Towards Automating Protein Structure Determination from NMR Data

by

Xin Gao

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Computer Science

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Xin Gao

Abstract

Nuclear magnetic resonance (NMR) spectroscopy technique is becoming exceedingly significant due to its capability of studying protein structures in solution. However, NMR protein structure determination has remained a laborious and costly process until now, even with the help of currently available computer programs. After the NMR spectra are collected, the main road blocks to the fully automated NMR protein structure determination are peak picking from noisy spectra, resonance assignment from imperfect peak lists, and structure calculation from incomplete assignment and ambiguous nuclear Overhauser enhancements (NOE) constraints.

The goal of this dissertation is to propose error-tolerant and highly-efficient methods that work well on real and noisy data sets of NMR protein structure determination and the closely related protein structure prediction problems.

One major contribution of this dissertation is to propose a fully automated NMR protein structure determination system, AMR, with emphasis on the parts that I contributed. AMR only requires an input set with six NMR spectra. We develop a novel peak picking method, PICKY, to solve the crucial but tricky peak picking problem. PICKY consists of a noise level estimation step, a component forming step, a singular value decomposition-based initial peak picking step, and a peak refinement step. The first systematic study on peak picking problem is conducted to test the performance of PICKY. An integer linear programming (ILP)-based resonance assignment method, IPASS, is then developed to handle the imperfect peak lists generated by PICKY. IPASS contains an error-tolerant spin system forming method and an ILP-based assignment method. The assignment generated by IPASS is fed into the structure calculation step, FALCON-NMR. FALCON-NMR has a threading module, an *ab initio* module, an all-atom refinement module, and

an NOE constraints-based decoy selection module. The entire system, AMR, is successfully tested on four out of five real proteins with practical NMR spectra, and generates 1.25Å, 1.49Å, 0.67Å, and 0.88Å to the native reference structures, respectively.

Another contribution of this dissertation is to propose novel ideas and methods to solve three protein structure prediction problems which are closely related to NMR protein structure determination. We develop a novel consensus contact prediction method, which is able to eliminate server correlations, to solve the protein inter-residue contact prediction problem. We also propose an ultra-fast side chain packing method, which only uses local backbone information, to solve the protein side chain packing problem. Finally, two complementary local quality assessment methods are proposed to solve the local quality prediction problem for comparative modeling-based protein structure prediction methods.

Acknowledgements

I would like to owe my greatest gratitude to all the people who made this thesis possible. I am deeply indebted to my supervisor Ming Li and my co-supervisor Jinbo Xu, whose guidance, encouragement, patience, and supports during my entire Ph.D. study enabled me to finish my dissertation.

It was Ming who led me into the area of bioinformatics. He provided me invaluable ideas and helped on every single goal I have accomplished. He is the best and the kindest teacher I have ever met. It was my great honor to have such a wonderful mentor who not only helped me build my own academic career, but also cared for me on everyday life.

It was Jinbo who opened the door of protein structure prediction to me. By participating CASPs with him and working on protein structure prediction problems with him, I have learned a great deal and published a number of papers on these topics.

Special thanks to my committee members, Professor Dong Xu, Professor Thorsten Dieckmann, Professor Brendan McConkey, and Professor Bin Ma, for taking their precious time to review this thesis and provide valuable comments.

I am grateful to my collaborators, Babak Alipanahi, Emre Karakoc, Shuai Cheng Li, Jing Zhang, Libo Yu, Guangyu Feng, and Frank Balbach at our lab, Professor Dongbo Bu at Chinese Academy of Sciences, and Professor Logan Donaldson at York University , for their wonderful collaboration works on thirteen academic papers. My friends, Xuefeng Cui, Lin He, Xi Han, Yuzhong Zhao, Hongyu Li, Jianwei Niu, Jianfei Niu, Zhiwei Wang, Fang Wei, Luosha Lu, Xiaowen Liu, Bo Wang, Can Tang, Feifei Lin, Zhenming Jiang, Nan Tang, Chen Chen, and Eric Chen, have made my stay at Waterloo truly enjoyable.

I would also like to thank Professor Cheryl Arrowsmith at the University of

Toronto, and Professor David Wishart and Professor Guohui Lin at the University of Alberta who provided me a huge amount of NMR data.

My great thanks go to the University of Waterloo and Cheriton Scholarship which provided me an ideal research environment and also provided me enough funding for my research.

Dedication

This is dedicated to the memory of my father who always encouraged me to achieve my goals, gave me correct directions, and supported me on my decisions. He was not only the best father, but also my best mentor and my best friend.

I would also like to dedicate this dissertation to my mother and my lovely wife, for their unconditional love and spiritual support of my Ph.D. study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

As of April 7th, 2009, there are about 55,000 protein structures solved in the Protein Data Bank (PDB) [19]. About 99.5% of these proteins are experimentally determined by either X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy technique [5]. Among them, about 86.6% of the structures are solved by X-ray crystallography and about 12.9% of the structures are solved by NMR spectroscopy. Although the number of protein structures determined by X-ray crystallography is still dominant, NMR protein structure determination has become extremely significant. The underlying reasons are: (i) NMR is the only physical method that can study dynamics and determine the three-dimensional (3D) structures of proteins in solution; (ii) NMR technique has been well-established, which can determine protein structures to atomic resolution; (iii) even when crystal structures exist, NMR spectral properties are sometimes applied to refine the structures determined by X-ray crystallography [10].

However, NMR protein structure determination in NMR laboratories remains a tremendously costly and laborious process until now. It usually takes an expe-

rienced NMR spectroscopist weeks to months to process the spectra and solve the structure of a target protein after the NMR spectra are collected. With no doubt, high-throughput structural genomics requires parallelizable high-resolution protein structure determination. NMR can be such a technique if its tedious process can be eliminated. Therefore, automating parts of or the entire NMR protein structure determination process with computational methods has become a very hot research area. However, there is still a huge gap between the capability of such computational methods and the requirements of NMR laboratories to apply them in practice.

Protein structure prediction, a very important and big area in bioinformatics, is closely related to the NMR protein structure determination problem. For example, protein structure prediction methods can be used as the structure calculation step in NMR protein structure determination process. Moreover, recent CASP (Critical Assessment of Techniques for Protein Structure Prediction) events have raised three research directions for protein structure prediction, i.e., inter-residue contact prediction, side chain packing, and local quality assessment. All of these three problems are also key steps in the NMR protein structure determination process. Therefore, novel methods and research on the three problems will directly accelerate and improve the NMR protein structure determination process.

In this dissertation, I will focus on developing an integrated fully-automated system, AMR, to determine accurate protein structures after the NMR spectra are collected. AMR contains three major and novel modules, i.e., a peak picking method PICKY, a resonance assignment method IPASS, and a structure calculation method FALCON-NMR. Furthermore, I will propose novel methods to solve the three closely related protein structure prediction problems.

1.2 Contributions

The dissertation proposes a fully-automated NMR protein structure determination protocol, AMR. AMR has three major components:

1. PICKY: PICKY is a novel NMR spectra peak picking method [9]. Picking peaks from experimental NMR spectra is a key unsolved problem for automated NMR protein structure determination. Such a process is a prerequisite for resonance assignment. NOE distance restraint assignment, and structure calculation tasks. Manual or semi-automatic peak picking, which is currently the prominent way used in NMR labs, is tedious, time-consuming, and costly. We develop a novel peak picking method, PICKY. PICKY includes noise level estimation, component forming and sub-division, singular value decomposition (SVD)-based initial peak picking, and peak pruning and refinement. Different from the previous research on peak picking, we provide the first systematic study on peak picking methods. PICKY is tested on 32 real 2D and 3D spectra of eight target proteins, and achieves an average of 88% *recall* and 74% *precision*. This is a joint work with Babak Alipanahi and Emre Karakoc. My contributions mainly focus on developing the component forming and sub-division method, and the peak refinement method; implementing the system; and conducting the experimental studies.

2. IPASS: IPASS is an error-tolerant NMR backbone resonance assignment method [8]. The automation of the entire NMR protein structure determination process requires a superior error-tolerant backbone resonance assignment method. Although a variety of assignment approaches have been developed, none works well on noisy automatically picked peaks. We develop IPASS as a novel integer linear programming (ILP)-based assignment method, that can optimally assign spin systems to the corresponding residues under our problem setup.

In order to reduce the size of the problem, IPASS further employs probabilistic spin system typing based on chemical shifts and secondary structure predictions. The experimental results demonstrate that IPASS significantly outperforms the previous assignment methods on the synthetic data sets. IPASS achieves an average of 99% *precision* and 96% *recall* on the synthesized spin systems, and an average of 96% *precision* and 90% *recall* on the synthesized peak lists. When applied on automatically picked peaks from experimentally derived data sets, it achieves an average *precision* and *recall* of 78% and 67%, respectively. In contrast, the next best method, MARS, achieves an average precision and recall of 50% and 40%, respectively. This is a joint work with Babak Alipanahi, Emre Karakoc, and Frank Balbach. My contributions mainly focus on developing the ILP model for the resonance assignment problem and fragment fixing method to reduce the search space; implementing the ILP module and the fragment fixing module of the system; and conducting the experimental and comparative studies.

3. FALCON-NMR: FALCON-NMR is a chemical shift assignment and NOE constraints-based protein structure determination method. Given the imperfect resonance assignment generated by IPASS, the traditional NMR protein structure generation methods are not able to generate accurate structures. Therefore, we encode our knowledge and previous work on protein structure prediction to improve the accuracy. FALCON [117] was previously developed by our lab as a protein structure prediction package. FALCON-NMR takes the resonance assignment as input. It first tries to find whether there are close homologs of the target protein in the database. If homologs can be found, FALCON-NMR will generate medium-resolution decoy structures by Modeller [158], and then directly call the all-atom refinement module of FALCON to iteratively refine the decoy structures. If no homologs can be found,

FALCON-NMR will call Frazor [116] to generate fragment candidates according to the chemical shift assignment, call the *ab initio* module of FALCON to generate decoy structures, and then call the all-atom refinement module of FALCON to iteratively refine the decoy structures. However, FALCON is not able to identify the best models from the decoy sets. Therefore, we develop an NOE distance constraints-based decoy selection method to select the best decoys from each round of FALCON and feed them back to FALCON to iteratively generate better models. FALCON-NMR has been successfully tested on proteins with sizes under 15kDa. This is a joint work with Shuai Cheng Li, Dongbo Bu, and Guangyu Feng. My contributions mainly focus on developing and implementing the NOE distance constraint-based decoy selection method; and conducting the experimental and comparative studies.

The dissertation also proposes novel ideas and methods to solve the three closely related protein structure prediction problems, i.e., protein inter-residue contact prediction, side chain packing, and local quality assessment.

I propose a novel consensus method to solve contact prediction problem [62]. Protein inter-residue contacts play a crucial role in the determination and prediction of protein structures. Previous studies on contact prediction indicate that although template-based consensus methods outperform sequence-based methods on targets with typical templates, such consensus methods perform poorly on new fold targets. However, I find out that even for new fold targets, the models generated by threading programs can contain many true contacts. The challenge is how to identify them. I develop an integer linear programming model for consensus contact prediction. In contrast to the simple majority voting method assuming that all the individual servers are equally important and independent, the newly developed method evaluates their correlation by using maximum likelihood estimation and extracts independent latent servers from them by using principal component

analysis. An integer linear programming method is then applied to assign a weight to each latent server to maximize the difference between true contacts and false ones. The proposed method is tested on the CASP7 data set. If the top $L/5$ predicted contacts are evaluated where L is the number of residues in the protein, the accuracy is 73%, which is much higher than that of any previously reported study. Moreover, if only the 15 new fold CASP7 targets are considered, our method achieves an accuracy of 37%, which is much better than that of the majority voting method, SVM-LOMETS (the best published consensus method), SVM-SEQ (the best reported study on new fold targets), and SAM-T06 (the best evaluated contact predictor on CASP7). These methods demonstrate an accuracy of 13%, 11%, 26% and 21%, respectively.

I propose a novel classification model to solve side chain packing problem [211]. High-accuracy protein structure modeling demands accurate and very fast side chain prediction since such a procedure must be repeatedly called at each step of structure refinement. Many known side chain prediction programs, such as SCWRL [44] and TreePack [203], depend on the philosophy that global information and pairwise energy function must be used to achieve a high accuracy. These programs are too slow to be used in the case when side chain packing has to be used thousands of times, such as protein structure refinement and protein design. We draw an unexpected conclusion that backbone information can determine side chain conformation accurately. LocalPack, our side chain packing program, which is based on only local backbone information, achieves equal accuracy as SCWRL and TreePack, yet runs 4-14 times faster, hence providing a key missing piece in our efforts to high-accuracy protein structure modeling. This is a joint work with Jing Zhang. My contributions mainly focus on formulating the side chain packing problem as a classification problem; building a multi-class support vector machine (SVM) model to solve the problem; implementing the system; and conducting ex-

perimental and comparative studies.

I propose two novel methods to solve local quality assessment problem [61, 63]. A protein model derived from automated prediction or determination methods is subject to various errors. As methods for structure prediction develop, a continuing problem is how to evaluate the quality of a protein model, especially to identify some well predicted regions of the model, so that the structural biology community can benefit from the automated structure prediction. It is also important to identify badly-predicted regions in a model so that some refinement measurements can be applied to. I develop two complementary techniques, FragQA and PosQA, to accurately predict local quality of a sequence-structure alignment generated by comparative modeling, i.e., homology modeling and threading. FragQA and PosQA predict local quality from two different perspectives. Different from existing methods, FragQA directly predicts RMSD between a continuously aligned fragment determined by an alignment and the corresponding fragment in the native structure, while PosQA predicts the quality of an individual aligned position. Both FragQA and PosQA use an SVM regression method to perform prediction using similar information extracted from a single given alignment. Experimental results demonstrate that FragQA performs well on predicting local fragment quality, and PosQA outperforms two top-notch methods, ProQres and ProQprof.

1.3 Overview of Dissertation

The rest of the dissertation is organized as follows. Chapter 2 introduces the background on NMR protein structure determination process and related protein structure prediction problems. Chapter 3 presents the overview of the fully automated NMR protein structure determination protocol, AMR. Chapter 4 presents the NMR spectra peak picking method, PICKY. Chapter 5 presents the ILP-based resonance

assignment method, IPASS. Chapter 6 presents the structure calculation module of AMR, FALCON-NMR, and evaluates the overall performance of AMR on five test proteins. In Chapter 7, I present the novel ideas and methods to solve three protein structure prediction problems that are closely related to NMR protein structure determination problem. Finally in Chapter 8, I summarize and conclude, and propose future work.

Chapter 2

Background

2.1 Protein Structure

DNA, RNA, and proteins are the three biological sequences that encode the function of life. DNA contains all the genetic information. It can be transcribed into RNA. In turn, RNA is the medium to transport the genetic information from DNA to proteins. It is proteins that play a vital role in keeping our bodies functioning properly.

Proteins are the basic building blocks of life. They form the basis of hormones which regulate metabolism, structures such as hair, wool, and muscle, and antibodies. In the form of enzymes, they are behind most the chemical reactions in the body. Protein structure is essential for correct function because it allows molecular recognition. There are four levels of protein structures. The first level is primary structure, which is also referred as the protein sequence. A protein is a linear chain of amino acids. Different amino acids have common backbone which contains a central carbon atom (C_α), an amino group, and a carboxyl group. Different amino acids have quite different side chain groups. Each amino acid is also called a residue in a protein. The second level is secondary structure. Secondary structure is pri-

marily stabilized by hydrogen bonds. There are three typical classes of secondary structures: alpha helix, beta sheet, and coil. Alpha helix is held in place by hydrogen bonds between backbone oxygen and hydrogen atoms of residue i and $i + 4$, whereas beta sheet is formed by beta strands and stabilized by hydrogen bonds between the beta strands. The third level is tertiary structure, which is also referred as three-dimensional (3D) structure. By understanding the tertiary structure and the folding process, researchers could develop supplemental proteins for people with deficiencies and gain more insight into diseases associated with misfolded proteins. In this dissertation, protein structure always refers to the 3D structure. The fourth level is quaternary structure. Quaternary structure is the arrangement of multiple folded protein molecules in a multi-subunit complex.

2.2 NMR Protein Structure Determination Pipeline

Until now, most of the protein structures are determined by either X-ray crystallography or NMR spectroscopy. However, NMR protein structure determination remains a costly and laborious process. Typically, it takes an experienced spectroscopist weeks to months for a target protein. Currently, most NMR laboratories are following the standard process proposed by Kurt W*ü*thrich [200] in 1986, which contains data collection, data processing, peak picking, resonance assignment, nuclear Overhauser enhancements (NOE) peak assignment, and finally, structure calculation. This process is designed under the basic assumption of NMR structure determination: the 3D structure of the target protein can be uniquely determined if enough proton-proton distance constraints are provided. Therefore, the entire process works in the following manner: (i) peak picking step analyzes the NMR spectra and identifies the important signals; (ii) resonance assignment step extracts chemical shift values and connectivity information from the peak lists

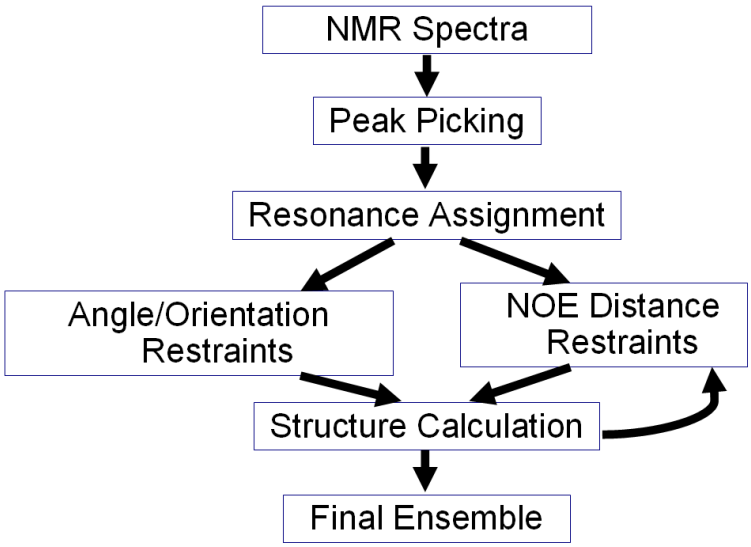


Figure 2.1: Standard process of NMR structure determination

identified from different spectra, combines connectivity information and sequence information together, and assigns those chemical shift values to the corresponding atoms; (iii) NOE peak assignment step identifies peaks from NOE spectra and generates ambiguous distance constraints according to the resonance assignment; and (iv) structure calculation step takes ambiguous distance constraints into consideration, and iteratively generates final structures while simultaneously satisfying as many distance constraints as possible.

Figure 2.1 shows the standard pipeline for NMR-based protein structure determination. The physical principle of NMR behind this is that when active nuclei such as ^{1}H, ^{13}C, and ^{15}N are placed in a strong magnetic field, such nuclei absorb at a frequency that is characteristic of the isotope. Depending on local chemical and geometric environments, different nuclei resonate at different frequencies. The frequencies can then be transformed into a magnetic field-independent term, which is so called chemical shift. Chemical shift is a measure of the dependence of the resonance frequency of the nucleus on its chemical environment, and is commonly indicated in parts per million (ppm) relative to a reference compound.

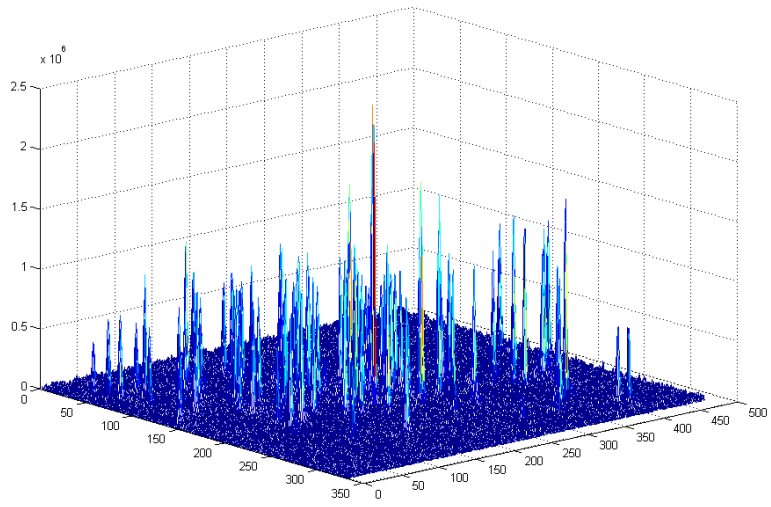


Figure 2.2: An illustration of an ^{15}N-HSQC spectrum.

As shown in Figure 2.1, the first step is to collect NMR spectra data. Data collection is a purely physical step that only depends on the target protein sample and the NMR spectrometer. In this step, NMR spectroscopists prepare the purified and isotope enriched protein samples, put them in the tubes, place the tubes in NMR spectrometers, and collect the output spectra. During the last two decades, the development of new cryogenic probe-heads and high magnetic field spectrometer has significantly reduced data collection time to several days, and simultaneously, hugely improved spectra resolution. After data collection, one can employ Fourier transformation to transfer data into frequency domain. One of the commonly used tools is NMRPipe [41], which is a scripts based NMR spectral processing and analysis package. In this dissertation, by NMR spectra we mean NMR spectra in frequency domain. An NMR spectrum describes the coupling nuclei. A spectrum is stored in the format of a multi-dimensional matrix. The index of each dimension is the discrete chemical shift values of a certain nucleus, and the entries of the matrix are the intensity values. Figure 2.2 shows an ^{15}N-HSQC spectrum.

Apparently, the remaining steps, i.e., peak picking, resonance assignment, NOE

peak assignment, and structure calculation, are the most time-consuming parts in the entire NMR protein structure determination process, which attract huge attention for years [41, 223, 103, 37, 75, 144, 104, 162, 34, 219, 107, 96, 73, 126, 161, 18, 197, 121, 179, 27, 194, 168, 196, 114, 189].

2.2.1 Peak Picking

Automating this entire NMR structure determination process can provide a powerful tool for high-throughput structural genomics, and mitigate costs substantially [195, 74]. Clearly, peak picking is a prerequisite for all the other steps. Peak picking is a well-known "tricky" step in the NMR structure determination process [10]. In a d-dimensional spectrum, a signal, which is often referred to as a "peak", represents a group of d nuclei that can be coupled through bonds (scalar coupling) or through space (spin-spin coupling). In the frequency domain, the coordinate of each dimension of the peak denotes the chemical shift value of the corresponding nucleus. Thus, the task of peak picking is to identify all the signals in an NMR spectrum, such as ^{15}N-HSQC, HNCO, HNCA, CBCA(CO)NH, HNCACB, ^{15}N-edited NOESY, and TOCSY. Peak picking has been investigated for about twenty years. A variety of techniques, such as neural networks [36, 25], Bayesian methods [156, 12], three-way decomposition [144, 104], and spectrum- and peak property-based methods [101, 64, 93, 103], have been developed to identify peaks.

AUTOPSY [103] is one of the most well-known peak picking programs. It differs from previous methods in that not only are the data points around a potential peak taken considered, but also further data points, near the local maximum, are taken into account. Given a spectrum, AUTOPSY first estimates the noise level, which is modeled as the sum of a global base noise and an additional local noise. After

all the data points that have intensities lower than the noise level are removed, AUTOPSY applies a "flood-fill" algorithm to decompose the remaining data points into connected regions. The easily separable peaks are first identified by considering the symmetry and peak shape properties. Lineshapes are then extracted from these peaks. The underlying mathematical assumption is that a well-separable peak shape (a 2D or 3D intensity matrix) can be approximated by the outer product of 1D lineshapes (a 1D intensity vector) times an intensity matrix. For resolving overlapping peaks, AUTOPSY then clusters lineshapes of the separated peaks. In a region with possible overlapping peaks, AUTOPSY tries to interpret this region by a linear combination of all the potential "layers", each of which is constructed from different combinations of lineshapes that overlap with that region. Finally, integration, symmetrization, and filtering modules are called to refine the peak lists.

Later, Orekhov *et al.* [144, 104] proposed a multi-dimensional NMR spectra interpretation method, MUNIN, which can only be applied to 3D or higher-dimensional NMR spectra. The idea of MUNIN is similar to that of AUTOPSY: both are based on the assumption that the spectra can be interpreted by a linear combination of different "layers", each of which is the outer product of 1D lineshapes. However, instead of solving this multi-layer problem in each separated region, which AUTOPSY does, MUNIN deals with the entire spectrum. Thus, each "layer" of the MUNIN method might contain several peaks. Also, it is very likely that several such "layers" are required to describe a single peak. MUNIN has some advantages over AUTOPSY. For example, MUNIN can be applied to frequency-domain or time-domain data, and does not depend on any assumptions about the lineshapes of the peaks. It is worth noticing that MUNIN is not a peak picking method, but it can have a straightforward add-on module for processing the results of decomposition.

Resonance assignment is not the only step in the NMR structure determination process that requires highly accurate peak picking results. The performance of the

NOE peak assignment step also depends on peak picking. NOE peak picking problem is easier than multi-dimensional spectra peak picking, because the resonance assignment information is given as the input for NOE peak picking, which can greatly reduce the chance of picking artifacts. Consequently, NOE peak picking method is usually combined with the iterative NOE peak assignment and structure calculation part. For instance, ATNOS [79] incorporates NOE peak picking and assignment into structure calculation, and refines both sides simultaneously.

However, NMR labs currently do not mainly use any automated peak-picking software. Both AUTOPSY and MUNIN are tested on only one 2D/3D ^{15}N-edited NOESY spectrum in their papers. AUTOPSY cannot be successfully run on any of our experimental spectra by its default parameters, and MUNIN is not publicly available. Regarding all of these impediments: peak picking in the NMR community is accomplished manually, and sometimes semi-automatically with the help of assistant software such as SPARKY [67] and NMRView [93], which can achieve restricted peak picking when the chemical shift values are given. Thus, peak picking is a substantial road block to automated NMR protein structure determination.

2.2.2 Resonance Assignment

The backbone resonance assignment also known as chemical shift assignment plays a vital role in the entire NMR protein structure determination process. Here, the goal is to assign the picked peaks from NMR spectra to their corresponding nuclei of the target protein. Furthermore, backbone resonance assignment acts as an indispensable prerequisite for the NOE assignment. In fact, backbone resonance assignment is the part of the entire NMR process that has attracted the most computational attention for the last ten years [13, 14, 223, 75, 34, 96, 126, 197, 194, 114, 189].

Typically, the backbone resonance assignment is divided into three sub-problems: forming spin systems, linking spin systems into fragments, and mapping the fragments to the target sequence. A "spin system" denotes a group of coupled nuclei that can be observed as cross-peaks in one or more spectra. Usually spin systems contain both inter-residue and intra-residue information. The existing methods can be classified into two groups: assignment methods that require spin systems [34, 96, 126, 114] and assignment methods that do not require spin systems [223, 197, 194, 189]. However, the latter assignment methods always require high quality peak lists with a very small number of missing or false peaks and little difference in the chemical shift of the same nucleus in different spectra. Therefore, for most cases, the experiments carried out in such studies are based on either the manually picked and refined peak lists by spectroscopists, or the synthetic peak lists formed by assigned chemical shifts in a known protein database such as BioMagResBank (BMRB) [163].

Also, according to whether or not an assignment method needs human intervention, existing methods can be classified as "semi-automated" assignment methods [75, 34] or "fully-automated" assignment methods [223, 96, 126, 197, 194, 114, 189]. AUTOASSIGN [223] is a fully-automated multi-stage expert system. The idea of AUTOASSIGN is the best first search, which assigns the strongest fragment matches first, and then gradually relaxes restrictions to assign weaker matches. MAPPER [75] and PACES [34] are semi-automated methods that are also based on the best first search concept. Both of them employ exhaustive search strategy to map the fragments to target proteins. AUTOLINK [126] is an attempt to mimic human logic by a fuzzy logic and relative hypothesis prioritization method. AUTOLINK is the first assignment method that extracts spin system connectivity information from the NOESY data. Wu *et al.* [197] later proposed a weighted maximum independent set formulation for the assignment problem. They provided

a comprehensive summary of the different sources of the spectra errors in the lab experiments, and further simulated these errors on perfect datasets, extracted from BMRB.

MARS [96], one of the widely acknowledged assignment methods, is different from its ancestors in that it applies the consensus idea to multiple runs of assignments, where each run is carried out to optimize different objective functions. For the local assignment, MARS uses the best first search to find the local fit of the fragments, comprising as many as five spin systems. For global assignment, however, MARS optimizes the global pseudo-energy function, which measures how well a spin system matches a residue in the target protein. The pseudo-energy is based on the likelihood of observing a certain chemical shift for an amino acid type in the BMRB database.

Recently, [114] and [189] proposed two sophisticated methods to solve the resonance assignment problem on the most up-to-date NMR spectra. ABACUS [114] takes unassigned peaks from NOESY, COSY (correlation spectroscopy), and TOCSY (total correlation spectroscopy), as well as database-derived likelihoods, as the input. A multi-canonical Monte Carlo procedure, Fragment Monte Carlo (FMC), is used to perform sequence-specific assignments. In MATCH method [189], both the global and local optimization strategies are applied, and 6D APSY spectrum [81, 58] is used as the input.

2.2.3 NOE Peak Assignment and Structure Calculation

As mentioned previously, NOE peak assignment usually combines with structure calculation in an iterative manner, because NOE assignment usually contains too much ambiguity, which can never be completely and correctly eliminated without considering the final 3D structures. Although there are many studies on NOE

peak assignment and structure calculation [142, 120, 69, 70, 78, 219, 73, 107, 71], most of them call either XPLOR [162, 161] or DYANA [78] as structure calculation subroutine. Both XPLOR and DYANA are molecular dynamic methods which apply torsion angle dynamics, simulated annealing, or gradient-based minimization to minimize a target function containing covalent geometry, torsion angle restraints, and non-bonded distance constraints extracted from NMR spectra.

CYANA [73] is a structure calculation program which takes resonance assignment and NOE peak positions and volumes as input, and iteratively applies NOE peak assignment and DYANA molecular dynamic engine. CYANA first generates ambiguous distance constraints based on chemical shift assignment under certain threshold values. A network anchoring technique is then applied to find a self-consistent subset in the constraint network. This is feasible because wrong assignments are usually random, and thus are unlikely to form self-consistent subset. After network anchoring, CYANA re-combines different sets of ambiguous constraints in two ways, which can significantly reduce the number of wrong assignment sets. The newly combined constraints are encoded into the target function and feed to torsion angle dynamic method DYANA to generate structures. Feedbacks are extracted from structures generated and are then applied to improve NOE assignment accuracy, until convergence criteria are satisfied.

However, in practice, it is sometimes very difficult to get almost-complete chemical shift assignment, and thus NOE assignment accuracy will be low because of the missing chemical shift values, especially when aliphatic and aromatic protons are missing. Methods that require high accuracy of assignment may fail on such target proteins. Therefore, various studies are done to overcome this problem [107, 69, 70, 71]. PASD (probabilistic assignment algorithm for automated structure determination) [107] aims to generate protein structures with a high ratio of incorrect NOE assignment. The main difference between PASD and other

iterative methods, such as CYANA, is that results from the successive cycles are not biased by the global fold of structures generated in the preceding cycles. Their experiments show that PASD can tolerate as much as 80% of long-range NOE assignments to be wrong. On the other hand, Grishaev and Llinas [70, 69, 71] attempted to generate protein structures without chemical shift assignment. They tried to identify NOE assignment by forming fragment of spin systems without assigning them.

2.2.4 Automation of NMR Structure Determination Pipeline

With all these efforts on different steps of NMR structure determination process, one can expect that fully automated pipeline is coming of age. In fact, there are very comprehensive surveys that summarize the current bottlenecks, and discuss the feasibility of a fully automated pipeline [136, 10]. An all-encompassing data model has also been proposed by CCPN project [60] to meet researchers' needs.

Recently, significant attention has been paid to the automation of NMR structure determination [219, 27, 168, 196, 121, 179]. Zheng *et al.* [219] combined AutoAssign [223] and AutoStructure [85, 86] together to generate medium accuracy protein backbone structures. Their experiments on some high quality spectra demonstrate that medium-resolution backbone structures (around 3Å to experimentally determined structures) can be acquired. FLYA [121, 179] is proposed as an automated structure determination package, that combines existing programs together, such as AUTOPSY, NMRView, GARANT [13, 14], and CYANA. However, the peak picking step of FLYA requires manually adjusted parameters, and the peak picking subroutine of FLYA demonstrates a quite low performance [121, 179].

Besides the attempts that try to explore the best combinations of existing automated programs, there are pioneer studies along another avenue [27, 168, 196].

The basic assumption of these methods is that chemical shifts carry sufficient information to determine protein structures. Previous studies have shown that chemical shifts are not only highly correlated with hydrogen bonds [190], secondary structures [167], and aromatic properties [26], but can also efficiently guide the selection of structural fragments [37]. Cavalli *et al.* [27] encoded chemical shifts information into the ROSETTA [21] energy function, and calls ROSETTA for *ab initio* structure prediction. In this manner, they avoided the most time-consuming and unreliable step, NOE assignment. CS-Rosetta [168] further encodes chemical shift restraints into more subroutines of structure calculation step, such as chemical shift based fragment selection and chemical shift based decoy selection. Recently, Wishart *et al.* developed CS23D server [196], which also takes chemical shift assignment as input, and applies homology modeling, chemical shift threading, as well as *de novo* structure prediction to generate final structures. CS23D can rapidly converge (around 15 minutes) when close homologs or chemical shift homologous templates can be found in the database.

Although very promising results have been shown by these pioneer studies on accelerating NMR structure determination process, they all have an obvious bottleneck, that is they all require almost complete and perfect data as input, such as manually picked peaks or manually assigned chemical shift. In wet lab experiments, some parts of target proteins, especially long loop regions, will have extremely weak signals in spectra, because they are rapidly moving in solution. Thus, these methods are very likely to fail on these targets. On the other hand, all the studies reviewed here can not efficiently manage peak picking in a good manner, which makes them still far away from fully automating the entire NMR structure determination pipeline.

2.3 Some Relevant Protein Structure Prediction Problems

Although NMR protein structure determination is mainly based on experimental data, sometimes such experimental data is not enough to accurately determine the protein structures due to the experimental errors, such as poor sample quality, missing peaks, and misassignments. Therefore, methods for protein structure prediction are considered as good complements to NMR based protein structure determination. For example, it is common in NMR labs that there are not enough NOE contacts that can be extracted from NOE spectra, which will probably result in the failure of structure calculation step; it is also very difficult to refine the structures to the atomic resolution because side chain chemical shift assignment is much more complicated than the backbone assignment; and more than 90% of the NMR structures are solved based on known structural homologs, which means it is crucial to identify well conserved regions from the homologs.

Meanwhile, existing genome sequencing techniques have led to the identification of millions of proteins. Thus, there is a huge gap between the number of identified protein sequences and the number of solved protein structures. Computational protein structure prediction has made great progress in the last three decades [205, 206]. Given a protein sequence, the goal of protein structure prediction is to predict the tertiary structure of the protein. There are two typical classes of protein structure prediction methods: template-based methods (comparative modeling or threading) and template-free methods (*ab initio* modeling). Template-based methods try to identify homologs from the PDB by either sequence-sequence alignments (comparative modeling) [11] or sequence-structure alignments (threading) [94, 99, 204], whereas template-free methods assemble new structures according to the physical- or statistical-based energy functions [21, 216].

The biennial CASP (Critical Assessment of Structure Prediction) [141, 139, 140, 138, 68, 33, 137] is the most important and objective event for protein structure prediction. CASP results demonstrate that more than 95% of the newly solved proteins have a known structural homolog. Thus, template-based methods should be able to solve most of the new target proteins. However, although template-based methods are capable of generating reasonable predictions for approximately 70% of new proteins, the predictions of such methods are still not good enough for structural biology community use. Therefore, CASP meetings have raised three research directions for protein structure prediction, *i.e.*, inter-residue contact prediction, side chain packing, and local quality assessment. Not surprisingly, all of these three problems have direct and important use in NMR protein structure determination.

2.3.1 Inter-residue Contact Prediction

Protein inter-residue contact prediction is one of the problems being actively studied in the structure prediction community. Recent CASP events have demonstrated that a few true contacts, extracted from template-based models, can provide very important information for protein structure refinement, especially on targets without good templates in PDB. For example, Misura *et al.* [134] revised the widely-used *ab initio* folding program, ROSETTA [31], by incorporating inter-residue contact information as a component of ROSETTA's energy function, and shown that the revised ROSETTA exhibits not only a better computational efficiency, but also a better prediction accuracy. For some test proteins, the models built by this revised ROSETTA are more accurate than their template-based counterparts, which is rarely seen before [137]. Zhang-server [212] and TASSER [214] perform very well in both CASP7 and CASP8. One of the major advantages of these two programs over the others is that both depend on contacts and distance restraints, extracted

from multiple templates, to refine the template-based models. It has been shown by Zhang *et al.* that *ab initio* prediction methods can benefit from contact predictions with an accuracy that is higher than 22% [215].

Protein inter-residue contact was first studied by [135, 174, 72, 84] to calculate the mean force potential. Göbel *et al.* [66] formally proposed the problem of contact prediction, and showed that correlated mutation (CM) is useful information to predict inter-residue contacts. The fundamental assumption is that if two residues are in contact with each other, during evolution, if one residue mutates, the other one has a high chance to mutate as well. Thus, by analyzing residue mutation information from multiple sequence alignments, it can be predicted whether or not two residues are in contact. Since then, different correlated mutation statistical methods have been carefully examined [170, 181, 182, 143, 76, 106].

According to whether structural templates information is taken into consideration, contact prediction methods can be classified into two categories: sequence-based methods and template-based methods. Among all the sequence-based methods, some rely solely on correlated mutation information calculated by different statistical approaches [66, 143, 76, 106], while others encode the correlated mutation, together with other features such as secondary structure and solvent accessibility, into machine learning models [54, 53, 172, 150, 218, 77, 152, 199]. Although the correlated mutation performs well on local contact prediction, which is usually defined to be two residues within six amino acids from each other in the protein sequence, it usually fails for non-local contacts. Therefore, other information such as evolutionary information and secondary structure information, has been applied to improve the performance of contact prediction methods [54, 53, 172, 150, 218, 77, 152, 164, 199]. In [54], Fariselli *et al.* encoded four types of features into a neural network based server (CORNET): 1) correlated mutation, 2) evolutionary information, 3) sequence conservation, and 4) predicted secondary structure.

They defined that two residues are in contact if the Euclidean distance between the coordinates of their C_β atoms (C_α atom for Glycine) is smaller than 8Å, and the sequence separation between the two residues is at least seven to eliminate the influence of local alpha-helical contacts. CORNET achieves an average accuracy of 21%. Other features have been investigated since then [172, 218]. PROFcon [152], one of the best three contact prediction servers in CASP6 [68], encodes more information into its neural network model, including solvent accessibility and secondary structure over the regions between the two residues, as well as the properties of the entire protein. PROFcon performs impressively on small proteins or alpha/beta proteins with an accuracy of more than 30%. Recently, Shackelford and Karplus [164] proposed a neural network based method to calculate the correlated mutation by using the statistical significance of the mutual information between the columns of multiple sequence alignment. Their SAM-T06 server outperforms all the other contact prediction servers in CASP7, and achieves an average accuracy of 45% for all CASP7 target proteins, which is higher than that of any previously reported study.

In contrast to these sequence-based methods, which encode correlated mutation information and other sequence-derived information, there are some studies on predicting inter-residue contacts from structural templates [134, 209, 166, 198, 199]. The underlying assumption for such methods is that contacts are usually very conserved during evolution. Consequently, templates, with structures similar to that of the target protein, usually contain common contacts, such that consensus methods work well. Bystroff *et al.* [209, 166] considered folding pathways, and predicted contacts by employing HMMSTR [23], a hidden Markov model for local sequence-structure correlation. LOMETS [198], a majority voting based consensus method, takes nine state-of-the-art threading programs as inputs. LOMETS predicts contacts by attempting to select the best input model.

Recently, two Support Vector Machines (SVMs) based contact prediction methods, SVM-SEQ and SVM-LOMETS, are proposed by Wu *et al.* [199]. SVM-SEQ only takes sequence-derived information into consideration, whereas SVM-LOMETS, a consensus method, is based on structural templates. SVM-LOMETS differs from its ancestor, LOMETS, in that it carefully trains contact frequency, C_α distances, and template quality by an SVM model. The inputs for SVM-LOMETS are nine state-of-the-art threading programs: FUGUE [169], HHSEARCH [175], PAINT, PPA-I, PPA-II [198], PROSPECT2 [207], SAM-T02 [98], SP3 and SPARKS2 [222]. Both SVM-SEQ and SVM-LOMETS are tested on a set of 554 proteins, on which they achieve an average accuracy of 29% and 53%, respectively. Although it is widely acknowledged that a method usually has different performance on different data sets, one can still expect that a consensus contact prediction method will outperform the individual servers. Instead of testing on the entire CASP7 data set, these two programs are further tested on the 15 new fold (NF) targets of CASP7. The average accuracies are 26% and 11%, respectively. Through a comprehensive comparison of sequence-based and structure-based methods, including SVM-SEQ, SVMCON [89], SVM-LOMETS, LOMETS, and SAM-T06 server, Wu *et al.* concluded that template-based methods are better than sequence-based methods on template-based modeling (TBM) targets, but worse on new fold targets.

2.3.2 Side Chain Prediction

Protein side chain packing is a key step towards accurate protein structure modeling and has been studied for three decades [91, 20, 128, 178]. Given the backbone conformation of a protein, side chain prediction determines the coordinates of all the side chain atoms. Accurate and very fast side chain prediction is vital to accurate protein structure modeling since such a procedure needs to be repeatedly called at each step of the entire protein structure refinement process, which usually sam-

ples a very large number of backbone conformations. Protein side chain packing is also an indispensable component of protein design, which finds a protein sequence that can fold into a given three-dimensional protein structure [42, 39]. Whenever a protein backbone conformation (in protein structure modeling) or its primary sequence (in protein design) is changed, side chain packing has to be conducted to re-determine the coordinates of the affected side chain atoms or even all the side chain atoms. Many known side chain prediction programs, such as SCWRL [44] and TreePack [203], predict the positions of side chain atoms using global information and pairwise energy function, in order to achieve high accuracy. Thus these programs are too slow to be called tens of thousands of times in high-accuracy protein structure modeling or protein design. Therefore, an ultra-fast side chain prediction method is urgently needed.

An important discovery on side chain conformation is that the side chains have a few frequently occurred conformations (referred to as rotamers) [91, 128, 44, 45, 201]. Thus, most side chain prediction methods assume side chains can only take several highly probable rotamers, while others consider conformations sampled around rotamers.

Problem Description Given a finite set of side chain rotamers for each amino acid type, and a backbone conformation b. Let p denote a possible side chain conformation vector indicating the rotamer choice for each residue position. Traditional side chain prediction problem can be formulated as a combinatorial search problem:

$$p^* = \arg\min_p [E_{SS}(p,p) + E_{SB}(p,b) + E_{BB}(b,b)] \tag{2.1}$$

where p^* denotes the optimal side chain conformation, $E_{SS}(p,p)$ is a pairwise energy item representing interactions among side chain atoms, $E_{SB}(p,b)$ represents interaction energy between side chain atoms and backbone atoms, and $E_{BB}(b,b)$ represents backbone-backbone interaction energy. Among them, $E_{BB}(b,b)$ can be considered as a constant since the backbone conformation is fixed.

Following this formulation, almost all the side chain prediction methods employ a pairwise energy function and a rotamer library, then apply a global or local search strategy to find the optimal solution for this combinatorial problem.

Rotamer Libraries A rotamer library is a finite set of rotamers, each of which has an occurring probability. Rotamer libraries can be either backbone-independent [20, 28, 15, 151, 102, 124] or backbone-dependent [91, 128, 44, 45, 46, 160, 47], according to whether the occurring probability of a rotamer is estimated based on backbone information. Chandrasekaran *et al.* developed the first backbone-independent library [28]. Janin *et al.* [91] and McGregor *et al.* [128] examined the relationship between side chain conformation and secondary structure, and then developed a secondary-structure-dependent rotamer library. Dunbrack *et al.* developed the first backbone dihedral angle based rotamer library [46] and refined it by Bayesian statistical analysis [45].

Backbone-dependent rotamer library is widely used to predict side chain conformations [203, 119, 24, 147, 29, 100, 90, 208]. Rotamer library not only can make side chain prediction a discrete-optimization problem, but can also provide the probability of each rotamer in energy function calculation. However, since many side chain prediction methods use rotamer probabilities in their energy functions, their performance is sensitive to these values which are hard to be estimated accurately.

Energy Functions The energy function is considered to be a bottleneck of the existing side chain prediction methods. Although many studies aim to improve the accuracy of side chain packing energy functions [119, 208, 155, 176, 133], all side chain predictors claim that their methods can perform much better if the energy function is more accurate. As mentioned above, energy functions used in side chain prediction contain both side chain-backbone interaction energy and side chain-side chain interaction energy.

Roitberg *et al.* [155] used a mean field approximation, which probably has the same global minimum as the original system, to direct their search strategy. A much more accurate energy function was developed by Liang *et al* [119]. Their energy function contains contact surface, volume overlap, backbone dependency, electrostatic interactions, and desolvation energy. In [208], ROSETTA's energy function [154], which is the sum of Lennard-Jones potential, rotamer energy, atomic clash penalty, and hydrogen-bonding potential, was improved by the tree-reweighted belief propagation (TRBP) technique.

Search Methods A large number of search methods have been developed to optimize the energy function and find the side chain conformation with the minimum energy, such as Metropolis Monte Carlo [82], Gibbs sampling Monte Carlo [188], genetic algorithm [187], dead-end elimination (DEE) [124, 43], neural networks [88], simulated annealing [88, 112], graph theory methods [203, 24], semidefinite programming [29], and integer linear programming [100, 50].

Besides the energy function, search strategy is another bottleneck for side chain prediction. The side chain prediction problem has been proved to be NP-hard [7, 149] if pairwise or multi-body energy function is used. Heuristics such as Monte Carlo or genetic algorithm can find local minimum of an energy function relatively quickly, but cannot guarantee to find the optimal solution of the energy function.

On the other hand, some global search methods can find the global optimal solution at the cost of running time. For example, the widely-used program, SCWRL3.0 [24], can optimize its energy function to its global optimum by first decomposing a protein backbone structure into some substructures and then employing a divide-and-conquer strategy to determine the positions of side chain atoms. SCWRL is not fast enough to be used for iterative refinements and protein design. Another global search method, TreePack [203], achieves similar accuracy as SCWRL3.0, but runs much faster. In contrast to SCWRL, TreePack can decompose a protein structure into much smaller substructures without losing accuracy, and thus reduce running time dramatically. However, both SCWRL and TreePack are likely to fail in the case when the backbone conformation implies heavy steric atomic clashes and thus cannot be cut into small substructures without losing accuracy.

2.3.3 Local Quality Prediction

The biennial CASP events have demonstrated that the three-dimensional structures of many new target proteins can be predicted at a reasonable resolution, although in most cases, the predicted models are still not accurate enough for functional study. In particular, comparative modeling methods can generate reasonably good models for approximately 70% of target proteins in recent CASP events. Even for those free modeling (FM) targets, a structural model generated by protein threading usually contains some good local regions, although the overall conformation of the model is incorrect [216].

As methods for structure prediction develop, a continuing problem is how to evaluate the quality of a protein model in detail. The challenge is to distinguish a good model from a bad one (as referred to global quality assessment), as well as correctly-predicted residues from badly-predicted ones (as referred to local quality

assessment). To make automated structure prediction really useful for the structural biology community, a reliable model quality evaluation program is indispensable when hundreds of models are predicted for a single target protein.

Global quality prediction has been an active research topic for two decades [109, 6, 145, 132, 159, 111, 148, 171, 122, 125, 56, 57, 220, 65, 127, 191, 17, 113, 22, 217, 221, 222, 202, 52, 177, 183, 193, 157, 129]. This kind of programs can be used to pick up the best few from a bunch of models generated by different structure prediction programs, which enables structure biologists to focus on the most native-like models. However, a structural model is not able to provide enough information for functional study if it is bad quality [192].

A common practice taken by some human predictors or consensus-based automatic predictors to further improve the accuracy of the structure prediction is to identify correctly-predicted regions from each structural model and then assemble them together to obtain a better overall model for the target protein; for example, TASSER [216] and 3D-SHOTGUN [59] are two such top-ranked methods in CASPs. This kind of refinement method often performs better than the classical threading-based protein structure prediction methods. The key factor underlying the success of these refinement methods is identifying the correctly-predicted regions in a structural model. Besides being used to examine and improve the accuracy of a protein model, local quality prediction methods can also be used to recognize functional residues in a protein model [184, 16].

Local quality assessment methods are either structure-based [123, 173, 35, 48, 193, 129, 146] or alignment-based [184, 192, 55, 61, 153]. ERRAT [35] is a program that uses only structural information. This program employs a Gaussian error function based on the statistics of non-bonded interactions to predict incorrect regions in a protein model. Such methods can recognize incorrect structural regions which obviously deviate from their native structure. There are also programs using

alignment information to predict local quality. Tress *et al.* developed a method to evaluate local quality of a given alignment and tested the method on alignments generated by five comparative modeling methods [184]. The results indicate that an alignment position with a high profile-derived alignment score often has good quality. Wallner *et al.* developed four neural network-based methods, *i.e.*, ProQres, ProQprof, ProQlocal and Pcons-local, to identify correct regions in a protein model, using either structural information or alignment information [192]. ProQres uses structural information in a protein model; while ProQprof uses alignment information such as profile-profile scores, information scores, and gap penalty. ProQlocal combines ProQres and ProQprof together to achieve better performance. Pcons-local is a consensus-based local quality predictor, taking as input protein models generated by different structure prediction programs. These four methods evaluate local quality by comparing the sequence alignments used to build the models with the optimal structure alignments. However, to make local quality assessment methods really useful for structure prediction and refinement approaches, it is crucial to assess the real quality of regions of the structural models. Meanwhile, it is also important to evaluate the single residue position quality, so that local refinement strategies can be applied as well.

Chapter 3

Overview of AMR

Currently, the NMR protein structure determination process is done manually in NMR labs, even with the help of available programs. However, all of such programs require high-quality input data. Thus, they are not good enough for the purpose of fully automating the entire NMR process. To solve this problem and accelerate the NMR protein structure determination process, we believe all these steps are inter-related with each other, and should only be considered as a whole. Therefore, we develop a fully automated NMR protein structure determination protocol, AMR, which is short for automated NMR protocol. AMR fully automatically generates accurate final structures for the target proteins after the NMR spectra are collected.

Figure 3.1 shows the flowchart of AMR. Given the input NMR spectra, AMR first calls PICKY to automatically pick peaks from the spectra. The peak lists of PICKY are then given to IPASS for resonance assignment. After that, FALCON-NMR is developed to take the resonance assignment and iteratively generate final structures.

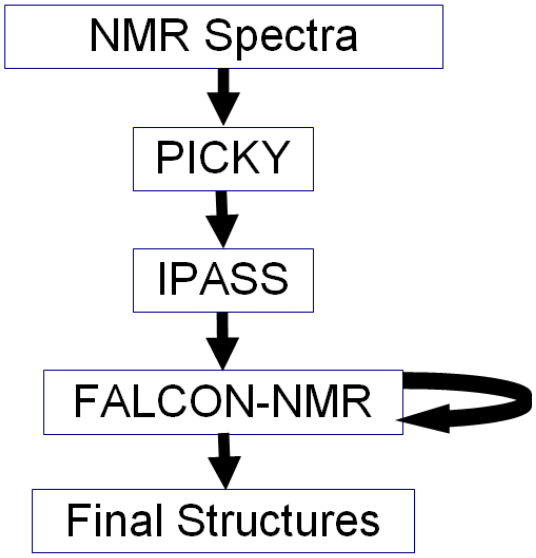


Figure 3.1: Flowchart of AMR

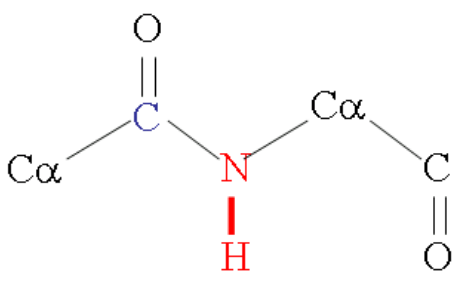
3.1 Input NMR Spectra for AMR

One goal of AMR is to use only a minimum set of the most commonly used NMR spectra to determine the final protein structures. At the current stage, AMR requires six NMR spectra as input: ^{15}N-HSQC, HNCO/HNCA, CBCA(CO)NH, HNCACB, HCCONH-TOCSY, and ^{15}N-edited NOESY. Among them, ^{15}N-HSQC is a 2D spectrum, while the others are 3D spectra. All the input spectra of AMR are through-bond spectra, except ^{15}N-edited NOESY, which is a through-space spectrum.

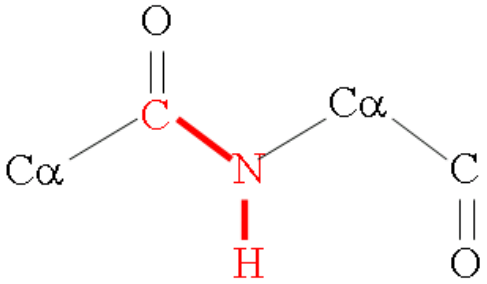
Figure 3.2 shows the illustration of the coupling nuclei that can be detected by the six through-bond spectra. ^{15}N-HSQC detects the coupling between the backbone nitrogen atom and the hydrogen atom that is attached to this nitrogen. Thus, in ideal case, there should be only one peak corresponding to one residue in ^{15}N-HSQC. HNCO detects the coupling of the backbone nitrogen atom, the hydrogen atom that is attached to this nitrogen, and the carbon atom of the carboxyl group of the previous residue. In ideal case, there should be only one peak corresponding to

one residue in HNCO. HNCA detects the coupling of the backbone nitrogen atom, the hydrogen atom that is attached to this nitrogen, and the carbon alpha atoms (if any) of both this residue and the previous residue. In ideal case, there should be two peaks corresponding to one residue in HNCA. HNCACB detects the coupling of the backbone nitrogen atom, the hydrogen atom that is attached to this nitrogen, and the carbon alpha and the carbon beta atoms (if any) of both this residue and the previous residue. In ideal case, there should be four peaks corresponding to one residue in HNCACB. CBCA(CO)NH detects the coupling of the backbone nitrogen atom, the hydrogen atom that is attached to this nitrogen, and the carbon alpha and the carbon beta atoms (if any) of the previous residue. In ideal case, there should be two peaks corresponding to one residue in CBCA(CO)NH. HCCONH-TOCSY detects the coupling of the backbone nitrogen atom, the hydrogen atom that is attached to this nitrogen, and the hydrogen atoms of the previous residue. These six through-bond spectra are mainly used for resonance assignment, which is the indispensable step for NOE assignment and structure calculation.

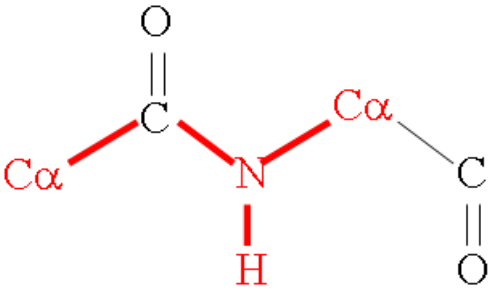
Other than the through-bond spectra, AMR also requires ^{15}N-edited NOESY as an input. As shown in Figure 3.3, ^{15}N-edited NOESY is a through-space spectrum. It detects the coupling of the backbone nitrogen atom, the hydrogen atom that is attached to this nitrogen, and any other hydrogen atoms that are close to this hydrogen atom in Euclidean space.



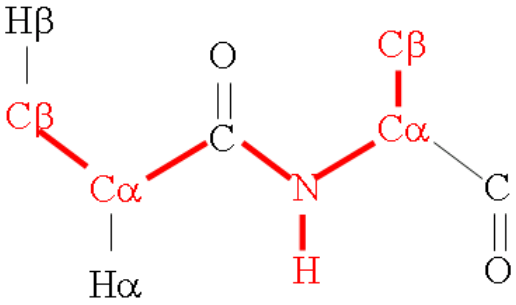
(a) ^{15}N-HSQC



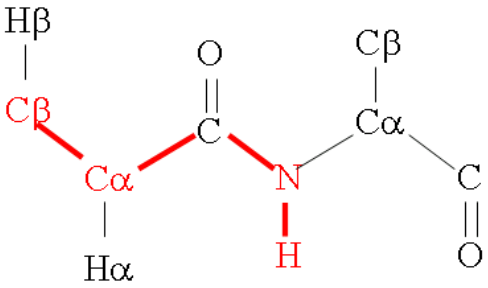
(b) HNCO



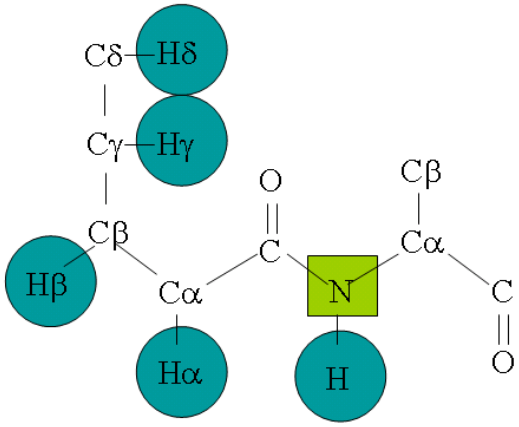
(c) HNCA



(d) HNCACB



(e) CBCA(CO)NH



(f) HCCONH-TOCSY

Figure 3.2: The six input through-bond spectra: ^{15}N-HSQC, HNCO/HNCA, CBCA(CO)NH, HNCACB, and HCCONH-TOCSY. Note that AMR only requires one of HNCO and HNCA as input.

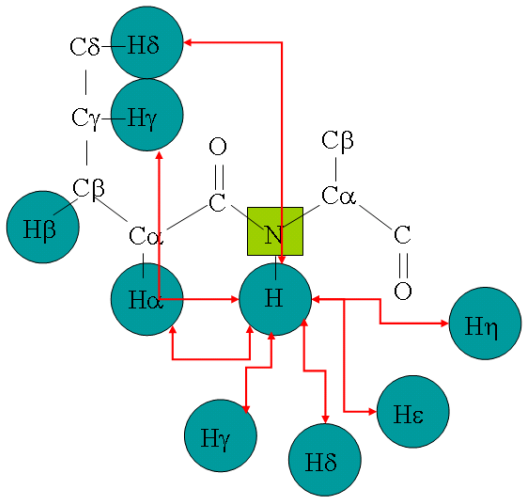


Figure 3.3: The input through-space spectrum of AMR: ^{15}N-edited NOESY

Chapter 4

Peak Picking of NMR Spectra

Peak picking is a crucial step in the entire NMR protein structure determination process. However, existing peak picking methods suffer from two bottlenecks, *i.e.*, high false positive rates and slow speed. We develop a novel peak picking method, PICKY. PICKY adapts a noise level estimation method to efficiently estimate the noise. A component forming algorithm is then developed to divide the spectra into very small and simple components. Then, we find out that singular value decomposition (SVD) can be naturally employed to decompose such small and simple components and get the initial peak lists. Finally, a novel multi-stage refinement procedure is applied to refine the initial peak lists.

4.1 Methods

4.1.1 Method Outline

PICKY consists of four sequential steps:

- **Noise level estimation:** The noise is assumed to be Gaussian and uniform. By estimating an accurate value for the noise level, most of the noisy data points

can be easily filtered out.

- **Component forming and subdivision:** After the elimination of the noisy points, a spectrum looks like a set of discrete components. Instead of processing all these components simultaneously, a novel and efficient component forming and subdividing algorithm is developed to form components that are as small and simple as possible.
- **Initial peak picking:** Because all the components are very small and simple, singular value decomposition (SVD) is found to be powerful enough to naturally solve the peak picking problem inside each component. In this step, each component is decomposed into the outer product of a set of lineshapes (equal to the dimension of the spectrum) by SVD. Then these lineshapes are searched for local maxima, *i.e.*, peaks.
- **Peak pruning and refinement:** The initial peak lists generated by SVD still contain many false positive peaks, yet they miss many true peaks. A powerful and intelligent multi-stage peak refinement step is developed to further refine the initial peak lists. This refinement step can significantly increase the peak picking accuracy. It can also be directly applied to refine the peak lists of any other peak picking methods.

4.1.2 Noise Level Estimation

There are several sources of noise in NMR spectra, including measurement noise and spectral artifacts, such as phase twisting and water bands. For example, water bands affect only a small part of the spectra. Thus, Koradi *et al.* considered a local noise in AUTOPSY [103]. The noise level is estimated within a small region of the entire spectrum. This results in the estimated noise level being much smaller than the actual noise level. Here, a uniform Gaussian noise throughout the spectrum is considered. The only information available is the observed intensity s_i, which can

be written as

$$s_i = t_i + \eta_i, \tag{4.1}$$

where t_i represents the actual intensity, and $\eta_i \sim N(0, \sigma_\eta^2)$ represents the i.i.d white Gaussian noise. The desired noise level is the standard deviation of η_i. However, it is not possible to directly compute the standard deviation of η_i. Instead, a term called observed intensity estimation error is introduced as follows

$$\hat{\eta}_i = s_i - \tfrac{1}{n} \sum_{j \in N_i} s_j, \tag{4.2}$$

where N_i is the set of all the direct neighbors of data point i, *i.e.*, all the points where their indices in all dimensions differ by at most one. In a d-dimensional spectrum, each point has $3^d - 1$ such neighbors. For example, in 2D and 3D spectra, the number of direct neighbors ($n = |N|$) is 8 and 26, respectively. The term $\hat{\eta}_i$ evaluates the error of estimating the observed intensity of a data point by the average of the observed intensities of all of its direct neighbors. By replacing s_i and s_j in Eq. (4.2) by Eq. (4.1), we can get the relationship between $\hat{\eta}$ and η as follows

$$\hat{\eta}_i = \eta_i + \epsilon_i - \tfrac{1}{n} \sum_{j \in N_i} \eta_j, \tag{4.3}$$

where $\epsilon_i = t_i - \frac{1}{n} \sum_{j \in N_i} t_j$, is the actual intensity estimation error, which evaluates the error of estimating the actual intensity of a data point by the average of the actual intensities of all of its direct neighbors.

By Eq. (4.3), the relationship between the variance of $\hat{\eta}$ and that of η can be derived as

$$\sigma_{\hat{\eta}}^2 = \tfrac{n+1}{n} \sigma_\eta^2 + \sigma_\epsilon^2. \tag{4.4}$$

Since the actual intensities are assumed to be much smoother than the observed

intensities, σ_ϵ^2 should be much smaller than $\sigma_{\hat{\eta}}^2$ and σ_η^2. Therefore, the noise level can be estimated by

$$\sigma_\eta = \sqrt{\frac{n}{n+1}}\sigma_{\hat{\eta}}. \tag{4.5}$$

$\sigma_{\hat{\eta}}$ is calculated by a two-round estimation. After $\sigma_{\hat{\eta}}$ is computed over all the original data points, all the $\hat{\eta}_i$ samples are again examined and omitted, if $|\hat{\eta}_i| > \mathrm{O_{TH}} \times \sigma_{\hat{\eta}}$. The outlier threshold, $\mathrm{O_{TH}}$, is set to 5 by default, since only about 0.000029% of the values are expected to be at least five standard deviations away from the mean. Then, an updated $\sigma_{\hat{\eta}}$ is computed, and the noise level σ_η is calculated according to Eq. (4.5).

After the noise level is calculated, all the data points with the absolute values of the observed intensities, less than the noise-threshold ($\mathrm{N_{TH}}$) times the noise level, *i.e.*, ($|s_i| < \mathrm{N_{TH}} \times \sigma_\eta$), are omitted (the intensities are set to 0). If the spectrum is supposed to contain only positive intensities, such as the CBCA(CO)NH spectrum, all the negative points are discarded (the intensities are set to 0).

For more details about the noise level estimation, please refer to [9].

4.1.3 Component Forming and Subdivision

After the noise level estimation step, most of the original data points are eliminated. The spectrum looks like a set of disconnected components. Previous peak picking methods, such as AUTOPSY [103] and MUNIN [144, 104], try to interpret all these components simultaneously. This results in a very slow speed of such methods when applied on the state-of-the-art spectra, which contain hundreds times more data points than previous spectra. Therefore, we develop a novel component forming and subdivision algorithm that can efficiently form very small and simple components. The peak picking step thus can very easily identify peaks from each component separately.

There are three steps for forming components. At the first step, all the connected components are identified by applying a modified version of the flood-fill algorithm in a similar manner as the one used in AUTOPSY [103]. The algorithm iteratively classifies a point as in the same component as its direct neighbors (if at least one of its neighbors has been already assigned), and forms a new component, otherwise. The component forming algorithm generates hundreds of components, especially for 3D components, and many of them contain only a small number of noisy data points which have not been completely eliminated by the noise estimation step. Furthermore, the components that have fewer than $3^d - 1$ points are discarded. Another problem is that some of the components are significantly large. For example, in 2D spectra, such as ^{15}N-HSQC, several overlapping peaks can form a large component. Figure 4.1 shows an example of a region of an ^{15}N-HSQC spectrum after the noise level estimation step. It is clear that some data points in this region are eliminated by the noise level estimation and this region contains several potential peaks, while two of them are highly overlapped and one of them is separate. Figure 4.2 shows the result of this region after the flood-fill algorithm. It can be seen that all the remaining data points are identified to be in the same large component.

The second step further divides the large components into smaller ones. A component is defined to be large if it contains more than one local maxima. A local maximum is defined to be a data point that has the intensity higher than all its first- and second-tier neighbors. The subdivision algorithm is conducted on each large component separately: each local maximum is labeled with a different component index, and then all of its direct neighbors are labeled with the same index and pushed into a priority queue (PQ). PQ is a list of points which are sorted according to their intensities from the highest to the lowest. For the entire algorithm, only the points that have been already assigned with a sub-component

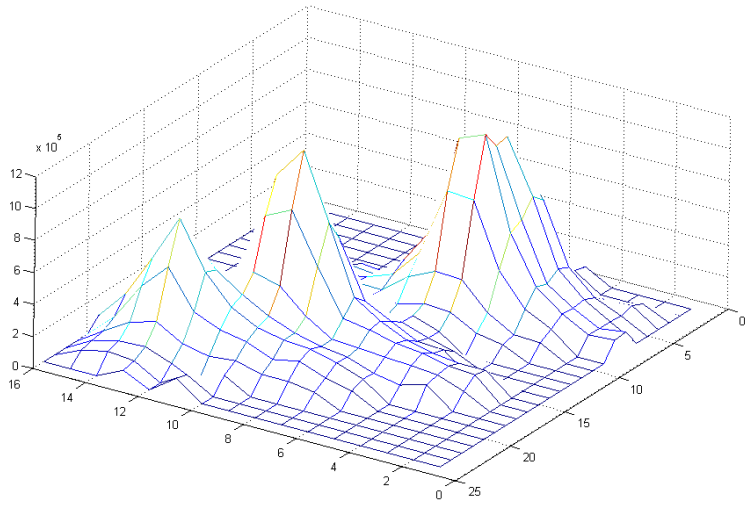


Figure 4.1: A region of an ^{15}N-HSQC spectrum after noise level estimation

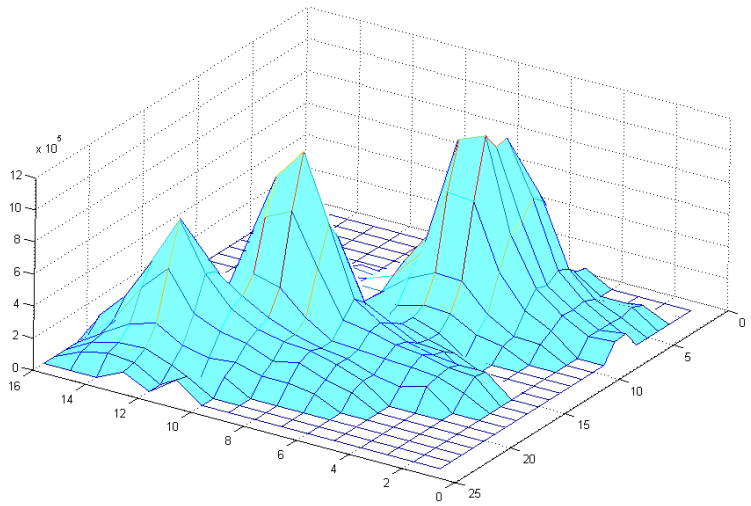


Figure 4.2: Illustration of the result of the flood-fill component forming algorithm

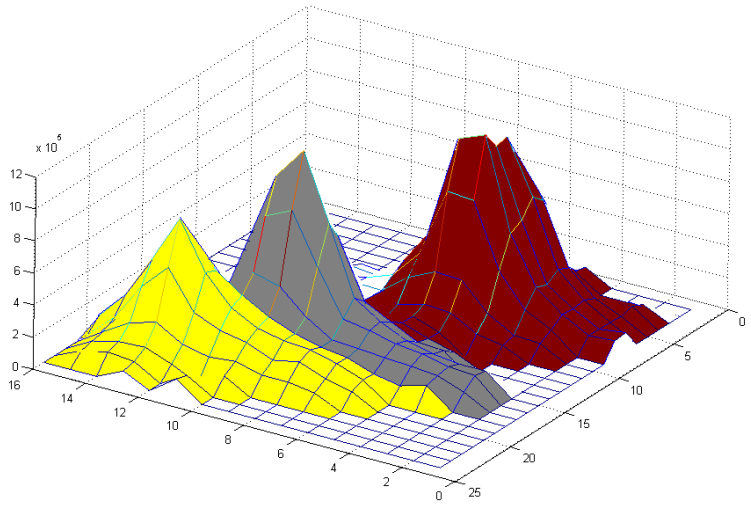


Figure 4.3: Illustration of the result of the component subdivision algorithm

index can be pushed into PQ. According to the definition of local maxima, the distance between any two local maxima is at least two data points, thus there is no conflict in assigning labels to the neighbors at the beginning of the algorithm. Then, each point in PQ is popped out in the order of its intensity. All the neighbors of this point, which have not been assigned any index, are assigned by this point's index, and then pushed into PQ. This process stops until the queue is empty. It is clear that this sub-division algorithm can detect the border of two components within one data point shift from the optimal solution. Figure 4.3 illustrates the result of the subdivision algorithm on the large component shown in Figure 4.2. The original component is divided into three smaller ones.

It can be seen in Figure 4.3 that two small components (the two on the left side of the figure) are highly overlapped with each other. Thus, it is not wise to deal with them separately because otherwise none of them will maintain an almost complete peak shape. Thus, the highly overlapped small components should be merged back again. In AUTOPSY [103], the number of data points within each sub-component is used as the criterion of merging them. An alternative way is to

analyze the points on the border of the two sub-components. If the intensities of those points are negligible, compared with the intensities of the two corresponding local maxima, there is no need to merge again; otherwise, it means the two potential peaks are highly overlapped, and thus, they should merge again. Thus, if the ratio defined in (4.6) is larger than *merge-threshold* ($\mathrm{M_{TH}}$), then the two sub-components merge and a larger sub-component is created.

$$\frac{\max_{k \in B_{i,j}}\{s_k\}}{\min\{m_i, m_j\}} > \mathrm{M_{TH}}, \tag{4.6}$$

where $B_{i,j}$ is the set of points on the border of sub-components i and j, and m_i and m_j are the intensities of the corresponding local maxima, respectively. $\mathrm{M_{TH}}$ is set to 1/2 in PICKY. Different settings are tested by comparing receiver operating characteristic (ROC) curves on a set of six spectra. However, very little difference is observed. Thus, $\mathrm{M_{TH}}$ is set to 1/2 by default, but the users can set different values manually in PICKY. For a large component that contains more than two local maxima, this process is applied on each pair of connected local maxima. Figure 4.4 shows the final result of the component forming and subdivision algorithm on the region shown in Figure 4.1. Each of the resulting components is very small and simple. It either contains a strong and obvious peak, or contains a few number of highly overlapped peaks.

4.1.4 Initial Peak Picking

After the component forming and subdivision step, each component is treated separately to identify the initial peaks. Instead of searching for peaks directly from each component, we assume that each component can be approximated by the outer product of d lineshapes. Each lineshape is a 1D intensity vector. This assumption is also used in [103, 144, 104]. Thus, if a component can be decomposed

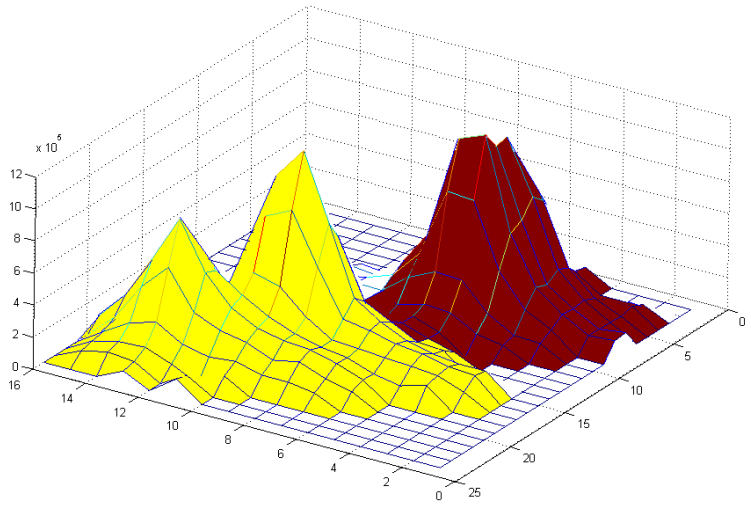


Figure 4.4: Illustration of the final result of the component forming and subdivision algorithm

into lineshapes, these lineshapes can be used to search for peaks. For example, a 2D component, $\mathcal{P} \in \mathbb{R}^{p \times q}$, can be approximated by

$$\mathcal{P} \approx \mathbf{u} \otimes \mathbf{v}, \tag{4.7}$$

where $\mathbf{u} \in \mathbb{R}^{p \times 1}$ and $\mathbf{v} \in \mathbb{R}^{q \times 1}$ are column vectors, called lineshapes, and $\otimes$ denotes the outer product. Similarly, a 3D $\mathcal{P} \in \mathbb{R}^{p \times q \times r}$ component is a tensor that can be expressed as

$$\mathcal{P} \approx \mathbf{u} \otimes \mathbf{v} \otimes \mathbf{w}, \tag{4.8}$$

where $\mathbf{u} \in \mathbb{R}^{p \times 1}$, $\mathbf{v} \in \mathbb{R}^{q \times 1}$, and $\mathbf{w} \in \mathbb{R}^{r \times 1}$ are the column vectors.

We find out that since each component is very small and simple, SVD can be naturally applied to decompose the component into lineshapes. For 2D spectra, standard SVD is applied, and for higher-dimensional spectra, higher-order SVD (HOSVD) is used. More surprisingly, rank-1 approximation is accurate enough for the peak picking problem. That is, a component can be accurately approximated

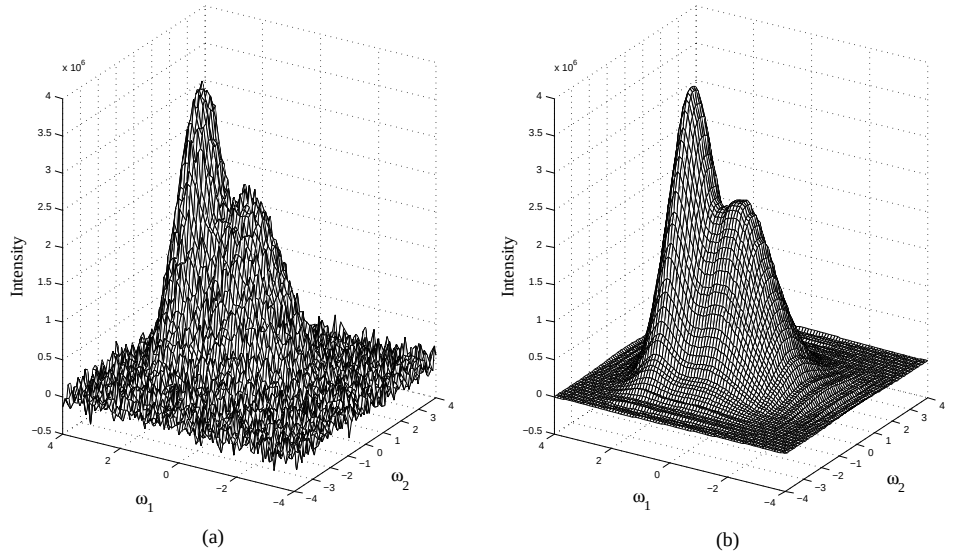


Figure 4.5: Noise reduction using SVD for a 2D component in an ^{15}N-HSQC spectrum: (a) the original component of two highly overlapping peaks, (b) the reconstruction of (a) by the vectors, corresponding to the largest singular value.

by the outer product of the vectors corresponding to the largest singular value in the SVD. The reason is that for such simple components, the largest singular value is larger than 80% of the sum of all the singular values for most cases. For more details about SVD, please refer to [9].

Figure 4.5(b) represents the reconstruction of Figure 4.5(a) by the vectors, corresponding to the largest singular value. It is clear that Figure 4.5(b) is a very good approximation which not only discovers all the potential peaks, but also smooths the original component. Thus, for most cases, a rank-1 approximation results in an accurate approximation. In other words, the lineshapes found by SVD are reliable enough to be searched for the possible locations of the peaks, because the lineshapes demonstrate the inherent characteristics of the component, while reducing the noise.

4.1.5 Peak Refinement

The initial peak lists generated by SVD still contain many false peaks, yet miss many true peaks. Thus, we develop a multi-stage peak refinement method. The initial peak lists are first given to the peak pruning step, which eliminates false peaks that do not satisfy local maximum requirement and discovers new potential peaks. Then, the peaks from different spectra, which share some common nuclei are cross-referenced to further remove the false peaks. Finally, an intensity-based peak filtering method is applied to make sure that a reasonable number of peaks is remained. This refinement method is able to significantly refine the peak lists generated by any peak picking method because it does not depend on the method itself.

Peak Pruning

A peak should be a local maximum, which means it should be larger (or smaller if the component corresponds to a negative-intensity peak) than its first- and second-tier neighbors ($5^d - 1$ points in total). If a peak from the initial peak lists fails to satisfy this local maximum requirement, it is not discarded directly. Instead, the peak location is corrected by a recursive jumping procedure, which keeps jumping to the highest intensity neighbor until the local maximum requirement is satisfied. Thus, an initial peak that fails to satisfy the requirement will either jump to a new peak location that satisfies the requirement or jump to an existing peak. This procedure can not only correct an existing false peak, but can also recover new potential peaks that are not detectable by SVD.

Cross-referencing

Most spectra share common nuclei. For example, ^{15}N-HSQC, HNCO, HNCA, HN-CACB, CBCA(CO)NH, HCCONH-TOCSY, and ^{15}N-edited NOESY all share backbone ^{15}N and ^{1}H attached to ^{15}N. Therefore, a cross-referencing refinement method is developed to further remove the false peaks, if the spectra, sharing some common nuclei, are available. ^{15}N-HSQC, HNCO, and HNCA are considered to be the most sensitive spectra. CBCA(CO)NH and HNCACB are considered to be less sensitive. Thus, if the peak list of HNCO is available, it is used as cross-referencing information to refine the peak list of ^{15}N-HSQC. If the peak list of HNCO is not available, HNCA is used and so on. The goal is to compare the ^{15}N-HSQC peaks with the most sensitive spectrum available. By this cross-referencing refinement, the artifacts and the peaks caused by the side chains of amino acids such as Asparagine and Tryptophan are eliminated. After ^{15}N-HSQC peak list is refined, its peaks are used to, first, compensate for the shifts in $(\mathrm{N}, \mathrm{H^N})$ values of all the NH-based spectra. Then, if the $(\mathrm{N}, \mathrm{H^N})$ value of a peak in these spectra does not correspond to any peak in ^{15}N-HSQC, this peak will be discarded.

Intensity-based Filtering

For most of the through-bond spectra, the number of expected peaks is known. For example, in an CBCA(CO)NH spectrum of a protein with n residues, there should be around $2n - 1$ peaks corresponding to the C_α and C_β nuclei. Therefore, after the peak pruning and cross-referencing steps, all the remaining peaks are sorted according to their intensities. In a spectrum that has N_r expected peaks, the top $K \cdot N_r$ peaks are kept as the final peak list, where K is set to 1.2 in PICKY, because false peaks can be further eliminated in the following assignment step, but missing peaks cannot be recovered in the following step. Here, the confidence score of a

peak is defined as the ratio of its intensity to the estimated noise level. If the number of the expected peaks is unknown for a spectrum, such as in ^{15}N-edited NOESY and HCCONH-TOCSY, the peaks with confidence score below a ceratin threshold (R_{TH}) are discarded. R_{TH} is set to 25 by default.

Therefore, for any input spectrum, PICKY finally outputs a list of a reasonable number of final peaks with confidence scores. Although there are other possible ways to define confidence scores, such as the portion of the peak shape that is overlapped with other peaks, none of them performs better than the intensity-based one in terms of the final precision and recall values.

4.2 Results

4.2.1 Peak Picking Accuracy on Raw Spectra Data

There are two traditional accuracy measures that can objectively evaluate the performance of a peak picking method: the *recall* value or the measure of completeness, the ability to discover true peaks; and the *precision* value or the measure of exactness, the ability to reject false peaks. Assume that in a given spectrum, there are N_r true peaks and a peak picking method picks N_o peaks, where T_p of them are true peaks. Then, *recall* and *precision* are defined as $recall = T_P/N_r$ and $precision = T_P/N_o$, respectively. Apparently, there is a trade-off between *recall* and *precision*. For instance, if the peak picking criteria are loose, the *recall* is high but a large number of false peaks pass through the filter, and result in a low *precision*.

PICKY's performance is evaluated on 32 spectra of eight proteins from Donaldson's lab at York University and Arrowsmith's lab at the University of Toronto. All the data are noisy raw spectra in the frequency domain, taken by NMR spec-

trometers from these two labs. In Table 4.1, the first four proteins, TM1112, YST0336, RP338, and ATC1776, are provided by Arrowsmith's lab, and the other four, COILIN, VRAR, HACS1, and CASKIN, are from Donaldson's lab. Since the peak lists that are manually picked by these experienced spectroscopists are not always available for all these spectra, and it is very common that spectroscopists sometimes do not pick some obvious peaks or fail to pick some highly-overlapped or buried-in-noise peaks, we generate "ideal peak lists" as the "correct answer", based on the final manually assigned chemical shift tables, established by these labs, to fairly compare the PICKY's peaks. For example, for residue i of a target protein, a peak for ^{15}N-HSQC at position $(\mathrm{N_i}, \mathrm{H_i^N})$ and a peak for HNCO at position $(\mathrm{N_i}, \mathrm{H_i^N}, \mathrm{C_{i-1}})$ are generated, where $\mathrm{N_i}$, $\mathrm{H_i^N}$, and $\mathrm{C_{i-1}}$ are experimentally assigned chemical shift values of backbone N and $\mathrm{H^N}$ atoms for residues i, and the chemical shift value of the backbone C atom for residue $i-1$, respectively.

Figure 4.6 illustrates PICKY's performance on the ^{15}N-HSQC spectrum of YST0336. The original spectrum is a challenging one, because it contains a huge and crowded region which contains many potential peaks. After PICKY's noise filtering, only about 2% of the data points remain. PICKY then forms components, picks peaks, and refines these peaks by peak pruning, cross-referencing, and intensity filtering. It can be seen that most of the overlapping peaks are found, whereas some obvious peaks are eliminated in the refinement process (most of which are caused by histidine-tags and side chains).

It is indicated in Table 4.1 that PICKY achieves 100% *recall* on 2 out of the 32 spectra, more than 85% *recall* on 22 out of 32 spectra, while more than 85% *precision* on 6 spectra. The underlying reason for this difference between *recall* and *precision* is that we prefer *recall* to *precision* in the intensity-based filtering step. Note that $1.2N_r$ peaks for a spectrum are retained in the intensity-based

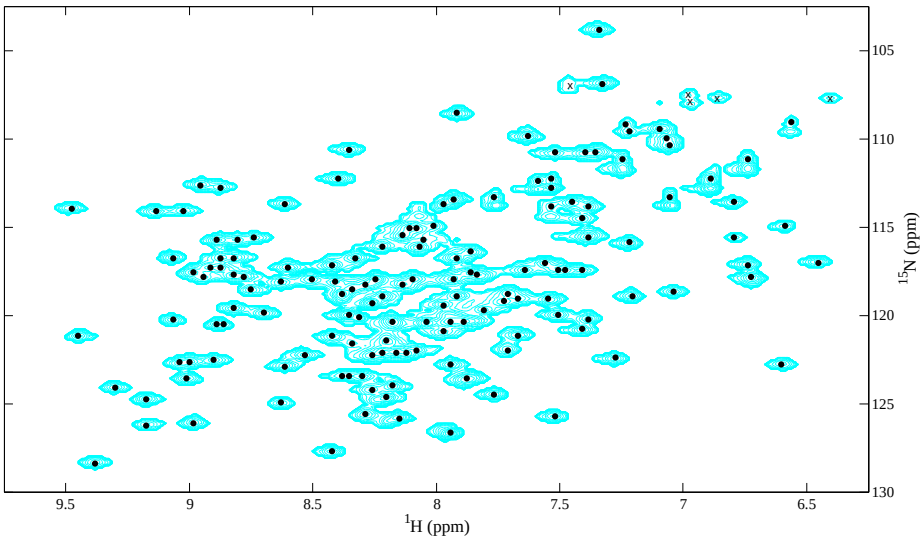


Figure 4.6: Illustration of PICKY's performance on the 2D ^{15}N-HSQC spectrum of YST0336. All the data points with intensities higher than 1.5×10^5, which is automatically determined by PICKY, are set to cyan. Peaks are shown by the black dots. Some strong peaks (shown by crosses), caused by side chains, are filtered by cross-referencing.

filtering step, where N_r is the ideal peak number of this spectrum. Consequently, even if PICKY picks all the true peaks correctly (100% *recall*), *precision* is only $\frac{N_r}{1.2N_r} = 83\%$. Sometimes, the peak pruning and cross-referencing processes can be used to eliminate most of the false peaks, resulting in no more than $1.2N_r$ peaks after the intensity filtering. This explains why we have more than 83% *precision* in some cases. Note here, all the spectra data that are investigated are real data with a high ratio of different sources of noise, artifacts, water bands, and even peaks caused by the histidine-tags attached to the target proteins. Consequently, it is likely that some "expected" peaks in the ideal peak list do not exist in a real spectrum, and some peaks, caused by histidine-tags or side chains, can appear to be strong peaks. Therefore, all the *recall* and *precision* values in Table 4.1 are

actually the lower bounds. A higher accuracy is expected from PICKY in practice. The missing "expected" peaks are also the main reason for the differences of *recall* and *precision* of PICKY on different proteins.

Table 4.1: Performance (*recall*/*precision*) of PICKY on the 32 spectra of the eight target proteins.

Protein	Len	NHSQC	HNCO	HNCA	CBCACONH	HNCACB	*Ave.*
TM1112	89	96 / 89	-	93 / 88	98 / 88	91 / 83	94 / 87
YST0336	146	91 / 84	96 / 79	84 / 79	86 / 69	-	89 / 76
RP3384	64	94 / 86	100 / 82	85 / 70	91 / 76	-	93 / 79
ATC1776	101	78 / 82	89 / 73	79 / 75	78 / 66	-	81 / 74
COILIN	98	97 / 70	97 / 58	-	86 / 54	78 / 54	90 / 59
VRAR	72	87 / 93	89 / 84	-	83 / 71	69 / 72	82 / 80
HACS1	74	95 / 67	94 / 62	-	94 / 61	82 / 52	91 / 61
CASKIN	67	100 / 93	85 / 72	-	91 / 68	70 / 75	86 / 77
Average	-	92 / 83	93 / 73	85 / 78	89 / 69	78 / 67	88 / 74

Len: the length of the protein; NHSQC: ^{15}N-HSQC; CBCACONH: CBCA(CO)NH; *Ave.*: *Average*. All the *recall*/*precision* values are in percentiles.

However, we are not able to make a comparison between PICKY and the previously published peak picking methods. In fact, AUTOPSY is the only automated peak picking program in the literature that is available for public users. Also, AUTOPSY is the most well-known and cited peak picking method. AUTOPSY was tested by using only one 2D-NOESY NMR spectrum and it was shown to be a useful tool for improving the manual peak picking process [103]. We test AUTOPSY with the spectra in our benchmark set. However, AUTOPSY fails to produce peaks by using its default parameters. Thus, the performance of AUTOPSY depends on how to manually set different parameters for different proteins, which is beyond the scope of our knowledge.

Another contribution of this work is to set a comparable benchmark set for automatic peak picking methods. In either AUTOPSY paper [103] or MUNIN paper [144], the demonstrated experiments contain only one spectrum, which is not

publicly available. Thus, it is difficult for other researchers to conduct a fair comparison. Our data set contains 32 spectra, which covers a wide range of commonly used spectra. This data set is publicly available. To the best of our knowledge, this is the first systematic study on the peak picking problem.

4.2.2 Efficiency of PICKY

PICKY is efficient. PICKY is run on a set of 46 spectra (^{15}N-HSQC, HNCO, HNCA, CBCA(CO)NH, HNCACB, ^{15}N-edited NOESY, and HCCONH-TOCSY) derived from the eight proteins. Eight of these spectra are 2D spectra. The remaining spectra are 3D in nature and are subdivided into 30 correlated experiments and 8 NOESY-based experiments. The total time required by PICKY to process these 46 spectra is 721 seconds, which gives an average runtime of 15.7 seconds per spectrum. This indicates that PICKY is very efficient. We also observed that the time required to process individual spectrum is directly related to the resolution of the spectrum.

4.3 Discussion

PICKY mainly differs from previous peak picking methods in the way it interprets the spectra. Both AUTOPSY and MUNIN try to accurately interpret a spectrum by a linear combination of different layers, whereas PICKY efficiently divides the spectrum into small and simple components, and takes advantage of the natural power of SVD, which can inherently find most of the overlapping peaks in such components. Then, a powerful refinement process reveals more peaks, corrects their locations, and significantly reduces false peaks, which makes PICKY very fast and accurate in practice.

PICKY is written in a flexible manner, so that expert experience can be taken as input, and users can easily modify peaks generated by PICKY. Thus, PICKY can hopefully lead to a better interactive strategy for rapid peak picking, *i.e.*, users would very rapidly pick the true peaks and then only have to manually sort through more questionable ones.

PICKY has not been tested on spectra with dimensions higher than three, because such spectra data are not at hand. However, all the four steps of PICKY can be trivially extended to higher dimensions. On the other hand, higher dimensional spectra contain significantly fewer overlapping peaks. Consequently, it can be expected that PICKY will be consistently successful for any spectra.

Chapter 5

Backbone Resonance Assignment

Most of the previously proposed resonance assignment methods are designed to deal with high quality data sets. Therefore, none of these methods work well on the imperfect peak lists generated by automatic peak picking methods. We develop a superior error-tolerant assignment method, IPASS, for automated peak-picking results. IPASS applies Integer Linear Programming (ILP) to optimally solve the assignment problem under our problem setup. Moreover, IPASS contains a new spin system forming step, an improved probabilistic spin system typing step, and a novel connectivity extraction step.

5.1 Methods

5.1.1 Problem Formulation

The resonance assignment problem is to assign the chemical shift values extracted from peaks of different spectra to the corresponding atoms in the protein. Due to the fact that most peaks detect chemical shifts of the nuclei that couple through covalent bonds, peaks are usually used to form spin systems which contain both

inter-residue and intra-residue information, and the spin systems are then assigned to the corresponding residues of the protein.

Define a protein sequence with n residues to be $r_1 r_2 \dots r_n$, let R denote $\{r_1, r_2, \dots, r_n\}$. Define spin system set to be $S = \{\mathbf{s}_1, \dots, \mathbf{s}_m\}$. Each spin system contains N, H^N, C^α and C^β chemical shifts such that

$$\mathbf{s}_\mathrm{j} = (\mathrm{N_j}, \mathrm{H_j^N}, \mathrm{C_j^\alpha}, \mathrm{C_j^\beta}, \tilde{\mathrm{C}}_\mathrm{j}^\alpha, \tilde{\mathrm{C}}_\mathrm{j}^\beta). \tag{5.1}$$

If $\mathbf{s}_\mathrm{j}$ is mapped to residue i, then $\tilde{\mathrm{C}}$ denotes Carbon chemical shifts of residue $i-1$.

The assignment problem is to find the correct mapping between the spin system set and the residue set, expressed as $f : S \to R$. Note that due to the imperfect NMR spectra, peak picking, and spin systems forming, the number of spin systems can be smaller, larger, or equal to the number of residues.

5.1.2 Method Outline

IPASS consists of two essential steps:

• **Spin system forming:** This is a pre-processing step for resonance assignment. During the spin system forming process, the chemical shifts from the peaks are grouped to form spin systems. Spin systems are viewed as the building blocks of the backbone assignment process. A novel clustering-based method is developed to group the peaks of different spectra into spin systems. The input to spin system forming module is the peak lists of ^{15}N-HSQC, HNCA, CBCA(CO)NH, and HNCACB spectra, and the output is a set of spin systems.

• **Integer linear programming:** After the spin systems are formed, the problem is to find the correct mapping between the spin system set and the residue set. We prove this problem is NP-hard. Therefore, we develop an ILP model to solve this

problem. However, sometimes the original problem size is too big for the state-of-the-art ILP solvers. Thus, additional information, such as chemical shift statistics and spin system connectivity information, is used to reduce the ILP problem size. Finally, the ILP model can be solved very fast to find the globally optimal resolution under our problem setup. The two helping steps are described as follows:

Spin system typing: Since the chemical shift of a nucleus only depends on its local chemical and local geometric environment, atoms of different amino acids and different secondary structures have quite different chemical shift distributions. Such distributions can be acquired by statistics from the known chemical shift database. A probabilistic model is then derived to estimate how likely a given spin system can be mapped to a certain residue.

Connectivity information extraction: Two spin systems are connected if they can be mapped to two consecutive residues. The connections are detected by both inter-residue and intra-residue information. Since there are always shifts in the chemical shift values of the same nucleus in different spectra, exactly matched chemical shift values are not expected from peaks of different spectra. Thus, a threshold is needed to define the connectivity. Consequently, a low threshold results in many undetected true connections, whereas a large threshold results in many false connections. In IPASS, two sets of connections are defined: a set of highly reliable connections based on the C^{α} and C^{β} chemical shifts and the information extracted from the ^{15}N-edited NOESY peaks. Furthermore, a set of loose connections are detected by a larger threshold. By using reliable connections, a set of fragments is determined and the combinations of them are enumerated.

5.1.3 Spin System Forming

A highly error-tolerant resonance assignment method requires a highly error-tolerant spin system forming step. The NMR spectra used in the spin system forming step are 2D ^{15}N-HSQC and triple resonance experiments HNCA, HNCACB, and CBCA(CO)NH. In ideal case, since peaks from ^{15}N-HSQC, HNCA, HNCACB, and CBCA(CO)NH share common backbone N and H^N nuclei, if all these peaks are projected to the 2D $N - H$ space, the peaks from the same spin system should overlap with each other. A simple clustering method can then be easily applied to find these clusters and form spin systems accordingly. However, when deal with real peaks, it is always the case that there are shifts of the chemical shift values of the same nucleus in different spectra. Thus, the traditional clustering method does not work on practical peaks. We develop a two-stage clustering method to form spin systems based on imperfect peaks generated by PICKY.

The problem of forming spin systems is modeled as a graph theory problem. Typically, a shift as high as 0.5 ppm is expected in the ^{15}N and ^{13}C chemical shifts, and a shift as high as 0.05 ppm in the ^{1}H chemical shifts. To solve this problem, each peak is denoted as a point in the multidimensional space, where each dimension corresponds to a certain type of nuclei such as ^{15}N, ^{1}H, or ^{13}C.

At the first stage, the peaks within each 3D spectrum are connected according to their N and H^N chemical shifts. Each spectrum provides multiple peaks for the same residue, and these peaks should be in the small vicinity of each other. Given two peaks with root pairs $\mathrm{P_x} = (\mathrm{N_x}, \mathrm{H_x^N})$ and $\mathrm{P_y} = (\mathrm{N_y}, \mathrm{H_y^N})$, the distance between them is defined as

$$d_{\mathrm{P_x},\mathrm{P_y}} = \sqrt{(\mathrm{N_x} - \mathrm{N_y})^2 + \omega^2(\mathrm{H_x^N} - \mathrm{H_y^N})^2}, \tag{5.2}$$

where ω is the scaling factor for the compensation of the difference in the resolution between ^{1}H and ^{15}N. Usually, ^{1}H chemical shifts are 10 times more sensitive than the ^{15}N chemical shifts, and so the default value of ω is 10. According to the distance defined in Eq. (5.2), each peak, P, in a given spectrum is associated with its nearest neighbor, $\mathrm{P_{NN}}$. An edge is created between P, and any peak that is within $2 \times d_{\mathrm{P,P_{NN}}}$ distance to P. The edges between the peaks are directional with the starting point to be the reference peak, P. The peaks which are connected to each other represent the peaks from the same N and $\mathrm{H^N}$ root.

At the second stage, the peaks from different spectra are connected. Peaks from two 3D spectra are connected according to their N, $\mathrm{H^N}$, and C chemical shifts, whereas a peak from a 2D spectrum is connected with a peak from a 3D spectrum according to N and $\mathrm{H^N}$ chemical shifts. For example, the distance between $\mathrm{P_x} = (\mathrm{N_x}, \mathrm{C_x}, \mathrm{H_x^N})$ in CBCA(CO)NH spectrum and $\mathrm{P_y} = (\mathrm{N_y}, \mathrm{C_y}, \mathrm{H_y^N})$ in HNCA is defined as

$$D_{\mathrm{P_x,P_y}} = \sqrt{(\mathrm{N_x} - \mathrm{N_y})^2 + (\mathrm{C_x} - \mathrm{C_y})^2 + \omega^2(\mathrm{H_x^N} - \mathrm{H_y^N})^2} \tag{5.3}$$

The distance between $\mathrm{P_x} = (\mathrm{N_x}, \mathrm{C_x}, \mathrm{H_x^N})$ in CBCA(CO)NH spectrum and $\mathrm{P_z} = (\mathrm{N_z}, \mathrm{H_z^N})$ in ^{15}N-HSQC is defined as

$$D_{\mathrm{P_x,P_z}} = \sqrt{(\mathrm{N_x} - \mathrm{N_z})^2 + \omega^2(\mathrm{H_x^N} - \mathrm{H_z^N})^2} \tag{5.4}$$

Similar to the aforementioned process, the edges can be created between P and its close vicinity peaks in other spectra, which are within $2 \times D_{\mathrm{P,P_{NN}}}$ distance to P. All of the created edges are directional. If there are two edges in both directions between two nodes, two edges are replaced by a non-directional edge.

After these two stages, each connected component represents a cluster that

corresponds to a spin system in the resulting general peak graph. The primary advantage of this approach is its generalization. It can be applied to any set of available NMR spectra. After the connected components are found, each cluster contains similar H^N and N values such that these values are taken from the ^{15}N-HSQC spectrum. The problem is how to detect C^α, C^β, $\tilde{C}^\alpha$, and $\tilde{C}^\beta$. The clusters are usually incomplete as a result of the missing peaks, and over-crowded as a result of the overlapping peaks.

A brute force method is applied, which searches all the possible combinations of the chemical shift values for different C^α and C^β nuclei in each cluster. If a unique combination of the chemical shifts exists and does not conflict with the peaks in the cluster, a spin system is generated. After C^α and C^β are identified, $\tilde{C}^\alpha$ and $\tilde{C}^\beta$ can be easily identified.

5.1.4 An ILP Model to Solve the Assignment Problem

After the spin systems are formed, the next step is to assign the spin systems to the corresponding residues. We first prove that the assignment problem is NP-hard, then adopt additional information, such as statistics and connectivity information, to reduce the problem size, and finally propose a novel ILP model for the assignment problem.

The backbone resonance assignment problem can be represented by a graph $G(V, E)$. Here, each node in V corresponds to a mapping between a spin system and a certain residue, and the edges in E represent the connections between the spin systems.

NP-hardness Proof of the Assignment Problem

Initially, any of the m spin systems can be mapped to any of the n residues. Recall that spin systems contain both inter-residue and intra-residue information, thus, there is connectivity information between different spin systems, *i.e.*, if two spin systems i and j are mapped to two consecutive residues, then there should be a connectivity edge from i to j. Figure 5.1 illustrates the graph of the original assignment problem. It is clear that any spin system can be mapped to any residue, and if there is an edge between spin system k and l, there should be $n-1$ duplications of that edge. The goal of the resonance assignment problem is to find the mapping between the spin systems and the residues that results in the largest total weights of the connectivity edges used.

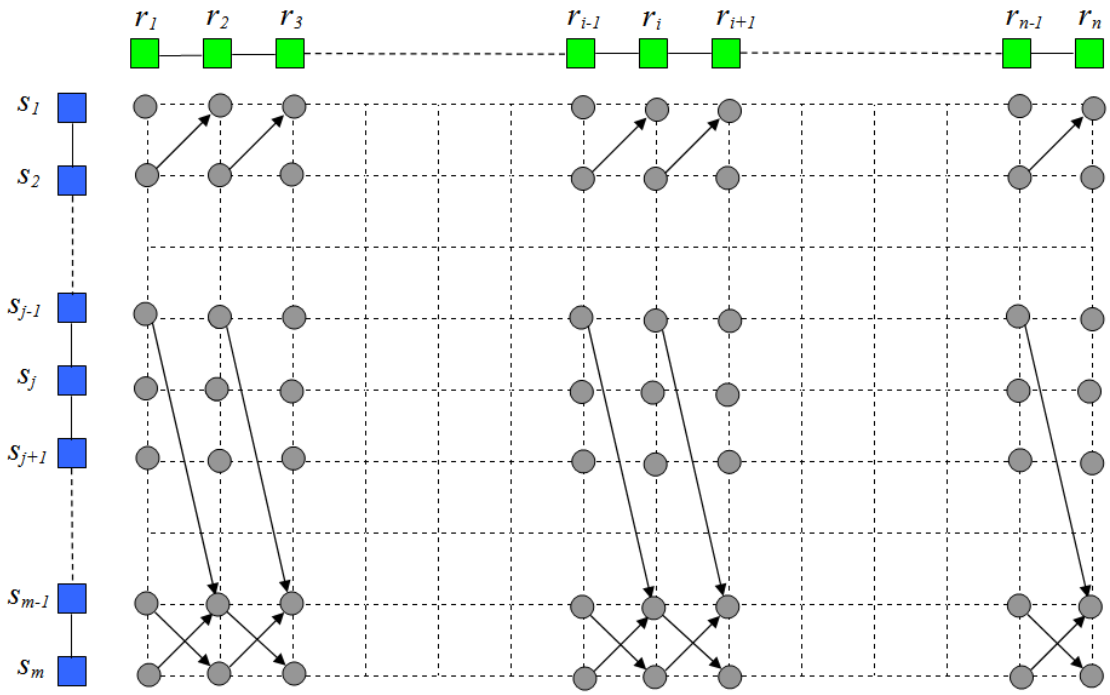


Figure 5.1: Illustration of the original problem setup of the assignment problem.

Theorem 1. *Backbone resonance assignment problem, under the proposed graphical representation is NP-hard.*

Proof. The NP-hardness of the backbone resonance assignment problem is through a reduction from the *Hamiltonian path* problem which is known to be NP-hard. The Hamiltonian path problem is defined as follows: Given a graph, $G'(V', E')$, decide whether there exists a path in $G'(V', E')$ that visits each vertex exactly once. For an instance of the Hamiltonian path problem, a new graph $G''(V'', E'')$, which is a product of the $\{1, 2, ..., n\} \times G'$, where $n = |V'|$, is constructed. Thus the new graph, $G''(V'', E'')$, has nodes (i, v), where $v \in V'$ and $1 \leq i \leq n$, and an edge between (i, v) and (j, w) if $j = i + 1$ and there is an edge between v and w in G', where $w \in V'$ and $1 \leq j \leq n$. Here, the edge weights are defined as 1 for all the edges in G''. Each spin system corresponds to a vertex in G' and the residues correspond to the $\{1, 2, ..., n\}$ set. G'' has a very similar topology as the graph shown in Figure 5.1.

G' has a hamiltonian path, if and only if there exists an assignment solution with total edge weights $n - 1$ for the backbone resonance assignment problem. For each i, the vertices are connected to their adjacent vertices in the graph with weight 1. A backbone resonance assignment solution with total weight $n-1$ corresponds to a mapping, where each spin system is used once, and every residue has a spin system that is mapped to it. As a result, the assignment with total weight $n - 1$ visits each vertex in G' exactly once which corresponds to a Hamiltonian path. Similarly, if there is a Hamiltonian path visiting vertices $v_1, v_2, ..., v_n$, it corresponds to an assignment of mapping spin system v_i to residue i. This assignment has total weight $n - 1$, and is thus an optimal solution for the assignment problem.

Therefore, the resonance assignment problem is NP-hard. □

Spin System Typing

Right now, every spin system can be mapped to every residue. However, this is not the case in practice. Each spin system is a vector of chemical shift values. Chemical shift value of a nucleus only depends on the local chemical and local geometric environment. Therefore, atoms of different amino acids and different secondary structures have quite different chemical shift distributions. The goal of spin system typing is to reduce the number of candidate spin systems for each residue, based on the chemical shift information. A statistical analysis of the deposited chemical shifts in the BMRB database reveals correlation between the chemical shifts, and the amino acid types and secondary structures. These statistics are used to build a probabilistic model to estimate how likely a spin system can be mapped to a certain residue.

Collecting Statistics The statistics is based on the deposited proteins with experimentally assigned chemical shift values in BMRB database. A restriction is that only proteins which has corresponding structure entries in PDB database are considered, because only for these proteins, their secondary structure information is known. There are 1168 proteins left after redundancy elimination at sequence identity level 40%. DSSP [97] is called to determine the secondary structure types for these proteins. Since the N and H^N chemical shifts exhibit similar statistics for all amino acids and all secondary structures, N and H^N chemical shifts are not informative for typing the spin system purpose. Thus, only mean and covariance matrices for C^α and C^β chemical shifts are estimated for each amino acid and secondary structure type.

A Probabilistic Typing Model A probabilistic model is developed to estimate $\Pr\{r_i \mid \mathbf{s}_j\}$. That is, how likely spin system s_j can be mapped to residue r_i. Two

vectors are extracted for spin system $\mathbf{s}_\mathrm{j}$: $\mathbf{c}_\mathrm{j} = (\mathrm{C}_\mathrm{j}^\alpha, \mathrm{C}_\mathrm{j}^\beta)^T$ and $\tilde{\mathbf{c}}_\mathrm{j} = (\tilde{\mathrm{C}}_\mathrm{j}^\alpha, \tilde{\mathrm{C}}_\mathrm{j}^\beta)^T$. Recall that N and H^N chemical shifts are not informative for spin system typing. Thus, $\Pr\{r_i \mid \mathbf{s}_\mathrm{j}\}$ is equal to the probability that $\mathbf{c}_\mathrm{j}$ and $\tilde{\mathbf{c}}_\mathrm{j}$ are mapped to r_i and r_{i-1}, respectively, which is represented by Eq. (5.5).

$$\Pr\{r_i \mid \mathbf{s}_\mathrm{j}\} = \Pr\{r_i = a_p, r_{i-1} = a_q \mid \mathbf{c}_\mathrm{j}, \tilde{\mathbf{c}}_\mathrm{j}\}, \tag{5.5}$$

where $a_p, a_q \in A$, and A is the set of twenty amino acids.

We assume that $\mathbf{c}_\mathrm{j}$ and $\tilde{\mathbf{c}}_\mathrm{j}$ are independent, apply the Bayes' rule, and finally get

$$\Pr\{r_i \mid \mathbf{s}_\mathrm{j}\} = \Pr\{r_i = a_p \mid \mathbf{c}_\mathrm{j}\} \times \Pr\{r_{i-1} = a_q \mid \tilde{\mathbf{c}}_\mathrm{j}\} \tag{5.6}$$

$$= \frac{\Pr\{\mathbf{c}_\mathrm{j} \mid r_i = a_p\}\Pr\{r_i = a_p\}}{\Pr\{\mathbf{c}_\mathrm{j}\}} \times \frac{\Pr\{\tilde{\mathbf{c}}_\mathrm{j} \mid r_{i-1} = a_q\}\Pr\{r_{i-1} = a_q\}}{\Pr\{\tilde{\mathbf{c}}_\mathrm{j}\}}. \tag{5.7}$$

In Eq. (5.7), $\Pr\{r_i = a_p\}$ and $\Pr\{r_{i-1} = a_q\}$ only depend on the amino acid types, but not the positions in the protein sequence. Therefore, they can be easily estimated by the abundance of amino acid a_p and a_q in the entire BMRB database. By using the total probability law,

$$\Pr\{\mathbf{c}_\mathrm{j}\} = \sum_{a_\ell \in A} \Pr\{\mathbf{c}_\mathrm{j} \mid r_i = a_\ell\}\Pr\{r_i = a_\ell\}, \tag{5.8}$$

$$\textit{and} \quad \Pr\{\tilde{\mathbf{c}}_\mathrm{j}\} = \sum_{a_\ell \in A} \Pr\{\tilde{\mathbf{c}}_\mathrm{j} \mid r_{i-1} = a_\ell\}\Pr\{r_{i-1} = a_\ell\}.$$

By using the total probability law again,

$$\Pr\{\mathbf{c}_\mathrm{j} \mid r_i = a_\ell\} = \sum_{k=1}^{3} \Pr\{\mathbf{c}_\mathrm{j} \mid r_i = a_\ell, \gamma_i = \sigma_k\} \Pr\{\gamma_i = \sigma_k\}, \tag{5.9}$$

where γ_i denotes the secondary structure of r_i. For $k = 1, 2$, and 3, σ_k denotes α-helix, β-strand, and random coil, respectively. PSIPRED is used to estimate $\Pr\{\gamma_i = \sigma_k\}$ values [130], whereas $\Pr\{\mathbf{c}_\mathrm{j} \mid r_i = a_\ell, \gamma_i = \sigma_k\}$ is calculated according to the previously extracted chemical shift statistics.

$\Pr\{r_i \mid \mathbf{s}_\mathrm{j}\}$ values are calculated for every spin system and residue pair. For a residue, the spin systems that have lower than 0.05 probabilities of mapping to this residue are eliminated. For more details about the spin system typing, please refer to [8].

Connectivity Information Extraction

Although the spin system typing step is able to significantly reduce the number of candidate spin systems for each residue, the remaining problem size is still very large. Therefore, some highly reliable fragments are assigned and fixed. Since spin systems contain both inter-residue and intra-residue information, connectivity information can be extracted between spin systems. We define two types of connections, reliable connections and loose connections.

Spin system s_j can be reliably followed by spin system s_k if and only if at least two of the following three conditions are satisfied:

1. $|\mathrm{C}^{\alpha}_\mathrm{j} - \tilde{\mathrm{C}}^{\alpha}_\mathrm{k}| \leq \delta_\alpha$,
2. $|\mathrm{C}^{\beta}_\mathrm{j} - \tilde{\mathrm{C}}^{\beta}_\mathrm{k}| \leq \delta_\beta$,
3. $(\mathrm{N}_\mathrm{j}, \mathrm{H}^{\mathrm{N}}_\mathrm{k}, \mathrm{H}^{\mathrm{N}}_\mathrm{j})$ and $(\mathrm{N}_\mathrm{k}, \mathrm{H}^{\mathrm{N}}_\mathrm{j}, \mathrm{H}^{\mathrm{N}}_\mathrm{k})$ peaks exist in the ^{15}N-edited NOESY spectrum,

where $\delta_\alpha = \delta_\beta = 0.05$ ppm. The first two conditions require that two reliably connected spin systems should agree on their shared chemical shift values, while

the third condition requires that if two spin systems are assigned to two consecutive residues on the target protein sequence, their hydrogen atoms of the amide groups should be close in 3D space, providing a peak in the ^{15}N-edited NOESY spectrum.

For the loose connections, we set $\delta_\alpha = \delta_\beta = 0.5$ ppm. Two spin systems $\mathbf{s}_\mathrm{j}$ and $\mathbf{s}_\mathrm{k}$ are loosely connected if they can satisfy one of the first two conditions without violating the other one. Note that the third condition itself is not enough to judge a connection because $\mathrm{H}^\mathrm{N}_\mathrm{j}$ can be from a residue that is far from residue k in the protein sequence, but close in 3D space.

Both the reliable connections and the loose connections are position specific. The reason is that the spin system typing step has significantly reduced the number of the residues that a certain spin system can be mapped to. Thus, a connection between two spin systems s_i and s_j can only occur if s_i can be mapped to a certain residue r_k and s_j can be mapped to r_{k+1}.

After all the reliable connections are determined, they are enumerated for all the possible fragments, *i.e.*, if s_i and s_j is a reliable connection, s_j and s_k is a reliable connection, then there is a fragment that contains s_i, s_j, and s_k. Suppose there are p reliable fragments, $F_1, \ldots, F_p$, with lengths $l_1, \ldots, l_p$, respectively. Each fragment is denoted as $F_q = (\mathbf{s}_{e_1}, \mathbf{s}_{e_2}, \ldots, \mathbf{s}_{e_{l_q}})$, where $\mathbf{s}_{e_j}$ is connected to $\mathbf{s}_{e_{j+1}}$ for $j = 1, \ldots, l_q - 1$. Fragments shorter than three spin systems, or fragments that are contained by other fragments are discarded. Since it is possible that a fragment can have more than one mapping positions in the protein sequence, we define the score of mapping a fragment F_q to the i-th position in the target sequence as

$$T_i^{(q)} = -\sum_{k=1}^{l_q} \log\left(1 - \Pr\left\{r_{i+k-1} \mid \mathbf{s}_{e_k}\right\}\right), \quad 1 \le i \le n - l_q + 1, \tag{5.10}$$

where $\Pr\left\{r_{i+k-1} \mid \mathbf{s}_{e_k}\right\}$ can be calculated by the probabilistic model mentioned before. All the combinations of the reliable fragments are enumerated according

to the requirement that in a combination, any two fragments should not be in conflict, i.e., they should not share any spin systems, and their mapped positions in the sequence should not overlap. Then, all the fragments within the combination are fixed. For example, if in a combination, F_q is mapped to the sequence region starting from the i-th position, then all the spin systems contained in F_q are removed from the candidate spin system sets of residues outside this sequence region. All possible reliable combinations are enumerated, and the following ILP model is built for each combination separately. Finally, the solution of the ILP model with the highest assignment score is selected as the final solution.

Integer Linear Programming Model for the Problem

After typing the spin systems and fixing a combination of reliable fragments, the problem size of the backbone resonance assignment is significantly reduced. That is, for a residue r_i, for any spin system s_j such that $\Pr\{r_i \mid \mathbf{s}_\mathrm{j}\} \neq 0$, there is a node $v_{i,j} \in V$ in G. There is a directional edge from $v_{i,j}$ to $v_{l,k}$ if and only if $l = i+1$ and spin system j can be followed by spin system k by either a reliable connection or a loose connection. The edge between $v_{i,j}$ to $v_{i+1,k}$ is denoted as $e_{i,j,k}$, the weight of which is defined as

$$\begin{aligned} w_{i,j,k} &= \log\left(\Pr\{r_i;\ r_{i+1} \mid \mathbf{s}_\mathrm{j},\ \mathbf{s}_\mathrm{k}\}\right) \\ &= \log\left(\Pr\{r_i \mid \mathbf{s}_\mathrm{j}\}\right) + \log\left(\Pr\{r_{i+1} \mid \mathbf{s}_\mathrm{k}\}\right), \end{aligned} \tag{5.11}$$

where $w_{i,j,k}$ corresponds to the probability of mapping two spin systems to two consecutive residues.

For each node $v_{i,j} \in V$, define a boolean variable $x_{i,j}$ to be 1 if spin system j is mapped to residue i, and 0 otherwise. For each edge $e_{i,j,k} \in E$, define a boolean variable $y_{i,j,k}$ to be 1 if the connection between node $v_{i,j}$ and $v_{i+1,k}$ is selected, and

0 otherwise.

Figure 5.2 shows an illustration of the assignment problem setup. Note that a spin system can still be mapped to multiple residues with different probabilities which are calculated in spin system typing step, and for each residue, there are only a few spin system candidates left which can be possibly mapped to it.

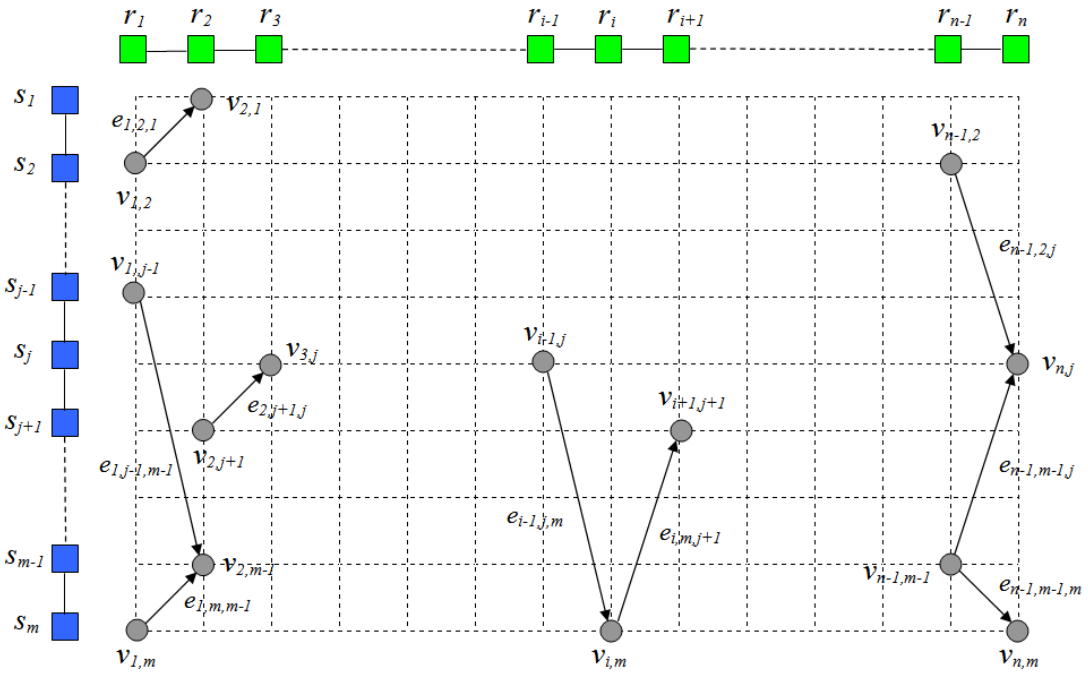


Figure 5.2: Illustration of the problem setup of the assignment problem. There is a node $v_{i,j}$ (shown by gray circles) corresponding to residue r_i and $\mathbf{s}_\mathrm{j}$ if and only if $\Pr\{r_i|\mathbf{s}_\mathrm{j}\} \neq 0$.

Therefore, the backbone resonance assignment problem can be formulated by the following ILP model:

$$\max_{y_{i,j,k}} \sum_{e_{i,j,k} \in E} (w_{i,j,k} + \lambda)\, y_{i,j,k}, \tag{5.12}$$

$$\text{subj. to } \forall i \in \{1, \ldots, n\},\ \textstyle\sum_{j=1}^{m} x_{i,j} \leq 1, \tag{5.13}$$

$$\forall j \in \{1, \ldots, m\},\ \textstyle\sum_{i=1}^{n} x_{i,j} \leq 1, \tag{5.14}$$

$$\forall e_{i,j,k} \in E \quad y_{i,j,k} \leq x_{i,j};\ \ y_{i,j,k} \leq x_{i+1,k}, \tag{5.15}$$

$$\text{and } \ x_{i,j} \in \{0, 1\}, \quad y_{i,j,k} \in \{0, 1\}. \tag{5.16}$$

Constraint (5.13) ensures that a residue can be assigned by, at most, one spin system. Constraint (5.14) ensures that a spin system can be assigned to, at most, one residue. Constraint (5.15) ensures that an edge can be selected, only if both of its ends are selected. The objective function is to find the assignment which maximizes the total weight of the selected edges. Since all the original edge weights are negative, the objective function adjusts all the edge weights to be positive values by adding a fixed term, $\lambda = -\min_{i,j,k}(w_{i,j,k})$, which enables the maximization to be meaningful.

CPLEX9.1 is used to solve the ILP model. For each combination of the reliable fragments, an ILP is generated and the solution is attained. The assignment with the highest score is reported as the final assignment.

5.2 Experimental Results

To evaluate the performance of IPASS, several experiments are conducted. Two performance measures are used in the following parts: *precision* and *recall*. *Precision* measures the ability to reject false assignments, whereas *recall* measures the ability to discover true assignments. Assume that for the target protein, there are

N_r manually assigned residues, and a resonance assignment program assigns N_o residues, where T_p of them are assigned correctly. Then, *recall* and *precision* are defined as T_P/N_r and T_P/N_o, respectively.

5.2.1 Performance on Real Data Sets

In practice, the input for resonance assignment is not "perfect". Instead, the input peak lists contain various sources of errors, such as the chemical shift difference of the same nucleus in different spectra and false peaks, picked during the peak picking step. Therefore, an assignment method is practical only if it works on "low quality" real noisy input data sets.

In the NMR lab experiments, the spectroscopists usually conduct the entire NMR process altogether, i.e., the resonance assignment, NOE assignment, structure calculation information, as well as information from the various kinds of other spectra, which are used as feedback to refine the peak lists. Thus, the final peak lists provided by NMR labs are always "almost perfect", and do not represent the original peaks picked by spectroscopists. Therefore, we apply IPASS on the peak lists generated by automated peak picking method, PICKY, to evaluate its performance on the real NMR lab data sets. Since IPASS requires HNCACB at one of the input spectra, only five of the eight proteins from PICKY's experiments are used as test proteins for IPASS, *i.e.*, TM1112, COILIN, VRAR, HACS1, and CASKIN.

Table 5.1 summarizes the performance of RIBRA, MARS, and IPASS for the five real proteins. Since MARS cannot take the peak lists as inputs, the spin systems formed by the spin system forming step of IPASS are given to MARS as inputs. The performance of MARS and IPASS are compared on the same set of spin systems. RIBRA takes the peak lists of ^{15}N-HSQC, CBCA(CO)NH, and

HNCACB as inputs, so the performance of RIBRA and IPASS are compared on the same peak lists. Table 5.1 clearly shows that IPASS significantly outperforms RIBRA and MARS on all of the five proteins in terms of the number of correctly assigned residues. One thing to notice is that when the input peak list quality is as good as TM1112, IPASS can generate assignments, which are almost as good as the manual assignment. In Table 5.1, the number of Glycine and Proline residues are shown. The Proline residues cut the fragments and make the assignment more challenging. The Glycine residues are favorable in a way that can be typed very easily due to their distinct C^{α} values. However, The Glycine residues shorten the fragments, because they do not have any C^{β} chemical shifts, and hence, no reliable connections. It is noticeable that when a protein has a large number of Glycine and Proline residues, such as COILIN, HACS1, and CASKIN, the number of incorrectly assigned residues by IPASS is also quite high. This reveals the importance of the reliable fragments because such residues break long reliable fragments.

Table 5.1: Performance of RIBRA, MARS, and IPASS on target proteins TM1112, COILIN, VRAR, HACS1, and CASKIN.

Protein	Len	Man	SS	Gly/Pro	RIBRA[1]	MARS[2]	MARS[3]	IPASS
TM1112	89	83	81 / 85	4 / 5	40 / 54	6 / 45	55 / 63	71 / 73
COILIN	98	71	60 / 73	4 / 9	6 / 38	23 / 28	23 / 28	36 / 64
VRAR	72	60	47 / 47	1 / 0	4 / 13	6 / 17	6 / 17	34 / 42
HACS1	74	61	48 / 61	7 / 5	5 / 11	15 / 16	15 / 16	24 / 36
CASKIN	67	54	47 / 48	7 / 4	12 / 21	23 / 25	23 / 25	31 / 41

Len: protein length; Man: number of manually assigned residues; SS: number of correct/total spin systems discovered by the spin system forming step of IPASS; Gly/Pro: number of Glycine and Proline residues in the sequence. For each protein, the performance of each method is shown in "number of correctly assigned residues/total number of assigned residues" format.

[1] RIBRA's performance with ^{15}N, ^{13}C threshold values of 0.5 and ^{1}H threshold value of 0.05. No residue can be assigned if the default values are used. The parameters are set according to IPASS, which makes the comparison fair.

[2] MARS with the first set of default parameters: $\delta_{\alpha} = 0.5ppm$ and $\delta_{\beta} = 0.5ppm$.

[3] MARS with the second set of default parameters: $\delta_{\alpha} = 0.2ppm$ and $\delta_{\beta} = 0.4ppm$.

5.2.2 Performance on Simulated Data Sets

Although the goal is to develop a backbone resonance assignment method which works on real data sets of automatically picked peak lists, performance of IPASS is also evaluated on some previously used benchmark sets.

Simulated Spin Systems as Input

First, the IPASS performance is evaluated on a simulated data set, used by [194], which contains 12 proteins. For each protein, the spin systems are simulated, based on the BMRB deposited chemical shift assignments of the protein, and used as the input for all of these programs. Each spin system contains N, H^N, C^α, C^β, $\tilde{C}^\alpha$, and $\tilde{C}^\beta$ chemical shifts. Since RANDOM and CISA are not available, the *precision* and *recall* values are selected from [194]. The accuracy of RANDOM, MARS, and CISA is calculated according to two different sets of threshold values, because these programs are sensitive to different threshold values. Note that in these experiments, the input for IPASS is simulated spin systems, so the spin system forming step is not tested here.

As shown in Table 5.2, IPASS performs very well and significantly better than any other program regardless of the set of threshold settings. The average *precision* of IPASS is 99%, and IPASS achieves a 100% *precision* on seven out of 12 target proteins. Meanwhile, IPASS can also achieve a high *recall* value of 96%. It is noteworthy that although MARS also has a high *precision* value on this data set, it has a relatively low *recall* value, compared to that of IPASS. On the other hand, Table 5.2 demonstrates that RANDOM, MARS, or CISA are sensitive to the threshold settings. For this simulated data set, a smaller threshold value can give a much better accuracy. However, in practice, researchers do not know the quality of the spin systems. It is challenging to determine the potential difference in the chemical

shift values of the same nucleus in different spectra. In contrast, IPASS does not rely on any parameter settings and its parameters are chosen without training on any data set.

Table 5.2: Performance (precision/recall) of RANDOM, MARS, CISA, and IPASS on the 12 protein data set with simulated spin systems.

			$\delta_\alpha = 0.2ppm,\ \delta_\beta = 0.4ppm$			$\delta_\alpha = 0.4ppm,\ \delta_\beta = 0.8ppm$			
Protein	Len	Man	RAN	MARS	CISA	RAN	MARS	CISA	IPASS
bmr4391	66	59	67 / 63	100 / 76	97 / 97	58 / 55	100 / 75	91 / 91	93 / 90
bmr4752	68	66	40 / 35	100 / 97	96 / 94	36 / 30	100 / 97	90 / 88	100 / 94
bmr4144	78	68	36 / 33	100 / 91	100 / 99	33 / 31	100 / 69	100 / 99	98 / 85
bmr4579	86	83	54 / 51	99 / 98	98 / 98	34 / 32	96 / 90	80 / 80	100 / 98
bmr4316	89	85	42 / 36	100 / 100	100 / 99	35 / 30	99 / 91	83 / 83	99 / 98
bmr4288	105	94	62 / 55	100 / 99	98 / 98	42 / 38	98 / 97	91 / 91	100 / 98
bmr4929	114	110	68 / 63	100 / 100	93 / 91	46 / 43	100 / 99	96 / 94	100 / 100
bmr4302	115	107	66 / 64	100 / 100	96 / 95	47 / 45	100 / 100	91 / 91	100 / 99
bmr4670	120	102	67 / 62	100 / 100	96 / 95	43 / 39	100 / 100	88 / 87	98 / 97
bmr4353	126	98	48 / 43	95 / 55	96 / 95	47 / 43	95 / 55	90 / 90	99 / 93
bmr4027	158	148	43 / 32	100 / 99	100 / 99	40 / 30	100 / 99	88 / 85	100 / 97
bmr4318	215	191	40 / 38	99 / 99	87 / 84	25 / 22	100 / 95	74 / 70	100 / 98
Average	112	101	53 / 48	99 / 93	96 / 95	41 / 37	99 / 89	88 / 87	99 / 96

Len: protein length; Man: number of residues that are manually assigned in the BMRB file; RAN: RANDOM. The accuracy of RANDOM, MARS, and CISA is calculated based on two sets of thresholds and listed in percentiles.

Simulated Peak Lists as Input

The IPASS performance is further tested on the same data set, but with simulated peak lists. Spin system forming step is also tested in this experiment. However, the CISA paper [194] does not provide such a comparison on RANDOM, MARS, and CISA. Furthermore, RANDOM and CISA are not available. As a result, IPASS is compared with two available programs: MARS and RIBRA. MARS takes only formed spin systems as inputs and RIBRA takes the peak lists as inputs. RIBRA is used directly, and IPASS's spin system forming method is applied to form spin systems for MARS.

Table 5.3 shows that both MARS and IPASS perform well on the simulated peak lists, and are better than RIBRA. MARS achieves higher *precision* and lower *recall* values than IPASS.

Table 5.3: Performance (precision/recall) of RIBRA, MARS, and IPASS on the 12 protein data set with simulated peak lists.

Protein	Len	Man	SS	Gly/Pro	RIBRA[1]	MARS[2]	MARS[3]	IPASS
bmr4391	66	59	55	6/1	91 / 76	93 / 43	94 / 46	91 / 85
bmr4752	68	66	65	6/1	91 / 90	100 / 94	100 / 94	100 / 92
bmr4144	78	68	63	3/5	62 / 45	100 / 58	100 / 41	98 / 85
bmr4579	86	83	80	5/2	87 / 67	99 / 87	99 / 83	100 / 94
bmr4316	89	85	80	13/3	99 / 88	99 / 83	99 / 73	88 / 79
bmr4288	105	94	93	5/10	100 / 97	99 / 95	100 / 97	99 / 97
bmr4929	114	110	108	10/2	82 / 78	100 / 83	99 / 68	99 / 98
bmr4302	115	107	107	5/2	100 / 92	100 / 96	99 / 97	96 / 95
bmr4670	120	102	92	9/5	98 / 86	99 / 87	100 / 87	93 / 79
bmr4353	126	98	97	8/10	98 / 93	99 / 90	100 / 91	97 / 90
bmr4027	158	148	146	11/8	90 / 82	99 / 94	99 / 92	97 / 94
bmr4318	215	191	188	9/12	74 / 63	99 / 93	99 / 86	98 / 90
Average	112	101	98	5/8	89 / 80	99 / 84	99 / 80	96 / 90

Len: protein length; Man: number of residues that are manually assigned in the BMRB file; SS: number of correct spin systems discovered by the spin system forming step of IPASS; Gly/Pro: number of Glycine and Proline residues in the sequence. The accuracy is listed in percentiles.
[1] RIBRA's performance with ^{15}N and ^{13}C threshold values of 0.5 and ^{1}H threshold value of 0.05. Those parameters are set according to IPASS for the sake of fair comparison.
[2] MARS with the first set of default parameters $\delta_\alpha = 0.2ppm$, and $\delta_\beta = 0.4ppm$.
[3] MARS with the second set of default parameters, $\delta_\alpha = 0.5ppm$, and $\delta_\beta = 0.5ppm$ which is the same as IPASS.

5.3 Discussion

IPASS is implemented in C++. It takes IPASS fewer than 5 minutes to achieve the result for a practical noisy data set of a medium size protein (70-100 residues in length). In addition, the whole process requires only five seconds for a simulated data set. The difference in speed stems from the fact that for the simulated data

sets, most of the fragments are fixed. Consequently, ILP problem size is very small.

Chapter 6

Structure Calculation and Decoy Selection

Although PICKY performs very well on the real NMR spectra and IPASS significantly outperforms the state-of-the-art assignment methods on automatically picked peaks. There is still one question to answer: since IPASS is not able to generate complete and perfect assignment, is the assignment of IPASS good enough for the ultimate goal, *i.e.*, automatically determining the high resolution structures of proteins? FALCON-NMR is developed, as the third module of AMR, to answer this question.

6.1 Methods

Figure 6.1 shows the flowchart of FALCON-NMR. Given the target protein sequence and the backbone assignment done by IPASS, FALCON-NMR first tries to look for close homologs by FALCON-Threading, the threading module of FALCON [117]. If there are close homologs, they are used to build the initial structural models for the target protein by Modeller [158]. If there is no close homolog found, the target

protein is considered as an *ab initio* target, and FALCON-AbInitio, the *ab initio* module of FALCON, is called to generate the initial structural models for the target. FALCON-AbInitio first calls Frazor [116] to generate fragment candidates for each small region of the target protein sequence according to chemical shift information. The torsion angles of these fragments are used to build torsion angle distributions. A hidden Markov model (HMM)-based torsion angle sampling method is developed to sample the protein conformational space based on the torsion angle distributions. However, both FALCON-Threading and FALCON-AbInitio are not able to identify the best decoys from the large number of decoys generated. Therefore, we develop an NOE contact-based score function to select the best decoys for FALCON. Such decoys are selected and fed into FALCON-Refinement to conduct all-atom level refinement. The refinement process is iterative by selecting the best decoys by the NOE contact-based score function at each round, and feeding back to FALCON-Refinement to further refine the models, until convergence.

6.1.1 FALCON

The protein structure prediction modules used in FALCON-NMR, *i.e.*, FALCON-Threading, FALCON-AbInitio, and FALCON-Refinement, are parts of the FALCON package [117, 116], which was previously developed by our lab.

FALCON-Threading is a threading method that tries to identify the best templates from PDB for a target protein by evaluating not only the sequence similarity, but also how well it is to align the target protein to the structural environments of the templates [115]. Since most of the newly solved protein structures have close homologs in PDB [141, 139, 140, 138, 137], it can be expected that FALCON-Threading can detect good templates for most of the target proteins.

If there is no close homolog in PDB or FALCON-Threading fails to detect any

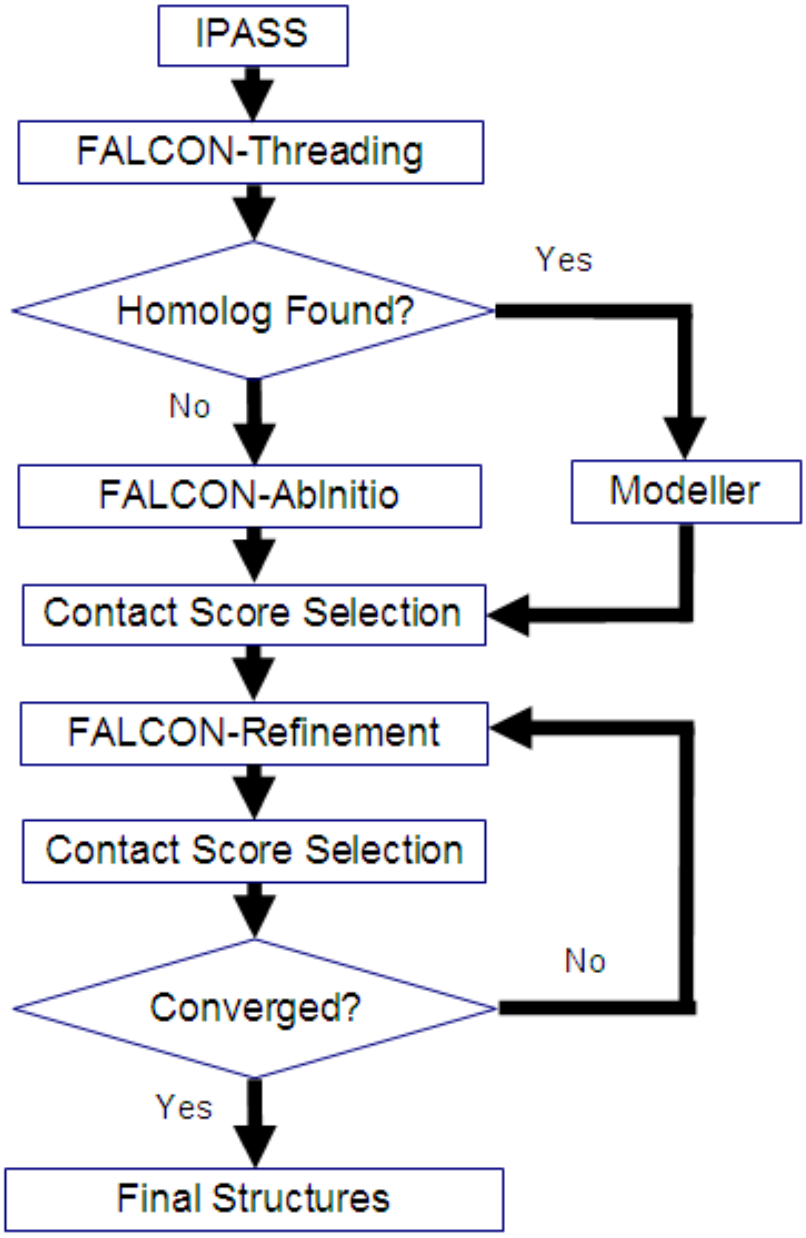


Figure 6.1: Flowchart of FALCON-NMR

homologs, the target protein is considered to be an *ab initio* target, for which FALCON-AbInitio is developed to predict the structure from scratch. FALCON-AbInitio first calls Frazor [116] to generate fragment candidates. Frazor takes the protein sequence and the backbone resonance assignment as input. The backbone assignment can be either complete or incomplete. Frazor then applies a local threading technique to select 9-mer fragments based on both the chemical shift information and the local threading information, such as solvent accessibility and secondary structure. Local threading is similar to the threading technique, which is widely applied in protein structure prediction community. Local threading tries to find local template fragments that have similar structures to a sequence fragment of a target protein with unknown structure, according to not only the sequence similarity between the template fragment and the sequence fragment, but also how well it is to put the sequence fragment into the local structural environment of the template fragment. Frazor further encodes chemical shift information into its local threading score function. For more details about Frazor, please refer to [116]. After the 9-mer fragment candidates are generated for each length-9 sliding window of the target protein, the torsion angles (ϕ and ψ angles) of these fragments are extracted to build torsion angle distributions for each residue. A high-order HMM-based torsion angle sampling method is applied to sample the conformational space directed by an empirical energy function that is used in ROSETTA [21]. For more details about the high-order HMM-based sampling method, please refer to [115, 117].

Although FALCON-Threading can identify good templates and FALCON-AbInitio can generate medium-resolution structural models for most target proteins, the quality of such models are still far away from the accuracy of the experimentally determined structures. Therefore, FALCON-Refinement is developed to try to refine the medium-resolution structural models to high-resolution ones. Similar to FALCON-AbInitio, FALCON-Refinement first extracts torsion angle distributions

from the input medium-resolution models. However, the distributions are much tighter than the ones extracted from the fragment candidates selected by Frazor. FALCON-Refinement then conducts an all-atom level refinement by sampling the torsion angles according to these very tight distributions.

6.1.2 A Contact-based Score Function

However, a common bottleneck of FALCON-Threading, FALCON-AbInitio, FALCON-Refinement, and any other protein structure prediction methods is that they are not able to select the best decoys from the large number of decoys generated. The most commonly used clustering-based decoy selection methods usually trap into popular but bad-quality models. We develop an NOE contact-based score function to select the best decoys for the structure generation methods. The main idea is that since NMR data is the major source to determine the structure of a target protein, a good-quality structural model must agree with the experimental evidence.

For a target protein, PICKY is called to pick peaks for the ^{15}N-edited NOESY and HCCONH-TOCSY spectra. A simple process is then applied to map protons determined by HCCONH-TOCSY peaks to their corresponding residues, according to the consistency between $(\mathrm{N}, \mathrm{H^N})$ values of HCCONH-TOCSY peaks and that of backbone resonance assignment. A similar process is then called to explain each ^{15}N-edited NOESY peak. For an ^{15}N-edited NOESY peak $(\mathrm{N_i}, \mathrm{H_j}, \mathrm{H_i^N})$, the residue with the closest $(\mathrm{N_k}, \mathrm{H_k^N})$ values are first found, and all residues that contain protons with chemical shift values close to $\mathrm{H_j}$ are kept to form "ambiguous" NOE assignments, *i.e.*, each NOE assignment contains a set of possibly correct contact residue pairs. The basic idea is that there should be at least one correct contact pair inside each assignment. For a decoy on one NOE assignment, it scores 1 if it satisfies at least one pairwise contact in this "ambiguous" assignment, and 0 otherwise. All

decoys are ranked according to how well they agree with the NOE assignments, and the top ranked models are selected as the inputs for iterative refinement process. If the RMSD between the models selected by two consecutive rounds of the refinement process is smaller than 0.05Å, the refinement process is considered to be converged and the top-ranked model is outputted as the final structure.

6.2 Results

FALCON-NMR is applied to determine the structures for the five target proteins, *i.e.*, TM1112, COILIN, VRAR, HACS1, and CASKIN. Among them, HACS1 and CASKIN have close homologs identified by FALCON-Threading, whereas TM1112, COILIN, and VRAR are solved by FALCON-AbInitio. For HACS1 and CASKIN, the initial models are built by Modeller based on the alignments between the target protein and its homologs detected by FALCON-Threading. For each of the three *ab initio* targets, FALCON-AbInitio generates 10,000 initial structural models. The NOE contact-based score function is applied to select the best decoys. The best decoys are then fed into FALCON-Refinement to iteratively refine the structures until convergence.

6.2.1 Final Prediction Models

The final prediction models for these five proteins have RMSD, without variable regions, of 1.25Å, 6.95Å, 1.49Å, 0.67Å, and 0.88Å to the native reference structures, respectively. It is noticeable that targets with close homologs usually result in better prediction models than the *ab initio* targets. One reason for the failure on COILIN is that the resonance assignment generated by IPASS is bad on COILIN (with 36 correctly assigned residues over the total 64 assigned residues). Another reason is

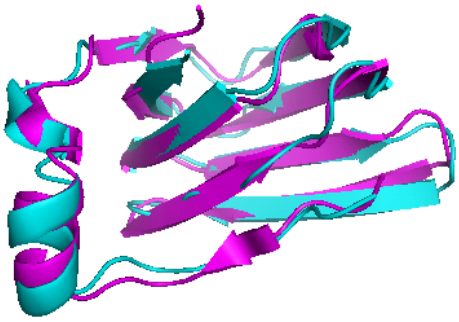


Figure 6.2: The superimposition between the finally selected model of FALCON-NMR (shown in cyan) and the crystal structure (shown in magenta) of TM1112. Backbone RMSD is 1.25Å.

that COILIN, which has 98 residues, is longer than the other four target proteins. FALCON-AbInitio is not able to handle such a long protein due to the flexibility in the torsion angle sampling process.

Figure 6.2, 6.3, 6.4, and 6.5 show the superimposition between the finally selected model and the native reference structure for TM1112, VRAR, HACS1, and CASKIN, respectively. It is clear that all the predicted models align very well to the native structures, except the variable loop regions which do not have fixed 3D structures.

6.2.2 Case Study of Contact-based Decoy Selection

To further illustrate the accuracy and the usefulness of the NOE contact-based score function, we present more details about NOE contact-based decoy selection on TM1112.

PICKY automatically picks 1,213 peaks for the ^{15}N-edited NOESY spectrum of TM1112, and 951 "ambiguous" NOE assignments are generated. Among them, 811 assignments contain at least one correct contact pair, which gives an accuracy of 85.3%. A contact pair is correct if the distance between H^N atoms of the two

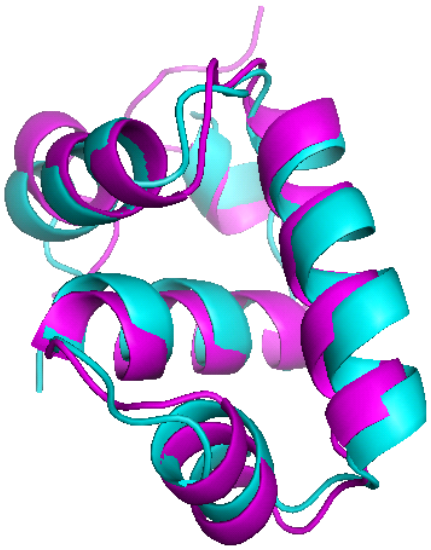


Figure 6.3: The superimposition between the finally selected model of FALCON-NMR (shown in cyan) and the reference NMR structure (shown in magenta) of VRAR. Backbone RMSD is 1.49Å.

Figure 6.4: The superimposition between the finally selected model of FALCON-NMR (shown in cyan) and the reference NMR structure (shown in magenta) of HACS1. Backbone RMSD without variable regions is 0.67Å.

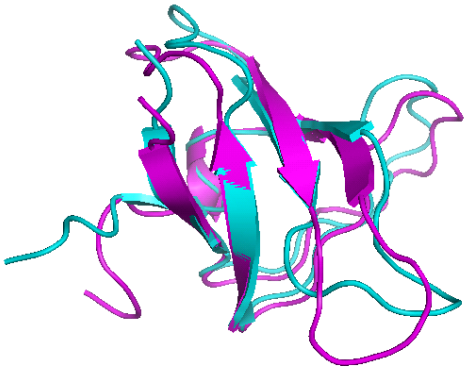


Figure 6.5: The superimposition between the finally selected model of FALCON-NMR (shown in cyan) and the reference NMR structure of (shown in magenta) CASKIN. Backbone RMSD without variable regions is 0.88Å.

residues are indeed smaller than 6Å in the crystal structure. More specifically, there are 207 correct non-local assignments. An NOE assignment is defined as non-local if the two residues in contact are at least 6 residues apart in the protein sequence. These non-local contacts are extremely important to determine the 3D structure of the protein.

This set of 951 NOE assignments is then applied on the decoys generated by FALCON-AbInitio and FALCON-Refinement. Figure 6.6 shows the correlation between the decoy quality, in terms of RMSD value to the crystal structure, and the NOE contact score, on the final decoy set of FALCON-Refinement before convergence. The decoy set contains 10,000 decoys. It can be seen that the best decoys are well identified by the NOE contact score. In fact, the best decoy (RMSD 1.25Å to the crystal structure) is ranked number one among all 10,000 decoys, while the five of the best six decoys are ranked as top five, which are 1.25Å, 1.43Å, 1.94Å, 1.36Å, and 1.74Å RMSD to the crystal structure, respectively. The second best decoy (RMSD 1.26Å to the crystal structure) is ranked number ten.

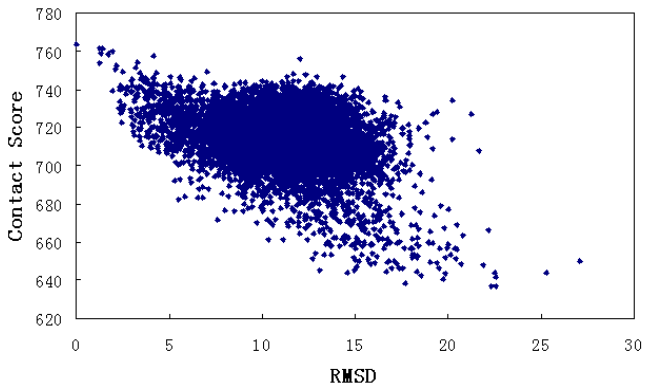


Figure 6.6: Correlation between decoy quality in terms of RMSD value to the crystal structure, and the NOE contact score, for TM1112. The blue point on y-axis represents the crystal structure, which has higher contact score than any decoy does.

6.3 Discussion

One may argue that since FALCON is a protein structure prediction method, it is possible that even without NMR data, *i.e.*, chemical shift information and NOE contacts, FALCON itself can still generate high-resolution structural models for the target proteins. To verify this, three more experiments are performed: 1) run FALCON with the experimentally determined resonance assignment based on the manually picked peaks from the NMR lab. The NOE contact-based score function is applied again for decoy selection; 2) simply run FALCON without any chemical shift information, and perform decoy selection by the default clustering-based method of FALCON; 3) run ROSETTA2.3.0 [21] without any chemical shift information, and perform decoy selection by the default clustering-based method of ROSETTA. Experimental results demonstrate that running FALCON with perfect assignment results in a slightly better model for the target protein, whereas simply running FALCON or ROSETTA with their default clustering-based decoy selection methods results in low-resolution models, especially for *ab initio* targets. For example, the finally selected decoys for TM1112 are 1.17Å, 11.84Å, and 12.13Å for those three

experiments, respectively. Similar results are obtained on other targets as well. This implies that replacing the manual peak picking process by PICKY and replacing the manual assignment process by IPASS do not affect the final structure accuracy. This also shows that without chemical shift information, neither FALCON nor ROSETTA is able to generate the final high-resolution structures.

Chapter 7

Some Related Protein Structure Prediction Works

7.1 Contact Prediction

To solve the protein inter-residue contact prediction problem, we propose a novel consensus contact prediction method to eliminate the effect of server correlation, and to discover true contacts even when they are not commonly found in the top templates. All the contacts, determined by structure prediction servers, are considered to be candidates. Our consensus method then assigns a confidence score to each contact candidate, while also taking correlated mutation information into consideration.

7.1.1 Methods

Recent CASP results have indicated that correlation exists in different protein structure prediction servers, because of the common information used by the servers such as PSI-BLAST [11] sequence profile and PSIPRED-predicted secondary struc-

ture [95]. Thus, it is very likely that a true contact is supported less than some false ones due to the server correlation. Our consensus method is capable of reducing the impact caused by the server correlation. The outline of our consensus method is as follows:

- A maximum likelihood (ML) method is applied to measure the correlation coefficient between two servers.
- Principal component analysis (PCA) technique is employed to extract a few independent latent servers from a set of correlated servers.
- An integer linear programming (ILP) method is then used to assign a weight to each latent server, by maximizing the difference between the true contacts and the false ones. Also, correlated mutation is treated as a latent server which assigns a probability value to each contact candidate. This results in a consensus contact predictor that can accurately assign confidence scores to all the contact candidates.

Notations

A *model* refers to a protein conformation, generated by a protein structure prediction server. In contrast to human experts, a *server* refers to an automated system which predicts a set of models for a given protein (also called a *target*), whereas a *contact predictor/server* refers to an automated system which predicts a set of contacts. Following the contact definition used by CASPs, two residues are in contact, if the distance between their C_β atoms (C_α atom for Glycine), is smaller than 8Å, and they are at least six residues apart in the sequence. We call a contact *native/true contact*, if the two residues are indeed in contact in the native structure of the target.

The prediction accuracy is the number of correctly predicted contacts divided by the total number of contacts predicted by a predictor, while the coverage is defined as the number of correctly predicted contacts divided by the total number of native contacts. If a contact predictor is a tertiary structure prediction server, all the contacts, contained in the structural models of this server, are considered to be the contact prediction results of this server.

Let ℓ denote the number of target proteins, and u denote the number of input contact prediction servers. Given a target t_l $(1 \leq l \leq \ell)$, a server S_i $(1 \leq i \leq u)$ outputs a set of models. The contacts, determined by these models, are extracted and considered as contact candidates, denoted as $C_{i,l} = \{c_{i,l,q} | 1 \leq q \leq n_{i,l}\}$, where $n_{i,l}$ is the number of contacts, predicted by server S_i for target t_l. The set of all contact candidates for target t_l is denoted as $C_l = \bigcup_i C_{i,l}$. A consensus server aims to assign a confidence score to each candidate in C_l.

Our consensus method is based on the following two assumptions:

- Server S_i generates its predictions based on a confidence measure; that is, for each contact $c \in C_l$, S_i has a confidence, $s_{i,c,l}$, on how likely it is for c to appear in the native structure. Since the initial confidence score is unavailable, it is approximated as follows: the number of models containing this contact divided by the total number of models generated by the server for this target.

- There are v implicitly independent latent servers H_j $(1 \leq j \leq v)$ dominating the explicit servers S_i. Given a target t_l, H_j assigns a value $h_{j,c,l}$ $(c \in C_l)$ as the confidence score on how likely c is a native contact.

Identifying independent latent servers is essential to reduce the negative effects of the server correlation and to reduce the dimensionality of the search space, as the number of latent servers is expected to be smaller than the number of original

servers. After deriving the latent servers, a new and more accurate prediction server S^* can be designed, by an optimal linear combination of the latent servers, which for each target t_l, assigns a confidence score to each contact candidate $c \in C_l$ as follows:

$$s^*_{c,l} = \sum_{j=1}^{v} \lambda^*_j h_{j,c,l}, \tag{7.1}$$

where λ^*_j is the weight of latent server H_j in S^*.

Maximum Likelihood Estimation of Server Correlation

Let $O_{i,j,l}$ denote the overlap set of $C_{i,l}$ and $C_{j,l}$; that is, $O_{i,j,l} = C_{i,l} \cap C_{j,l}$, and let $o_{i,j,l} = |O_{i,j,l}|$. For a given target, let $p_{i,j}$ be the probability that a contact, returned by server S_i, is the same as that returned by S_j. Under a reasonable assumption that targets t_l $(1 \leq l \leq \ell)$ are mutually independent, the likelihood that server S_i $(1 \leq i \leq u)$ generates contacts $c_{i,l,q}$ $(1 \leq q \leq n_{i,l})$ is

$$L(p_{i,j}) = \prod_{l=1}^{\ell} \binom{n_{i,l}}{o_{i,j,l}} p_{i,j}^{o_{i,j,l}} (1 - p_{i,j})^{n_{i,l} - o_{i,j,l}}. \tag{7.2}$$

Therefore, the maximum likelihood estimation of $p_{i,j}$ can be calculated as follows:

$$p_{i,j} = \frac{\sum_{l=1}^{\ell} o_{i,j,l}}{\sum_{l=1}^{\ell} n_{i,l}}. \tag{7.3}$$

Let P denote the matrix $[p_{i,j}]_{u \times u}$.

Uncovering Independent Latent Servers

Recall that on a target t_l, $s_{i,c,l}$ and $h_{j,c,l}$ are the confidence scores assigned by server S_i and H_j, respectively. Since the latent servers are mutually independent, it is

reasonable to assume that $s_{i,c,l}$ is a linear combination of $h_{j,c,l}(1 \leq j \leq v)$ such that

$$\overrightarrow{s_{i,l}} = \sum_{j=1}^{v} \lambda_{i,j} \overrightarrow{h_{j,l}}, \qquad \sum_{j=1}^{v} \lambda_{i,j} = 1, \qquad 1 \leq i \leq u, \qquad 1 \leq l \leq \ell, \tag{7.4}$$

where $\overrightarrow{s_{i,l}} =< s_{i,1,l}, s_{i,2,l}, \ldots, s_{i,|C_l|,l} >, 1 \leq i \leq u$, and $\overrightarrow{h_{j,l}} =< h_{j,1,l}, h_{j,2,l}, \ldots, h_{j,|C_l|,l} >$ $, 1 \leq j \leq v$. Here, $\lambda_{i,j}$ is the weight, and a larger $\lambda_{i,j}$ implies there is a higher chance that server S_i will adopt the contacts reported by H_j.

From the correlation matrix of prediction servers S_i, the factor analysis technique is employed to derive $\lambda_{i,j}$ and $\overrightarrow{h_{j,l}}$; that is, $\overrightarrow{h_{j,l}}$ can be represented as a linear combination of $\overrightarrow{s_{i,l}}$ as follows:

$$\overrightarrow{h_{j,l}} = \sum_{i=1}^{u} \omega_{j,i} \overrightarrow{s_{i,l}}, \qquad 1 \leq j \leq v, \qquad 1 \leq l \leq \ell, \tag{7.5}$$

where $< \omega_{j,1}, \omega_{j,2}, \cdots, \omega_{j,u} >$ is an eigenvector of $P^T P$.

ILP Model to Optimally Combine Latent Servers

After deriving the latent servers $H_j(1 \leq j \leq v)$, a new server S^* can be constructed as an optimal linear combination of the latent servers. For each target t_l, S^* assigns a score to each contact candidate $c \in C_l$ as in Eq. (7.1).

To determine a reasonable setting of coefficient λ_k^*, a training process is conducted on a data set $D = \{< t_l, C_l^+, C_l^- >, 1 \leq l \leq |D|\}$, containing $|D|$ training proteins, where t_l is a training protein, $C_l^+ \subseteq C_l$ denotes the set of native contacts, and $C_l^- \subseteq C_l$ denotes the set of false contacts. The learning process attempts to maximize the number of contacts that can be correctly identified by S^*.

More specifically, for each target t_l in the training data set, a score is assigned to each contact candidate by S^*. A good contact predictor should assign native

contacts with higher scores than those with false ones. The larger the gap between the scores of the native contacts and those of the false ones, the more robust this new prediction server is. In practice, a "soft margin" idea is adopted to take the outliers into account; that is, by allowing errors on some samples, we maximize the number of native contacts with a score that is higher than that of all the false ones, by at least a gap.

This optimization problem is formulated as an integer linear programming model. Let $x_{p,q}$ be an integer variable such that $x_{p,q} = 1$ if and only if the native contact p is assigned a score higher than that of the false contact q by at least ϵ by the new server; $x_{p,q} = 0$, otherwise. Here, ϵ is a parameter used as the lower bound of the gap between the score of a native contact and that of the false ones. Similarly, $y_{p,l} = 1$ if and only if the native contact p has a score higher than that of all the false contacts in C_l^-; $y_{p,l} = 0$, otherwise. The goal is to maximize the number of native contacts that have higher score than that of all the false contacts.

Consequently, the consensus contact prediction problem is formulated by the following ILP model:

$$\max_{y_{p,l}} \sum_{l=1}^{|D|} \sum_{p \in C_l^+} y_{p,l}, \tag{7.6}$$

$$\text{subj. to} \quad \forall p \in C_l^+, \forall q \in C_l^-, 1 \le l \le |D| \quad \sum_{j=1}^{v} \lambda_j^* h_{j,p,l} - \sum_{j=1}^{v} \lambda_j^* h_{j,q,l} - \epsilon \ge x_{p,q} - 1, \tag{7.7}$$

$$\forall p \in C_l^+, 1 \le l \le |D| \quad \frac{1}{|C_l^-|} \sum_{q \in C_l^-} x_{p,q} \ge y_{p,l}, \tag{7.8}$$

$$\sum_{j=1}^{v} \lambda_j^* = 1, \lambda_j^* \geq 0, \tag{7.9}$$

$$\text{and} \quad x_{p,q} \in \{0,1\} \quad y_{p,l} \in \{0,1\}. \tag{7.10}$$

Constraint (7.7) forces $x_{p,q}$ to be 0 if the gap between the scores, assigned to the native contact p and the false contact q, is smaller than ϵ. If a native contact p has a score not higher than all the false contacts, constraint (7.8) forces $y_{p,l}$ to be 0. Thus, there is no contribution to the objective function. Constraint (7.9) normalizes the weights, and constraint (7.10) restricts $x_{p,q}$ and $y_{p,l}$ to be either 0 or 1. The objective function is the number of native contacts that have higher scores than all the false contacts.

New Prediction Server

After the independent latent servers are derived and the optimal weights are trained, a new contact predictor is formed. Given a query target t_l, each server S_i produces a set of contact candidates, $C_{i,l}$. The set of all the candidates is denoted as $C_l = \bigcup_i C_{i,l}$. For each contact candidate $c \in C_l$, the confidence assigned by the latent server H_j is calculated by Eq. (7.5). Then, the new consensus server S^* assigns a confidence score to contact candidate c according to Eq. (7.1). S^* assigns a confidence score to each contact candidate, and picks up the top scored ones as the final predictions.

7.1.2 Results

Data Set

Server Selection To evaluate the performance of the proposed consensus method, six threading-based protein structure prediction servers are used: FOLDpro [30], mGenThreader [94, 131], RAPTOR [204, 202], FUGUE3 [169], SAM-T02 [98], and SPARK3 [222]. Although there are some servers, such as ROSETTA and Zhang-server, with a better performance than that of the six servers, they are not used because their models are already refined by predicted contacts.

Training and Test Data The biennial CASP competition provides us a comprehensive and objective data set. The CASP7 targets and models generated by the six servers are adopted as the training and test data. For each server on a target, the five submitted models are considered. All server models are downloaded from the CASP7 website, except for mGenThreader, which does not participate in CASP7. We submitted the CASP7 targets to the mGenThreader web server, and downloaded models from there before August 2006. Therefore, all these models are generated before the native structures of the CASP7 targets are released. Eighty nine CASP7 target proteins are used as valid targets for the CASP7 evaluation, while 104 protein sequences are released as targets. Redundancy is removed at the 40% sequence identity level by using CD-HIT [118], which results in 88 target proteins. Only T0346 is removed, because it shares 71% sequence identity with T0290. Furthermore, two targets (T0334 and T0385) are removed from the data set due to some errors in the models, generated by some of the six individual servers on these two targets; for example, the models generated by some servers only cover discontinued regions of the target proteins. To conduct a cross-validation, the 86 target proteins are randomly divided into four sets of 22, 21, 21, and 22 proteins, respectively. If one target belongs to a certain set, all of its models and contacts

Table 7.1: Average and deviation of contact accuracy and coverage of the six individual servers on the 86 CASP7 targets.

	Accuracy		*Coverage*	
	Average	*Deviation*	*Average*	*Deviation*
FOLDpro	45	8.2	48	9.3
mGenThreader	43	6.6	45	8.5
RAPTOR	48	6.6	52	7.0
FUGUE3	46	7.9	37	5.5
SAM-T02	53	6.5	37	5.5
SPARK3	48	7.3	51	7.6
Overall	12	7.2	80	2.3

All values are percentiles.

are in the same set.

Data Set Statistics The performance of the six individual servers are compared in terms of prediction accuracy and coverage. In evaluating the performance of a server, only the best models of each target are considered. If the number of contacts in a model is less than $L/5$, where L is the target size, both the accuracy and the coverage for this model are set to 0. As shown in Table 1, the average accuracy of the six servers ranges from 43% to 53%. The SAM-T02 server has the highest accuracy but the lowest coverage. The artificial server "Overall" in Table 1 means a server that generates the union set of all contacts contained in the best models. The accuracy of server "Overall" is very low (12%), compared to that of any individual server. Note that the server "Overall" consistently contains many more true contacts than any individual server does. Therefore, the low accuracy of the server "Overall" implies that the false contacts, generated by these individual servers, differ from each other in most cases, whereas the individual servers tend to generate common true contacts. This means the consensus method can probably be employed to differentiate true contacts from false ones.

As shown in Table 1, the average coverage of the six servers ranges from 37% to

Table 7.2: Pairwise correlation of the six individual servers.

Server	FDP	MGTH	RAP	FUG	SAM	SPK
FDP	1	0.34	0.43	0.25	0.30	0.41
MGTH	0.35	1	0.42	0.26	0.30	0.41
RAP	0.43	0.41	1	0.30	0.35	0.51
FUG	0.35	0.35	0.40	1	0.37	0.40
SAM	0.50	0.50	0.59	0.47	1	0.59
SPK	0.40	0.41	0.50	0.29	0.34	1

FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3.

52%. However, when they are combined, the coverage for server "Overall" is very high (approximately 80%). This indicates that some true contacts are predicted by only a small number of individual servers, and the different servers can predict a common subset of the true contacts. On the other hand, this implies that most of the native contacts are contained in the models, generated by threading programs. Thus, the challenge is how to identify them.

Server Correlation and Latent Servers

The correlation of the six individual servers is studied and the latent servers are derived. Table 2 lists the pairwise correlation of the six individual servers, which is calculated according to Eq. (7.3). Note that the matrix is not symmetric, because different servers predict different numbers of contacts. As shown in Table 2, the correlation between two different servers ranges from 0.25 to 0.59, which implies that some servers are more closely correlated than others in terms of contact prediction. Therefore, the majority voting based consensus methods, which simply apply majority voting and assume each server is independent, will not always work because some of the common components of these servers are over-expressed.

The relationship between the latent servers and the individual ones is then

Table 7.3: Relationship between the six individual servers and the independent latent servers.

Server	H1	H2	H3	H4	H5	H6
FOLDpro	0.37	-0.35	0.66	0.01	-0.08	-0.55
mGenThreader	0.37	-0.26	-0.75	-0.01	-0.02	-0.48
RAPTOR	0.42	-0.23	0.04	0.27	0.76	0.36
FUGUE3	0.37	0.82	0.04	0.37	0.01	-0.22
SAM-T02	0.49	0.20	0.03	-0.81	-0.04	0.23
SPARK3	0.41	-0.21	-0.02	0.36	-0.65	0.49

derived according to Eq. (7.4). Note here that the top five models for each target of each server are considered. The confidence score of a server on a contact candidate is estimated by the number of models in the top five, containing this contact, divided by the total number of models considered (five in this case). As shown in Table 3, different latent servers represent different individual servers. For example, H_1 represents the common characteristics shared by these individual servers, because the weights of H_1, on these individual servers, are about the same; H_2 differentiates FUGUE3 from the other servers; H_3 represents FOLDpro by a large positive weight, and represents mGenThreader by a large negative weight. Based on the eigenvalues, H_6 is eliminated, since the eigenvalue for H_6 is much smaller than that of the others. Thus, H_6 is considered as random noise.

The optimal weights for the latent servers are derived by the cross-validation of the four sets. Correlated mutation is considered to be another independent latent server, because it provides a target sequence-related probability for each contact candidate. Correlated mutation is calculated as previously described in [66, 143]. Each time the ILP model is trained on three of the four sets, and a set of weights is optimized by the ILP model, based on which a new prediction server is derived, named as S_1^*, S_2^*, S_3^*, and S_4^*, respectively. S^* refers to server S_i^* on test set i ($i = 1, 2, 3, 4$). Since the inputs are the six individual servers, after the optimal

Table 7.4: Linear combination representation of new server S^* on the six individual servers and correlated mutation.

S	FDP	MGTH	RAP	FUG	SAM	SPK	CM
S_1^*	0.29	-0.28	1.27	1.47	0.23	0.62	0.30
S_2^*	0.301	-0.27	1.35	1.35	0.22	0.58	0.37
S_3^*	0.38	-0.29	1.37	1.36	0.14	0.65	0.28
S_4^*	0.29	-0.44	1.29	1.39	0.12	0.56	0.23

FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3, CM: correlated mutation.

weights λ^* are calculated by the ILP model, each hidden server in Eq. (7.1) are further replaced by the linear combination of the original individual servers as calculated in Eq. (7.5). Table 4 shows the linear combination representation of S^* on the individual servers and correlated mutation. It is clear that the four sets of weights are very similar. Note that mGenThreader has negative weights. This implies that the contribution of mGenThreader is accounted for by the other individual servers.

CASP7 Evaluation

We first assess our consensus server S^* by Receiver Operating Characteristic (ROC) plots. They provide an intuitive way to examine the trade-off between the ability of a classifier to correctly identify positive cases and to incorrectly classify negative cases. Figure 7.1 depicts the performance of server S^* and the six individual servers on the four test sets.

As shown in Figure 7.1, server S^* performs better than any individual server on all the four test sets. For each server, the performance of this server on test set 1 is slightly better than that on the other three test sets, which means test set 1 is the easiest among those four. RAPTOR performs better than other individual servers on the first three test sets, and SPARK3 exhibits the best performance

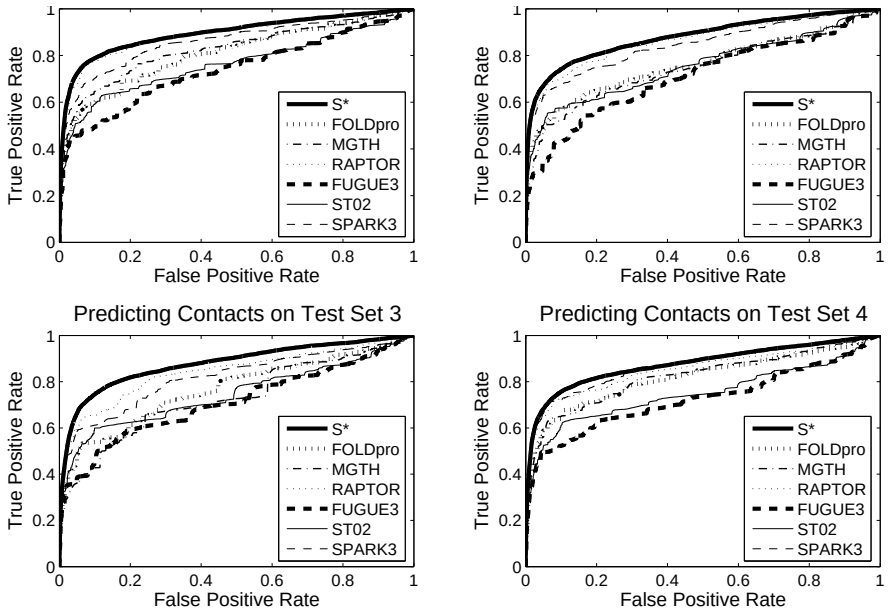


Figure 7.1: ROC curves for our method and the six individual servers

on test set 4. There are distinct performance differences between server S^* and the best individual server on test set 1, 2, and 4, when the false positive rate is below 0.3. However, the difference is not obvious on those three test sets, when the false positive rate is higher than 0.3. For test set 3, the most difficult test set, the performance of S^* is much better than that of any individual server all the time. It is also noticeable that the curve of S^* is much smoother than that of the individual servers.

Then, the accuracy of S^* is evaluated. Table 5 summarizes the average accuracy of S^* and the majority voting method on the four test sets, where different numbers of top contacts are evaluated. Recall S^* generates a confidence score for each contact candidate. The top contacts for each target are readily found by sorting the candidates according to their confidence scores. The majority voting method is implemented as follows: for each contact candidate, its confidence score by the majority voting method is calculated as the sum of the confidence scores assigned by the six servers. After the scores of all the contact candidates are calculated and

Table 7.5: Average accuracy of the top contacts predicted by S^* on different test sets, and the accuracy of the majority voting method.

# Contacts	$Accu_{set1}$	$Accu_{set_2}$	$Accu_{set3}$	$Accu_{set4}$	$Accu_{overall}$	$Accu_{mv}$
L	69	60	57	65	63	61
$L/2$	75	67	63	72	69	66
$L/5$	80	73	67	74	73	68
$L/10$	79	74	69	76	75	71

Contacts: the number of top contacts being considered. $Accu_{set1}$, $Accu_{set2}$, $Accu_{set3}$, and $Accu_{set4}$ show the accuracy of our method on the four test sets, respectively. $Accu_{overall}$: the overall accuracy of our method on all the four test sets. $Accu_{mv}$: the overall accuracy of the majority voting method on all the four test sets. All values are percentiles.

sorted, different numbers of the top candidates are chosen.

As shown in Table 5, the average accuracy increases when the number of the top contacts decreases, except for server S^* on test set 1, in which the accuracy for the top $L/10$ contacts is slightly lower than that for the top $L/5$ contacts. This occurs because $L/10$ is usually a small number (20-30 for most cases), and a few incorrectly predicted top contacts will influence the total accuracy significantly. The overall accuracy of S^* on all the four test sets is at least 63%, and is consistently higher than the accuracy of the majority voting method. For the top $L/5$ contacts, the accuracy of S^* is 73%, which is about 5% higher than that of the majority voting method, and much higher than the accuracy of any previously reported study.

Figure 7.2 reflects the prediction accuracy for the top $L/5$ contacts of S^* on each CASP7 target. It can be seen that the accuracy is higher than 80% on most targets. In fact, of the total 86 targets, S^* has an accuracy of 100% on 13 targets, higher than 90% on 39 targets, higher than 80% on 58 targets, and below 40% on only 16 targets. Note that S^* has an accuracy of 0 on two targets: T0309 (free modeling target) and T0335 (template based modeling target). We carefully look into these two cases. Both targets are very short. The target sequences, published by CASP7 for T0309 and T0335, have 76 and 85 residues, respectively. However,

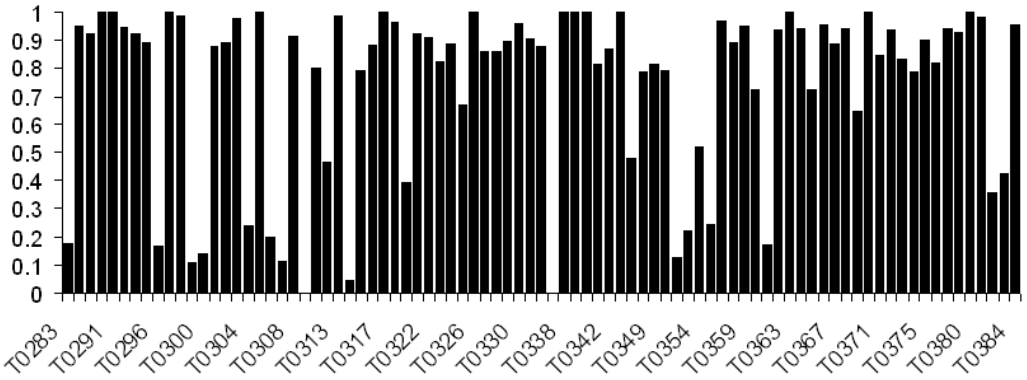


Figure 7.2: Prediction accuracy for the top $L/5$ contacts of S^* on each CASP7 target

the experimentally determined size used by CASP7 to evaluate these two targets are only 62 and 36, respectively. This conveys that some parts of the targets are not experimentally determinable or accurate enough. Thus, $L/5$ is only 12 and 7 for the two targets. Additionally, all the six individual servers perform poorly in terms of the contact prediction, which means there are only a few correct candidates among a large number of incorrect ones. This can explain the failure of S^* on T0309 and T0335.

To evaluate more carefully how much our consensus method can improve upon individual servers and the simple majority voting method, all the targets are divided into three categories: easy (high accuracy), medium (template based modeling), and hard (new fold), according to the CASP7 assessment [3]. Table 6 shows the average accuracy and deviation on the top $L/5$ contacts of S^*, individual servers, and the majority voting method. As shown in Table 6, for easy, medium, and hard targets, the accuracy of S^* on the top $L/5$ contacts is 94%, 76%, and 37%, respectively, and much higher than the best individual server, where the improvement is at least 17% for each case. On the other hand, server S^* always performs better

Table 7.6: Accuracy and deviation of top $L/5$ contacts of the six individual servers, the majority voting method, and the new server S^* on easy, medium, and hard target sets.

Server Name	Easy Targets		Medium Targets		Hard Targets	
	Accu.	*Dev.*	*Accu.*	*Dev.*	*Accu.*	*Dev.*
FOLDpro	**77**	1.1	44	5.3	10	5.8
mGenThreader	68	3.8	43	4.4	11	7.4
RAPTOR	75	1.3	50	3.9	13	7.1
FUGUE3	75	0.7	47	6.2	12	9.1
SAM-T02	75	1.3	**54**	5.2	**17**	14.7
SPARK3	76	1.4	48	4.7	11	7.4
Majority Voting	**92**	0.7	**71**	8.1	**13**	6.9
S^*	**94**	0.4	**76**	8.5	**37**	28.2

Accu.: Accuracy. *Dev.*: Deviation. All values are percentiles.

than the majority voting method, and the improvements are about 2%, 5%, and 24%, respectively. This exactly verifies the server correlation assumption because, for easy targets, individual servers usually do well, which means for a contact candidate, the more servers that support it, the more likely it is correct. However, the majority voting rule does not always work on medium and hard targets, because it suffers from the over-expressed common components of the input servers due to the server correlation. Thus, our consensus method does much better than the majority voting method on harder targets.

Depending on the sequence separation, contacts can be classified as short-range contacts (separation 6-11), medium-range contacts (separation 12-24), and long-range contacts (separation >24) [89, 199]. The performance of our method is further evaluated on different separation ranges for target protein classes with different difficulty levels. As shown in Table 7, the accuracy of a certain separation range decreases clearly when target proteins become harder. For easy targets, the accuracy of long-range contacts is higher than that of short- and medium-range contacts. This makes sense because for an easy target, it is very likely that all the individ-

Table 7.7: Performance of the new server S^* on different separation ranges of target protein classes with different difficulty levels.

Target Classes	Short-range	Medium-range	Long-range	All-range
Easy Targets	91	90	93	94
Medium Targets	73	74	70	76
Hard Targets	41	35	26	37
All Targets	72	71	68	73

Top $L/5$ contacts are considered. All values are percentiles.

ual servers predict models that have very similar topology to the native structure. Thus, these models contain common long-range contacts, which helps to determine the overall topology. For medium targets, our method achieves similar performance on different separation ranges. Not surprisingly, when applied on hard targets, the accuracy of long-range contacts is much worse than that of short- and medium-range contacts. This coincides with the fact that the individual servers are usually not able to generate models with correct folds, which causes most long-range contact candidates to be wrong ones.

Among the three categories of the test proteins, the new fold category is much more important than the other two for fairly evaluating the performance of a contact predictor, especially for template-based consensus methods. In fact, new fold targets are adopted as the assessment data set for the contact predictors by CASPs. Table 8 shows the average accuracy on the top $L/5$ contacts of the six individual servers, the majority voting method, and our method on the 15 new fold targets of CASP7. Note that the classification of the new fold targets comes from the assessors of CASP7, according to the criterion that no server could find the correct templates although there might be homologs in the PDB. Our method significantly outperforms the best individual server on eight out of the 15 targets, and performs worse than the best individual server on five targets. It is noticeable that SAM-T02 outperforms our method on four of these five targets. The reason is that SAM-T02

does not generate complete models for these targets. Instead, it generates structures only for some very conserved regions of the targets. The contacts predicted by SAM-T02 thus cover only a small portion of the targets. It can also be seen from Table 8 that our method performs much better than the majority voting method on new fold targets. More specifically, the accuracy of our method at least doubles that of the majority voting method on 10 of the 15 targets. On the other hand, only four of these 15 new fold targets lacked any homologs during CASP7 season, *i.e.*, T0287, T0309, T0314, and T0353 [89]. This implies that although other new fold targets have similar structures in the PDB, almost all structure prediction servers fail to detect them. Thus, by using our predicted contacts, one may be able to identify the similar structures for these target proteins, especially for the proteins on which our method achieves a high accuracy, such as T0316_D2, T0319, T0347_D2, T0350, T0356_D1, T0356_D3, and T0386.

We are not able to obtain top-notch contact predictors, such as SVM-LOMETS, the best published consensus method, SVM-SEQ, the best reported study on new fold targets, and SAM-T06 server, the best evaluated contact predictor on CASP7. Thus, the performance of these three methods is retrieved from [199]. When SVM-LOMETS, SVM-SEQ, and SAM-T06 server are applied to the 15 new fold targets of CASP7, each achieves an accuracy of 10.8%, 25.8%, and 21.2% on the top $L/5$ contact predictions, respectively. On the same data set, the accuracy of our method for the top $L/5$ contacts is 37%, which indicates that the improvements are significant. Recall that among all three methods, SVM-LOMETS is the only template-based consensus method. Although the input threading programs of our method are not the same as SVM-LOMETS, both methods contain some common input servers such as FUGUE and SAM-T02. The different input servers are within a similar range of accuracy in terms of structure prediction according to the CASP7 evaluation; three inputs for SVM-LOMETS, *i.e.* PAINT, PPA-I,

Table 7.8: Accuracy of top $L/5$ contacts of the six individual servers, the majority voting method, and the new server S^* on the 15 new fold targets of CASP7.

	FDP	MGTH	RAP	FUG	SAM	SP3	MV	S^*
$T0287$	8	6	9	5	12	6	9	33
$T0296$	2	25	7	3	29	7	8	17
$T0300$	16	4	6	10	3	6	8	18
$T0307$	3	6	10	15	18	10	7	12
$T0309$	22	3	6	6	32	5	8	0
$T0314$	12	6	7	8	3	6	8	5
$T0316\_D2$	15	18	14	24	31	16	21	88
$T0319$	9	9	15	29	0	8	11	40
$T0347\_D2$	13	3	14	5	46	28	26	48
$T0350$	11	8	27	9	35	26	21	80
$T0353$	17	23	24	29	26	18	12	22
$T0356\_D1$	4	18	6	3	0	5	8	36
$T0356\_D3$	6	10	12	9	12	10	11	79
$T0361$	4	4	19	4	9	10	6	21
$T0386$	12	15	23	18	5	11	25	56
Average	10	11	13	12	17	11	13	37

FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SP3: SPARK3, MV: majority voting, S^*: our method. All values are percentiles.

and PPA-II, are components of Zhang-server, which is ranked the best among all the structure prediction servers on CASP7. Thus, the huge improvement of our method over SVM-LOMETS demonstrates that by revealing the server correlation and optimizing the gap between the true and false contacts, superior contacts can be predicted than those of other consensus methods.

Case Study on Two CASP7 New Fold Targets

As shown in the previous section, our method significantly outperforms the other methods, especially on new fold targets. Two CASP7 new fold targets, T0319 and T0350, are investigated in this section. T0319 (PDB id 2j6a) is a zinc finger protein from the ERF1 methyltransferase complex [80] with 135 residues. T0350 (PDB id 2hc5) is protein yvyC from Bacillus subtilis [49] with 117 residues. Table 9 lists the TM-score [217], contact accuracy, and contact coverage of the best models among the five models submitted by each threading server on T0319 and T0350. All the six threading servers fail to detect correct templates. Typically, a TM-score lower than 0.17 indicates a random structure, and a TM-score higher than 0.4 indicates a meaningful structure [217]. Consequently, all the models predicted by these six servers are probably not meaningful structures.

The hardness of these two targets causes all the six threading servers to fail. Thus, the templates selected by the threading servers are almost random and significantly different from each other, which consequently leads to the failure of the majority voting consensus method. As shown in Figure 7.3, the majority voting method is even worse than some individual servers on these two targets, whereas our method performs significantly better than any individual server. In fact, the top $L/5$ accuracy of our method is 40.0% and 80.2% on these two targets, while the majority voting method achieves an accuracy of only 10.8% and 21.0%, respectively. One may argue that some of the true contacts picked up by our method are strongly

Table 7.9: TM-score, contact accuracy, and contact coverage of the best models by the six individual servers for T0319 and T0350.

		FDP	MGTH	RAP	FUG	SAM	SPK
T0319	$TMscore^a$	0.20	0.18	0.27	0.26	0.12	0.22
	$Accuracy^b$	9%	9%	15%	29%	0	8%
	$Coverage^c$	7%	6%	14%	13%	0	6%
T0350	$TMscore^a$	0.24	0.23	0.33	0.26	0.26	0.27
	$Accuracy^b$	11%	8%	27%	9%	35%	26%
	$Coverage^c$	15%	9%	29%	3%	12%	28%

FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3.
[a] TM-score of the best model among the five submitted models for each server.
[b] Contact accuracy of the best model for each server. All contacts contained in this model are evaluated.
[c] Contact coverage of the best model for each server. All contacts contained in this model are evaluated.

supported by correlated mutation. Even when we remove correlated mutation from our method, its accuracy decreases only slightly, to 39.3% on T0319 and 79.3% on T0350. The minor difference shows that correlated mutation information is not that important for these two targets. Therefore, the case study on these two new fold targets demonstrates that by removing the server correlation and optimizing the best combination of the individual servers, it is possible to select true contacts even if the majority of the individual servers does not support them.

We further demonstrate the usefulness of our method on protein structure prediction by applying it to model ranking, which is one of the major bottlenecks for the state-of-the-art protein structure prediction methods. The most widely used method for model ranking is clustering. However, although clustering based methods work well on easy and medium targets because for such targets, most of the models are high-quality ones and very similar to each other, such clustering methods usually fail on hard targets since the models usually have poor quality and are very different from each other. Thus, we test how well the contacts predicted

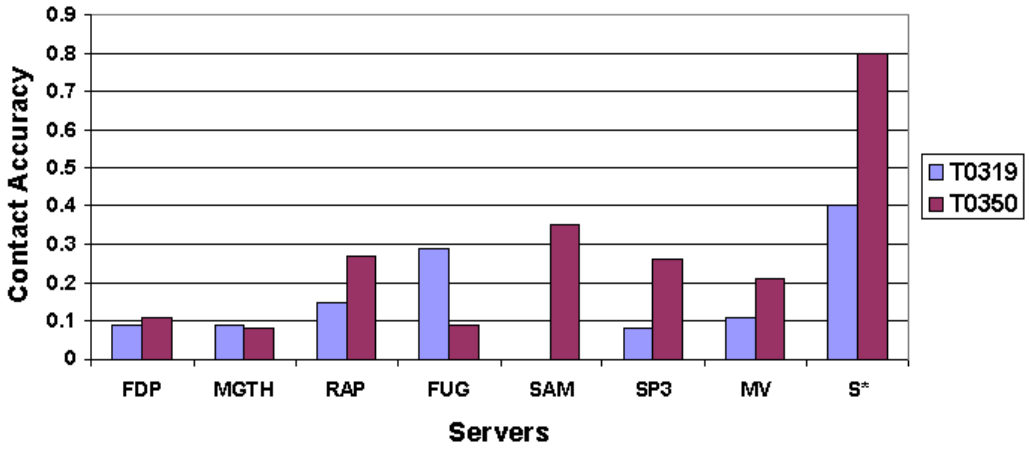


Figure 7.3: Accuracy of top $L/5$ contacts of the six threading servers, the majority voting method, and our method on T0319 and T0350. FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SP3: SPARK3, MV: majority voting, S^*: our method

by our method can rank the models on T0319 and T0350. We design a simple contact ranking score which is very similar to NOE contact-based score proposed in Section 6.1.2. Given the top $L/5$ contacts predicted by our method, for each contact, a model scores 1 if this model indeed contains this contact, and scores 0 otherwise. Table 10 shows the ranking of the best model, in terms of TM-score, of each individual server by their default model ranking method (according to the order of the models submitted to CASP7) and the ranking of the best model by our contact score. It is clear that our contact score has much better ranking of the best models for most cases. Additionally, for both T0319 and T0350, the best models generated by all these six individual servers, i.e., model 4 of RAPTOR for T0319 and model 4 of RAPTOR for T0350, are ranked first among all the models by our contact score. This demonstrates the potential applications of our contact prediction method to select better submitted models or to select good models at each iteration of the refinement process, especially for hard targets.

Table 7.10: The ranking of the best model (in terms of TM-score) for each individual server by its default ranking method and by our contact score for T0319 and T0350.

		FDP	MGTH	RAP	FUG	SAM	SPK
T0319	Default ranking	4	3	4	4	4	3
	Contact ranking	2	1	1	1	3	2
T0350	Default ranking	1	3	4	4	1	4
	Contact ranking	2	1	1	1	3	1

FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3.

7.1.3 Discussion

The experimental results demonstrate that by accounting for the correlation among different threading programs, our consensus method can successfully identify native contacts, even when these contacts are not contained in the majority of the models. It is worth noticing that the proposed method is quite different from the more direct linear combination or non-linear combination of the original individual servers. The underlying reason is that by detecting correlation among the individual servers and removing the last latent server which corresponds to the random noise, our ILP-based optimization process is able to find an optimal solution without the bias caused by the random noise.

A potential application of our contact prediction method is to provide highly conserved constraints for *ab initio* folding or protein structure refinement. Recent research has shown that by incorporating contacts predicted from template-based methods or sequence-based methods, a structural model generated by comparative modeling can be refined [32, 214, 212, 110]. However, if all the individual servers predict the structure for a target protein extremely well or very poorly, our consensus method will probably not help too much. In the former case, since almost all the contact candidates provided by these individual servers are correct ones, our method can only improve the accuracy slightly. In the latter case, since there are

very few correct contact candidates for our method to choose from, the refinement process can hardly benefit from our results. However, in any other case, contacts provided by our method should help with the folding simulation. The reason is that our method can generate a small number of highly conserved contacts. Considering only a small number of contacts can reduce the conformational search space, and thus increase the speed and reduce the chance of generating wrong models. Moreover, experimental results demonstrate that our method can generate contacts with a higher accuracy than both sequence-based and template-based methods. This can reduce the risk of generating models with incorrect contacts, which can reduce the risk of selecting incorrect models from the final decoy set, and thus, greatly increase the overall *ab initio* folding accuracy.

7.2 Side Chain Packing

To solve the side chain packing problem, we study the relationship between local backbone information and side chain conformations, and develop a side chain packing program LocalPack. LocalPack predicts the side chain conformations using local backbone information only and is as accurate as SCWRL, a program that uses pairwise energy function and global search method. We first reformulate side chain packing problem and then solve it using multi-class Support Vector Machines (multi-class SVM). Our method has the following three features: 1) Instead of using the occurring probabilities contained in a rotamer library, our method only uses the angle values of rotamer candidates. 2) Our method does not use any pairwise energy function. Instead, only local backbone information is employed to predict side chain positions. Furthermore, these local backbone features can be calculated extremely fast. 3) Our method does not need to optimize any energy function. By contrast, our method generates a set of linear classifiers based on local backbone

features and then use these classifiers to predict the side chain positions.

7.2.1 New Formulation for Side Chain Prediction

Given a position in a protein backbone sequence, we can calculate a set of backbone related local features at this position. Starting from a rotamer library, our basic assumption is that a certain set of local features can determine the correct rotamer of the side chain at this position.

Table 7.11: An example of the basic assumption: a backbone related feature vector A can determine the rotamer choice. Except for the last column, the first 6 columns show examples of possible backbone related feature vectors. The last column shows χ_1 rotamer values corresponding to the feature vectors.

Res	ϕ	ψ	SS	Solvent Access.	# Contacts	χ_1 Rotamer
ARG	60°	45°	Helix	82.75%	11	63°
PHE	112°	42°	Helix	10.23%	4	114°
GLN	34°	16°	Loop	8.65%	6	125°
MET	156°	107°	Sheet	65.22%	19	178°

Res: residue type; SS: secondary structure type.

Let $\mathcal{A} = \{A_1, A_2, \ldots, A_n\}$ denote the set of feature vectors for a given protein with length n, where vector $A_j = \{a_1^j, a_2^j, \ldots, a_k^j\}$ denote the set of backbone related features at the j-th position, either continuous values, such as solvent accessibility, or discrete values, such as secondary structure and amino acid type. Let $\mathcal{R} = \{r_1, r_2, ..., r_m\}$ denote an arbitrary rotamer set. Table 7.11 shows some examples of feature vectors, according to which the rotamer choice for each residue position is determined.

Based on our assumption, given a rotamer set $\mathcal{R}$, we can consider side chain predictor as a function $f(A_j)$ that maps from a given feature vector A_j to a rotamer.

$f(A_j)$ is defined to be

$$f(A_j) = \arg\max_{i, r_i \in R} h(A_j, r_i), j = 1, \ldots, n \tag{7.11}$$

where $h(A_j, r_i)$ is a scoring function that evaluates the score of assigning the rotamer r_i to the j-th position with feature vector $A_j \in \mathcal{A}$. We aim to find a function $h(A_j, r_i)$ such that $f(A_j)$ matches the correct rotamer choices as well as possible for all the position j.

The formulation (7.11) is based on a general rotamer library $\mathcal{R}$. Studies on backbone-dependent rotamer libraries [44, 45, 46, 47] show that side chains do prefer some rotamers for a fixed amino acid type and a fixed pair of ϕ, ψ backbone dihedral angles. This kind of rotamer libraries can also fit into our model easily by removing the features (*amino acid type*, ϕ, ψ) from vector A_j and finding h on a rotamer library which is an (*amino acid type*, ϕ, ψ)-dependent subset of the original rotamer library $\mathcal{R}$.

7.2.2 A Multi-class SVM Model for Side Chain Packing Problem

A Multi-class SVM Model

We consider side chain prediction problem as described in formulation (7.11) that is a linear function on feature vector A. That is, $h(A_j, r_i) = w_i \cdot A_j$, where w_i is a parameter vector for rotamer i that we want to learn. Thus, according to formulation 7.11, side chain prediction problem can be formulated as a classification problem:

$$f(A_j) = \arg\max_{i, r_i \in R} w_i \cdot A_j, \quad j = 1, \ldots, n, \tag{7.12}$$

in which we want to find such a f that matches correct rotamer choices as well as possible.

To learn the parameter vectors w_i from a training example set $S = \{(A^1, r^1), \ldots, (A^p, r^p)\}$ with size p, where A^j is the feature vector of a residue and r^j is the experimentally determined rotamer of this residue, we apply a multi-class Support Vector Machine (multi-class SVM) model. Multi-class SVM provide powerful approaches to deal with the general problem of learning a mapping from a high dimensional feature space to a discrete set [38]. However, traditional multi-class SVM do not directly fit into the side chain prediction problem. The reason is that the number of rotamer labels is usually very large in the real world, which will result in a large number of constraints in multi-class SVM. This will make the traditional quadratic programming based algorithm unfeasible to solve the side chain prediction problem.

To solve this large class problem, the idea of loss function $\triangle$ from structured SVM [185, 186], a generalized version of multi-class SVM, is applied. Different from multi-class SVM, which is developed to solve classification problems on discrete set $\mathcal{Y} = \{1, \ldots, k\}$, structured SVM is developed to solve classification problems that involve features extracted jointly from the inputs and the outputs, such as sequences, strings, graphs, or labeled trees. Loss function $\triangle$ is widely used in structured SVM [185, 186] to deal with the case in which $|\mathcal{Y}|$ is large. In our method, we use the concept of loss function and define it to be: $\triangle : \mathcal{R} \times \mathcal{R} \rightarrow \{0, 1\}$, where $\triangle(y', y)$ returns 0 if $y' = y$, and 1 otherwise. $\triangle(y', y)$ quantifies how "bad" it is to predict y' when y is the correct label.

Here we use the loss function $\triangle$ to re-scale the margin as proposed by Taskar *et al.* [180] and formulate the problem of finding parameter vectors $w_i,\ i = 1, \ldots, m$ in the form of the following optimization problem:

$$\min_{w_i,\xi_j} \frac{1}{2}\sum_{i=1}^{m} \|w_i\|^2 + \frac{C}{p}\sum_{j=1}^{p} \xi_j \tag{7.13}$$
$$\forall j, l \quad w_{r^j} \cdot A_j - w_l \cdot A_j \geq \triangle(l, r^j) - \xi_j$$

where m is the size of rotamer library, p is the size of training set, $\xi_j \geq 0$ are called *slack variables*. l and r^j are the predicted rotamer and the real rotamer for residue j in the training set. $\|w_i\|$ is the norm of vector w_i, which determines the size of margin in SVM. $C > 0$ is a tradeoff between training error minimization and margin maximization.

We then apply a cutting plane algorithm described in [185] to solve this optimization problem. The basic idea of the algorithm is to find a relatively small set of constraints without losing too much accuracy. They achieved this goal by building a nested sequence which successively tights relaxations of the original problem. It can be proved that:

- Accuracy: the cutting plane algorithm can compute arbitrarily close approximation to the optimal solution.
- Efficiency: the number of steps that the cutting plane algorithm needs to converge is polynomial on the number of data points.

In practice, the cutting plane algorithm works very well on solving our side chain prediction problem, which we will show later. For more details about the algorithm, please refer to [185].

Model Features

The relationship between side chain conformations and backbone dihedral angles (ϕ, ψ) has been well studied. Many side chain prediction programs use a backbone-

dependent or backbone-independent rotamer library. Here we use the backbone-dependent rotamer library [44, 45] developed by Dunbrack *et al.*. The major problem to be addressed is what kind of backbone structure features a side chain conformation depends on. Many works [128, 47, 51] have been done to analyze the relationship between side chain dihedral angles and local backbone features, such as backbone dihedral angles, secondary structure and solvent accessibility. Here we introduce the local structure features used in our prediction and show how to use them in training and testing.

Backbone Dihedral Angles Given an amino acid and a pair of (ϕ, ψ) angles, the backbone-dependent rotamer library can provide a set of candidate side chain conformations. We do not use backbone dihedral angles as features in the training. Instead, we divide training data point into many groups according to the amino acid types and ϕ, ψ angles, and develop a classifier for each group based on its corresponding rotamer subset.

Secondary Structure Secondary structure is local conformation of a protein backbone. Previous works [128] have shown that secondary structure is highly relevant to the distribution of side chain dihedral angles. We use P-SEA [108] to calculate the secondary structure of a given protein backbone. P-SEA can generate the secondary structure type for each backbone position. Since SVM can only take numerical values as input, we use the expected occurring probability of each secondary structure type as its feature value. Let $N(\alpha)$, $N(\beta)$, $N(loop)$ denote the numbers of residues in α-helices, β-sheets and loops in a training data group, and N denote their sum. The expected occurring probabilities are calculated as $N(\alpha)/N$, $N(\beta)/N$ and $N(loop)/N$, respectively.

Solvent Accessibility The accessible surface area is the area of a biomolecule's surface that is accessible to a solvent. It can be calculated by using a sphere of a certain radius to probe the surface of the molecule. A typical radius value is $1.4\mathring{A}$, which approximates the radius of a water molecule. Solvent-accessible surface of atoms have been used to predict conformations of side chains in [51], where they added this term into the energy function during the global optimization and calculated it iteratively. Their results show that the prediction accuracy can be significantly improved by adding the solvent term. This implies the importance of solvent accessibility in modeling side chain conformations. We use Naccess [87] to calculate the backbone solvent accessibility. The output of Naccess is normalized value and we use it as one of our features directly.

Contact Number The contact number of a residue in a protein structure is a quantity similar to, but different from solvent accessible surface area. The contact number of a given residue is defined as the number of C_α atoms within a predefined distance $D(= 8\mathring{A})$ to the C_α atom of this given residue. The contact numbers are scaled to values between 0 and 1 using a standard max-min normalization method, such that the smallest contact number becomes zero and the largest number becomes one.

7.2.3 Results

Implementation Details

LocalPack is implemented in C++. To improve the efficiency of feature calculation, we use a quick K-nearest-neighbor (KNN) algorithm [40, 165] to calculate contact numbers. After extracting backbone related features, such as solvent accessibility, secondary structure, and contact number, these features are encoded into a multi-

class SVM model as described in Section 7.2.2. The SVM model is trained using $SVM^{multiclass}$ [4] with linear kernel function, a program that solves multi-class SVM problem by applying cutting plane algorithm described in [185].

Ten-fold cross-validation is applied on the training set to estimate the best C (see Eq. (7.13)), a tradeoff between model parameter complexity and tolerable model training errors. A big C indicates that a small training error is tolerated but a big model parameter complexity allowed. A model trained using such a C may not generalize well to the test data. Hsu *et al.* showed in [83] that by testing on a sequence of exponentially growing C values, a good model can be identified in practice. Thus, $C = 2^{-5}, 2^{-4}, ..., 2^{20}$ are tried and the best C value is determined accordingly.

Training and Test Sets

Selecting reasonable training and test sets is very important for fairly evaluating the performance of machine learning methods. PDB20 is used as the training set, in which any two proteins do not share more than 20% sequence identity. The proteins in this set with resolution worse than 2Å are removed. This results in a data set of 3060 proteins. For test set, Dunbrack's benchmark set [44], which consists of 180 proteins, is used. Since the rotamer library used here is extracted from a set of 800 proteins [45], we examine the overlap among PDB20, the set of 800 proteins for rotamer library generation, and Dunbrack's benchmark set. It turns out that Dunbrack's benchmark set contains 87 proteins in PDB20 and 102 in the set of proteins for rotamer library generation. Thus, we remove all the overlapping proteins from Dunbrack's benchmark set and obtain a reduced benchmark set of 78 proteins. It can be seen from Figure 7.4 that both the PDB20 training set and the reduced test set are good samples of real world proteins in terms of amino acid composition.

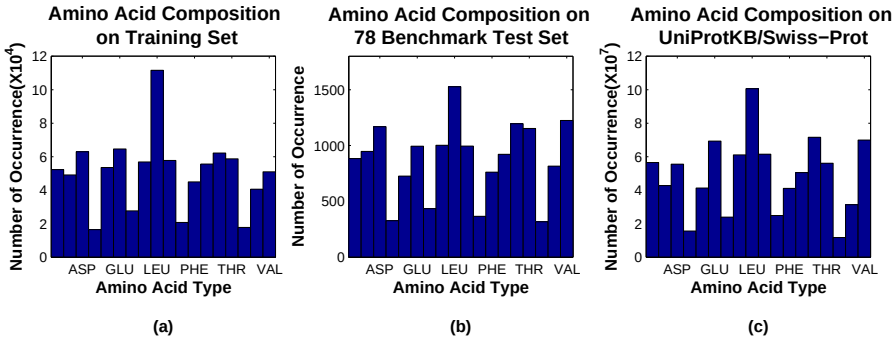


Figure 7.4: The amino acid compositions on PDB20 training set; (b) reduced 78 benchmark test set, and (c) the UniProtKB/Swiss-Prot protein knowledgebase. UniProtKB/Swiss-Prot protein knowledgebase [1] is one of the largest protein sequence databases. The statistics of UniProtKB/Swiss-Prot was taken at 283,454 protein sequences on September 11, 2007.

The performance of our method is evaluated on both this reduced benchmark set and Dunbrack's original benchmark set which has overlapping proteins to our training set. Not surprisingly, the accuracy of our method is approximately 8% higher on the Dunbrack's benchmark set than on the reduced set, while the accuracy of SCWRL3.0 is consistent on the two benchmark sets. Thus, in the following experimental studies, we will only evaluate our method on this reduced benchmark set.

Prediction Accuracy on Native Backbones

The accuracy of our method is compared to the most widely used program SCWRL3.0 in terms of χ_1 and χ_{1+2} accuracy. Other widely used programs, such as Modeller [158], SCAP [201], and TreePack [203], performs no better than SCWRL3.0 on both the 180 benchmark set and the 78 benchmark set. A prediction is considered to be correct if its value is within 40° from its experimental value. The prediction accuracy of one amino acid is calculated as the ratio of the number of

Table 7.12: Prediction accuracy of LocalPack and SCWRL 3.0 on the 78 benchmark set. A prediction of a side chain is correct if its deviation from the experimental value is no more than 40°. χ_1 accuracy of one amino acid is the ratio of the number of correctly predicted χ_1 angles to the total number of this amino acid type, while χ_{1+2} accuracy of one amino acid is the ratio of the number of side chains with both χ_1 and χ_2 being predicted correctly to the total number of this amino acid type.

	LocalPack		SCWRL 3.0	
amino acid	χ_1 accuracy	χ_{1+2} accuracy	χ_1 accuracy	χ_{1+2} accuracy
ARG	0.7701	0.6060	0.7558	0.6226
ASN	0.7888	0.7011	0.7956	0.6882
ASP	0.8322	0.7337	0.8218	0.6974
CYS	0.8497	0.8497	0.8915	0.8915
GLN	0.7493	0.5416	0.7449	0.5319
GLU	0.6841	0.5077	0.7084	0.5128
HIS	0.8226	0.7551	0.8382	0.7745
ILE	0.9172	0.7884	0.9114	0.8060
LEU	0.7851	0.7321	0.8996	0.8142
LYS	0.7678	0.5768	0.7199	0.5444
MET	0.8169	0.6097	0.8160	0.6720
PHE	0.8410	0.7740	0.9361	0.8774
PRO	0.8426	0.7701	0.8517	0.7879
SER	0.7556	0.7556	0.6883	0.6883
THR	0.9193	0.9193	0.8855	0.8855
TRP	0.8328	0.6851	0.8843	0.6688
TYR	0.9239	0.8616	0.9171	0.8615
VAL	0.8922	0.8922	0.9075	0.9075
overall	0.8205	0.7314	0.8266	0.7365

correctly predicted side chains to the total number of side chains of this amino acid type.

As shown in Table 7.12, the overall accuracy of our method is very close to that of SCWRL3.0. In fact, the χ_1 accuracy of our method is only 0.61% lower than that of SCWRL3.0, while the χ_{1+2} accuracy is 0.51% lower. Although our method is based on local backbone information only, it does not lose any accuracy while is much more computationally efficient. In fact, the χ_1 accuracy of our method is higher than SCWRL3.0 on nine out of the eighteen amino acids, especially Lysine

(LYS), Serine (SER) and Threonine (THR). However, our method is much worse than SCWRL3.0 on Cysteine (CYS), Leucine (LEU), Phenylalanine (PHE) and Tryptophan (TRP). Meanwhile, the χ_{1+2} accuracy of our method is higher than SCWRL3.0 on eight out of the eighteen amino acids. This means local backbone information can also determine χ_2 conformation accurately. On the other hand, results shown in Table 7.12 also demonstrate that the accuracy of our method is not worse than any global optimization methods.

We further examine the eight amino acids on which our method did not perform well (with χ_1 accuracy $\leq 82\%$). They are Arginine (ARG), Asparagine (ASN), Glutamine (GLN), Glutamic acid (GLU), Leucine (LEU), Lysine (LYS), Methionine (MET) and Serine (SER). Except for SER, all the other seven amino acids have large side chain groups as shown in Figure 7.5. This result is consistent with the model on which our method is built. Our method assumes that local backbone information can determine side chain conformations. However, if a side chain group is large, its position will be more likely to be impacted by other side chain groups around it and thus cannot be completely determined using only local information. Thus, for such cases, we probably need more information to determine side chain conformations. Interestingly, the global optimization method, SCWRL3.0, which considers all the side chain and backbone atoms around one side chain, performs worse than our method on four out of these seven amino acids as shown in red boxes in Figure 7.5.

Feature Importance Analysis

A key step in feature based machine learning study is to evaluate the importance of each feature encoded. The importance of each feature is evaluated by removing it from the whole set of features, and testing the accuracy on the rest of the feature set. Table 7.13 shows the χ_1 accuracy on different feature sets on amino acid

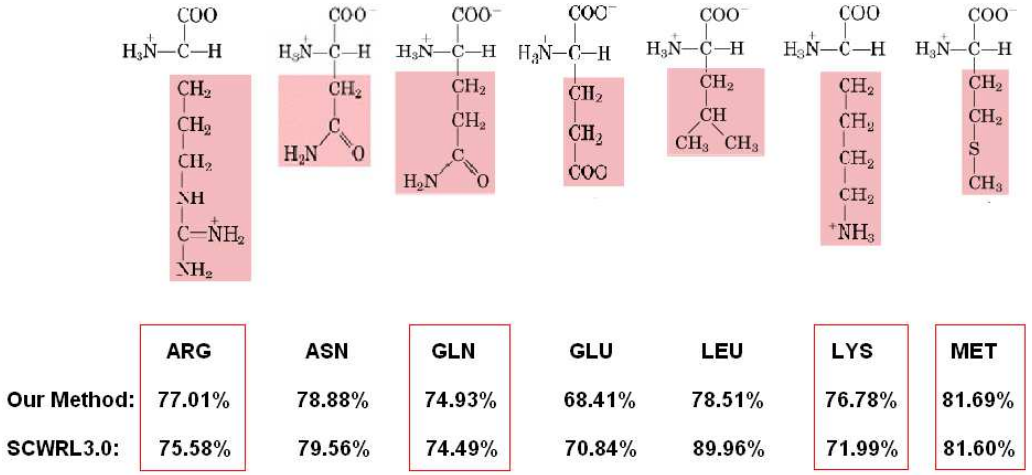


Figure 7.5: The χ_1 accuracy of LocalPack on amino acid types ARG, ASN, GLN, GLU, LEU, LYS, and MET. The four amino acids on which the accuracy of LocalPack is higher than that of SCWRL3.0 are marked in red boxes.

Arginine (ARG). It can be seen from Table 7.13 that all of the three features are important to our method. More specifically, removing solvent accessibility feature will reduce the accuracy by 4.8%, while removing secondary structure and contact number will reduce the accuracy by 3.5% and 3.8%, respectively. This means that solvent accessibility is the most important feature in our method, while secondary structure is the least. This makes sense becuase the backbone-dependent rotamer library [45] has already partially encoded secondary structure information by considering backbone ϕ, ψ angles in their statistics. Similar results are observed on other amino acids as well.

Table 7.13: Feature importance analysis on ARG.

	All	No as	No ss	No cn
χ_1 Accuracy	0.7701	0.7226	0.7352	0.7320

All: the χ_1 accuracy of LocalPack with all the three features; No as: the χ_1 accuracy on feature sets without solvent accessibility; No ss: the χ_1 accuracy on feature sets without secondary structure; No cn: the χ_1 accuracy on feature sets without contact number.

Performance on Non-native Backbones

We further evaluate the accuracy of our method on nonnative backbones. The χ_1 accuracy of our method is compared to four commonly used side chain prediction methods: MODELLER, TreePack, SCWRL3.0, and SCAP, on a nonnative backbone test set provide by Xu *et al.* in [203]. The test set contains prediction models generated by a protein threading program, RAPTOR [204], on 24 CASP6 test proteins [2]. RAPTOR generated good alignments for most of these targets. MODELLER [158] was called by RAPTOR to generate model backbones according to the alignments. Besides, MODELLER is also able to predict side chains based on a statistical method. SCAP was tested using the CHARMM force field with the heavy atom model and the largest rotamer library available to SCAP.

The overall χ_1 accuracy is shown in Table 7.14. The prediction accuracy of our method is the same as TreePack, and slightly worse than SCWRL3.0, while much better than MODELLER and SCAP. This indicates that our method also works well on nonnative backbones.

Table 7.14: The overall χ_1 accuracy of MODELLER, TreePack, SCWRL3.0, SCAP, and LocalPack on the 24 nonnative test proteins.

	MODELLER	TreePack	SCWRL3.0	SCAP	LocalPack
χ_1 Accuracy	0.428	0.520	0.530	0.488	0.520

Computational Efficiency

Since our method is based on only local backbone features, it can be expected that our method is much more computationally efficient. TreePack has been reported as one of the fastest methods for side chain prediction. Table 7.15 shows the total CPU time comparison of TreePack, SCWRL3.0, and our method on the 78 benchmark set. All three programs are tested on a Debian Linux box with a 1.7GHz CPU.

Table 7.15: CPU time comparison of TreePack, SCWRL3.0, and LocalPack on the 78 protein benchmark set.

	TreePack	SCWRL3.0	LocalPack
Time	186 seconds	657 seconds	46 seconds

From Table 7.15, it is clear that our method is much faster than both TreePack and SCWRL3.0. In fact, LocalPack is more than 14 times faster than SCWRL3.0, and more than 4 times faster than TreePack. The average CPU time of our method on one test protein is 0.58 seconds.

7.2.4 Discussion

We demonstrate that protein side chain positions can be predicted using local backbone information to the same accuracy as those programs employing pairwise energy functions and computationally-intensive optimization algorithms, such as SCWRL and TreePack. We hope our discovery will change the way researchers look at this problem and lead to rapid and accurate protein side chain packing programs, which are indispensable in high-accuracy protein structure modeling.

One of the major bottlenecks in protein structure refinement is how to quickly generate a huge number of possible all-atom conformations so that an all-atom energy function can be used to pick up the energetically most favorable conformations. Our method enables us to generate a good side chain packing extremely fast after a change of backbone conformation. Since our method depends on local backbone information only, it can be even much faster when only a local part of a protein structure is refined. This allows us to do side chain packing at each step of protein structure refinement and thus makes it feasible to apply an accurate full-atom energy function to each generated conformation.

7.3 Local Quality Assessment

To solve the local quality assessment problem, we develop two complementary methods FragQA and PosQA to accurately predict local quality of a sequence-structure alignment. Distinguishing itself from previous methods, FragQA directly predicts the quality of an ungapped region in the alignment. The quality is measured using the RMSD (*i.e.*, C_α-based RMSD) between two fragments corresponding to the ungapped region: one is the native structure of the region and the other one is the predicted model. Note that the quality measurement used here is "absolute" quality, which is independent of the optimal structure alignment. Furthermore, statistical significance is introduced to improve FragQA's performance. As opposed to RMSD, statistical significance can cancel out the impact of region length. Complementary to FragQA, recently developed PosQA predicts the quality of an individual aligned position in a given alignment. The single position quality is measured using a normalized RMSD described in [192]. FragQA and PosQA utilize only information in a single alignment. Structural information in the alignment-derived protein model is not directly used. However, in calculating features from an alignment, we use structural information in the template.

7.3.1 Methods

Development of FragQA

Our SVM regression model uses only features extracted from a single sequence-template alignment, generated by any comparative modeling program (*i.e.*, homology modeling and threading). To exploit the evolutionary information of proteins, sequence profiles of both the target protein and the template protein are utilized in calculating features. The sequence profile of the template, denoted by

$PSSM_{template}$ (position specific mutation matrix), is generated by PSI-BLAST [11] with five iterations; $PSSM_{template}(i, a)$ encodes mutation information for amino acid a at position i of the template. PSI-BLAST is also applied with five iterations to generate position specific frequency matrix, $PSFM_{target}$, for each target protein; $PSFM_{target}(j, b)$ encodes occurring frequency of amino acid b at position j of the target. Let $A(i)$ denote the aligned sequence position of template position i, and T_{temp} denote the set of template positions belonging to an aligned region. A variety of features extracted from the alignment are explored, and their relative importance is studied in Section 7.3.2. In summary, the following features are tested in FragQA:

1. *Mutation score*: Mutation score measures the sequence similarity between two segments of an aligned region: one corresponds to the target protein and the other to the template. The mutation score (S_m) of a region is calculated as:

$$S_m = \sum_{i \in T_{temp}} \sum_{a} PSFM_{target}(A(i), a) \times PSSM_{template}(i, a). \tag{7.14}$$

2. *Environmental fitness score*: This score measures how well to align one target protein region to the environment where the corresponding template region lies in. The environment consists of two types of local structure features.

   - Three types of secondary structure are used: α-helix, β-strand, and loop.
   - Solvent accessibility: There are three levels: buried (inaccessible), intermediate, and accessible. The Equal-Frequency discretization method is used to determine boundaries between these three levels. The calculated boundaries are 7% and 37%.

   Thus, there are nine environment combinations (denoted as *env*) in total.

Define $F(env, a)$ to be the environment fitness potential for amino acid a and environment combination env. $F(env, a)$ is calculated and taken from PROSPECT-II [99]. For more details about $F(env, a)$, please refer to [99]. The environment fitness score (S_e) for an aligned region is calculated as:

$$S_e = \sum_{i \in T_{temp}} \sum_{a} PSFM_{target}(A(i), a) \times F(env_i, a). \tag{7.15}$$

3. *Secondary structure score*: In addition to the secondary structure information encoded in environmental fitness score, we also use $SS(i, A(i))$, the secondary structure difference between position i in template and position $A(i)$ in target, to measure the quality of an ungapped region from another aspect. PSIPRED [95] is called to predict the secondary structure of the target protein. Let $\alpha(i)$, $\beta(i)$, and $loop(i)$ denote the predicted confidence levels of α-helix, β-sheet, and loop at sequence position i, respectively. If the secondary structure type at template position i is α-helix, then $SS(i, A(i)) = \alpha(A(i)) - loop(A(i))$. If the secondary structure type at template position i is β-sheet, then $SS(i, A(i)) = \beta(A(i)) - loop(A(i))$. Otherwise, we set $SS(i, A(i))$ to be 0. The secondary structure score (S_{ss}) of an ungapped region is calculated as:

$$S_{ss} = \sum_{i \in T_{temp}} SS(i, A(i)). \tag{7.16}$$

4. *Contact capacity score*: Contact capacity potentials describe the hydrophobic contribution of free energy, measured by the capability of a residue making a certain number of contacts with other residues in the protein. Two residues are in physical contact if the spatial distance between their C_β atoms (C_α for glycine) is smaller than 8Å. Let $CC(a, k)$ denote the contact potential of

amino acid a having k contacts. $CC(a,k)$ is calculated by statistics on PDB as:

$$CC(a,k) = -log\frac{N(a,k)N}{N(k)N'(a)}, \tag{7.17}$$

where $N(a,k)$ is the number of amino acid a with k contacts; $N(k)$ is the number of residues with k contacts; $N'(a)$ is the number of amino acid a; and N is the total number of residues in PDB. Let $C(i)$ denote the number of contacts at template position i. The contact capacity score (S_c) is calculated as:

$$S_c = \sum_{i \in T_{temp}} \sum_{a} PSFM_{target}(A(i), a) \times CC(a, C(i)). \tag{7.18}$$

5. *Aligned region length*: The RMSD between the two fragments of an ungapped region is relevant to its length. The longer the ungapped region is, the more likely larger the RMSD is.

6. *G-score*: G-score measures the overall quality of a sequence-structure alignment. An alignment with a good G-score likely contains more good ungapped regions. G-score is calculated by Xu's SVM module [202] as the predicted alignment accuracy normalized by the target protein size. G-score ranges from 0 to 1. G-score equal to 0 means the alignment is likely random, while 1 means it is probably a perfect alignment.

7. *Sequence identity*: The fraction of identical residues in the whole alignment is used to measure the sequence identity.

8. *Other sequential features*: Three separate sequential features are tested: template protein size, target protein size, and alignment length (*i.e.*, the number

of aligned positions).

Meanwhile, mutation score, environmental fitness score, secondary structure score, contact capacity score, and aligned region length are specific to the ungapped region; while G-score, sequence identity, and other sequential features are for the whole sequence-structure alignment.

Development of PosQA

Instead of directly using RMSD between the native C_α position and the predicted position of a residue, a normalized RMSD is used as the objective function of PosQA. Let D_i and d_i denote the normalized RMSD and RMSD at position i, respectively. Then D_i is defined as $1/(1+\frac{(d_i)^2}{(d_0)^2})$ [192] where d_0 is set to $\sqrt{5}$. Thus, the larger the D_i is, the higher the quality of this position is.

PosQA uses almost the same set of features as FragQA. In particular, PosQA tests the following information: (1) mutation score, (2) environmental fitness score, (3) secondary structure score, (4) contact capacity score, and (5) G-score. The only difference between PosQA and FragQA is that the values of the first four features are calculated at a single position.

7.3.2 Results

Problem description

We study the following two problems:

1. Given a sequence-structure alignment, what is the quality of an ungapped region in this alignment? The quality is defined as the RMSD between the native and the predicted local structures of the ungapped region, denoted

as "RMSD of an ungapped region", after they are optimally superimposed. Note that the two local structures are superimposed without taking into consideration other parts of the alignment. The alignment is cut into ungapped regions at gap positions. Thus, the fragments studied here are different from the fixed-length fragments studied in [55, 153]. FragQA is developed to solve this problem.

2. Given a sequence-structure alignment, what is the quality of a single aligned position in this alignment? To measure the quality of a single position, we optimally superimpose the predicted structural model, derived from this alignment, and the native structure, and then calculate RMSD at each position to measure its quality. The final quality measure is normalized RMSD as described in [192]. Different from the quality measure of an ungapped region, the single-point quality depends on the superimposition between the whole predicted model and its native structure. PosQA is developed to solve this problem.

FragQA Training

Training and Test Data. Alignments generated by RAPTOR default threading algorithm (with NoCore option) on the CASP7 target proteins are used as the training and test data. As suggested by Fasnacht *et al.* [55], CASP data set is the most practical and challenging set, which covers a very broad range of types of target proteins and local errors. There are 104 target proteins in CASP7 while 89 of them were considered as valid targets and were used for final assessment by CASP7 assessors. Eighty-eight target proteins are left after we removed redundancy at 40% sequence identity level using CD-HIT [118]. Only T0346 is removed because it shares 71% sequence identity with T0290. To do a cross validation, the 88 target

Table 7.16: Statistics of ungapped regions on the four data sets.

Set Name	# of proteins	# of fragments	Average RMSD	Deviation
1	22	1347	2.93Å	1.50Å
2	22	1108	2.57Å	1.46Å
3	22	1519	2.86Å	1.47Å
4	22	1461	2.73Å	1.49Å

Columns 2-5 show the number of target proteins, the number of fragments, the average quality in terms of RMSD of the fragments, and the standard deviation of RMSD of each set, respectively.

proteins are randomly divided into four sets. Top 10 alignments generated by RAPTOR are considered for each target protein. If one target protein belongs to a set, then all of its 10 alignments belong to this set. Each alignment is cut into a set of ungapped regions with cutting points being at the gap positions. The ungapped regions containing less than 5 residues are not considered in our experiments. Table 7.16 shows the statistics on the four sets. It is clear that the four data sets are very similar.

Training. SVM-light [92] with RBF (radial basis function) kernel is used to train FragQA. The parameter gamma in the RBF kernel function is trained using the leave-one-out error estimation method. Other parameters are set to their default values or calculated automatically by SVM-light. Experimental results indicate that the RBF kernel with its gamma parameter set to 0.2 can yield the best training performance. Other kernel functions such as linear kernel and polynomial kernel are also tested, but they cannot yield as good performance as the RBF kernel.

A 4-fold cross validation is applied. Each time three of the four data sets are used as the training set, and the other one is used for testing.

Performance of FragQA

After studying the relative importance of eight features, which will be discussed later, following features are encoded into FragQA: (1) length of the ungapped region, (2) G-score of the whole alignment, (3) mutation score of the region, (4) environmental fitness score of the region, and (5) secondary structure score of the region.

Comparing to ProQres To the best of our knowledge, FragQA is the first method to directly predict the quality of fragments that are automatically determined by the sequence-structure alignments rather than fragments with fixed length. Thus, there is no existing method for us to compare with. However, there are some well-known methods that predict local quality for each residue. So it is possible to convert the prediction on residues by such methods to a prediction of a fragment. Since the objective function of FragQA is RMSD, to fairly evaluate FragQA, FragQA is compared to a top-notch method ProQres [192], which uses a residue-based RMSD-related objective function. All three available methods by ProQ-group are tested in terms of the ability to predict fragment quality: ProQlocal, ProQres, and ProQprof. ProQres yields the best results (slightly better than ProQlocal and ProQprof in terms of RMSD prediction of fragments). For the sake of clearness, only the comparison between FragQA and ProQres is shown. The objective function of ProQres is $D_i = 1/(1 + \frac{(d_i)^2}{(d_0)^2})$ [192], where d_i denotes the RMSD at position i, and d_0 is set to $\sqrt{5}$. From the prediction of ProQres, d_i is calculated from D_i for each residue of a fragment, then the predicted RMSD by ProQres for the fragment is calculated by $RMSD = \sqrt{\frac{1}{n}\sum_{i=1}^{n} {d_i}^2}$, where n is the length of the fragment. Note RMSD calculated by this way has a slightly different meaning to the one used by FragQA, because this RMSD is based on the optimal superimposition between the whole target and the template on all similar regions,

while FragQA's RMSD is based on the optimal superimposition between two local regions. However, the superimposition between two aligned regions determined by the optimal superimposition of the whole target and template is usually very similar to the optimal one between the two regions, because aligned regions are usually similar. Thus, FragQA and ProQres are comparable from this point of view.

Prediction Error and Correlation Coefficient of FragQA The prediction error is defined as the absolute difference between the predicted RMSD value and the real one. Table 7.17 lists the average prediction errors of FragQA and ProQres, under different RMSD thresholds on the four test sets, together with the average fraction of fragments with real RMSD under such thresholds, and the correlation coefficient between the predicted and real RMSD by FragQA and ProQres on the four test sets. As shown in this table, the prediction error of FragQA ranges from 0.9Å to 1.6Å, whereas the error of ProQres ranges from 0.9Å to 2.4Å. In most cases, the prediction error of FragQA is much smaller than that of ProQres. In fact, when there is no restriction on RMSD, the error of FragQA is on average 0.5Å smaller than that of ProQres. The smallest error of FragQA happens when RMSD threshold is set to 3Å, which means FragQA is most accurate when dealing with fragments with RMSD to native smaller than 3Å. However, when the real RMSD is very small ($\leq 1\mathring{A}$), the prediction error tends to be big. In other word, it is hard to obtain an accurate prediction when RMSD is very small. As indicated in Table 7.17, the correlation coefficient between the predicted RMSD by FragQA and the real RMSD is about 0.5 for each test set, while that of ProQres is at most 0.22. This makes sense because FragQA is trained to predict the "absolute" quality of a fragment, while ProQres is trained to predict the displacement of a single residue.

Table 7.17: Comparison of prediction accuracy of FragQA and ProQres.

RMSD	Test Set 1		Test Set 2		Test Set 3		Test Set 4		Fra.
	FQA	PQr	FQA	PQr	FQA	PQr	FQA	PQr	
≤1Å	1.36	1.50	1.57	1.10	1.41	1.35	1.54	1.30	14%
≤2Å	1.11	1.06	1.28	0.90	1.08	1.01	1.18	1.01	42%
≤3Å	1.00	1.84	1.16	1.12	0.94	0.98	1.04	1.01	69%
≤4Å	1.03	1.79	1.12	1.23	0.97	1.21	1.04	1.13	85%
≤5Å	1.12	1.88	1.14	1.34	1.06	1.37	1.09	1.34	92%
≤6Å	1.20	1.98	1.19	1.46	1.16	1.50	1.20	1.51	95%
≤7Å	1.33	2.13	1.26	1.57	1.22	1.58	1.25	1.62	97%
≤8Å	1.41	2.20	1.32	1.68	1.29	1.67	1.31	1.72	98%
≤9Å	1.48	2.27	1.36	1.73	1.37	1.78	1.36	1.77	99%
≤10Å	1.57	2.37	1.39	1.77	1.41	1.84	1.41	1.83	99%
Correlation Coefficient	0.51	0.07	0.46	0.22	0.50	0.22	0.48	0.16	-

Column 1 lists different RMSD thresholds. Column 2-9 list prediction errors of FragQA (denoted as FQA) and ProQres (denoted as PQr), under different RMSD thresholds on the four test sets. Column 10 lists average fraction of fragments with real RMSD under such thresholds.

Sensitivity and Specificity Given a RMSD threshold, sensitivity is calculated as the fraction of ungapped regions with real RMSD smaller than the threshold, that are also predicted to be smaller than the threshold. Specificity measures the fraction of ungapped regions with predicted RMSD under a given threshold, that indeed have RMSD smaller than the threshold. Figure 7.6 illustrates the sensitivity and specificity of FragQA and ProQres under various RMSD thresholds on the four test sets.

As shown in Figure 7.6, there is no obvious difference between sensitivity or specificity of FragQA and ProQres when RMSD is larger than 4Å. When RMSD is smaller than 4Å, the sensitivity of ProQres is higher than that of FragQA for most cases, while the specificity of FragQA is higher than that of ProQres. In particular, when RMSD threshold is 2.5Å, approximately 70% of ungapped regions with predicted RMSD by FragQA under 2.5Å indeed have RMSD less than 2.5Å, while 70% of ungapped regions with real RMSD under this threshold are predicted

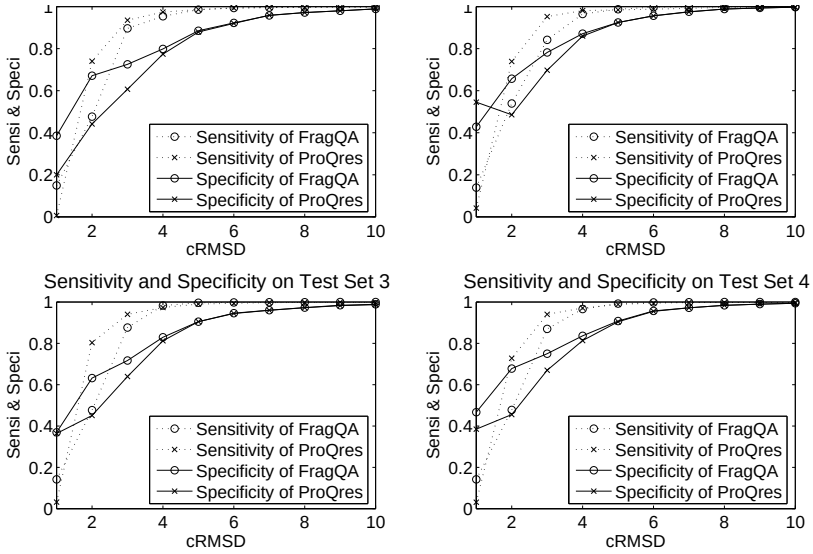


Figure 7.6: Comparison of sensitivity and specificity between FragQA and ProQres under different RMSD thresholds on the four test sets. Circle dotted line: sensitivity of FragQA, cross dotted line: sensitivity of ProQres, circle solid line: specificity of FragQA, cross solid line: specificity of ProQres.

by FragQA correctly. However, the sensitivity of ProQres on RMSD 2.5Å is about 80%, while the specificity is only 50%. This implies that ProQres has a strong trend to predict the RMSD of a fragment to be smaller than the real value. This makes ProQres to have a high sensitivity but a low specificity. As shown in Figure 7.6, the specificity curves of ProQres are not smooth sometimes, nor it is monotonous. It is clear that for both FragQA and ProQres, the sensitivity curves increase much more quickly than the specificity curves. However, all curves are quite low when RMSD is small. A possible explanation is that when the ungapped region is short, a small RMSD does not necessarily mean that this region has a good quality. Therefore, it is hard for FragQA to predict RMSD accurately under such cases. Later we will replace RMSD with its statistical significance and show that when statistical significance is high, even as high as 1, FragQA can still yield a good prediction.

Feature Selection for FragQA It is important to detect which features are closely relevant to the prediction capability of FragQA since unrelated features may introduce extra noise. The importance of each feature is investigated by excluding it from the feature set, training a new FragQA, and then testing the performance of this new predictor. Thus, the performance resulting from different sets of features can be compared, and the important features can be detected.

Table 7.18 lists the sensitivity and specificity of FragQA with different sets of features under different RMSD thresholds on test set 1. The results are similar on the other test sets. There is no obvious difference among different sets of features when RMSD threshold is larger than 3.75Å. As shown in this table, if the aligned region length is removed, the performance of FragQA will drop obviously, except for RMSD threshold larger than 2.75Å, the sensitivity of FragQA without fragment length is a little higher than that with all the features. This complies with a fact that RMSD itself is closely related to the length of an ungapped region. Removing mutation score or the overall G-score will also have an obvious reduction on the performance of FragQA, except for RMSD larger than 2.25Å, removing G-score will increase the sensitivity slightly and have no obvious influence on the specificity. This also makes sense: mutation score measures the sequence similarity in the aligned region, and G-score evaluates the overall quality of the alignment. An alignment with good overall quality often contains good aligned regions. However, when the overall quality of an alignment is poor (G-score is low), the fragments can be either good or bad. In such case, G-score will not be an influential factor any more. Removing environmental fitness score will decrease both the sensitivity and the specificity. Surprisingly, removing contact capacity score will increase both the sensitivity and the specificity. This implies contact score is a noisy feature. On the other hand, removing secondary structure score will decrease the specificity but increase the sensitivity slightly. Removing any other features, such as

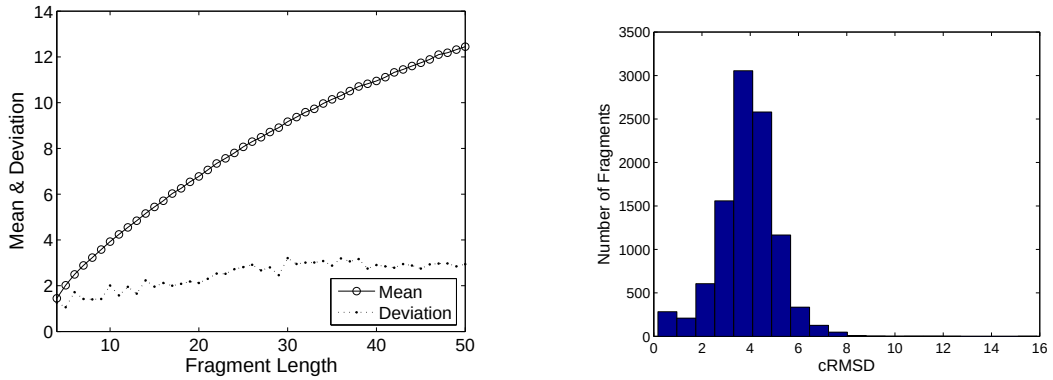
Table 7.18: Sensitivity and specificity of FragQA with different feature sets.

RMSD	All	$\bar{Len}$	$\bar{S_g}$	$\bar{S_m}$	$\bar{S_e}$	$\bar{S_c}$	$\bar{S_{ss}}$	$\bar{SeqId}$	$\bar{Other}$
≤1Å	12/19	0/0	4/10	9/17	11/16	13/32	13/17	12/18	12/18
≤1.25Å	16/28	1/22	8/20	15/27	14/22	22/43	18/27	16/28	15/28
≤1.5Å	25/42	4/23	16/37	19/35	2236	27/49	26/41	25/42	25/41
≤1.75Å	35/52	12/41	27/51	27/46	29/47	34/57	36/51	34/52	35/52
≤2Å	42/59	21/48	38/58	35/53	39/57	48/65	42/56	43/60	42/59
≤2.25Å	50/64	42/56	52/64	46/60	48/62	58/68	51/63	51/64	51/64
≤2.5Å	62/72	61/63	64/70	55/66	56/69	65/73	63/70	62/72	62/72
≤2.75Å	70/78	74/67	73/75	65/73	67/76	74/78	71/77	69/78	69/77
≤3Å	76/79	82/70	79/77	74/77	75/80	81/79	77/79	76/79	76/79
≤3.25Å	83/82	90/75	86/80	82/81	80/83	85/82	84/80	83/82	83/82
≤3.5Å	88/86	94/79	90/84	88/83	84/85	89/86	89/84	88/86	88/86

Column 1 lists different thresholds. Column 2 lists the sensitivity/specificity of FragQA with all features. Starting from column 3, each column lists the sensitivity/specificity when one feature is removed. Len: region length, S_g: G-score, S_m: mutation score, S_e: environmental fitness score, S_c: contact capacity score, S_{ss}: secondary structure score, $SeqId$: sequence identity, and $Other$: other sequential features. All values are percentiles.

sequence identity feature and other sequential features, does not obviously deteriorate either the sensitivity or the specificity. Thus, the final version of FragQA uses the following features: (1) aligned region length, (2) overall alignment G-score, (3) mutation score, (4) environmental fitness score, and (5) secondary structure score. Meanwhile, mutation score, G-score, and the region length are the most important factors in quality prediction.

Statistical Significance The RMSD between the predicted structure of an ungapped region and its native is closely relevant to the length of the region. Thus, a 5-residue ungapped region with 3Å RMSD may not be better than a 15-residue region with 4Å RMSD. To better evaluate the quality of a region, the statistical significance of its RMSD is calculated to reduce the bias introduced by region length. To calculate statistical significance, statistical distribution of RMSD for a



(a) (b)

Figure 7.7: (a) Mean (circle solid line) and standard deviation (point dotted line) of RMSD for random region sets with length from 5 residues to 50 residues. (b) The statistical distribution of RMSD calculated from 10,000 randomly sampled pairs of fragments with length 10.

given region length is empirically calculated as follows. For a given region length, 10,000 pairs of fragments of this length are randomly sampled from PDB30, and their pairwise RMSDs are calculated. PDB30 is a subset of PDB (the Protein Data Bank) [19], in which any two proteins share no more than 30% sequence identity. As shown in Figure 7.7(a), the mean of RMSD increases clearly with respect to the length, but the standard deviation increases much more slowly. The RMSD distribution looks like a normal distribution. Figure 7.7(b) shows the statistical distribution of RMSD calculated from 10,000 randomly sampled pairs of fragments with length 10. Fragments with different length give similar distributions. For a given ungapped region with length l and (real or predicted) RMSD r, its statistical significance (denoted as $StatSig$) is calculated as follows:

$$StatSig = \frac{\#random\ pairs\ of\ length\ l\ with\ RMSD \geq r}{10,000}. \tag{7.19}$$

Thus, the smaller the RMSD is, the larger its statistical significance is.

The sensitivity and specificity of FragQA in terms of statistical significance is

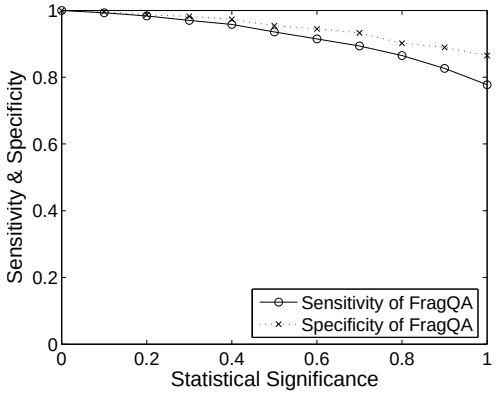


Figure 7.8: FragQA's sensitivity (circle solid line) and specificity (cross dotted line) in terms of statistical significance on test set 1.

calculated in a way similar to that calculates them in terms of RMSD. For each statistical significance threshold varying from 0 to 1, the sensitivity is defined as the percentage of ungapped regions with real statistical significance larger or equal than the threshold, that also have predicted values larger or equal than the threshold. The specificity is defined as the percentage of ungapped regions with predicted significance larger or equal than the threshold, that have real statistical significance better or equal than the threshold. Figure 7.8 illustrates the sensitivity and specificity of FragQA in terms of statistical significance on test set 1. Results are similar on the other three sets. As shown in this figure, when statistical significance is 0.8 (about 81% fragments in our test sets have such values), both the sensitivity and specificity is around 90%. Even when statistical significance threshold is 1 (about 48% fragments in our test sets have this value), the sensitivity is 78%, and the specificity is 88%.

The prediction error of FragQA in terms of statistical significance is also studied. As shown in Table 7.19, the prediction error decreases quickly from 0.26 to 0.05 when the statistical significance threshold increases from 0 to 1. When the threshold is 0.9, the prediction error is approximately 0.12. This indicates that FragQA is able to predict the statistical significance well when the ungapped region has a

Table 7.19: Prediction errors of FragQA in terms of statistical significance.

StatSig	Whole	High-quality	Medium-quality	Low-quality
≥0	0.26	0.21	0.25	0.28
≥0.1	0.23	0.20	0.23	0.25
≥0.2	0.21	0.19	0.21	0.22
≥0.3	0.19	0.16	0.18	0.20
≥0.4	0.17	0.14	0.17	0.18
≥0.5	0.15	0.12	0.16	0.16
≥0.6	0.14	0.10	0.15	0.14
≥0.7	0.13	0.08	0.14	0.14
≥0.8	0.12	0.08	0.14	0.13
≥0.9	0.12	0.08	0.14	0.13
=1.0	0.05	0.03	0.04	0.08

Column 1 lists different significance thresholds. Column 2 lists the overall prediction errors of FragQA. Columns 3-5 are the prediction errors on the three classes of alignments: "high-quality", "medium-quality", and "low-quality".

good quality. By contrast, FragQA is not able to accurately predict RMSD when it is small because a small RMSD does not imply a high-quality region. This result also shows that statistical significance is a better measure than RMSD. All the test alignments are further divided into three classes, "high-quality" alignments, "medium-quality" alignments, and "low-quality" alignments, based on their G-scores (calculated by RAPTOR) at cutting points 0.33 and 0.66. A "high-quality", "medium-quality", and "low-quality" alignment has G-score at least 0.66, between 0.33 and 0.66, and less than 0.33, respectively. Table 7.19 indicates that different sets have different prediction errors. The underlying reason may be that different sets have different distributions of ungapped regions under a given threshold.

On the other hand, the correlation coefficient of FragQA on each set in terms of statistical significance is higher than 0.60. This means statistical significance is probably a better way to measure the quality of a fragment.

PosQA Training

PosQA uses the same data source as FragQA to train and test the SVM model. The only difference is that a data entry in FragQA is an ungapped region while a data entry in PosQA is a single aligned position. If a residue in the target protein is aligned to a gap, the quality of this position is set to zero, and this residue is not used for training or test. The whole CASP7 data set is also divided into four sets as in FragQA. In summary, there are 26,432, 27,018, 26,982, and 26,831 entries in the four sets, respectively. Their average D_i's are 0.57, 0.51, 0.52 and 0.54, respectively.

The SVM-light software [92] is also applied to train PosQA with the RBF kernel, following almost the same procedure to train FragQA. Experimental results indicate that PosQA yields the best performance when the RBF kernel function is used with gamma being 0.3. After selecting features by using the similar approach used by FragQA, PosQA encodes the following features: (1) overall alignment G-score, (2) mutation score, (3) environmental fitness score, and (4) secondary structure score. Again, contact capacity score has no contribution to the performance of PosQA, and is thus not encoded in PosQA.

Performance of PosQA

Prediction Error of PosQA We compare the prediction error of PosQA, ProQres, and ProQprof, which is defined as the average absolute difference between the predicted D_i and its real value. Table 7.20 shows the prediction errors above different D_i thresholds. As shown in this table, the overall prediction errors for PosQA, ProQres, and ProQprof range from 0.13 to 0.29, 0.14 to 0.41, and 0.15 to 0.40, respectively. This implies that the overall prediction accuracy of PosQA is better than that of ProQres and ProQprof. When D_i increases, the overall prediction errors of PosQA decrease clearly, while the lowest errors of ProQres and ProQprof

Table 7.20: Comparison of prediction errors of PosQA (PA), ProQres (Pr), and ProQprof (Pp).

D_i	Whole			High-quality			Medium-quality			Low-quality		
	PA	Pr	Pp	PA	Pr	Pp	PA	Pr	Pp	PA	Pr	Pp
0	0.29	0.41	0.40	0.27	0.36	0.44	0.29	0.47	0.54	0.29	0.41	0.20
0.1	0.28	0.31	0.35	0.27	0.26	0.32	0.29	0.31	0.36	0.29	0.39	0.37
0.2	0.26	0.26	0.29	0.25	0.22	0.27	0.26	0.26	0.30	0.29	0.31	0.30
0.3	0.23	0.22	0.24	0.21	0.19	0.23	0.22	0.22	0.26	0.27	0.25	0.24
0.4	0.22	0.18	0.20	0.20	0.16	0.19	0.21	0.18	0.22	0.25	0.22	0.20
0.5	0.21	0.16	0.17	0.18	0.14	0.15	0.20	0.15	0.18	0.23	0.19	0.18
0.6	0.19	0.14	0.15	0.16	0.13	0.12	0.19	0.13	0.15	0.20	0.18	0.19
0.7	0.17	0.15	0.15	0.15	0.12	0.10	0.15	0.12	0.14	0.21	0.21	0.24
0.8	0.15	0.16	0.17	0.14	0.14	0.10	0.10	0.14	0.13	0.20	0.22	0.29
0.9	0.13	0.19	0.19	0.13	0.17	0.13	0.12	0.17	0.13	0.24	0.25	0.33

Column 1 lists different D_i thresholds. Columns 2-13 list the prediction errors of PosQA (denoted as PQA), ProQres (denoted as PQr), and ProQprof (denoted as PQp) on the whole set, "high-quality" alignments, "medium-quality" alignments, and "low-quality" alignments, respectively.

happen when D_i threshold is 0.6. Recall that a large D_i indicates a high-quality position. This means that PosQA predicts the well-aligned positions better than ProQres and ProQprof.

All the test alignments are also divided into three classes: "high-quality" alignments, "medium-quality" alignments, and "low-quality" alignments, based on their G-scores (calculated by RAPTOR) at cutting points 0.33 and 0.66. Table 7.20 shows the prediction errors of PosQA, ProQres, and ProQprof on the three classes of alignments. It is clear that different sets have different prediction errors, which means G-score is an informative factor for local quality. For all the three classes, the overall errors, which correspond to $D_i \geq 0$, and the errors on high-quality residues, which correspond to $D_i \geq 0.9$, of PosQA are better than those of ProQres and ProQprof. However, ProQres outperforms the other two methods on both "high-quality" and "medium-quality" alignments, whereas PosQA is the best method on "low-quality" alignments. This makes sense because ProQres and ProQprof are

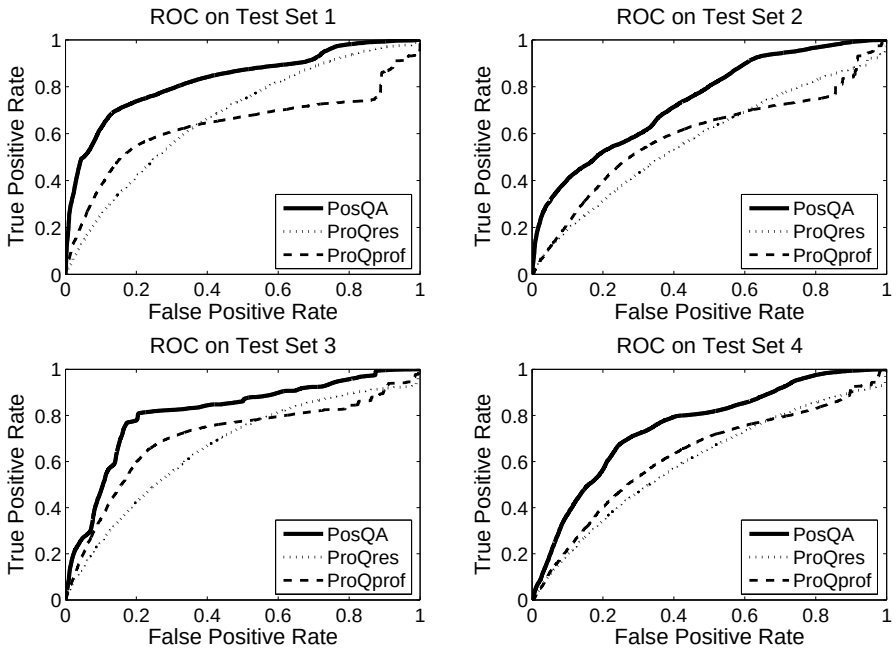


Figure 7.9: ROC curves for PosQA, ProQres, and ProQprof on the four test sets. Discrimination threshold 4Å.

both trained on high-quality models and alignments, while PosQA is trained on the comprehensive set of CASP7 targets, which contains high-quality (HA) targets, template based modeling (TBM) targets, as well as free modeling (FM) targets.

Sensitivity and Specificity Receiver Operating Characteristic (ROC) plots are used to evaluate the trade-off between the ability of PosQA, ProQres, and ProQprof to correctly identify positive cases and the number of negative cases that are incorrectly classified. Figure 7.9 shows the ROC curves for PosQA, ProQres, and ProQprof on the four cross-validation test sets. PosQA clearly outperforms the other two methods on all the four test sets. Meanwhile, the ROC curves also show that the performance for a method on test set 1 and 3 is higher than that on test set 2 and 4, which reveals test set 1 and 3 are probably easier than test set 2 and 4 in terms of single position quality assessment.

We further evaluate the performance of PosQA, ProQres, and ProQprof on "high-quality", "medium-quality", and "low-quality" alignment sets. As shown in Figure 7.10(a)-(c), ProQres outperforms PosQA and ProQprof on "high-quality" alignments, whereas PosQA is the best method on both "medium-quality" and "low-quality" alignments. It is noteworthy that PosQA performs significantly better than both ProQres and ProQprof on "low-quality" alignments. One may argue that the difference on the performance is the result of the settings of ROC discrimination thresholds. Thus, we draw the ROC curves of PosQA with different discrimination thresholds on test set 1 in Figure 7.10(d). Since there is almost no difference between different curves when false positive rate is higher than 0.4, only the ROC curves with false positive rate lower than 0.4 are shown. Again, the difference is not obvious when different discrimination thresholds are used. Similar observations are found on the other test sets and on the other two methods. Thus, all ROC curves shown here reveal the actual comparisons of the three methods regardless of the discrimination thresholds.

Prediction Examples of PosQA and ProQres Three representative alignments generated by RAPTOR in CASP7 are shown here, and the performance of PosQA and ProQres on them is carefully studied. ProQres has been used for protein structure prediction by its developer, a top-ranked group in the CASP events [192]. These three alignments are T0346 (target) vs. 1a33 (template), T0323 vs. 1dizA, and T0372 vs. 1sqhA; the structural models derived from these alignments have very different GDT_TS [210] scores 97.67, 53.69 and 24.75, respectively. For the sake of clearness, only the results of PosQA and ProQres are compared here, because ProQprof performs worse than ProQres on these three alignments. Since PosQA does not predict the quality of an unaligned position, to do a fair comparison between PosQA and ProQres, the average prediction errors for both PosQA

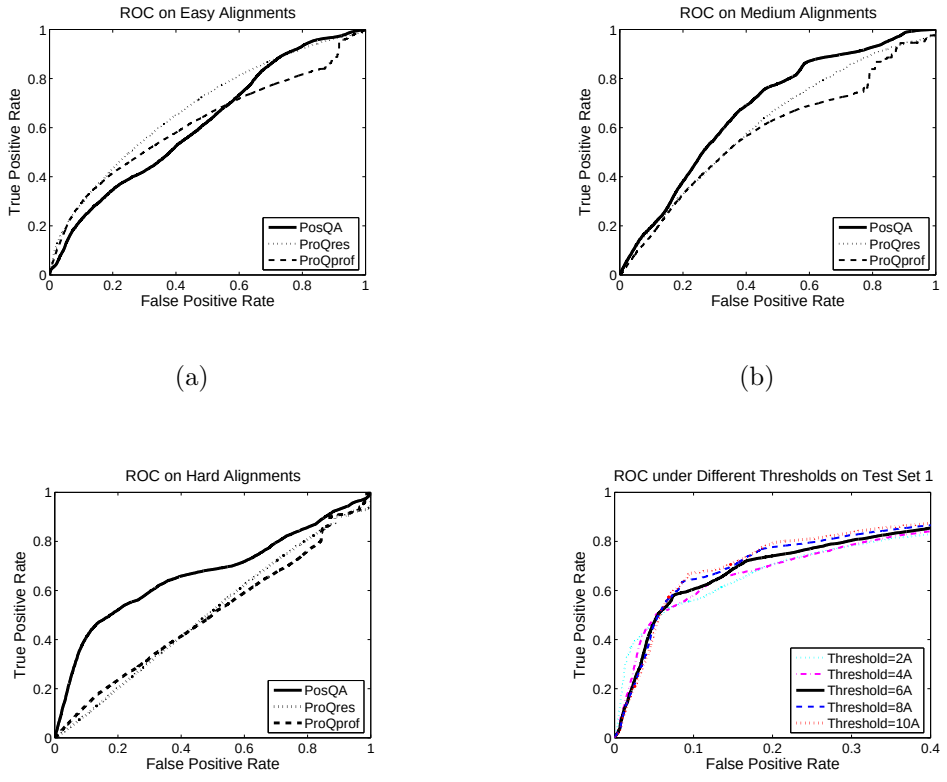


Figure 7.10: (a) ROC curves for PosQA, ProQres, and ProQprof on "high-quality" alignments (G-score$\leq$0.33). Discrimination threshold 2Å; (b) ROC curves for PosQA, ProQres, and ProQprof on "medium-quality" alignments (0.33<G-score$\leq$0.66). Discrimination threshold 4Å; (c) ROC curves for PosQA, ProQres, and ProQprof on "low-quality" alignments (0.66<G-score$\leq$1.0). Discrimination threshold 6Å; (d) ROC curves for PosQA with different discrimination threshold values on test set 1.

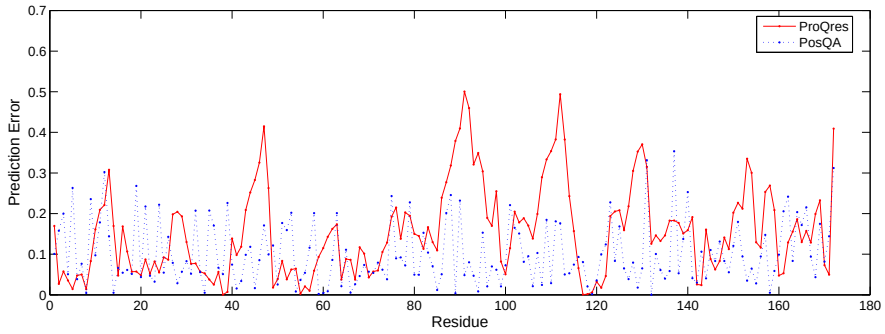
and ProQres are calculated on only the aligned positions. As shown in Figure 7.11, the prediction errors of both PosQA and ProQres are related to the overall alignment quality. The better the overall quality is, the smaller the prediction error is. PosQA performs better than ProQres on all these three test cases. The difference between the prediction errors of PosQA and ProQres is large on "high-quality" and "low-quality" alignments, *i.e.*, T0346 vs. 1a33 and T0372 vs. 1sqhA, but relatively small on "medium-quality" alignment, T0323 vs. 1dizA. The average prediction er-

rors of PosQA and ProQres are 0.10 and 0.15 for T0346 vs. 1a33, respectively, 0.24 and 0.27 for T0323 vs. 1dizA, respectively, and 0.39 and 0.47 for T0372 vs. 1sqhA, respectively. It is clear that for most residues of these alignments, the prediction errors of PosQA are smaller than that of ProQres. In particular, ProQres has obviously large prediction errors at some positions on the "high-quality" alignment between T0346 and 1a33, whereas PosQA's prediction errors are mostly contained within 0.3.

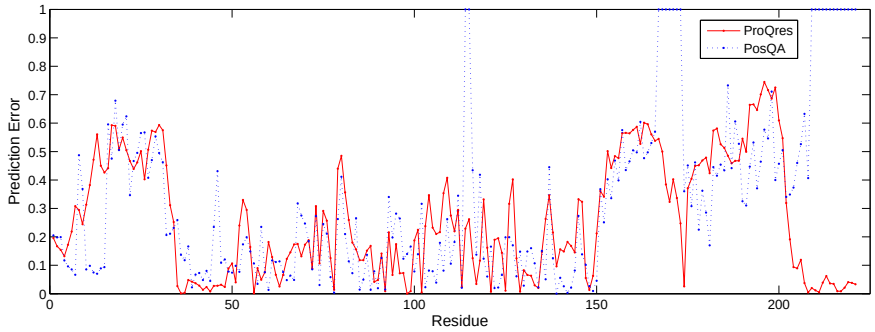
7.3.3 Discussion

Other than the RMSD and the statistical significance, there are other possible measurements for local quality assessment. For instance, torsion angles may be a good choice for measuring the local quality of a fragment. The performance of FragQA and PosQA on such measurements will be worth exploring in the future.

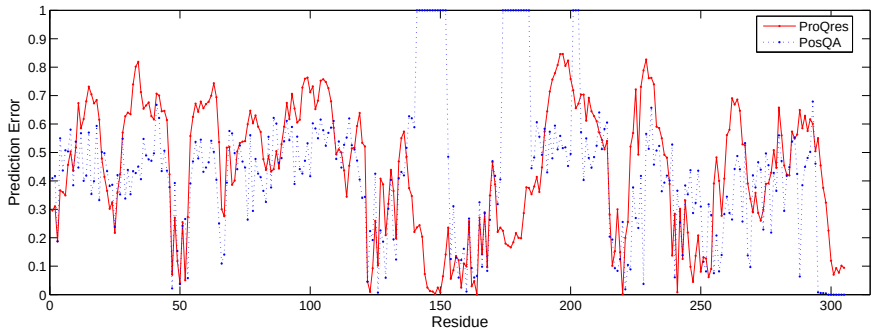
A potential application of local quality predictors such as FragQA and PosQA is that they can be used to identify those high-quality regions in an alignment. These high-quality regions can often cover a large portion of the target protein even if it is a hard target and thus, they can be refolded to obtain a better structural model for the target protein. For example, Zhang-server [212, 213] achieved an impressive performance in CASP7 and CASP8 by first cutting a threading-generated alignment into some ungapped regions, and then rearranging the physical orientations of these regions. Zhang-server uses all the ungapped regions without considering their quality. A further improvement over Zhang-server is to first predict the "absolute" quality of each region, and then refold only those high-quality regions to obtain a better structural model. FragQA provides such a powerful tool to directly evaluate the fragment quality cut from the alignments, which is independent of the optimal superimposition of the two whole structures.



(a) Prediction errors on T0346 vs. 1a33 (GDT_TS score 97.67). The average errors of PosQA and ProQres are 0.10 and 0.15, respectively.



(b) Prediction errors on T0323 vs. 1dizA (GDT_TS score 53.69). The average errors of PosQA and ProQres are 0.24 and 0.27, respectively.



(c) Prediction errors on T0372 vs. 1sqhA (GDT_TS score 24.75). The average errors of PosQA and ProQres are 0.39 and 0.47, respectively.

Figure 7.11: Prediction errors of PosQA and ProQres on three typical alignments generated by RAPTOR in the CASP7 event. Since PosQA does not predict the quality at unaligned positions, the prediction errors at these positions are set to 1.

Chapter 8

Concluding Remarks and Future Work

8.1 Conclusions

In this dissertation, we propose a fully automatic NMR protein structure determination protocol, AMR, which is able to generate high-resolution structures for short proteins from the raw NMR spectra. The major difference between AMR and previously proposed automatic NMR protein structure determination methods is that rather than combining existing methods together, AMR has all of its three steps developed altogether. Therefore, each step is highly error-tolerant to the imperfect output from the previous step. All the test proteins are acquired from our collaborators. These NMR spectra are thus obtained by different NMR spectrometers, different NMR spectroscopists, and different sample preparation processes, which make the spectra have quite different noise levels and different fractions of missing peaks. AMR is able to generate accurate structures for four of the five test proteins. Even for the one on which AMR failed, *i.e.*, COILIN, the results of automatic peak picking and resonance assignment are still reasonable. The main reason for the

failure on that protein is that COILIN is recognized by FALCON-NMR as an *ab initio* target, but the torsion angle sampling method applied by FALCON-NMR is not able to generate medium-resolution structural models due to the bad-quality fragments selected by Frazor and the flexibility in the sampling process.

We propose a novel peak picking method PICKY. The key idea is to form components that are as small and simple as possible. Decomposition techniques can then be applied to each component separately to detect peaks. PICKY is tested by the first systematic study on peak picking problem, and demonstrates fairly high *recall* and *precision* values. The refinement module of PICKY is method-independent. So it can be applied to the peak lists generated by any other peak picking methods. After peak lists are generated, we first prove that the resonance assignment problem is NP-hard. A novel assignment method, IPASS, is then proposed to deal with imperfect peaks generated by PICKY. As demonstrated in the experiments, IPASS is the only available assignment method that is able to tolerate errors in automatically picked peaks. IPASS has an error-tolerant spin system forming module. After the spin systems are formed, they are typed and reliable fragments are fixed. An integer linear programming model is proposed to globally optimize the assignment problem under our problem setup. Given the assignment done by IPASS, FALCON-NMR is developed to generate high-resolution structures based on chemical shift information and ambiguous NOE constraints. The idea is to use such NMR information as soft constraints rather than hard constraints. Therefore, chemical shift information is used to identify homologs if there is any or to select fragment candidates for each small region of the protein. The ambiguous NOE constraints are used to select the best decoys from each iteration of the structure calculation. The advantage of this method is that it can tolerate the high ambiguity rate on NOE constraints while still requiring the final structure to agree with the experimental data. However, the disadvantage is that NOE constraints

are not directly encoded into the sampling process, which results in the possibility of trapping into the local minimum of the energy function.

Another contribution of this dissertation is to propose novel methods to solve three key protein structure prediction problems which are closely related to NMR protein structure determination. These three protein structure prediction problems are inter-residue contact prediction, side chain packing, and local quality assessment. We propose a novel consensus method to accurately select true contacts from a large number of false ones. By eliminating server correlation, we are able to identify true contacts even when they are not supported by the majority of individual servers. Our method significantly outperform any other contact predictor, especially on new fold targets. Thus, this method can provide more true contacts to the NOE contact set in NMR protein structure determination. Especially when the quality of the NMR spectra is poor, our contact prediction method can possibly save the entire NMR structure determination process from failure. As demonstrated in the experiments of AMR, the all-atom refinement is an important step to achieve the final high-resolution structures. However, AMR currently does not contain a side chain assignment module. We propose an ultra-fast side chain packing method, which uses only backbone information. Our method is as accurate as the state-of-the-art global optimization methods, yet runs many times faster. This method can be hopefully applied to the all-atom refinement module of FALCON-NMR to accelerate the refinement process, thus to shorten the runtime of the entire system. In AMR, when a homolog can be found for a target protein, Modeller is called to generate initial structural models according to the alignment between the target and the homolog. However, the structural models generated from the alignment usually have bad regions, which are caused by the wrongly aligned positions. We propose two complementary local quality assessment methods to accurately predict the quality of local fragments and single aligned positions. Both of these two

methods perform better than the stat-of-the-art local quality predictors. These two methods can be applied to FALCON-NMR when homologs can be identified. According to the prediction results of these two methods, FALCON-NMR can refine the structural models by sampling based on strict torsion angle distributions for well-aligned regions and loose distributions for poorly-aligned regions. This will reduce the conformational search space for the sampling process, and thus increase the speed of convergence and reduce the risk of trapping into local minima.

8.2 Future Work

An immediate improvement of AMR is to increase the limit of the protein size that it can handle. Among all the three steps of AMR, peak picking and resonance assignment steps are standard which work consistently well on both short proteins and long proteins. The major bottleneck comes from the structure calculation step, especially when no homologs can be found to build reasonably good initial structural models. I plan to develop a new structure calculation method that is able to generate good structures for longer proteins. The key idea is to increase the weight of NOE contacts in the structure calculation step. There are two possible ways to achieve this goal, *i.e.*, encoding NOE contacts as soft constraints or as hard constraints. A direct way to improve FALCON-NMR is to encode NOE contacts into its energy function. By using the NOE contacts as soft constraints, the torsion angle sampling process will reduce the risk of generating structures with obviously wrong topologies. However, it is possible that due to the other terms in the energy function, some very bad structures can still be generated if they behave well on the other energy terms. Thus, another possibility is to encode the NOE contacts as hard constraints. Techniques such as constraint programming and multi-dimensional scaling might be used to achieve this goal. However, if the ambiguity of the NOE

contacts is too high, such methods will still be very easy to fail. Thus, a better method is needed to reduce the NOE ambiguity. CYANA [73] uses the ideas of network-anchoring and constraint combination to eliminate ambiguous distance constraints. I would like to develop a method to discover self-consistent distance constraint set based on geometric properties of the protein, such as bond length, bond angle, and secondary structure information.

Another way to improve our current system is to encode more knowledge and experience on protein structure prediction problems. We have proposed novel methods to solve three key structure prediction problems that are close related to NMR protein structure determination. These methods should be able to improve the performance of FALCON-NMR. For example, the consensus contact prediction method can be used as the complement to the NOE constraint extraction module of AMR, the side chain packing method can be used as a subroutine of FALCON-Refinement, and the local quality assessment methods can be encoded into the FALCON-Threading module. I plan to incorporate these methods or the ideas into FALCON-NMR. Thus, if there is enough information from NMR spectra, the weights of the prediction methods will be reduced and the structure calculation step will generate structures mainly according to the experimental data. Otherwise, if the quality of the NMR spectra is poor, or the resonance assignment returns results with low confidence, the weights of the prediction methods will be increased and the structure calculation step will generate structures according to both experimental data and prediction results.

Providing AMR as a fully automatic system to the NMR community is a long-term goal, because it still requires significant improvements on AMR. A more reasonable short-term goal is to provide PICKY and IPASS as interactive tools to assist NMR spectroscopists in the data processing steps. We have implemented PICKY as a plug-in in one of the most commonly used NMR user interfaces, SPARKY,

so that the users can easily call PICKY through SPARKY to pick initial peaks, and modify the initial peaks according to their expertise. We will also implement IPASS to be an interactive tool in the near future.

As methods for protein structure determination developing, a continuing problem is to model protein dynamics and protein-protein interaction or protein-ligand binding. The traditional way of modeling protein structures is to model them as rigid bodies. However, this is not true in nature. A protein has different structures under different conditions, such as different PH values or different temperatures. Even under the same condition, a protein is dynamically changing its conformation in solution. Therefore, a very important problem is to model protein dynamics and protein conformational changes during the protein-protein interaction or protein-ligand binding process. Many proteins undergo fairly large conformational changes when they bind to another molecule. These movements are often essential for binding and function, and are thus relevant to things like drug design. Understanding the mechanisms by which the proteins bind to each other or to ligands is crucial to control and alter protein associations. For example, the action of an enzyme and a single substrate has been extensively studied. The first hypothesis is "lock and key" model. That is, the enzyme serves as a lock and the substrate serves as a key. Thus, this hypothesis is based on the rigid shape assumption. However, experimental evidence shows that this model can not well explain the real action in nature. The induced-fit theory is then proposed [105], which assumes that the shape of the enzyme is actually partially flexible and the substrate is the main factor to determine the final shape of the enzyme. One major advantage of NMR over other structure determination techniques, such as X-ray crystallography and electron microscopy, is the fact that NMR is able to study protein structures in solution. Thus, NMR is a perfect tool to study protein dynamics and conformational changes. I would like to work on modeling protein dynamics and conformational

changes during protein-protein interaction or protein-ligand binding process. The automatic NMR tools proposed in this dissertation, such as PICKY and IPASS, can be directly applied to such studies. I believe this is an essential step towards the understanding of protein structures and their function.

References

[1] http://ca.expasy.org/sprot/relnotes/relstat.html. 118

[2] http://predictioncenter.org/casp6/Casp6.html. 122

[3] http://predictioncenter.org/casp7/. 101

[4] http://svmlight.joachims.org/svm_multiclass.html. 117

[5] http://www.rcsb.org/pdb/statistics/holdings.do. 1

[6] R. A. Abagyan and M. M. Totrov. Contact area difference (CAD): A robust measure to evaluate accuracy of protein models. *Journal of Molecular Biology*, 268:678–685, 1997. 30

[7] T. Akutsu. NP-hardness results for protein side-chain packing. *Genome Informatics*, 8:180–186, 1997. 28

[8] B. Alipanahi, X. Gao, E. Karakoc, F. Balbach, L. Donaldson, C. Arrowsmith, and M. Li. IPASS: error tolerant NMR backbone resonance assignment by linear programming. *University of Waterloo technical report, No. CS-2009-16*, 2009. http://www.cs.uwaterloo.ca/research/tr/2009/. 3, 65

[9] B. Alipanahi, X. Gao, E. Karakoc, L. Donaldson, and M. Li. PICKY: a novel SVD-based NMR spectra peak picking method. *Bioinformatics*, 25:i268–i275, 2009. 3, 40, 46

[10] A. S. Altieri and R. A. Byrd. Automation of NMR structure determination of proteins. *Current Opinion in Structural Biology*, 14:547–553, 2004. 1, 13, 19

[11] 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*, 25:3389–3402, 1997. 21, 87, 125

[12] C. Antz, K. P. Neidig, and H. R. Kalbitzer. A general Bayesian method for an automated signal class recognition in 2D NMR spectra combined with a multivariate discriminant analysis. *Journal of Biomolecular NMR*, 5:287–296, 1995. 13

[13] C. Bartels, M. Billeter, P. Güntert, and K. Wüthrich. Automated sequence-specific assignment of homologous proteins using the program GARANT. *Journal of Biomolecular NMR*, 7:207–213, 1996. 15, 19

[14] C. Bartels, P. Güntert, M. Billeter, and K. Wüthrich. GARANT - a general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra. *Journal of Computational Chemistry*, 18:139–149, 1997. 15, 19

[15] E. Benedetti, G. Morelli, G. Nemethy, and H. Scheraga. Statistical and energetic analysis of sidechain conformations in oligopeptides. *International Journal of Peptide and Protein Research*, 22:1–15, 1983. 27

[16] C. Berezin, F. Glaser, J. Rosenberg, I. Paz, T. Pupko, P. Fariselli, R. Casadio, and N. Ben-Tal. ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinformatics*, 20(8):1322–1324, 2004. 30

[17] A. Berglund, R. D. Head, E. A. Welsh, and G. R. Marshall. ProVal: a protein-scroing function for the selection of native and near-native folds. *Proteins*, 54:289–302, 2004. 30

[18] M. V. Berjanskii, S. Neal, and D. S. Wishart. PREDITOR: a web server for predicting protein torsion angle restraints. *Nucleic Acids Research*, 34:63–69, 2006. 13

[19] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, Bhat T. N., H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28:235–242, 2000. 1, 137

[20] T. N. Bhat, V. Sasisekharan, and M. Vijayan. An analysis of side-chain conformation in proteins. *International Journal of Peptide and Protein Research*, 14:170–184, 1979. 25, 27

[21] P. Bradley, K. M. S. Misura, and D. Baker. Toward high-resolution de novo structure prediction for small proteins. *Science*, 309:1868–1871, 2005. 20, 21, 79, 85

[22] N. V. Buchete, J. E. Straub, and D. Thirumalai. Orientational potentials extracted from protein structures inprove native fold recognition. *Protein Science*, 13:862–874, 2004. 30

[23] C. Bystroff, V. Thorsson, and D. Baker. HMMSTR: A hidden Markov model for local sequence-structure correlations in proteins. *Journal of Molecular Biology*, 301:173–190, 2000. 24

[24] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Science*, 12(9):2001–2014, 2003. 27, 28, 29

[25] E. A. Carrara, F. Pagliari, and C. Nicolini. Neural networks for the peak-picking of nuclear magnetic resonance spectra. *Neural Networks*, 6:1023–1032, 1993. 13

[26] D. A. Case. Calibration of ring-current effects in proteins and nucleic acids. *Journal of Biomolecular NMR*, 6:341–346, 1995. 20

[27] A. Cavalli, X. Salvatella, C. M. Dobson, and M. Vendruscolo. Protein structure determination from NMR chemical shift. *Proceedings of the National Academy of Sciences*, 104:9615–9620, 2007. 13, 19, 20

[28] R. Chandrasekaran and G. Ramachandran. Studies on the conformation of amino acids. XI. Analysis of the observed side group conformations in proteins. *International Journal of Protein Research*, 2:223–233, 1994. 27

[29] B. Chazelle, C. Kingsford, and M. Singh. A semidefinite programming approach to side chain positioning with new rounding strategies. *Informs Journal on Computing*, 16:380–392, 2004. 27, 28

[30] J. Cheng and P. Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22:1456–1463, 2006. 94

[31] D. Chivian, D. E. Kim, L. Malmström, 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. 22

[32] D. Chivian, D. E. Kim, L. Malmström, J. Schonbrun, C. Rohl, and D. Baker. Prediction of CASP6 structures using automated Robetta protocols. *Proteins*, 61(S7:1):57–66, 2005. 109

[33] N.D. Clarke, A. Valencia, J.M.G. Izarzugaza, M.L. Tress, and O. Graïa. CASP7 presentation on contact prediction. Retrieved March 6, 2009, from `http://predictioncenter.org/casp7/meeting/presentations/Presentations_assessors/CASP7_RR_Clarke.pdf`. 22

[34] B. Coggins and P. Zhou. PACES: protein sequential assignment by computer-assisted exhaustive search. *Journal of Biomolecular NMR*, 26:93–111, 2003. 13, 15, 16

[35] C. Colovos and T. O. Yeates. Verification of protein structures: patterns of nonbonded atomic interactions. *Protein Science*, 2:1511–1519, 1993. 30

[36] S. A. Corne and P. Johnson. An artificial neural network for classifying cross peaks in two-dimensional NMR spectra. *Journal of Magnetic Resonance*, 100:256–266, 1992. 13

[37] G. Cornilescu, F. Delaglio, and A. Bax. Protein backbone angle restraints from searching a database for chemical shift and sequence homology. *Journal of Biomolecular NMR*, 13:289–302, 1999. 13, 20

[38] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. 113

[39] B. Dahiyat and S. Mayo. Protein design automation. *Protein Science*, 5:895–903, 1996. 26

[40] B.V . Dasarathy. *Nearest neighbor (NN) norms: NN pattern classification techniques*. Los Alamitos: IEEE Computer Society Press, 1990. 116

[41] F. Delaglio, S. Grzesiek, G. W. Vuister, G. Zhu, J. Pfeifer, and A. Bax. NMRPipe: amultidimensional spectral processing system based on UNIX pipes. *Journal of Biomolecular NMR*, 6:277–293, 1995. 12, 13

[42] J. Desjarlais and T. Handel. De novo design of the hydrophobic cores of proteins. *Protein Science*, 4:2006–2018, 1995. 26

[43] J. Desmet, M. Maeyer, B. Hazes, and I. Laster. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356:539–542, 1992. 28

[44] R. L. Dunbrack. Rotamer libraries in the 21st century. *Current Opinion in Structural Biology*, 12:431–440, 2002. 6, 26, 27, 112, 115, 117

[45] R. L. Dunbrack and F. Cohen. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Science*, 6:1661–1681, 1997. 26, 27, 112, 115, 117, 121

[46] R. L. Dunbrack and M. Karplus. Backbone-dependent rotamer library for proteins: Application to side-chain prediction. *Journal of Molecular Biology*, 230:543–574, 1993. 27, 112

[47] R. L. Dunbrack and M. Karplus. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. *Nature Structral Biology*, 1:334–340, 1994. 27, 112, 115

[48] D. Eisenberg, R. Lüthy, and J. U. Bowie. VERIFY3D: assessment of protein models with three-dimensional profiles. *Methods Enzymol*, 277:396–404, 1997. 30

[49] A. Eletsky, G. Liu, H.S. Atreya, D. Sukumaran, D. Wang, K. Cunningham, H. Janjua, L.-C. Ma, R. Xiao, J. Liu, M. Baran, T.B. Acton, B. Rost, G.T.

Montelione, and T. Szyperski. Solution NMR Structure of Bacillus subtilis Hypothetical Protein yvyC. Retrieved March 18, 2007, from `http://www.pdb.org/pdb/explore.do?structureId=2HC5`. 106

[50] O. Eriksson, Y. Zhou, and A. Elofsson. Side chain-positioning as an integer programming problem. In *Proceedings of the First Workshop on Algorithms in Bioinformatics (WABI)*, pages 128–141, 2001. 28

[51] E. Eyal, R. Najmanovich, R. J. Mcconkey, M. Enelman, and V. Sobolev. Importance of solvent accessibility and contact surfaces in modeling side-chain conformations in proteins. *Journal of Computational Chemistry*, 25:712–724, 2004. 115, 116

[52] Q. J. Fang and D. Shortle. A consistent set of statistical potentials for quantifying local side-chain and backbone interactions. *Proteins*, 60:90–96, 2005. 30

[53] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering*, 14(11):835–843, 2001. 23

[54] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins*, 5:157–162, 2001. 23

[55] M. Fasnacht, J. Zhu, and B. Honig. Local quality assessment in homology models using statistical potentials and support vector machines. *Protein Science*, 16:1557–1568, 2007. 30, 129

[56] M. Feig and C. L. Brooks. Evaluating CASP4 predictions with physical energy functions. *Proteins*, 49:232–245, 2002. 30

[57] A. K. Felts, E. Gallicchio, A. Wallqvist, and R. M. Levy. Distinguishing native conformations of proteins from decoys with an effective free energy estimator based on the OPLS all-atom force field and the surface generalized born solvent model. *Proteins*, 48:404–422, 2002. 30

[58] F. Fiorito, S. Hiller, G. Wider, and K. Wüthrich. Automated resonance assignment of protein: 6D APSY-NMR. *Journal of Biomolecular NMR*, 35:27–37, 2006. 17

[59] D. Fischer. 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. *Proteins*, 51:434–441, 2003. 30

[60] R. Fogh, J. Ionides, E. Ulrich, W. Boucher, W. Vranken, J. P. Linge, M. Habeck, W. Rieping, T.N. Bhat, J. Westbrook, K. Henrich, and Berman H. Gilliland, G., J. Thornton, M. Nilges, J. Markley, and E. Laue. The CCPN project: an interim report on a data model for the NMR community. *Nature Structural Biology*, 9:416–418, 2002. 19

[61] X. Gao, D. Bu, S. C. Li, J. Xu, and M. Li. FragQA: predicting local fragment quality of a sequence-structure alignment. *Genome Informatics*, 19:27–39, 2007. 7, 30

[62] X. Gao, D. Bu, J. Xu, and M. Li. Improving consensus contact prediction via server correlation reduction. *BMC Structural Biology*, 9(28), 2009. 5

[63] X. Gao, J. Xu, S. C. Li, and M. Li. Predicting local fragment quality of a sequence-structure alignment. *Journal of Bioinformatics and Computational Biology*, 2009. in press. 7

[64] D. S. Garret, R. Powers, A. M. Gronenborn, and G. M. Clore. A common sense approach to peak picking in two-, three-, and four-dimensional spectra

using automatic computer analysis of contour diagrams. *Journal of Magnetic Resonance*, 95:214–220, 1991. 13

[65] K. Ginalski, A. Elofsson, D. Fischer, and L. Rychlewski. 3D-Jury: A simple approach to improve protein structure predictions. *Bioinformatics*, 19:1015–1018, 2003. 30

[66] U. Göbel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins*, 18:309–317, 1994. 23, 97

[67] T. D. Goddard and D. G. Kneller. SPARKY 3. *University of California, San Francisco*, 2007. 15

[68] O. Graïa, D. Baker, R.M. MacCallum, J. Meiler, M. Punta, B. Rost, M.L. Tress, and A. Valencia. CASP6 assessment of contact prediction. *Proteins*, 61:214–224, 2005. 22, 24

[69] A. Grishaev and M. Llinás. CLOUDS: a protocol for deriving a molecular proton density via NMR. *Proceedings of the National Academy of Sciences*, 99:6707–6712, 2002. 18, 19

[70] A. Grishaev and M. Llinás. Sorting signals from protein NMR spectra: SPI, a Bayesian protocol for uncovering spin systems. *Journal of Biomolecular NMR*, 24:203–213, 2002. 18, 19

[71] A. Grishaev and M. Llinás. BACUS: a Bayesian protocol for the identification of protein NOESY spectra via unassigned spin systems. *Journal of Biomolecular NMR*, 28:1–10, 2004. 18, 19

[72] T. Grossman, R. M. Farber, and A. S. Lapedes. Neural net representations of empirical protein potentials. In *Proceedings of the Third International*

Conference on Intelligent Systems for Molecular Biology (ISMB), pages 154–161, 1995. 23

[73] P. Güntert. Automated NMR structure calculation with CYANA. *Methods in Molecular Biology*, 278:353–378, 2004. 13, 18, 151

[74] P. Güntert. Automated structure determination from NMR spectra. *European Biophysics Journal*, 38:129–143, 2009. 13

[75] P. Güntert, M. Salzmann, D. Braun, and K. Wüthrich. Sequence-specific NMR assignment of proteins by global fragment mapping with the program MAPPER. *Journal of Biomolecular NMR*, 18:129–137, 2000. 13, 15, 16

[76] I. Halperin, H. Wolfson, and R. Nussinov. Correlated mutations: advances and limitations. A study on fusion proteins and on the Cohesin-Dockerin families. *Proteins*, 63:832–845, 2006. 23

[77] N. Hamilton, K. Burrage, M. A. Ragan, and T. Huber. Protein contact prediction using patterns of correlation. *Proteins*, 56:679–684, 2004. 23

[78] T. Herrmann, P. Güntert, and K. Wüthrich. Protein NMR structure determination with automated NOE assignment using the new software CANDID and the torsion angle dynamics algorithm DYANA. *Journal of Molecular Biology*, 319:209–227, 2002. 18

[79] T. Herrmann, P. Güntert, and K. Wüthrich. Protein NMR structure determination with automated NOE-identification in the NOESY spectra using the new software ATNOS. *Journal of Biomolecular NMR*, 24:171–189, 2002. 15

[80] V. Heurgue-Hamard, M. Graille, N. Scrima, N. Ulryck, S. Champ, H. van Tilbeurgh, and R. H. Buckingham. The zinc finger protein Ynr046w is pluri-

functional and a component of the eRF1 methyltransferase in yeast. *Journal of Biological Chemistry*, 281:36140–36148, 2006. 106

[81] S. Hiller, F. Fiorito, K. Wüthrich, and G. Wider. Automated projection spectroscopy (APSY). *Proceedings of the National Academy of Sciences*, 102:10876–10881, 2005. 17

[82] L. Holm and C. Sander. Fast and simple monte carlo algorithm for side chain optimization in proteins: application to model building by homology. *Proteins*, 14:213–223, 1992. 28

[83] C. W. Hsu, C. C. Chang, and C. J. Lin. A practical guide to support vector classification. Technical report, 2003. 117

[84] E. S. Huang, S. Subbiah, and M. Levitt. Recognizing native folds by the arrangement of hydrophobic and polar residues. *Journal of Molecular Biology*, 249:493–507, 1995. 23

[85] Y. J. Huang. *Automated determination of protein structures from NMR data by iterative analysis of self-consistent contact patterns*. Ph.D. thesis, Rutgers University, New Brunswick, NJ, 2001. 19

[86] Y. J. Huang, G. V. Swapna, P. K. Rajan, H. Ke, B. Xia, K. Shukla, M. Inouye, and G. T. Montelione. Solution NMR structure of ribosome binding factor A (RbfA), a cold-shock adaptation protein from Escherichis coli. *Journal of Molecular Biology*, 327:521–536, 2003. 19

[87] S. J. Hubbard and J. M. Thornton. 'NACCESS', Computer Program. Department of Biochemistry and Molecular Biology, University College London. 1993. 116

[88] J. Hwang and W. Liao. Side-chain prediction by neural networks and simulated annealing optimization. *Protein Engineering*, 8:363–370, 1995. 28

[89] J. M. Izarzugaza, O. Graña, M. L. Tress, A. Valencia, and N. D. Clarke. Assessment of intramolecular contact predictions for CASP7. *Proteins*, 69:152–158, 2007. 25, 102, 104

[90] T. Jain, D. Cerutti, and J. McCammon. Configurational-bias sampling techinique for predicting side-chain conformations in proteins. *Protein Science*, 15:2029–2039, 2007. 27

[91] J. Janin, S. Wodak, M. Levitt, and B. Maigret. The conformation of amino acid side chains in proteins. *Journal of Molecular Biology*, 125:357–386, 1978. 25, 26, 27

[92] T. Joachims. Making large-scale support vector machine learning practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998. 130, 140

[93] B. A. Johnson and R. A. Blevins. NMR View: A computer program for the visualization and analysis of NMR data. *Journal of Biomolecular NMR*, 4(5):603–614, 1994. 13, 15

[94] D. T. Jones. GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *Journal of Molecular Biology*, 287:797–815, 1999. 21, 94

[95] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 1999. 88, 126

[96] Y. Jung and M. Zweckstetter. Mars - robust automatic backbone assignment of proteins. *Journal of Biomolecular NMR*, 30:11–23, 2004. 13, 15, 16, 17

[97] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1993. 63

[98] Barrett C. Karplus, K. and R. Hughey. Hidden Markov Models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998. 25, 94

[99] 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. 21, 126

[100] C. Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 21:1028–1036, 2005. 27, 28

[101] G. Kleywegt, R. Boelens, and R. Kaptein. A versatile approach toward the partially automatic recognition of cross peaks in 2D ^1H NMR spectra. *Journal of Magnetic Resonance*, 88:601–608, 1990. 13

[102] H. Kono and J. Doi. A new method for side-chain conformation prediction using a hopfield network and reproduced rotamers. *Journal of Computational Chemistry*, 17:1667–1683, 1996. 27

[103] R. Koradi, M. Billeter, M. Engeli, P. Güntert, and K. Wüthrich. Automated peak picking and peak integration in macromolecular NMR spectra using AUTOPSY. *Journal of Magnetic Resonance*, 135:288–297, 1998. 13, 38, 40, 41, 43, 44, 52

[104] D. M. Korzhnev, I. V. Ibraghimov, M. Billeter, and V. Y. Orekhov. MUNIN: application of three-way decomposition to the analysis of heteronuclear NMR relaxation data. *Journal of Biomolecular NMR*, 21:263–268, 2001. 13, 14, 40, 44

[105] D. E. Koshland. Application of a theory of enzyme specificity to protein synthesis. *Proceedings of the National Academy of Sciences*, 44:98–104, 1958. 152

[106] P. J. Kundrotas and E. G. Alexov. Predicting residue contacts using pragmatic correlated mutations method: reducing the false positives. *BMC Bioinformatics*, 7:503–511, 2006. 23

[107] J. Kuszewski, C. D. Schwieters, D. S. Garrett, A. Byrd, N. Tjandra, and G. M. Clore. Completely automated, highly error-tolerant macromolecular structure determination from multidimensional nuclear Overhauser enhancement spectra and chemical shift assignment. *Journal of American Chemical Society*, 126:6258–6273, 2004. 13, 18

[108] G. Labesse, N. Colloc'h, J. Pothier, and J. P. Mornon. P-SEA, a new efficient assignment of secondary structure from C_α trace of proteins. *Computer Applications in the Biosciences*, 13:291–295, 1997. 115

[109] R. A. Laskowski, M. W. Macarthur, D. S. Moss, and J. M. Thornton. PROCHECK: a program to check the stereochemical quality of protein structures. *Journal of Applied Crystallography*, 26:283–291, 1993. 30

[110] D. Latek and A. Kolinski. Contact prediction in protein modeling: scoring, folding and refinement of coarse-grained models. *BMC Structural Biology*, 8:36–50, 2008. 109

[111] T. Lazaridis and M. Karplus. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. *Journal of Molecular Biology*, 288:477–487, 1999. 30

[112] C. Lee and S. Subbiah. Prediction of protein side-chain conformation by packing optimization. *Journal of Molecular Biology*, 217:373–388, 1991. 28

[113] M. C. Lee and Y. Duan. Distinguish protein decoys by using a scoring function based on a new AMBER force field, short molecular dynamics simulations, and the generalized born solvent model. *Proteins*, 55:620–634, 2004. 30

[114] A. Lemak, C. A. Steren, C. H. Arrowsmith, and M. Llinás. Sequence specific resonance assignment via Multicanonical Monte Carlo search using an ABACUS approach. *Journal of Biomolecular NMR*, 41:29–41, 2008. 13, 15, 16, 17

[115] S. C. Li. *New Approaches to Protein Structure Prediction*. Ph.D. thesis, University of Waterloo, Waterloo, ON, 2009. 77, 79

[116] S. C. Li, D. Bu, X. Gao, J. Xu, and M. Li. Designing succinct structural alphabets. *Bioinformatics*, 24(13):i182–i189, 2008. 5, 77, 79

[117] S. C. Li, D. Bu, J. Xu, and M. Li. Fragment-HMM: a new approach to protein structure prediction. *Protein Science*, 17(11):1925–1934, 2008. 4, 76, 77, 79

[118] W. Li and A. Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein for nucleotide sequences. *Bioinformatics*, 22:1658–1659, 2006. 94, 129

[119] S. Liang and N. Grishin. Side-chain modeling with an optimized scoring function. *Protein Science*, 11:322–331, 2002. 27, 28

[120] J. P. Linge, S. O'Donoghue, and M. Nilges. *Nuclear Magnetic Resonance of Biological Macromolecules*. London: Academic Press, 2001. 18

[121] B. López-Méndez and P. Güntert. Automated protein structure determination from NMR spectra. *Journal of the American Chemical Society*, 128(40):13112–13122, 2006. 13, 19

[122] H. Lu and J. Skolnick. A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins*, 44:223–232, 2001. 30

[123] R. Luthy, J. U. Bowie, and D. Eisenberg. Assessment of protein models with 3-dimensional profiles. *Nature*, 356:83–85, 1992. 30

[124] M. Maeyer, J. Desmet, and I. Lasters. All in one: a highly detailed rotamer library improves both accuracy and speed in the modelling of sidechains by dead-end elimination. *Folding and Design*, 2:53–66, 1997. 27, 28

[125] M. A. Marti-Renom, M. S. Madhusudhan, A. Fiser, B. Rost, and A. Sali. Reliability of assessment of protein structure prediction methods. *Structure*, 10:435–440, 2002. 30

[126] J. E. Masse and R. Keller. Autolink: automated sequential resonance assignment of biopolymers from NMR data by relative-hypothesis-prioritization-based simulated logic. *Journal of Magnetic Resonance*, 174:133–151, 2005. 13, 15, 16

[127] B. McConkey, V. Sobolev, and M. Edelman. Discrimination of native protein structures using atom-atom contact scoring. *Proceedings of the National Academy of Sciences*, 100(9):3215–3220, 2003. 30

[128] M. McGregor, S. Islam, and M. Sternberg. Analysis of the relationship between side-chain conformation and secondary structure in globular proteins. *Journal of Molecular Biology*, 198:295–310, 1987. 25, 26, 27, 115

[129] L. J. McGuffin. The ModFOLD server for the quality assessment of protein structural models. *Bioinformatics*, 24:586–587, 2008. 30

[130] L. J. Mcguffin, K. Bryson, and D. T. Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000. 65

[131] L. J. McGuffin and D. T. Jones. Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics*, 19:874–881, 2003. 94

[132] F. Melo and E. Feytmans. Assessing protein structures with a non-local atomic interaction energy. *Journal of Molecular Biology*, 277:1141–1152, 1998. 30

[133] J. Mendes, H. Nagarajaram, C. Soares, T. Blundell, and M. Carrondo. Incorporating knowledge-based biases into an energy-based side-chain modeling method: application to comparative modeling of protein structure. *Biopolymers*, 59:72–86, 2001. 28

[134] K. M. S. Misura, D. Chivian, C. A. Rohl, D. E. Kim, and D. Baker. Physically realistic homology models built with Rosetta can be more accurate than their templates. *Proceedings of the National Academy of Sciences*, 103:5361–5366, 2006. 22, 24

[135] S. Miyazawa and R. L. Jernigan. Estimation of effective interresidue contact energies from protein crystal-structures quasi-chemical approximation. *Macromolecules*, 18:534–552, 1985. 23

[136] H. N. Moseley and G. T. Montelione. Automated analysis of NMR assignments and structures for proteins. *Current Opinion in Structural Biology*, 9:635–642, 1999. 19

[137] J. Moult, K. Fidelis, A. Kryshtafovych, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure prediction (CASP):Round VII. *Proteins*, 69:3–9, 2007. 22, 77

[138] J. Moult, K. Fidelis, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure prediction (CASP):Round VI. *Proteins*, 61:3–7, 2005. 22, 77

[139] J. Moult, K. Fidelis, A. Zemla, and T. Hubbard. Critical assessment of methods of protein structure prediction (CASP):Round IV. *Proteins*, 45:2–7, 2001. 22, 77

[140] J. Moult, K. Fidelis, A. Zemla, and T. Hubbard. Critical assessment of methods of protein structure prediction (CASP):Round V. *Proteins*, 53:334–339, 2003. 22, 77

[141] J. Moult, T. Hubbard, K. Fidelis, and J. Pedersen. Critical assessment of methods of protein structure prediction (CASP):Round III. *Proteins*, 37:2–6, 1999. 22, 77

[142] M. Nilges, M. J. Macias, S. Odonoghue, and H. Oschkinat. Automated NOESY interpretation with ambiguous distance restraints: the reined NMR solution structure of the pleckstrin homology domain from beta-spectrin. *Journal of Molecular Biology*, 269:408–422, 1997. 18

[143] O. Olmea and A. Valencia. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Folding and Design*, 2:S25–32, 1997. 23, 97

[144] V. Y. Orekhov, I. V. Ibraghimov, and M. Billeter. MUNIN: a new approach to multi-dimensional NMR spectra interpretation. *Journal of Biomolecular NMR*, 20:49–60, 2001. 13, 14, 40, 44, 52

[145] B. H. Park, E. S. Huang, and M. Levitt. Factors affecting the ability of energy functions to discriminate correct from incorrect folds. *Journal of Molecular Biology*, 266:831–846, 1997. 30

[146] M. Pawlowski, M. J. Gajda, R. Matlak, and J. M. Bujnicki. MetaMQAP: a meta-server for the quality assessment of protein models. *BMC Bioinformatics*, 9(403), 2008. 30

[147] R. Peterson, P. Dutton, and A. Wand. Improved side-chain prediction accuracy using an *ab initio* potential energy function and a very large rotamer library. *Protein Science*, 13:735–751, 2004. 27

[148] D. Petrey and B. Honig. Free energy determinats of tertiary structure and the evaluation of protein models. *Protein Science*, 9:2181–2191, 2000. 30

[149] N. Pierce and E. Winfree. Protein design is NP-hard. *Protein Engineering*, 15:779–782, 2002. 28

[150] 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:S62–70, 2002. 23

[151] J. Ponder and F. Richards. Tertiary templates for proteins. use of packing criteria in the enumeration of allowed sequences for different structural classes. *Journal of Molecular Biology*, 193:775–791, 1987. 27

[152] M. Punta and B. Rost. PROFcon: novel prediction of long-range contacts. *Bioinformatics*, 21:2960–2968, 2005. 23, 24

[153] H. Rangwala and G. Karypis. fRMSDPred: Predicting local RMSD between structural fragments using sequence information. *Proteins*, 72:1005–1018, 2007. 30, 129

[154] C. Rohl, C. Strauss, D. Chivian, and D. Baker. Modeling structurally variable regions in homologous proteins with rosetta. *Proteins*, 55:656–677, 2004. 28

[155] A. Roitberg and R. Elber. Modeling side chains in peptides and proteins: Application of the locally enhanced sampling and the simulated annealing methods to find minimum energy functions. *Journal of Chemical Physics*, 95:9277–9287, 1991. 28

[156] A. Rouh, A. Louis-Joseph, and J. Y. Lallemand. Bayesian signal extraction from noisy FT NMR spectra. *Journal of Biomolecular NMR*, 4:505–518, 1994. 13

[157] M. I. Sadowski and D. T. Jones. Benchmarking template selection and model quality assessment for high-resolution comparative modeling. *Proteins*, 69:476–485, 2007. 30

[158] A. Sali and T. L. Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234:779–815, 1993. 4, 76, 118, 122

[159] R. Samudrala and J. Moult. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of Molecular Biology*, 275:895–916, 1998. 30

[160] H. Schrauber, F. Eisenhaber, and P. Argos. Rotamers: To be or not to be? An analysis of amino acid sidechain conformations in globular proteins. *Journal of Molecular Biology*, 230:592–612, 1993. 27

[161] C. D. Schwieters, J. J. Kuszewski, and G. M. Clore. Using Xplor-NIH for NMR molecular structure determination. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 48:47–62, 2006. 13, 18

[162] C. D. Schwieters, J. J. Kuszewski, N. Tjandra, and G. M. Clore. The XPLOR-NIH NMR molecular structure determination package. *Journal of Magnetic Resonance*, 160:65–73, 2003. 13, 18

[163] B. R. Seavey, E. A. Farr, W. M. Westler, and J. Markley. A relational database for sequence-specific protein NMR data. *Journal of Biomolecular NMR*, 1:217–236, 1991. 16

[164] G. Shackelford and K. Karplus. Contact prediction using mutual information and neural nets. *Proteins*, 69:159–164, 2007. 23, 24

[165] G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-neighbor methods in learning and vision: theory and practice (neural information processing)*. The MIT Press, 2006. 116

[166] Y. Shao and C. Bystroff. Predicting interresidue contacts using templates and pathways. *Proteins*, 53:497–502, 2003. 24

[167] Y. Shen and A. Bax. Protein backbone chemical shifts predicted from searching a database for torsion angle and sequence homology. *Journal of Biomolecular NMR*, 38:289–302, 2007. 20

[168] Y. Shen, O. Lange, F. Delaglio, P. Rossi, J. M. Aramini, G. Liu, A. Eletsky, B. Wu, K. K. Singarapu, A. Lemak, A. Ignatchenko, C. Arrowsmith, T. Szyperski, G. T. Montelione, D. Baker, and A. Bax. Consistent blind protein structure generation from NMR chemical shift data. *Proceedings of the National Academy of Sciences*, 105:4685–4690, 2008. 13, 19, 20

[169] J. Shi, T. L. Blundell, and K. Mizuguchi. FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of Molecular Biology*, 310(1):243–257, 2001. 25, 94

[170] I. N. Shindyalov, N. A. Kolchanov, and C. Sander. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Engineering*, 7:349–358, 1994. 23

[171] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer. MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, 16(9):776–785, 2000. 30

[172] M. S. Singer, G. Vriend, and R. P. Bywater. Prediction of protein residue contacts with a PDB-derived likelihood matrix. *Protein Engineering*, 15:721–725, 2002. 23, 24

[173] M. Sippl. Recognition of errors in three-dimensional structures of proteins. *Proteins*, 17:355–362, 1993. 30

[174] M. J. Sippl. Calculation of conformational ensembles from potentials of mean force. *Journal of Molecular Biology*, 213:859–883, 1990. 23

[175] J. Soding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21:951–960, 2005. 25

[176] A. Street and S. Mayo. Intrinsic beta-sheet propensities result from van der waals interactions between side chains and the local backbone. *Proceedings of the National Academy of Sciences*, 96:9074–9076, 1999. 28

[177] C. M. Summa, M. Levitt, and W. F. DeGrado. An atomic environment potential for use in protein structure prediction. *Journal of Molecular Biology*, 352:986–1001, 2005. 30

[178] N. L. Summers and M. Karplus. Construction of side-chains in homology modeling: Application to the c-terminal lobe of rhizopuspepsin. *Journal of Molecular Biology*, 210:785–810, 1989. 25

[179] M. Takeda, T. Ikeya, P. Güntert, and M. Kainosho. Automated structure determination of proteins with the SAIL-FLYA NMR method. *Nature Protocols*, 2:2896–2902, 2007. 13, 19

[180] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *Proceedings of the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS)*, pages 26–33, 2003. 113

[181] W. R. Taylor and K. Hatrick. Compensating changes in protein multiple sequence alignments. *Protein Engineering*, 7:341–348, 1994. 23

[182] D. J. Thomas, G. Casari, and C. Sander. The prediction of protein contacts from multiple sequence alignments. *Protein Engineering*, 9:941–948, 1996. 23

[183] S. C. Tosatto. The victor/FRST function for model quality estimation. *Journal of Computational Biology*, 12:1316–1327, 2005. 30

[184] M. Tress, D. T. Jones, and A. Valencia. Predicting reliable regions in protein alignments from sequence profiles. *Journal of Molecular Biology*, 330:705–718, 2003. 30, 31

[185] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *The* 21^{st} *International Conference on Machine Learning*, volume 69, pages 104–111, 2004. 113, 114, 117

[186] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005. 113

[187] P. Tuffery, C. Etchebest, S. Hazout, and R. Lavery. A new approach to the rapid determination of protein side chain conformations. *Journal of Biomolecular Structure and Dynamics*, 8:1267–1289, 1991. 28

[188] M. Vasquez. An evaluation of discrete and continuum search techniques for conformational analysis of side-chains in proteins. *Biopolymers*, 36:53–70, 1995. 28

[189] J. Volk, T. Herrmann, and K. Wüthrich. Automated sequence-specific protein NMR assignment using the memetic algorithm MATCH. *Journal of Biomolecular NMR*, 41:127–138, 2008. 13, 15, 16, 17

[190] G. Wagner, A. Pardi, and K. Wüthrich. Hydrogen-bond lenght and H-1-NMR chemical shifts in proteins. *Journal of American Chemical Society*, 105:5948–5949, 1983. 20

[191] B. Wallner and A. Elofsson. Can correct protein models be identified? *Protein Science*, 12(5):1073–1086, 2003. 30

[192] B. Wallner and A. Elofsson. Identification of correct regions in protein models using structural, alignment, and consensus information. *Protein Science*, 15:900–913, 2005. 30, 31, 124, 128, 129, 131, 143

[193] B. Wallner and A. Elofsson. Prediction of global and local model quality in CASP7 using Pcons and ProQ. *Proteins*, 69(S8):184–193, 2007. 30

[194] X. Wan and G. Lin. CISA: combined NMR resonance connectivity information determination and sequential assignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4:336–348, 2007. 13, 15, 16, 72, 73

[195] M. P. Williamson and C. J. Craven. Automated protein structure calculation from NMR data. *Journal of Biomolecular NMR*, 43:131–143, 2009. 13

[196] D. S. Wishart, D. Arndt, M. Berjanskii, P. Tang, J. Zhou, and G. Lin. CS23D: a web server for rapid protein structure generation using NMR chemical shifts and sequence data. *Nucleic Acids Research*, 36:496–502, 2008. 13, 19, 20

[197] K. P. Wu, J. M. Chang, J. B. Chen, C. F. Chang, W. J. Wu, T. H. Huang, T. Y. Sung, and W. L. Hsu. RIBRA - an error-tolerant algorithm for the NMR backbone assignment problem. *Journal of Computational Biology*, 13:229–244, 2006. 13, 15, 16

[198] S. Wu and Y. Zhang. LOMETS: a local meta-threading-server for protein structure prediction. *Nucleic Acids Research*, 35:3375–3382, 2007. 24, 25

[199] S. Wu and Y. Zhang. A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. *Bioinformatics*, 24:924–931, 2008. 23, 24, 25, 102, 104

[200] Kurt Wüthrich. *NMR of Proteins and Nucleic Acids*. John Wiley & Sons, New York, 1986. 10

[201] Z. Xiang and B. Honig. Extending the accuracy limits of prediction for side-chain conformations. *Journal of Molecular Biology*, 311:421–430, 2001. 26, 118

[202] J. Xu. Protein fold recognition by predicted alignment accuracy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(2):157–165, 2005. 30, 94, 127

[203] J. Xu and B. Berger. Fast and accurate algorithms for protein side-chain packing. *Journal of the ACM*, 53:533–557, 2006. 6, 26, 27, 28, 29, 118, 122

[204] J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *Journal of Bioinformatics and Computational Biology*, 1:95–117, 2003. 21, 94, 122

[205] Y. Xu, D. Xu, and J. Liang. *Computational Methods for Protein Structure Prediction and Modeling (Volume 1)*. Springer, New York, 2007. 21

[206] Y. Xu, D. Xu, and J. Liang. *Computational Methods for Protein Structure Prediction and Modeling (Volume 2)*. Springer, New York, 2007. 21

[207] Y. Xu, D. Xu, and V. Olman. A practical method for interpretation of threading scores: an application of neural networks. *Statistica Sinica Special Issue on Bioinformatics*, 12:159–177, 2002. 25

[208] C. Yanover, O. Schueler-Furman, and Y. Weiss. Minimizing and learning energy functions for side-chain prediction. *Journal of Computational Biology*, 15(7):899–911, 2008. 27, 28

[209] M. J. Zaki, S. Jin, and C. Bystroff. Mining residue contacts in proteins using local structure predictions. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(5):789–801, 2003. 24

[210] A. Zemla, C. Venclovas, J. Moult, and K. Fidelis. Processing and evaluation of predictions in CASP4. *Proteins*, 45:13–21, 2001. 143

[211] J. Zhang, X. Gao, J. Xu, and M. Li. Rapid and accurate protein side chain prediction with local backbone information. In *Proceedings of the Twelfth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 285–299, 2008. 6

[212] Y. Zhang. Template-based modeling and free modeling by I-TASSER in CASP7. *Proteins*, 8:108–117, 2007. 22, 109, 145

[213] Y. Zhang. I-TASSER server for protein 3D structure prediction. *BMC Bioinformatics*, 9(40), 2008. 145

[214] Y. Zhang, A. Arakaki, and J. Skolnick. TASSER: an automated method for the prediction of protein tertiary structures in CASP6. *Proteins*, 61(S7):91–98, 2005. 22, 109

[215] Y. Zhang, A. Kolinski, and J. Skolnick. TOUCHSTONE II: a new approach to ab initio protein structure prediction. *Biophysical Journal*, 85:1145–1164, 2003. 23

[216] Y. Zhang and J. Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *Proceedings of the National Academy of Sciences*, 101(20):7594–7599, 2004. 21, 29, 30

[217] Y. Zhang and J. Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins*, 57:702–710, 2004. 30, 106

[218] Y. Zhao and G. Karypis. Prediction of contact maps using Support Vector Machines. In *Proceedings of the Third IEEE International Conference on Bioinformatics and Bioengineering (BIBE)*, pages 26–33, 2003. 23, 24

[219] D. Zheng, Y. J. Huang, H. N. Moseley, R. Xiao, J. Aramini, G. V. Swapna, and G. T. Montelione. Automated protein fold determination using a minimal NMR constraint strategy. *Protein Science*, 12:1232–1246, 2003. 13, 18, 19

[220] 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 Science*, 11:2714–2726, 2002. 30

[221] H. Zhou and Y. Zhou. Single body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins*, 55:1005–1013, 2004. 30

[222] 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. 25, 30, 94

[223] D. E. Zimmerman, C. A. Kulikowski, Y. Huang, W. Feng, M. Tashiro, S. Shimotakahara, C. Chien, R. Powers, and G. T. Montelione. Automated analysis of protein NMR assignments using methods from artificial intelligence. *Journal of Molecular Biology*, 269:592–610, 1997. 13, 15, 16, 19