteins, rather than to search the database of templates. The alignment scores for query-template pairs are used as similarity measures. Furthermore, based on alignments (e.g. profile-profile) between query and template, we further extract pairwise structural compatibility features by checking the predicted secondary structure, solvent accessibility, contact map, and beta-strand pairings of the query protein against the tertiary structure of the template protein. Second, these alignment and structural similarity scores as well as other sequence and structural features derived using 3 standard similarity measures (cosine, correlation, and Gaussian kernel) are fed into support vector machines (SVMs) (Vapnik, 1998) to learn a relevance function to evaluate whether the query and template belong to the same fold. Finally, the continuous output scores produced by the SVMs are used to rank the templates with respect to the query. The top-ranked templates can be used to model the structure of the query.

7.2 Methods

7.2.1 Feature Extraction

We extract five categories of pairwise features (similarity scores) for each query-template pair associated with sequence or family information, sequence alignment, sequence-profile (or profile-sequence) alignment, profile-profile alignment, and structure (Table 7.1).

Sequence/family information features. To compare the sequences of query and template proteins, we compute their single amino acid (monomer) and ordered pair of amino acids (dimer) compositions. The composition vectors x and y of the query and template are compared and transformed into six similarity scores using the cosine ($\frac{x \cdot y}{|x||y|}$), correlation ($\frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$), and Gaussian kernel ($e^{-||x-y||^2}$) respectively. We apply the same techniques to the monomer and dimer residue composition vectors of the family of sequences associated with the query and the template to extract another set of six similarity measures, to measure the family composition similarity. The sequences for both query and template families are derived from multiple sequence alignments generated by searching the NCBI non-redundant sequence database (NR release 1.21, 28-Apr-2003) using PSI-BLAST(Altschul et al., 1997). The e-value (-e option) threshold for inclusion in the profile is set to 0.001; the cut-off threshold (-h option) for building iterative profiles is set to $e - 10$; and the number of iteration (-j option) is set to 3. Thus the sequence/family information feature subset includes 12 (6+6) pairwise features in total.

Sequence-sequence alignment features. Two sequence alignment tools, PALIGN (Ohlson et al., 2004) and CLUSTALW (Thompson et al., 1994), are used to extract pairwise features associated with sequence alignment scores. PALIGN uses local alignment methods and produces a score and an e-value. The score is divided by the length of the query to remove any length bias. CLUSTALW generates a global sequence alignment score between the query and the template. This score is also normalize by the length of the query sequence. Thus the sequence alignment feature subset includes 3 pairwise features.

Sequence-profile (or profile-sequence) alignment features. We use three different profile-sequence alignment tools (PSI-BLAST, HMMER-hhmsearch (Eddy, 1998), and IMPALA (Schaffer et al., 1999)) to extract profile-sequence alignment features between the query profile and the template sequence. The profiles (or multiple alignments) for queries are generated by searching the NR database using PSI-BLAST, as described above. Identical sequences in the multiple alignments are removed. No sophisticated weighting scheme is used. The multiple alignments are used by all profile alignment tools directly, or as the basis for building customized profiles. For instance, the HMM models in HMMER are built from the multiple alignments by using the hmmbuild and hmmcalibrate tools of HMMER. Note that, instead of using these tools to search sequence databases, we use them to align individual query profiles against individual template sequences to extract pairwise features. The alignment score normalized by the query length, the logarithm of the e-value, and the alignment length normalized by the query length from the most significant PSI-BLAST and IMPALA local alignment are used as features. The alignment scores, normalized by the length of the query sequence, and the logarithm of the e-value produced by hmmsearch alignments are used as features too. Thus the profile-sequence alignment tools generate 8 pairwise features.

For sequence-profile alignments, we use RPS-BLAST in the PSI-BLAST package and hmmpfam in the HMMER package to align the query sequence with the template profiles. The template profiles are generated in the same way as the query profiles. In this way, RPS-BLAST generates 3 features similar to PSI-BLAST. The logarithm of the e-value produced by hmmfam is also used as one feature. Thus the subset of profile-sequence (or sequence-profile) alignment features includes 12 (8 + 4) pairwise features in total.

Profile-profile alignment features. We use five profile-profile alignment tools including CLUSTALW, COACH of LOBSTER (Edgar and Sjolander, 2004), COMPASS (Sadreyev and Grishin, 2003), HHSearch (Söding, 2005), and PRC (Profile Compiler by M. Madera, http://supfam.org/PRC) to align query and template profiles. The global alignments produced by CLUSTALW and LOBSTER and the most significant local alignments produced by COMPASS, PRC, and HHSearch are used to extract the pairwise features. Specifically, CLUSTALW aligns query multiple alignments with template multiple alignments. COACH aligns query HMMs with template HMMs built from the multiple alignments produced by LOBSTER. HHSearch also aligns query HMMs with template HMMs generated from the multiple alignments using the hhmake function of HHSearch. The alignment scores produced by CLUSTALW and HHSearch are normalized by query length and used as pairwise features. The alignment scores produced by LOBSTER are not used directly as features because their dependence on template length would introduce a bias toward long templates.

PRC, an HMM profile-profile alignment tool, is used with two different kinds of profiles: HMM models built by HMMER and chk profiles built by PSI-BLAST. In each case, PRC produces 3 scores (co-emission, simple, and reverse), which are normalized by query length. COMPASS, which uses internally a log-odds ratio score and a sophisticated sequence weighting scheme, is used to align query multiple alignments with template multiple alignments. The Smith-Waterman local alignment score normalized by query length and the logarithm of the e-value from the COMPASS alignments are used also as pairwise features. Thus the subset of profile-profile alignment features includes 10 pairwise features in total.

Structural features. Based on the global profile-profile alignment between query and template obtained with LOBSTER, we use predicted 1D and 2D structural features including secondary structure (3-class: alpha, beta, loop), relative solvent accessibility (2-class: exposed or buried at 25% threshold), contact probability map at 8Å and 12Å, and beta-sheet residue pairing probability map to evaluate the compatibility between query and template structures. These structural features for query proteins are predicted using the SCRATCH suite (Pollastri et al. (2002a,b); Pollastri and Baldi (2002); Cheng et al. (2005a); Cheng and Baldi (2005), http://www.igb.uci.edu/servers/psss.html).

The predicted secondary structure (SS) and relative solvent accessibility (RSA) of the query residues are compared with the nearly exact SS and RSA of the aligned residues in the template structure. The fractions of correct matches for both SS (as in (Jones, 1999a; Xu et al., 2003b)) and RSA are used as two pairwise features. The SS and RSA composition (helix, strand, coil, exposed, and buried) are transformed into four similarity scores by cosine, correlation, Gaussian kernel, and dot product. So this 1D structural feature subset has 6 features in total.

For the aligned residues of the template which have sequence separation > 5 and are in contact at 8Å threshold (resp. 12Å), we compute the average contact probability of their counterparts in the predicted 8Å (resp. 12Å) contact probability map of the query. The underlying assumption is that the counterparts of the contact residues in the template should have high contact probability in the query contact map if the query and template share similar structure. Similarly, for each paired beta-strand residues in the template structures, we compute the average pairing probability of their beta-strand counterparts in the predicted beta-strand paring probability map of the query, assuming that two proteins will share similar beta-sheet topology if they belong to the same fold.

Moreover, we compute the contact order (sum of sequence separation of contacts) and contact number (number of contacts) for each aligned residue in both query and templates. This information is easy to derive for the template sequences since their tertiary structure is known. For the query sequence, we let the contact order for residue i to be $\sum_{|i-j|>5} C_{ij}|i-j|$, where C_{ij} is the predicted contact probability for residues i and j. The contact number for residue i in the query is defined as the sum of the contact probabilities $\sum_{|i-j|>5} C_{ij}$. The contact order and contact number vectors of the aligned residues are not used directly as features. Instead, they are compared and transformed into pairwise similarity scores using the cosine and correlation functions. For both the 8Å and 12Å contact maps, 8 pairwise features of contact order and contact number are extracted. So the 2D structural feature subset has 11 features in total. Thus the entire 1D and 2D structural feature subset has 17 features in total.

The entire feature set contains 54 pairwise features measuring query-template similarity (Table 7.1). Initially we used a larger set of 74 features (not shown) that included also non-pairwise features, such as the proportion of helices and strands in each chain. Information gain analysis (see Results) and experiments led us to remove the 20 least informative or most biased features to optimize performance. All the alignment tools for extracting pairwise features are run with default parameters, except the e-value thresholds (-e option) of PSI-BLAST, RPS-BLAST, and IMPALA which are set to larger values (100,50,20 respectively), to ensure that alignments between sequences with very little similarity are generated in most cases. If no features are generated by these tools, the corresponding similarity features are set to 0.

Table 7.1: Features used in fold recognition. cos/corr/Gauss denote cosine, correlation, and Gaussian kernel functions. SS and RSA represent secondary structure and relative solvent accessibility respectively.

Category	Feature	Method	Num
Seq&Family Info.	Seq monomer compo	cos/corr/Gauss	3
	Seq dimer compo	cos/corr/Gauss	3
	Fam monomer compo	cos/corr/Gauss	3
	Fam dimer compo	cos/corr/Gauss	3
Seq-Seq Align.	Local alignment	PALIGN	2
	Global alignment	CLUSTALW	1
Seq-Prof Align.	Prof vs. seq	PSI-BLAST	3
	Prof vs. seq	IMPALA	3
	Prof vs. seq	HHMER	2
	Seq vs. prof	RPS-BLAST	3
	Seq vs. prof	HMMER	1
Prof-Prof Align.	Multiple alignment	CLUSTALW	1
	PSSM	COMPASS	2
	HMM prof	PRC	6
	HMM prof	HHSearch	1
Structural Info.	SS&RSA match	ratio	2
	SS&RSA compo	cos/corr/Gauss	4
	Contact probability	average	2
	Residue contact order	cos/corr	4
	Residue contact num	cos/corr	4
	Beta-sheet pair prob.	average	1
Total	-	-	54

7.2.2 Fold Recognition with Support Vector Machines (SVMs)

Each feature vector associated with a pair of proteins in a given training set correspond to a positive or negative example, depending on whether the two proteins are in the same fold or not. These feature vectors in turn can be used to train a binary classifier. Here we train SVMs and learn an optimal decision function $f(x)$ to classify an input feature vector x into two categories ($f(x) > 0$: same fold; $f(x) < 0$: different fold). The decision function $f(x) = \sum_{x_i \in S} \alpha_i y_i K(x, x_i) + b$ is a weighted linear combination of the similarities $K(x_i, x)$ between the input feature vector x and the feature vectors x_i in the training dataset S. Here K is a user-defined kernel function that measures the similarity between the feature vectors x_i and x corresponding in general to *four* proteins. α_i is the weight assigned to the training feature vector x_i and y_i is the corresponding label (+1:positive, -1:negative). All protein pairs in the same fold are labeled as positive examples, and the remaining ones as negative examples. We use SVM-light (Joachims, 1999) to learn the SVM parameters. The continuous value $f(x)$ is indicative of how likely the corresponding sequences are in the same fold, and therefore it is used to evaluate the structural relevance and rank all the templates for a given query. We tested polynomial, tanh, and Gaussian radial basis kernels (RBF: $e^{-\gamma||x-y||^2}$). We report the results obtained with the RBF kernel which worked best for this task, with $\gamma = 0.015$. Preliminary tests indicated that the results are robust with respect to γ. All other SVM parameters are set to their default values. A thorough parameter optimization may help further improve the accuracy.

7.2.3 Training and Benchmarking

To compare the performance of our method with other well-established methods, we use the large benchmark dataset (Lindahl and Elofsson, 2000) derived from the SCOP (Murzin et al., 1995) database. The Lindahl's dataset includes 976 proteins. The pairwise sequence identity is $<=$ 40%. We extract a feature vector for all 976×975 distinct pairs. In this dataset, 555 sequences have at least one match at the family level, 434 sequences have at least one match at the superfamily level, and 321 sequences have at least one match at the fold level.

We split all protein pairs evenly into 10 subsets for 10-fold cross validation purposes. All the query-template pairs associated with the same query protein are put into the same subset. Nine subsets are used for training and the remaining subset is used for validation. The pairs in the training dataset that use queries in the test dataset as templates are removed. The procedure is repeated 10 times and the sensitivity/specificity results are computed across the 10 experiments. Training takes about 3 days for a single data-split on a single node with dual Pentium processors, hence 3 days for the entire 10-fold cross-validation experiment using 10 nodes in a cluster. Using the same evaluation procedure as in Lindahl and Elofsson (2000); Shi et al. (2001); Zhou and Zhou (2004), we evaluate the sensitivity by taking the top 1 or the top 5 templates in the ranking associated with each test query. Furthermore, as in Lindahl and Elofsson (2000); Shi et al. (2001), we also evaluate the performance of our method for all positive matches using specificity-sensitivity plots.

7.3 Results

Table 7.2 lists the 20 top features ranked using the information gain measure (Yang and Pedersen, 1997). The table shows that profile-profile alignment features are the most informative. For instance, the alignment features of HHSearch, COMPASS, and PRC are ranked first, second, and third respectively. Thus profile-profile alignment methods have the strongest discriminative power in fold recognition, consistently with previous studies (Rychlewski et al., 2000; Wallner et al., 2004; Ohlson et al., 2004). Profile-sequence (or sequence-profile) alignment features and some structural features based on the LOBSTER alignment between queries and templates have also strong discriminative power according to the information gain measure. For instance, the e-values of HMMer pfam and HMMer search are ranked fifth and seventh respectively. Our results, confirm also the importance of predicted structural features. The dot product of secondary structure and solvent accessibility composition vectors, and the secondary structure match ratio, rank sixth and eighth respectively.

Other profile-sequence (or sequence-profile) alignment features such as PSI-BLAST, IMPALA, BLAST and structural features such as the cosine of the residue contact number lead also to significant information gains. On the other hand, compared with other local profile-profile alignment scores, the CLUSTALW global profile-profile alignment score carries a lesser weight. This suggest that CLUSTALW is optimized for alignment, but not for direct fold recognition, which is consistent with previous results (Marti-Renom et al., 2004). Since the pairwise sequence identity in the dataset is less than 40%, sequence alignment and sequence/family information features have a lesser, albeit still noticeable, impact.

We evaluate the performance of our FOLDpro method against 11 other fold recognition methods. The 11 other methods are: PSI-BLAST, HMMER, SAM-T98 (Karplus et al., 1998), BLASTLINK, SSEARCH, SSHMM (Hargbo and Elofsson, 1999), THREADER (Jones et al., 1992), FUGUE (Shi et al., 2001), RAPTOR (Xu et al., 2003b), SPARKS (Zhou and Zhou, 2004), and SP^3 (Zhou and Zhou, 2005).

Table 7.2: Twenty top-ranked features using information gain

Feature	Information Gain
HHSearch score	0.0375
COMPASS evalue	0.0370
PRC reverse score on chk profile	0.0354
PRC reverse score on HMM profile	0.0341
HMMer pfam evalue	0.0287
Dot product of SS and RSA vectors	0.0266
HMMer search evalue	0.0264
SS match ratio	0.0263
Correlation of SS and RSA vectors	0.0263
PRC simple score on HMM profile	0.0248
Cosine of SS and RSA vectors	0.0246
Gaussian kernel on SS and RSA vectors	0.0237
COMPASS score	0.0235
PRC coemis score on HMM profile	0.0220
PSI-BLAST evalue	0.0205
IMPALA evalue	0.0181
RPS-BLAST evalue	0.0180
SA match ratio	0.0154
Cosine of residue contact num (8Å)	0.0150
HMMer search score	0.0142

SPARKS, for instance, was one of the top predictors during the sixth edition of the CASP evaluation (Moult et al., 2005). The results for PSI-BLAST, HMMER, SAM-T98, BLASTLINK, SSEARCH, SSHMM, and THREADER are taken from Lindahl and Elofsson (2000). The results for the other methods are taken from the corresponding articles. One caveat is that the sequence databases used to generate the profiles are being updated continuously, and so are some of the methods. Thus the comparative analysis is only meant to provide a broad, rough assessment of performance rather than a precise and stable ranking.

Table 7.3 shows the sensitivity of FOLDpro and the other methods at the family, superfamily, and fold levels, for the top 1 and top 5 predictions respectively. Here sensitivity is defined by the percentage of query proteins (with at least one possible hit) having at least one correct template ranked 1st, or within the top 5 (Lindahl and Elofsson, 2000). It shows that in almost all situations the performance of FOLDpro is better than that of other well-established methods such as SPARKS, SP^3, FUGUE, and RAPTOR.

Specifically, at the family level, the sensitivity of FOLDpro for the top 1 or 5 predictions is 85.0% and 89.9%, about 2-4% higher than FUGUE, SPARKS, and SP^3, and significantly higher than all other methods. At the superfamily level, the sensitivity of FOLDpro for the top 1 or 5 predictions is 55.5% and 70.0%, slightly higher than SPARKS and SP^3, and significantly higher than all other methods. At the fold level, the sensitivity of FOLDpro for the top 1 predictions is 26.5%, about 2% lower than SP^3, 1-3% higher than RAPTOR and SPARKS, and significantly higher than all other methods. For the top 5 predictions, at the fold level, the sensitivity of FOLDpro is 48.3%, about 0.6-3% higher than RAPTOR, SPARKS, and SP^3, and significantly higher than all other methods

The performance of FOLDpro is significantly better than pure sequence- or

Table 7.3: The sensitivity of 12 methods on the Lindahl's benchmark dataset at the family, superfamily, and fold levels.* denotes the best results.

	Family (%)		Superfamily (%)		Fold (%)	
Method	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
PSI-BLAST	71.2	72.3	27.4	27.9	4.0	4.7
HMMER	67.7	73.5	20.7	31.3	4.4	14.6
SAM-T98	70.1	75.4	28.3	38.9	3.4	18.7
BLASTLINK	74.6	78.9	29.3	40.6	6.9	16.5
SSEARCH	68.6	75.5	20.7	32.5	5.6	15.6
SSHMM	63.1	71.7	18.4	31.6	6.9	24.0
THREADER	49.2	58.9	10.8	24.7	14.6	37.7
FUGUE	82.2	85.8	41.9	53.2	12.5	26.8
RAPTOR	75.2	77.8	39.3	50.0	25.4	45.1
SPARKS	81.6	88.1	52.5	69.1	24.3	47.7
SP^3	81.6	86.8	55.3	67.7	28.7*	47.4
FOLDpro	85.0*	89.9*	55.5*	70.0*	26.5	48.3*

profile-based approaches, such as PSI-BLAST, HMMER, SAM-T98, and BLASTLINK. It is also significantly better than threading approaches, such as THREADER, in all 3 categories. For example, compared to PSI-BLAST, FOLDpro is about 14%, 28%, and 23% more sensitive at recognizing members of the same family, superfamily, and fold, respectively, using the top 1 predictions; using the top 5 predictions, these improvements are 18%, 42%, and 44% respectively.

As in Lindahl and Elofsson (2000), we also compare the performance of FOLDpro using specificity-sensitivity plots (Figures 7.1, 7.2, 7.3), to better assess the trade-offs between specificity and sensitivity, using the Lindahl's dataset. We compute the sensitivity and specificity of FOLDpro for different thresholds applied to the SVM scores. Specificity is defined as the percentage of predicted positives (above threshold) that are true positives (in the same family, superfamily, or fold). Sensitivity is defined as the percentage of true positives that are predicted as positives (above threshold). The advantage of the specificity-sensitivity plots is that they measure the ability of a method to reliably identify all positive matches in the dataset beyond the top hits. Sensitivity-specificity results for 7 of the 11 methods above were kindly provided by Dr. Elofsson (http://www.sbc.su.se/∼arne/protein-id/).

In the family category (Figure 7.1), FOLDpro consistently outperforms all other methods by more than 10% for almost all specificity values. However, like SAM-T98, the sensitivity of FOLDpro drops rapidly when the specificity is close to 1. This suggests that some false positives may be receiving very high scores. However, after manually inspecting the dozen of "false positives" with high scores, some of them turn out to be true positives that were misclassified in the original dataset. For instance, the pair (1XZL,3TGL) belongs to the same superfamily and fold (alpha/beta-Hydrolases) in the latest SCOP 1.69 release, while 1XZL was wrongly classified into another fold (Flavodoxin-like) in the Lindahl's dataset based on the old SCOP 1.37 release. This shows that FOLDpro is capable of correcting some human annotation errors and that "false positives" with high scores must be verified carefully. Although these wrongly classified pairs with high scores lead one to slightly under-estimate the performance of FOLDpro, we did not attempt to correct them in the evaluation, due to their small effect and to maintain consistency with previous evaluations.

At the superfamily level (Figure 7.2), FOLDpro has more than twice the sensitivity of the second best method for almost all specificity levels. For instance, at 50% specificity, the sensitivity of FOLDpro is 30%, about 20% higher than the second best method, PSI-BLAST. At the fold level (Figure 7.3), fold recognition remains challenging for all methods. However, FOLDpro's performance is significantly better than all other methods, including the second best method THREADER, a threading method specifically designed for this purpose. For instance, at 5% specificity, FOLDpro achieves sensitivity of 28%, about 23% higher than THREADER, while the sensitivity of all other methods is close to 0.

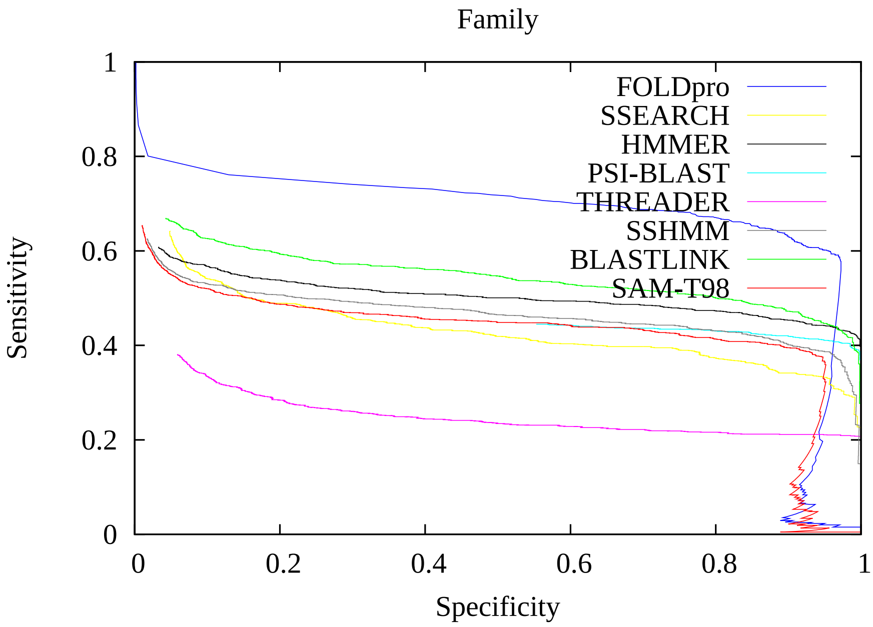


Figure 7.1: Specificity-sensitivity plot at the family level

The specificity-sensitivity plots show that FOLDpro significantly outperforms a variety of different methods in all categories, indicating that the integration of complementary alignment tools and sequence and structural information can improve fold recognition across the board.

7.4 Discussion

We have presented a general information retrieval framework for the fold recognition problem that leverages similarity methods at two fundamental levels. Rather than directly classifying individual proteins, we first consider pairs of proteins and derive a set of pairwise features (feature vector) consisting of many different similarity scores (e.g. profile-profile alignment scores). We then apply supervised classification methods (e.g., SVM) to these feature vectors to learn a relevance function to mea-

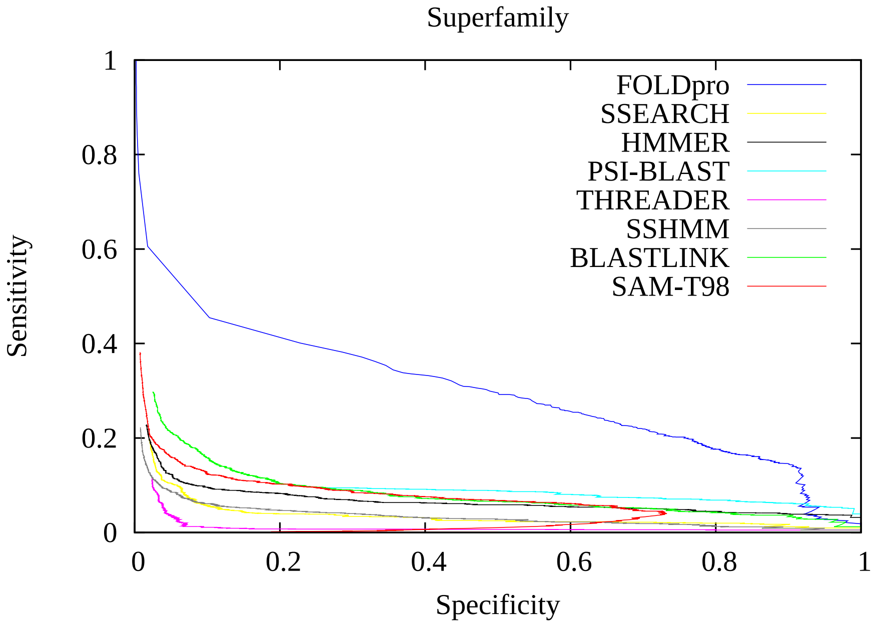


Figure 7.2: Specificity-sensitivity plot at the superfamily level

sure whether or not the query-template pairs are structurally relevant (same versus different fold). For a given query, the continuous relevance values are used to rank the templates.

The learning process involves measuring the similarity between pairs of feature vectors associated with *four proteins*, which differs from the two-protein comparison of traditional classification approaches. From the standpoint of using structural information in fold recognition, our approach differs also from traditional threading approaches, which use structural information to produce alignments and compute statistical contact potentials to evaluate sequence-structure fitness. In contrast, our approach employs sequence-based profile-profile alignment tools to align a query against the possible templates, without using structural information. Then, based on these alignments, it checks the predicted secondary structure, solvent accessibility, contact probability map, and beta-sheet pairings of the query against the template structures to evaluate fitness.

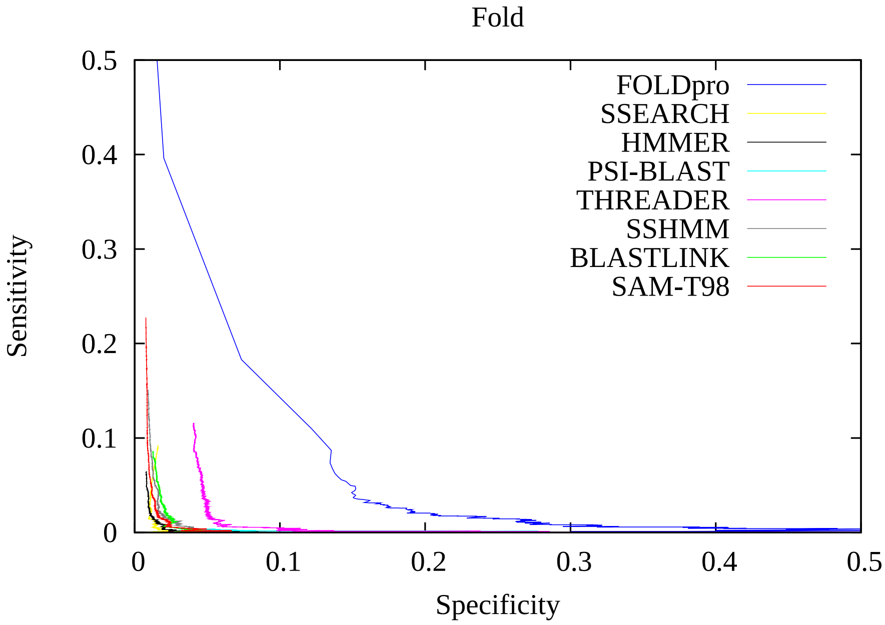


Figure 7.3: Specificity-sensitivity plot at the fold level

The approach used in FOLDpro has several advantages in terms of integration, scalability, simplicity, reliability, and performance. First the approach readily integrates complementary streams of information, from alignment to structure, and additional features can easily be added. It is worth pointing out this integrative approach is slower than some individual alignment methods such as PSI-BLAST. However, it can scan a fold library with about 10,000 templates in a few hours, for an average-size query protein, on a server with two Pentium processors. Second, most features can readily be derived using publicly available tools. This is simpler than trying to develop a new, specialized, alignment tool for fold recognition as in SPARKS, SP^3, FUGUE, and RAPTOR, which usually requires a lot of expertise. Third, our approach can be included in a meta-server but, unlike a meta-server, it is self-contained and does not rely on external fold-recognition servers. Unlike meta servers, this approach produces a full ranking of all the templates and does not discard any templates early on during the recognition process. Finally, the approach delivers state-of-the-art performance on current benchmarking datasets. And while fold recognition remains a challenging problem, the approach provides clear avenues of exploration to improve the performance, such as adding new features to the feature vector, enlarging the training set, using different machine learning tools to learn the relevance function, and leveraging ensembles.

Chapter 8

Protein Bioinformatics Software and Servers

8.1 Introduction

Publicly available bioinformatics web servers and software allow researchers from around the world to apply tools developed in other laboratories to their own data and fully automated systems provide a framework for high-throughput proteomics and protein engineering projects. We have developed a software suite (SCRATCH) for protein bioinformatics, to predict protein tertiary structure and structural features. A number of bioinformatics tools in the suite are available to the scientific community in the form of web servers or dowloadable packages (www.igb.uci.edu/servers/psss.html).

The SCRATCH software suite includes predictors for secondary structure, relative solvent accessibility, disordered regions, domains, disulfide bridges, single point mutation stability, residue contacts, and tertiary structure. The user simply provides an amino acid sequence and selects the desired predictions, then submits to the server. Results are emailed to the user. Most predictors can also be downloaded for local use from the SCRATCH web site.

8.2 Protein Bioinformatics Software and Servers

The SCRATCH suite combines machine learning methods, graph algorithms, evolutionary information in the form of profiles, fragment libraries extracted from the Protein Data Bank (PDB) (Berman et al., 2000), and energy functions to predict protein structural features and tertiary structures. The suite includes the following main modules:

- SSpro (Pollastri et al., 2002b; Cheng et al., 2005a): three class secondary structure prediction.

- ACCpro (Pollastri et al., 2002a; Cheng et al., 2005a): relative solvent accessibility prediction.

- DOMpro (Cheng et al., 2005d): domain boundary prediction.

- DISpro (Cheng et al., 2005c): disordered region prediction.

- MUpro (Cheng et al., 2006a): prediction of the stability change of a single amino acid mutation.

- CMAPpro (Pollastri and Baldi, 2002; Cheng et al., 2005a) : residue-residue contact map prediction.

- DIpro (Baldi et al., 2005; Cheng et al., 2006b): disulfide bridge prediction.

- BETApro (Cheng and Baldi, 2005): beta-sheet architecture prediction.

- FOLDpro (Cheng and Baldi, 2006): template-based tertiary structure prediction.

- 3Dpro (Cheng et al., 2005a; Cheng and Baldi, 2006): template-based and *ab initio* tertiary structure prediction.

SSpro, ACCpro, DOMpro, DISpro, and MUpro are 1D predictors to predict 1-dimensional structural features. CMAPpro, DIpro, and BETApro are 2D predictors to predict 2-dimensional structural features. 3Dpro and FOLDpro are 3D predictors to predict 3-dimensional (tertiary) structure.

All 1D and 2D predictors are trained in a supervised fashion using curated, non-redundant, datasets extracted from the PDB. SSpro, ACCpro, DISpro and DOMpro use ensembles of one-dimensional recursive neural network (1D-RNN) architectures (Baldi and Pollastri, 2003). MUpro uses feed-forward neural networks and SVMs to predict mutation stability changes from protein sequences or structures.

CMAPpro and DIpro predictors use ensembles of 2D-RNN architectures (Pollastri and Baldi, 2002; Baldi and Pollastri, 2003). In addition to the standard 2D-RNN architectures to predict the entire contact map in one step, a variant architecture of CMAPpro is used to predict contacts from low-sequence separation to high-sequence separation step by step. The predicted contact maps at lower sequence separation are used as inputs for the prediction of contact maps at higher sequence separation. The raw output of CMAPpro is a matrix of contact probabilities for all residue pairs. DIpro also uses support vector machines (SVMs) to discriminate proteins with disulfide bonds from proteins without disulfide bonds, and graph matching algorithms to pair the cysteines. BETApro uses 2D-RNN, alignment, and graph algorithms to predict protein β-residue pairings, β-strand pairings, and β-sheet architecture in three stages.

All 1D and 2D predictors except for MUpro, directly leverage evolutionary information in the form of input profiles derived using PSI-BLAST (Altschul et al., 1997) to include all homologous proteins (Pemberton and Jones, 1999; Przybylski and Rost, 2002). In addition, for SSpro and ACCpro, very high levels of local homology to known structures are used either directly or in combination with the prediction output to improve accuracy. Whenever possible and useful, predictors leverage the output of the other predictors. For instance, DOMpro uses the outputs of SSpro and ACCpro to improve domain boundary prediction.

We use both *ab-initio* and template-based methods (FOLDpro) for tertiary structure prediction. FOLDpro makes structure prediction in three steps. First, it uses a machine learning information retrieval approach to rank template proteins for the query protein, integrating a variety of similarity features including sequence/family information, sequence-sequence alignment, sequence-profile (profile-sequence) alignment, profile-profile alignment, and structural features. Second, it generates profile-profile alignments between the query protein and the top ranked template proteins. Multiple templates are used to improve both the alignment and the structure modeling if necessary. Third, based on the query-template alignments and 3D structures of the templates, Modeller (Sali and Blundell, 1993) is used to generate structure models for the query protein. Five models are generated for the query. The models ranked higher are generated based on the templates ranked higher. So they are presumably, but not always, better than the models ranked lower.

In addition to template-based component FOLDpro, 3Dpro includes an *ab-initio* component, which combines the use of predicted structural features (Pollastri et al., 2002b,a; Pollastri and Baldi, 2002; Cheng et al., 2005a), a fragment library (Simons et al., 1997), and energy terms derived from the PDB statistics. The structural features used are secondary structure, relative solvent accessibility, and a residue level contact map at a distance cut-off of 12 Å. The predicted structural features are used in the energy function. We include a contact-map energy term (Vendruscolo et al., 1997) based on a binary map derived from the matrix of contact probabilities predicted by CMAPpro. To select the contacts, we use a variable, band-dependent, threshold determined by estimating the total number of contacts in a band from the sum of all the predicted contact probabilities in that band. During the search of conformational space using simulated annealing, many models are produced using different seeds randomly. The single model with the lowest score is returned as the prediction.

8.3 Inputs, Outputs, and Performance

The standalone predictors except for MUpro take as input a sequence in the FASTA format and produce the predicted structural features or structures in self-explanatory format. MUpro requires additional information including mutation position, original residue, and substitute residue.

The input to the server is provided by the user through a simple HTML form. The user must enter an email address for the results to be sent to and the single letter code for an amino acid sequence. The user may also enter a name for the submission. The user may select multiple predictions for the same submission. MUpro is the exception to this simple input format. Input for MUpro is the single letter code for an amino acid sequence, single mutation site, and a new residue to use for replacement. The user may also provide a structure file in the PDB format, but the field is optional. The MUpro prediction results are displayed directly in the browser shortly after submission.

The predictions of all predictors except for MUpro are returned to the email address provided by the user. Here we describe the output of the individual predictors.

- SSpro: helix, strand, coil (loop).
- ACCpro: exposed, buried.

- DISpro: O ordered, D disordered.

- DOMpro: First and last residue of each domain.

- MUpro: In the classification mode, it predicts whether the protein stability is predicted to be increased or decreased by the mutation, and a confidence score. A score near 0 means unchanged stability. Score near -1 means high confidence in decreased stability. Score near +1 means high confidence in increased stability. In the regression model, it returns the real-value energy change ($\triangle\triangle G$).

- DIpro: Two class prediction of whether or not the target has disulfide bonds. Predicted bonding state of each cysteine in the protein. Predicted cysteine pairs.

- BETApro: β-residue pairing probabilities, β-strand pairing energy matrix, β-sheet number and architecture. The images of β-residue pairings and β-sheet pairings are also returned.

- CMAPpro: The contact map predictions are included as an attachment to the the returned email. Predictions come as attached raw files, with extension contact_map.8a and contact_map.12a, for thresholds of 8 and 12$\mathring{A}$, respectively. If the query is N amino acids long the files are composed of N lines, each containing N space-separated real numbers. The j-th number on line i-th represents the estimated probability that amino acids i and j are in contact (i.e. of their C_α atoms being closer than the threshold).

- FOLDpro: Outputs include the top 10 ranked templates with their ranking scores and the predicted 3D structures in PDB format. The ranking scores are generated by Support Vector Machine algorithms. A positive score usually indicates the query and template are significantly related. The higher the score, the more significant the match is. The PDB files and their corresponding alignment files used to generate the structures are attached with the returned email. The alignment files are in the PIR format.

- 3Dpro: Models generated by FOLDpro have the same format as described above. The predicted *ab-initio* model is a PDB file. The PDB file contains only the carbon alpha trace. To obtain an all-atom model a user may use other software to add the backbone, such as MaxSprout (Holm and Sander, 1991), and side chains, such as MaxSprout and SCWRL (Canutescu et al., 2003).

The predictors in the SCRATCH suite produce the state of the art results in general and have been widely used by the scientific community. The SCRATCH system has handled 300 000 jobs since March 2000, including submissions from more than 90 countries. The standalone predictors has been downloaded more than 3500 times since August 2005. Here is a brief summary of the performance of the SCRATCH predictors.

The three class per-amino acid accuracy (Q3) of SSpro 4.0 is 78% (Cheng et al., 2005a). SSpro 4.0 has been extensively evaluated on EVA and has been consistently ranked as one of the top secondary structure prediction servers (Eyrich et al., 2001). The accuracy of ACCpro 4.0 is 78% at the 25% exposure threshold. The prediction accuracies are based on targets where template homology is not used directly, and both systems perform better when template homology can be applied. DOMpro predicts the correct number of domains 69% of the times and it was ranked among the top *ab-initio* domain predictors in the Critical Assessment of Fully Automated Structure Prediction 4 (CAFASP-4)(Fischer et al., 1999; Saini and Fischer, 2005).

The precision and recall of disordered regions of DISpro are 75.4 and 38.8%, respectively. For DIpro, the prediction accuracy of cysteine bonding states is 87% (Baldi et al., 2005). The average disulfide bond prediction accuracy of DIpro is 53% (Baldi et al., 2005). The accuracy of mutation stability prediction of MUpro is 84%. On a test set of proteins with length <100, CMAPpro predicted contacts with 49% accuracy and non-contacts with 96% accuracy (Pollastri and Baldi, 2002). To our best knowledge, BETApro is the only method existing so far that can predict all aspects of β-sheet structure: β-residue pairings, β-strand pairings, and β-sheet architecture.

FOLDpro is a template-based 3D structure predictor. It is most appropriate to use with targets having suitable structural templates. Compared to 11 other fold recognition methods, FOLDpro yields the best results in almost all standard categories on a comprehensive benchmark dataset (Cheng and Baldi, 2006). Using predictions of the top-ranked template, the sensitivity is about 85%, 56%, and 27% at the family, superfamily, and fold levels respectively. Using the 5 top-ranked templates, the sensitivity increases to 90%, 70%, and 48%.

3Dpro is a combination of FOLDpro and an *ab initio* predictor. For targets without good structural templates, 3Dpro use the *ab-initio* predictor. The *ab-initio* structure prediction methods are evaluated by the Critical Assessment of Structure Prediction (CASP) experiments held every two years (Moult et al., 2003). CASP evaluates predictions in three broad categories: CM, fold recognition (FR) and new fold (NF). The easiest targets to predict are categorized as CM-easy, while the hardest are categorized as NF. There is a continuous spectrum of difficulty and these categories blur at the edges as do the methods that work best on different types of targets. We took part in the most recent experiment, CASP6 (for complete results see http://predictioncenter.llnl.gov/). Our *ab-initio* tertiary structure predictor baldi-group-server performed well on hard targets (those in the NF and

FR/A categories) compared with other fully automated predictors.

8.4 Software Structure and System Training

All standalone software packages are packed in the similar format including a configuration (installation) script, a bin sub-directory of Linux shell scripts of launching the applications, a script sub-directory of the Perl source files, a server sub-directory of binary executables complied from C++ code, sub-directories of third party tools such as PSI-BLAST, an optional data sub-directory of local databases, a test sub-directory of test examples, and a model sub-directory of the pre-trained models for neural networks and support vector machines.

DISpro, DOMpro, SSpro, and ACCpro use 1D-RNN neural network architectures. The models are trained by setting the number of hidden units to 15 and the number of output units to 9 for both forward and backward neural networks. The learning rate is set to 0.001. DIpro and BETApro use 2D-RNN architectures. The models are trained by setting the number of hidden units to 9 and the number of output units to 9 for all four hidden planes. The learning rate is set to 0.008. The stochastic learning process is applied for both 1D-RNN and 2D-RNN in the sense that the training examples are randomly shuffled in each epoch.

MUpro and FOLDpro use support vector machine to train models. Both systems use Gaussian kernel. The detailed parameter settings are described in Chapter 2 and Chapter 7, respectively. The generalized performance of these systems is estimated using 10-fold cross validation.

Bibliography

R. Abagyan, D. Frishman, and P. Argos. Recognition of distantly related proteins through energy calculations. *Proteins*, 19:132–140, 1994.

R.A. Abagyan and S. Batalov. Do aligned sequences share the same fold? *J. Mol. Biol.*, 273:355–368, 1997.

V.I. Abkevich and E.I. Shankhnovich. What can disulfide bonds tell us about protein energetics, function and folding: simulations and bioinformatics analysis. *J. Math. Biol.*, 300:975–985, 2000.

B. Al-Lazikani, F.B. Sheinerman, and B. Honig. Combining multiple structure and sequence alignments to improve sequence detection and alignment: application to sh2 domains of janus kinase. *PNAS*, 98:14796–14801, 1998.

S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

C.B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181

(96):223–230, 1973.

M. Asogawa. Beta-sheet prediction using inter-strand residue pairs and refinement with hopfield neural network. In *Proceeding of International Conference on Intelligent System of Molecular Biology*, volume 5, pages 48–51. 1997.

T.L. Bailey and M. Gribskov. Score distributions for simultaneous matching to multiple motifs. *J. Computational Biol.*, 4:45–59, 1997.

A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Res.*, 28:45–48, 2000.

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

P. Baldi and S. Brunak. *Bioinformatics: the machine learning approach.* MIT Press, Cambridge, MA, 2001. Second edition.

P. Baldi, Y. Chauvin, T. Hunkapillar, and M. McClure. Hidden markov models of biological primary sequence information. *PNAS USA*, 91(3):1059–1063, 1994.

P. Baldi, J. Cheng, and A. Vullo. Large-scale prediction of disulphide bond connectivity. In L. Bottou L. Saul, Y. Weiss, editor, *Advances in Neural Information Processing Systems (NIPS04 Conference)*, volume 17, pages 97–104. MIT press, Cambridge, MA, 2005.

P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures-DAG-RNNs and the protein structure prediction problem. *Journal of Machine Learning Research*, 4:575–602, 2003.

P. Baldi, G. Pollastri, C.A.F. Andersen, and S. Brunak. Matching protein β-sheet partners by feedforward and recurrent neural networks. In *Proceedings of the 2000 Conference on Intelligent Systems for Molecular Biology (ISMB00), La Jolla, CA*, pages 25–36. AAAI Press, Menlo Park, CA, 2000.

P. Baldi and M. Rosen-Zvi. On the relationship between deterministic and probabilistic directed graphical models. *Journal of Machine Learning Research*, 2005. In press.

E.N. Baldwin, I.T. Weber, R.S. Charles, J. Xuan, E. Appella, M. Yamada, K. Matsushima, B.F.P. Edwards, G.M. Clore, A.M. Gronenborn, and A. Wlodawar. Crystal structure of interleukin 8: Symbiosis of NMR and crystallography. *Proc. Nat. Acad. Sci. USA*, 88:502–506, 1991.

R.L. Baldwin. The nature of protein folding pathways: the classical versus the new view. *J Biomol NMR*, 5:103–109, 1995.

P.A. Bash, U.C. Singh, R. Langridge, and P.A. Kollman. Free-energy calculations by computer simulation. *Science*, 256:564–568, 1987.

P.A. Bates, L.A. Kelley, R.M. MacCallum, and M.J. Sternberg. Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins*, 45 (Suppl. 5):39–46, 2001.

Y. Bengio and P. Frasconi. Input-output HMM's for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, September 1996.

H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The Protein Data Bank. *Nucl. Acids Res.*, 28: 235–242, 2000.

S. Betz. Disulfide bonds and the stability of globular proteins. *Proteins, Struct., Function Genet.*, 21:167–195, 1993.

P.J. Bjorkman and P. Parham. Structure, function and diversity of class I major histocompatibility complex molecules. *Ann. Rev. Biochem*, 59:253–288, 1990.

T.L. Blundell and L.H. Johnson. *Protein Crystallography.* Academic Press, New York, 1976.

T.L. Blundell, B.L. Sibanda, M.J. Sternberg, and J.M. Thornton. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, 326: 347–352, 1987.

D.N. Bolon, J.S. Marcus, S.A. Ross, and S.L. Mayo. Prudent modeling of core polar residues in computational protein design. *J Mol Biol*, 329:611–622, 2003.

A.J. Bordner and R.A. Abagyan. Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations. *Proteins:Structure, Function, and Bioinformatics*, 57:400–413, 2004.

J.W. Bowie, R. Luthy, and D. Eisenberg. A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253:164–170, 1991.

Sir Lawrence Bragg. The development of x-ray analysis. G. Bell and Sons, London, 1975.

C. Branden and J. Tooze. *Introduction to protein structure;2nd edition.* Garland Publishing, New York, NY, 1999.

W.J. Browne, A.C. North, D.C. Philips, K. Brew, T.C. Vanaman, and R.L. Hill. A possible three-dimensional structure of bovine alpha-lactalbumin based on that of hen.s egg-white lysozyme. *J. Mol. Biol.*, 42:65–86, 1969.

S.H. Bryant and C.E. Lawrence. An empirical energy function for threading protein sequence through the folding motif. *Proteins*, 16:92–112, 1993.

K. Bryson, L.J. McGuffin, R.L. Marsden, J.J. Ward, J.S. Sodhi, and D.T. Jones. Protein structure prediction servers at University College London. *Nucleic Acids Research*, 33:w36–38, 2005.

C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.

A.A. Canutescu, A.A. Shelenkov, and R.L. Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Sci.*, 12, 2003.

E. Capriotti, P. Fariselli, and R. Casadio. A neural network-based method for predicting protein stability changes upon single point mutations. In *Proceedings of the 2004 Conference on Intelligent Systems for Molecular Biology (ISMB04), Bioinformatics(Suppl.1)*, volume 20, pages 190–201. Oxford University Press, 2004.

C.W. Carter, B.C. LeFebvre, S.A. Cammer, A. Torpsha, and M.H. Edgell. Four-body potentials reveal protein-specific correlations to stability changes caused by hydrophobic core mutations. *J Mol Biol*, 311:625–638, 2001.

R. Casadio, M. Compiani, P. Fariselli, and F. Vivarelli. Predicting free energy contributions to the conformational stability of folded proteins from the residue sequence with radial basis function networks. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 3, pages 81–88. 1995.

A. Ceroni, R. Frasconi, A. Passerini, and A. Vullo. Predicting the disulphide bonding state of cysteines with combinations of kernel machines. *VLSI Signal Processing*, 35, 2003.

J.M. Chandonia and S.E. Brenner. The impact of structural genomics: expectations and outcomes. *Science*, 311:347–351, 2006.

J. Cheng and P. Baldi. Three-stage prediction of protein beta-sheets by neural networks, alignments, and graph algorithms. *Bioinformatics*, 21(suppl 1):i75–i84, 2005.

J. Cheng and P. Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22(12):1456–1463, 2006.

J. Cheng, A. Randall, and P. Baldi. Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins: Structure, Function, Bioinformatics*, 62(4):1125–1132, 2006a.

J. Cheng, A.Z. Randall, M.J. Sweredoski, and P. Baldi. SCRATCH: a protein structure and structural feature prediction server. *Nucleic Acids Research*, 33 (web server issue):w72–76, 2005a.

J. Cheng, H. Saigo, and P. Baldi. Large-scale prediction of disulphide bridges using kernel methods, two-dimensional recursive neural networks, and weighted graph matching. *Proteins: Structure, Function, Bioinformatics*, 62(3):617–629, 2006b.

J. Cheng, L. Scharenbroich, P. Baldi, and E. Mjolsness. Sigmoid: Towards a generative, scalable, software infrastructure for pathway bioinformatics and systems biology. *IEEE Intelligent Systems*, 20(3):68–75, 2005b.

J. Cheng, M.J. Sweredoski, and P. Baldi. Accurate prediction of protein disordered regions by mining protein structure data. *Data Mining and Knowledge Discovery*, 11(3):213–222, 2005c.

J. Cheng, M.J. Sweredoski, and P. Baldi. DOMpro: Protein domain prediction using profiles, secondary structure, relative solvent accessibility, and recursive neural networks. *Data Mining and Knowledge Discovery*, 13(1):1–10, 2005d.

D. Chivian, D.E. Kim, L. Malmstrom, P. Bradley, T. Robertson, P. Murphy, C.E. Strauss, R. Bonneau, C.A. Rohl, and D. Baker. Automated prediction of CASP-5 structures using the Robetta server. *Proteins*, 53(S6):524–533, 2003.

C. Chothia. One thousand folds for the molecular biologist. *Nature*, 357:543–544, 1992.

J. Clarke and A.R. Fersht. Engineered disulfide bonds as probes of the folding pathway of barnase - increasing stability of proteins against the rate of denaturation. *Biochemistry*, 32:4322–4329, 1993.

N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines.* Cambridge University Press, Cambridge, UK, 2000.

B.I. Dahiyat. In silico design for protein stabilization. *Curr Opin Biotech*, 10: 387–390, 1999.

L.X. Dang, K.M. Merz, and P.A. Kollman. Free-energy calculations on protein stability: Thr-157 Val-157 mutation of T4 lysozyme. *J. Am Chem Soc*, 111: 8505–8508, 1989.

R. David, M.J. Korenberg, and I.W. Hunter. 3D-1D threading methods for protein fold recognition. *Pharmacogenomics*, 1:445–455, 2000.

M.O. Dayhoff, W.C. Barker, and L.T. Hunt. Establishing homologies in protein sequences. *Methods Enzymol*, 91:524–545, 1983.

W.F. DeGrado. De novo design and structural characterization of proteins and metalloproteins. *Ann Rev Biochem*, 68:779–819, 1999.

L. Demetrius. Thermodynamics and kinetics of protein folding: an evolutionary perpective. *J. Theor. Biol.*, 217:397–411, 2000.

E.W. Dijkstra. A note on two problems in connexion with graphs. *Nemerica Mathematik*, 1:269–271, 1959.

K.A. Dill. Dominant forces in protein folding. *Biochemistry*, 31:7134–7155, 1990.

A. Doig and M. Sternberg. Side chains, conformational entropy in protein folding. *Protein Sci.*, 4:2247–2251, 1995.

H. Domingues, J. Peters, K.H. Schneider, H. Apeler, W. Sebald, H. Oschkinat, and L. Serrano. Improving the refolding yield of interleukin-4 through the optimization of local interactions. *J. Biotechnol.*, 84:217–230, 2000.

Y. Dou, P. Baisnee, G. Pollastri, Y. Pecout, J. Nowick, and P. Baldi. ICBS: a database of interactions between protein chains mediated by beta-sheet formation. *Bioinformatics*, 20(16):2767–2777, 2004. submitted.

H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In T. Petsche M.C. Mozer, M.I. Jordan, editor, *Advances in Neural Information Processing Systems*, volume 9, pages 155–161. MIT Press, Cambridge, MA, 1997.

A.K. Dunker, C.J. Brown, J.D. Lawson, L.M. Iakoucheva, and Z. Obradovic. Intrinsic disorder and protein function. *Biochemistry*, 41(21):6573–6582, 2002.

S.R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14:755–763, 1998.

R.C. Edgar and K. Sjolander. Simultaneous sequence alignment and tree construction using hidden Markov models. *Bioinformatics*, 19:1404–1411, 2003.

R.C. Edgar and K. Sjolander. COACH: profile-profile alignment of protein families using hidden markov models. *Bioinformatics*, 20:1309–1318, 2004.

J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17: 449–467, 1965.

A. Elofsson and et al. Local moves: An efficient algorithm for simulation of protein folding. *Proteins: Structure, Function, and Genetics*, 23:73–82, 1995.

A. Elofsson, D. Fischer, D.W. Rice, S.M. Le Grand, and D. Eisenberg. A study of combined structure/sequence profiles. *Fold Des.*, 1:451–461, 1996.

S. Even. *Graph Algorithms.* Computer Science Press, Rockville, MD, 1979.

V.A. Eyrich, M.A. Marti-Renom, D. Przybylski, M.S. Madhusudhan, A. Fiser, F. Pazos, A. Valencia, A. Sali, and B. Rost. EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics*, 17:1242–1243, 2001.

P. Fariselli and R. Casadio. Prediction of disulfide connectivity in proteins. *Bioinformatics*, 17:957–964, 2001.

P. Fariselli and R. Casadio. Prediction of disulfide connectivity in proteins. *Bioinformatics*, 17:957–964, 2004.

P. Fariselli, P.L. Martelli, and R. Casadio. A neural network-based method for predicting the disulfide connectivity in proteins. 6th International Conference on Knowledge-Based Intelligent and Engineering Systems, 2002.

P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, 13:835–843, 2001.

P. Fariselli, P. Riccobelli, and R. Casadio. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. *Proteins*, 36:340–346, 1999.

F. Ferre and P. Clote. Disulfide connectivity prediction using secondary structure information and diresidue frquencies. *Bioinformatics*, 21:2336–2346, 2005.

D. Fischer. Hybrid fold recognition: combining sequence derived properties with evolutionary information. In R.B. Altman, A.K. Dunker, Hunter, K. Lauderdale, and T.E. Klein, editors, *Pacific Symp. Biocomputing*, pages 119–130. World Scientific, New York, 2000.

D. Fischer. 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins*, 51:434–441, 2003.

D. Fischer, C. Barret, K. Bryson, A. Elofsson, A. Godzik, D. Jones, K.J. Karplus, L.A. Kelley, R.M. MacCallum, K. Pawowski, B. Rost, L. Rychlewski, and M. Sternberg. CAFASP-1: Critical assessment of fully automated structure prediction methods. *Proteins*, Suppl 3:209–217, 1999.

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

D. Fisher and D. Eisenberg. Protein fold recognition using sequence-derived predictions. *Prot. Sci.*, 5:947–955, 1996.

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

P. Frasconi, A. Passerini, and A. Vullo. A two stage SVM architecture for predicting the disulfide bonding state of cysteines. In *Proceedings of IEEE Neural Network for signal processing conference*. IEEE Press, 2002.

C. Frenz. Neural network-based prediction of mutation-induced protein stability changes in staphylococcal nuclease at 20 residue positions. *Proteins*, 59(2):147–151, 2005.

J. Funahashi, K. Takano, and K. Yutani. Are the parameters of various stabilization factors estimated from mutant human lysozymes compatible with other proteins? *Protein Eng.*, 14:127–134, 2001.

H.N. Gabow. An efficient implementation of Edmond's algorithm for maximum weight matching on graphs. *Journal of the ACM*, 23(2):221–234, 1976.

R.A. George and J. Heringa. SnapDRAGON: a method to delineate protein structural domains from sequence data. *Journal of Molecular Biology*, 316:839–851, 2002.

J.E. Gewehr and R. Zimmer. SSEP-Domain:protein domain prediction by alignment of secondary structure elements and profiles. *Bioinformatics*, page In press, 2005.

D. Gilis and M. Rooman. Predicting protein stability changes upon mutation using database-derived potentials: solvent accessibility determines the importance of local versus non-local interactions along the sequence. *J. Mol. Biol.*, 272:276–290, 1997.

D. Gilis and M. Rooman. Prediction of stability changes upon single-site mutations using database-derived potentials. *Theor Chem Acc*, 101:46–50, 1999.

D. Gillis and M. Rooman. Stability changes upon mutation of solvent-accessible residues in proteins evaluated by database-derived potentials. *J. Mol. Biol.*, 257: 1112–1126, 1996.

K. Ginalski. Comparative modeling for protein structure prediction. *Current Opinion in Structural Biology*, 16(2):172–177, 2006.

K. Ginalski, A. Elofsson, D. Fischer, and L. Rychlewski. 3D-Jury: A simple approach to improve protein structure predictions. *Bioinformatics*, 19:1015–1018, 2003a.

K. Ginalski, J. Pas, L.S. Wyrwicz, M. vonGrotthuss, J.M. Bujnicki, and L. Rychlewski. ORFeus: Detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acids Res*, 31:3804–3807, 2003b.

A. Godzik, J. Skolnick, and A. Kolinski. Topology fingerprint approach to the inverse protein folding problem. *J. Mol. Biol*, 227:227–238, 1992.

J. Gough, K. Karplus, R. Hughey, and C. Chothia. Assignment of homology to genome sequences using a library of hidden markov models that represent all proteins of known structure. *J Mol Biol*, 313:903–919, 2001.

A. Grant, D. Lee, and C. Orengo. Progress towards mapping the universe of protein folds. *Genome Biol.*, 5:107, 2004.

J. Greer. Comparative modeling methods: Application to the family of the mammalian serine proteases. *Proteins*, 7:317–334, 1990.

S. Griffiths-Jones and A. Bateman. The use of structure information to increase alignment accuracy does not aid homologue detection with profile HMMs. *Bioinformatics*, 18:1243–1249, 2002.

M. Gromiha, J. An, H. Kono, M. Oobatake, H. Uedaira, P. Prabakaran, and A. Sarai. ProTherm, version 2.0: thermodynamic database for proteins and mutants. *Nucleic Acids Res*, 28:283–285, 2000.

R. Guerois, J.E. Nielsen, and L. Serrano. Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J. Mol. Biol.*, 320:369–387, 2002.

J. Hargbo and A. Elofsson. A study of hidden markov models that use predicted secondary structures for fold recognition. *Proteins*, 36:68–87, 1999.

P.M. Harrison and M.J.E. Sternberg. Analysis and classification of disulfide connectivity in proteins. *J. Mol. Biol.*, 244:448–463, 1994.

A. Heger and L. Holm. Exhaustive enumeration of protein domain families. *Journal of Molecular Biology*, 328:749–767, 2003.

S. Henikoff and J.G. Henikoff. Amino acid substitutes matrices from protein blocks. *PNAS*, 89:10915–10919, 1992.

U. Hobhom, M. Scharf, P. Schneider, and C. Sander. Selection of representative protein data sets. *Prot. Sci.*, 1:409–417, 1992.

U. Hobohm and C. Sander. Adaptive mixtures of local experts. *Protein Science*, 3, 1994.

L. Holm and C. Sander. Database algorithm for generating protein backbone and side-chain co-ordinates from the c alpha trace application to model building and detection of co-ordinate errors. *J. Mol. Biol.*, 218:183–194, 1991.

L. Holm and C. Sander. Parser for protein folding units. *Proteins*, 19:256–268, 1994.

L. Holm and C. Sander. Dictionary of recurrent domains in protein structures. *Proteins*, 33:88–96, 1998a.

L. Holm and C. Sander. Touring protein fold space with Dali/FSSP. *Nucleic Acids Research*, 26:316–319, 1998b.

B. Honig. Protein folding: from the Levinthal paradox to structure prediction. *J. Mol. Biol.*, 293:283–293, 1999.

T.J. Hubbard. Use of β-strand interaction pseudo-potentials in protein structure prediction and modelling. In R. H. Lathrop, editor, *Proceedings of the Biotechnology Computing Track, Protein Structure Prediction MiniTrack of the 27th HICSS*, pages 336–354. IEEE Computer Society Press, 1994.

R. Hughey and A. Krogh. Hidden Markov models for sequence analysis:extension and analysis of the basic method. *Comput. Appl Biosci*, 12:95–107, 1996.

E.G. Hutchinson, R.B. Sessions, J.M. Thornton, and D.N. Woolfson. Determinants of strand register in antiparallel beta-sheets of proteins. *Protein Sci*, 7(11):287–300, 1998.

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

M. Jacobson and A. Sali. Comparative protein structure modeling and its applications to drug discovery. In J. Overington, editor, *Annual reports in medical chemistry*, pages 259–276. Academic Press, Lodon, 2004.

L. Jaroszewski, L. Rychlewski, B. Zhang, and A. Godzik. Fold prediction by a hierarchy of sequence, threading, and modeling methods. *Protein Sci.*, 7:1431–1440, 1998.

T. Joachims. *Making large-scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola(ed.).* MIT Press, 1999.

T. Joachims. *Learning to Classify Text Using Support Vector Machines. Dessertation.* Springer, 2002.

D.T. Jones. GenTHREADER: An efficient and reliable protein fold recognition method for genomic sequences. *J. Molecular Biology*, 287:797–815, 1999a.

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

D.T. Jones, W.R. Taylor, and J.M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1992.

D. Juan, O. Grana, F. Pazos, P. Fariselli, R. Casadio, and A. Valencia. A neural network approach to evaluate fold recognition results. *Proteins*, 50:600–608, 2003.

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

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

L.A. Kelley, R.M. MacCallum, and M.J. Sternberg. Enhanced genome annotation using structural profiles in the program 3D-PSSM. *J. Mol. Biol.*, 299:499–520, 2000.

J.C. Kendrew, R.E. Dickerson, B.E. Strandberg, R.J. Hart, D.R. Davies, D.C. Phillips, and V.C. Shore. Structure of myoglobin: a three-dimensional fourier synthesis at 2å resolution. *Nature*, 185:422–427, 1960.

D. Kim, D. Xu, J. Guo, K. Ellrott, and Y. Xu. PROSPECT II: Protein structure prediction method for genome-scale applications. *Protein Engineering*, 16(9):641–650, 2003.

J.L. Klepeis and C.A. Floudas. Prediction of β-sheet topology and disulfide bridges in polypeptides. *J. Comput. Chem.*, 24:191–208, 2003.

T.A. Klink, K.J. Woycechosky, K.M. Taylor, and R.T. Raines. Contribution of disulfide bonds to the conformational stability and catalytic activity of ribonuclease A. *Eur. J. Biochem.*, 267:566–572, 2000.

P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.*, 239:249–275, 1994.

K.K. Koretke, R.B. Russell, and A.N. Lupas. Fold recognition from sequence comparisions. *Proteins*, Suppl. 5:68–75, 2001.

T. Kortemme, M. Ramirez-Alvarado, and L. Serrano. Design of a 20-amino acid, three-stranded β-sheet protein. *Science*, 281:253–256, 1998.

A. Krogh, M. Brown, I. S.Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, (235):1501–1531, 1994.

J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceeding of the American Mathematical Society*, volume 7, pages 48–50. 1956.

R. Kuang, E. Ie, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for detection of remote homologs and discriminative motifs. *J Bioinform Computa Biol*, 2005.

B. Kuhlman, G. Dantas, G.C. Ireton, G. Varani, B.L. Stoddard, and D. Baker. Design of a novel globular protein fold with atomic-level accuracy. *Science*, 302: 1364–1368, 2003.

J.M. Kwasigroch, D. Gillis, Y. Dehouck, and M. Rooman. Popmusic, rationally designing point mutations in protein structures. *Bioinformatics*, 18:1701–1702, 2002.

E. Lacroix, A.R. Viguera, and L. Serrano. Elucidating the folding problem of alpha-helices: local motifs, long-range electrostatics, ionic-strength dependence and prediction of NMR parameters. *J. Mol. Biol*, 284:173–191, 1998.

G.R.G. Lanckriet, N. Cristianini, M.I. Jordan, and W.S. Noble. Kernel-based integration of genomic data using semidefinite programming. In *Kernel Methods in Computational Biology*. MIT Press, 2004.

R.A. Laskowski, J.D. Watson, and J.M. Thornton. From protein structure to biochemical function? *J. Struct. Funct. Genomics*, 4:167–177, 2003.

T. Lazaridis and M. Karplus. Effective energy functions for protein structure prediciton. *Curr. Opin. Struct. Biol.*, 10:139–145, 2000.

C. Lee. Testing homology modeling on mutant proteins: predicting structural and thermodynamic effects in the Ala98-Val mutants of T4 lysozyme. *Fold Des*, 1: 1–12, 1995.

C. Lee and M. Levitt. Accurate prediction of the stability and activity effects of site-directed mutagenesis on a protein core. *Nature*, 352:448–451, 1991.

C. Leslie, E. Eskin, A. Cohen, J. Weston, and W. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.

C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: a string kernel for SVM protein classification. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauerdale, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 564–575. World Scientific, 2002.

C. Levinthal. Are there pathways for protein folding? *J. Chim. Phys.*, 65:44–45, 1968.

M. Levitt. Accurate modeling of protein conformation by automatic segment matching. *J. Mol. Biol.*, 226:507–533, 1992.

M. Levitt and C. Chothia. Structural patterns in globular proteins. *Nature*, 261 (5561):552–558, 1976.

M. Lexa and G. Valle. PRIMEX: rapid identification of oligonucleotide matches in whole genomes. *Bioinformatics*, 19:2486–2488, 2003.

X. Li, P. Romero, M. Rani, A.K. Dunker, and Z. Obradovic. Predicting protein disorder for N-, C-, and internal regions. *Genome Inform.*, 42:38–48, 1999.

L. Liao and W.S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *Proceedings of the Sixth International Conference on Computational Molecular Biology*, 2002.

S. Lifson and C. Sander. Specific recognition in the tertiary structure of beta-sheets of proteins. *Journal of Molecular Biology*, 139(4):627–639, 1980.

E. Lindahl and A. Elofsson. Identification of related proteins on family, superfamily and fold level. *J. Molecular Biology*, 295:613–625, 2000.

R. Linding, L.J. Jensen, F. Diella, P. Bork, T.J. Gibson, and R.B. Russell. Protein disorder prediction: Implications for structural proteomics. *Structure*, 11(11): 1453–1459, Nov 2003a.

R. Linding, R.B. Russell, V. Neduva, and T.J. Gibson. GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Research*, 31:3701–3708, 2003b.

J. Liu and B. Rost. Sequence-based prediction of protein domains. *Nucleic Acids Research*, 32(12):3522–3530, 2004.

L.L. Looger, M.A. Dwyer, J.J. Smith, and H.W. Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature*, 423:185–190, 2003.

J. Lundstrm, L. Rychlewski, J. Bujnicki, and Ar. Elofsson. Pcons: A neural network based consensus predictor that improves fold recognition. *Protein Science*, 10:

2354–2362, 2001.

R. MacCallum. Striped sheets and protein contact prediction. *Bioinformatics*, 20 (Supplement 1):i224–i231, 2004.

M. Madera and J. Gough. A comparison of profile hidden markov model procedures for remote homology detection. *Nucleic Acids Res*, 30:4321–4328, 2002.

Y. Mandel-Gutfreund, S.M. Zaremba, and L.M. Gregoret. Contributions of residue pairing to beta-sheet formation:conservation and covariation of amino acid residue pairs on antiparallel beta-strands. *Journal of Molecular Biology*, 305(2):1145–1159, 2001.

A. Marchler-Bauer, J.B. Anderson, C. DeWeese-Scott, N.D. Fedorova, L.Y. Geer, S. He, D.I. Hurwitz, J.D. Jackson, A.R. Jacobs, C.J. Lanczycki, C.A. Liebert, C. Liu, T. Madej, G.H. Marchler, R. Mazumder, A.N. Nikolskaya, A.R. Panchenko, B.S. Rao, B.A. Shoemaker, V. Simonyan, J.S. Song, P.A. Thiessen, S. Vasudevan, Y. Wang, R.A. Yamashita, J.J. Yin, and S.H. Bryant. CDD: a curated Entrez database of conserved domain alignments. *Nucleic Acids Research*, 31(1):383–387, 2003.

R.L. Marsden, L.J. McGuffin, and D.T. Jones. Rapid protein domain assignment from amino acid sequence using predicted secondary structure. *Protein Science*, 11:2814–2824, 2002.

P.L. Martelli, P. Fariselli, L. Malaguti, and R. Casadio. Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. *Protein Sci.*, 11(11): 2735–9, 2002.

M. Marti-Renom, A. Stuart, A. Fiser, R. Sanchez, F. Melo, and A. Sali. Comparative protein structure modeling of genes and genomes. *Annu Rev Biophys Biomol Struct*, 29:291–325, 2000.

M.A. Marti-Renom, M. Madhusudhan, and A. Sali. Alignment of protein sequences by their profiles. *Protein Sci*, 13:1071–1087, 2004.

M. Matsumura, G. Signor, and B. W. Matthews. Substantial increase of protein stability by multiple disulfide bonds. *Nature*, 342:291–293, 1989.

J. Mendes, R. Guerois, and L. Serrano. Energy estimation in protein design. *Curr Opin Struct Biol*, 12:441–446, 2002.

J.S. Merkel and L. Regan. Modulating protein folding rates in vivo and in vitro by side-chain interactions between the parallel beta strands of green fluorescent protein. *J Biol Chem*, 275:29200–29206, 2000.

S. Mika and B. Rost. UniqueProt: creating representative protein sequence sets. *Nucleic Acids Research*, 31(13):3789–3791, 2003.

D.L. Minor and S. Kim. Context is a major determinant of beta-sheet propensity. *Nature*, 371((6494)):264–267, 1994.

D. Mitelman, R. Sadreyev, and N. Grishin. Probabilistic scoring measures for profile-profile comparison yield more accurate short seed alignments. *Bioinformatics*, 19: 1531–1539, 2003.

S. Miyazawa and R.L. Jernigan. Protein stability for single substitution mutants and the extent of local compactness in the denatured state. *Protein Eng.*, 7: 1209–1220, 1994.

J. Moult, K. Fidelis, A. Tramontano, B. Rost, and T. Hubbard. Critical assessment of methods of protein structure prediction (CASP) - round VI. *Proteins*, 2005. In press.

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

K.R. Müller, G. Rätsch S. Mika, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181–201, 2001.

V. Munoz and L. Serrano. Development of the multiple sequence approximation within the AGADIR model of alpha-helix formation: comparison with Zimm-Bragg and Lifson-Roig formalisms. *Biopolymers*, 41:495–509, 1997.

A.G. Murzin and A. Bateman. Distance homology recognition using structural classification of proteins. *Proteins*, Suppl. 1:105–112, 1997.

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

N. Nagarajan and G. Yona. Automatic prediction of protein domains from sequence information using a hybrid learning system. *Bioinformatics*, 20:1335–1360, 2004.

S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48:443–453, 1970.

C. Notredame, D. Higgins, and J. Heringa. T-Coffee: A novel method for multiple sequence alignment. *J Mol Biol*, 302:205–217, 2000.

T. Ohlson, B. Wallner, and A. Elofsson. Profile–profile methods provide improved fold–recognition. a study of different profile-profile alignment methods. *Proteins*, 57:188–197, 2004.

N. Ohsen, I. Sommer, and R. Zimmer. Profile-profile alignment: a powerful tool for protein structure prediction. *PSB*, 2003.

C.A. Orengo, J.E. Bray, D.W. Buchan, A. Harrison, D. Lee, F.M. Perl, I. Sillitoe, A.E. Todd, and J.M. Thornton. The CATH protein family database: a resource for structural and functional annotation of genomes. *Proteomics*, 2:11–21, 2002.

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

O. O'Sullivan, K. Suhre, C. Abergel, D.G. Higgins, and C. Notredame. 3DCoffee: Combing protein sequences and structures within multiple sequence alignment. *J Mol Biol*, 340:385–395, 2004.

L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank citation ranking: Bringing order to the web.* Stanford University, 1998. Technical report.

A.R. Panchenko, A. Marchler-Bauer, and S.H. Bryant. Combination of threading potentials and sequence profiles improves fold recognition. *J. Mol. Biol*, 296: 1319–1331, 2000.

J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol Biol*, 284:1201–1210, 1998.

L. Pauling and R.B. Corey. The pleated sheet, a new layer configuration of the polypeptide chain. *Proc. Nat. Acad. Sci. USA*, 37:251–256, 1951.

L. Pauling, R.B. Corey, and H.R. Branson. The structure of proteins: Two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Nat. Acad. Sci. USA*, 37:205–211, 1951.

W.R. Pearson and D.J. Lipman. Improved tools for biological sequences analysis. *PNAS*, 85:2444–2448, 1988.

S.G. Pemberton and B. Jones. Ichnology of the pleistocene ironshore formation, grand cayman island, british west-indies. *J. Paleontol.*, 62:495, 1999.

M.F. Perutz, M.G. Rossmann, A.F. Cullis, G. Muirhead, G. Will, and A.T. North. Structure of haemoglobin: a three-dimensional fourier synthesis at 5.5å resolution, obtained by x-ray analysis. *Nature*, 185:416–422, 1960.

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

D. Petrey, Z. Xiang, C.L. Tang, L. Xie, M. Gimpelev, T. Mitros, C.S. Soto, S. Goldsmith-Fischman, A. Kernytsky, and A. Schlessinger. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. *Proteins*, 53 (Suppl. 6):430–435, 2003.

C. Pettitt, L.J. McGuffin, and D.T. Jones. Improving sequence-based fold recognition by using 3d model quality assessment. *Bioinformatics*, 21:3509–3515, 2005.

J.W. Pitera and P.A. Kollman. Exhaustive mutagenesis in silico: multicoordinate free energy calculations on proteins and peptides. *Proteins*, 41:385–397, 2000.

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

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

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

M. Prevost, S.J. Wodak, B. Tidor, and M. Karplus. Contribution of the hydrophobic effect to protein stability: analysis based on simulations of the Ile-96-Ala mutation in barnsase. *Proc. Natl. Acad. Sci.*, 88:10880–10884, 1991.

D. Przybylski and B. Rost. Alignments grow, secondary structure prediction improves. *Proteins*, 46:197–205, 2002.

M. Punta and B. Rost. Toward good 2D predictions in proteins. *FEBS*, 2005. in press.

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

J.J. Rocchio. *Document retrieval systems - optimization and evaluation.* Harvard University, 1966. Ph.D Thesis.

B. Rost, J. Liu, D. Przybylski, R. Nair, K.O. Wrzeszczynski, H. Bigelow, and Y. Ofran. Prediction of protein structure through evolution. In J. Gasteiger and T. Engel, editors, *Handbook of Chemoinformatics - From Data to Knowl-*

edge, pages 1789–1811. Wiley, New York, 2003.

B. Rost and S.I. O'Donoghue. Sisyphus and prediction of protein structure. *CABIOS*, 13:345–356, 1997.

B. Rost and C. Sander. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 90(16):7558–7562, 1993a.

B. Rost and C. Sander. Prediction of protein secondary structure at better than 70 % accuracy. *Journal of Molecular Biology*, 232(2):584–599, 1993b.

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

B. Rost, R. Schneider, and C. Sander. Protein fold recognition by prediction-based threading. *J. Mol. Biol.*, 270:471–480, 1997.

I. Ruczinski, C. Kooperberg, R. Bonneau, and D. Baker. Distributions of beta sheets in proteins with application to structure prediction. *Proteins*, 48:85–97, 2002.

L. Rychlewski, L. Jaroszewski, W. Li, and A. Godzik. Comparison of sequence profiles. strategies for structural predictions using sequence information. *Protein Sci.*, 9:232–241, 2000.

R. Sadreyev and N. Grishin. COMPASS: A tool for comparison of multiple protein alignments with assessment of statistical significance. *J Mol Biol*, 326:317–336, 2003.

H. Saigo, J-P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004.

H.K. Saini and D. Fischer. Meta-DP: domain prediction meta server. *Bioinformatics*, 21:2917–2920, 2005.

A. Sali and T.L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *J Mol Biol*, 234:779–815, 1993.

A. Sali, E. Shakhnovich, and M. Karplus. How does a protein fold? *Nature*, 369: 248–251, 1994.

C. Sander and R. Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, 9:56–68, 1991.

F. Sanger and E.O. Thompson. The amino-acid sequence in the glycyl chain of insulin. I. the identification of lower peptides from partial hydrolysates. *J. Biochem*, 53(3):353–366, 1953a.

F. Sanger and E.O. Thompson. The amino-acid sequence in the glycyl chain of insulin. II. the investigation of peptides from enzymic hydrolysates. *J. Biochem*, 53(3):366–374, 1953b.

J. Saven. Combinatorial protein design. *Curr Opin Struct Biol*, 12:453–458, 2002.

A.A. Schaffer, Y.I. Wolf, C.P. Ponting, E.V. Koonin, L. Aravind, and S.F. Altschul. IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics*, 15:1000–1011, 1999.

B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, 2002.

B. Schölkopf, K. Tsuda, and J-P. Vert. *Support vector machine applications in computational biology.* MIT Press, Cambridge, MA, 2004.

T. Schwede, J. Kopp, N. Guex, and M.C. Peitsch. SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Res.*, 31:3381–3385, 2004.

Y.B. Shan, G.L. Wang, and H.X. Zhou. Fold recognition and accurate query-template alignment by a combination of PSI-BLAST and threading. *Proteins*, 42:23–37, 2001.

Y. Shao and C. Bystroff. Predicting inter-residue contacts using templates and pathways. *Proteins*, 53(Supplement 6):497–502, 2003.

B.E. Shapiro, A. Levchenko, E.M. Meyerowitz, B.J. Wold, and E.D. Mjolsness. Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. *Bioinformatics*, 19(5):677–678, 2002.

J. Shi, T.L. Blundell, and K. Mizuguchi. FUGUE: Sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Molecular Biology*, 310:243–257, 2001.

K.T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *J Mol Biol*, 268:209–225, 1997.

K.T. Simons, C. Strauss, and D. Baker. Prospects for ab initio protein structural genomics. *J. Mol. Biol.*, 306:1191–1199, 2001.

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

J. Skolnick and et al. Momster: A method for folding globular proteins with a small number of distance restraints. *J. Mol. Biol.*, 265:217–241, 1997.

J. Skolnick and D. Kihara. Defrosting the frozen approximation: PROSPECTOR-a new approach to threading. *Proteins*, 42:319–331, 2001.

J. Skolnick and A. Kolinski. Dynamic monte carlo simulations of a new lattice model of globular protein folding, structure and dynamics. *J Mol Biol*, 221:449–531, 1991.

C.K. Smith and L. Regan. Guidelines for protein design: The energetics of β sheet side chain interations. *Science*, 270(5238):980–982, 1995.

C.K. Smith and L. Regan. Construction and design of beta-sheets. *Acc Chem Res*, 30:153, 1997.

T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

A. Smola and B. Scholkopf. A tutorial on support vector regression. In *NeuroCOLT Technical Report NC-TR-98-030*. Royal Holloway College, University of London, UK, 1998.

J. Söding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21:951–960, 2005.

R.E. Steward and J.M. Thornton. Prediction of strand pairing in antiparallel and parallel beta-sheets using information theory. *Proteins: Structure, Function, and Genetics*, 48:178–191, 2002.

A.G. Street and S.L. Mayo. Computational protein design. *Struct Fold Des*, 7: R105–R109, 1999.

N. Taddei, F. Chiti, T. Fiaschi, M. Bucciantini, C. Capanni, and M. Stefani. Stabilisation of alpha-helices by site-directed mutagenesis reveals the importance of secondary structure in the transition state for acylphosphatase folding. *J. Mol. Biol.*, 300:633–647, 2000.

K. Takano, M. Ota, K. Ogasahara, Y.Yamagata, K. Nishikawa, and K. Yutani. Experimental verification of the stability profile of mutant protein (spmp) data using mutant human lysozymes. *Protein Eng.*, 12:663–672, 1999.

C.L. Tang, L. Xie, I.Y. Koh, S. Posy, E. Alexov, and B. Honig. On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. *J Mol Biol*, 334:1043–1062, 2003.

W.R. Taylor. Towards protein tertiary fold prediction using distance and motif constraints. *Protein Eng.*, 4:853–870, 1991.

J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22: 4673–4680, 1994.

B. Tidor and M. Karplus. Simulation analysis of the stability mutant R96H of T4 lysozyme. *Biochemistry*, 30:3217–3228, 1991.

C.M. Topham, N. Srinivasan, and T.L. Blundell. Prediction of the stability of protein mutants based on structural environment-dependent amino acid substitution and propensity tables. *Prot. Eng*, 101:46–50, 1997.

A. Travers. DNA conformation and protein binding. *Ann. Rev. Biochem*, 58:427–452, 1989.

V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, Berlin, Germany, 1995.

V. Vapnik. *Statistical Learning Theory.* Wiley, New York, NY, 1998.

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

J. Vert, K. Tsuda, and B. Scholkopf. A primer on kernel methods. In J. Vert B. Scholkopf, K. Tsuda, editor, *Kernel Methods in Computational Biology*, pages 55–72. MIT Press, Cambridge, MA, 2004.

V. Villegas, A.R. Viguera, F.X. Aviles, and L. Serrano. Stabilization of proteins by rational design of alpha-helix stability using helix/coil transition theory. *Fold. Des.*, 1:29–34, 1996.

M. Vingron and M.S. Waterman. Sequence alignment and penalty choice. review of concepts, case studies and implications. *J. Mol. Biol.*, 235:1–12, 1994.

N. von Ohsen, I. Sommer, R. Zimmer, and T. Lengauer. Arby: automatic protein structure prediction using profile-profile alignment and confidence measures. *Bioinformatics*, 20:2228–2235, 2004.

A. Vullo and P. Frasconi. A recursive connectionist approach for predicting disulfide connectivity in proteins. In *Proceedings of the 18th annual ACM symposium on applied computing (SAC 2003)*, pages 67–71, 2003.

A. Vullo and P. Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20:653–659, 2004.

B. Wallner, H. Fang, T. Ohlson, J. Frey-Skott, and A. Elofsson. Using evolutionary information for the query and target improves fold recognition. *Proteins*, 54: 342–350, 2004.

G. Wang and RL Jr Dunbrack. Scoring profile-profile sequence alignments. *Protein Sci*, 13:1612–1626, 2004.

J.J. Ward, J.S. Sodhi, L.J. McGuffin, B.F. Buxton, and D.T. Jones. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *Journal of Molecular Biology*, 337(3):635–645, Mar 2004.

W.J. Wedemeyer, E. Welkler, M. Narayan, and H.A. Scheraga. Disulfide bonds and protein-folding. *Biochemistry*, 39:4207–4216, 2000.

J. Westbrook and P.M. Fitzgerald. The PDB format, mmCIF formats and other data formats. *Structural Bioinformatics*, 2003.

S.J. Wheelan, A. Marchler-Bauer, and S.H. Bryant. Domain size distributions can predict domain boundaries. *Bioinformatics*, 16(7):613–618, 2000.

J. Wootton. Non-globular domains in protein sequences: automated segmentation using complexity measures. *Computational Chemistry*, 18:269–285, 1994.

M.A. Wouters and P.M.G. Curmi. An analysis of side-chain interactions and pair correlations within antiparallel beta-sheets: the differences between backbone hydrogen-bonded and non-hydrogen-bonded residue pairs. *Proteins Struct. Funct. Genet.*, 22:119–131, 1995.

P.E. Wright and H.J. Dyson. Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. *Journal of Molecular Biology*, 293(2):321–331, Oct 1999.

K. Wuthrich. *NMR of Proteins and Nucleic Acids.* John Wiley & Sons, New York, 1986.

J. Xu, M. Li, G. Lin, D. Kim, and Y. Xu. Protein threading by linear programming. *Pac Symp Biocomput*, pages 264–275, 2003a.

J. Xu, Y. Xu, G. Lin, D. Kim, and M. Li. Protein structure prediction by linear programming. In *Pac Symp Biocomput.* 2003b.

Y. Xu, D. Xu, and E.C. Uberbacher. An efficient computational method for globally optimal threadings. *J. Computational Biology*, 5:597–614, 1998.

Y. Yang and J.P. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML97)*, pages 412–420. 1997.

G. Yona and M. Levitt. Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *J. Mol. Biol.*, 315:1257–1275, 2002.

S.M. Zaremba and L.M. Gregoret. Context-dependence of amino acid residue pairing in antiparallel β-sheets. *Journal of Molecular Biology*, 291(2):463–479, 1999.

E.M. Zdobnov and R. Apweiler. InterProScan–an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, 17:847–848, 2001.

C. Zhang and S. Kim. The anatomy of protein beta-sheet topology. *Journal of Molecular Biology*, 2(4):1075–1089, 2000.

Y. Zhang, A. Kolinski, and J. Skolnick. TOUCHSTONE II: A new approach to ab initio protein structure prediction. *Biophysics*, 85:1145–1164, 2003.

H. Zhou and Y. Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.*, 11:2714–2726, 2002.

H. Zhou and Y. Zhou. Quantifying the effect of burial of amino acid residues on protein stability. *Proteins*, 54:315–322, 2004.

H. Zhou and Y. Zhou. Fold recognition by combining sequence profiles derived from evolution and from depth-dependent structural alignment of fragments. *Proteins*, 58:321–328, 2005.

H. Zhu and W. Braun. Sequence specificity, statistical potentials, and three-dimensional structure prediction with self-correcting. *Protein Sci.*, 8:326–342, 1999.