University of Alberta

AN INTEGER PROGRAMMING APPROACH FOR PROTEIN NMR STRING ASSIGNMENT

by

Junfeng Wu 

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 2007

Your file Votre référence
ISBN: 978-0-494-30040-4
Our file Notre référence
ISBN: 978-0-494-30040-4

To my family,
Thanks for years of support.

Abstract

String assignment is one of the key steps in protein three-dimensional structure determination via NMR (Nuclear Magnetic Resonance) spectroscopy. It addresses the problem of how to map non-overlapping strings of spin systems to the target protein sequence, and it has been proven that finding an optimal solution for string assignment is NP-hard. Many methods have been developed to solve this problem. However, most of them have speed and scalability issues which make them unsuitable for large proteins, or high volume input data. We have developed a new approach to solving the string assignment based on integer programming. Experimental results have shown that our approach has advantages over other methods in both speed and scalability. Moreover, we applied the idea into a web application which provides high performance protein NMR string assignment function to users all over the world.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

NMR	Nuclear Magnetic Resonance
IP	Integer Programming
GARANT	General Algorithm for Resonance AssignmeNT
PACES	Protein sequential Assignment by Computer-assisted Exhaustive Search
RIBRA	Relaxation and Iterative Backbone Resonance Assignment
GLPK	GNU Linear Programming Kit
PSAtip	Protein NMR Sequential resonance Assignment though integer program
FCFS	First Come First Serve
HBSS	Histogram Based Scoring Scheme

Chapter 1

Introduction

Proteins play a vital role in supporting our daily physiological activities. Their functions ranged from building muscle tissues and connective tissues, hydrolyzing the polymers in food, protecting against diseases, to transferring oxygen and other sustenance over the body. Some functions can be done by a single protein, and others require interactions of multiple proteins. Recent research has shown that the functions of a protein and how it interacts with other proteins are mainly determined by its three-dimensional structure (Thornton et al., 2000).

Obtaining the three-dimensional structure of a protein is not easy. With years of study, researchers have already built the whole databases of genome sequences for many species including the human being, and are able to extract the amino acid sequences of proteins from the databases. Unfortunately, predicting how the protein sequence folds in a three-dimensional space remains as an extremely difficult task due to the complex interactions between amino acids. Many computational prediction methods have been developed to address this problem in the last few decades, such as molecular dynamics, secondary structure prediction, homology and pattern recognition, energy minimization on lattice models, and knowledge-based approaches, e.g., protein threading (Clote & Backofen, 2000). These methods have been proven to be able to produce impressive prediction results. However, they still cannot replace the experimental techniques for the protein three-dimensional structure determination for two reasons. First reason is that none of the computational methods can guarantee to produce the correct result for a particular protein. Second, for a new protein, all the computational methods can only "predict" its three-dimensional structure and we still need experimental results to verity the prediction.

Two experimental techniques, X-Ray crystallography and NMR spectroscopy, are the dominant methods for protein structure determination. They both have their own advantages and disadvantages. In general, X-Ray crystallography provides higher accuracy for protein three-dimensional structure than NMR spectroscopy, but it may take from several months to several years to determine the structure of one protein. Compared with X-Ray crystallography, NMR spectroscopy wins with its efficiency and lower cost. Therefore, when we need to determine the protein three-dimensional structures of a large dataset, for instance, at the genomic scale, NMR spectroscopy is a more suitable solution than X-Ray crystallography.

Since 1985, the first time that NMR was used in determining the complete structure for a protein (proteinase inhibitor IIA from bull seminal plasma) (Williamson et al., 1985), much effort has been expended to make NMR protein structure determination process au-

tomatic. Modern computing techniques can help biologists analyze NMR data to determine the protein three-dimensional structure with high accuracy.

In general, the whole process of using NMR spectroscopy for the protein three-dimensional structure determination can be done in three stages: peak picking, protein NMR sequential resonance assignment, and structure determination. In NMR spectra, peaks are generated from chemical shifts in a protein molecule and reflect the properties of the atoms that form the chemical shifts. The stage of peak picking is to identify the peaks of atoms in a protein molecule. Then the observed peaks along with the protein sequence are used in the next stage, protein NMR sequential resonance assignment, in short, sequential resonance assignment, to produce mappings of observed peaks and atoms of residues in the target protein. Finally, the three-dimensional structure of the protein can be determined at the structure determination step, by combining the protein sequence with the positions of atoms in three-dimensional space.

The process of sequential resonance assignment consists of three steps and a scoring scheme (see Figure 1.1). The first step is peak grouping. In the peak grouping step, spin systems are generated from observed peaks. Each spin system contains chemical shifts of one residue and some other information, such as coupling constants. Then, the step of connectivity determination estimates the connections between every two spin systems according to their chemical shift values. If two spin systems are believed from two neighboring residues, a directed connection is made from one to the other according to the order of residues. Several sequentially connected spin systems form a segment of spin systems, which is referred as a string. Finally, guided by a scoring scheme, the step of string assignment maps strings to correct positions on the target protein sequence. The scoring scheme evaluates every possible spin system/residue mapping and provides the confidence information for each mapping. It plays an important role in the sequential resonance assignment because the quality of the scoring scheme mainly determines the accuracy of the sequential resonance assignment.

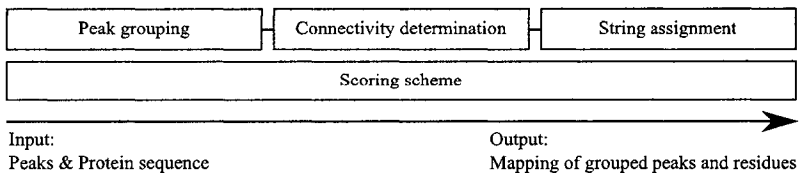


Figure 1.1: Protein NMR sequential resonance assignment

In our research, an Integer Programming (IP) approach is designed to solve the string

assignment. Compared to existing methods, our approach not only speeds up the string assignment process but also can handle large proteins. It takes only a few seconds to solve string assignment for protein with more than 700 residues while other methods may take hours, even days. In additional, some experiments which are not feasible to other methods can be easily done because of the efficiency of our approach, for instance, scoring scheme evaluation on a large dataset. Scoring scheme is very important to string assignment; however, without an efficient method, researchers can only test scoring schemes on small dataset with limited protein size. What's more, a web application based on the IP approach is designed and developed to provide public string assignment service to users all over the world.

Chapter 2

Related Work

Many approaches have been proposed for sequential resonance assignment, one of the key stages in the NMR protein three-dimensional structure determination. Several programs/systems have been developed to accomplish this task. In these programs/systems, different algorithms were used for the string assignment process and are summarized as below.

2.1 GARANT

GARANT (General Algorithm for Resonance AssignmeNT) (Bartels et al., 1997) is a program which uses a combination of genetic algorithm and local optimization routine for sequential resonance assignment. It contains three major elements: graph representation, a scoring scheme, and a combination algorithm.

GARANT represents NMR resonance assignment as a graph matching problem between two graphs: (1) expected graph which represents correlations of the atoms of the protein and expected cross peaks (graph A in Figure 2.1 (Bartels et al., 1997)), and (2) observed graph which represents , correlations of the chemical shifts and observed cross peaks (graph B in Figure 2.1).

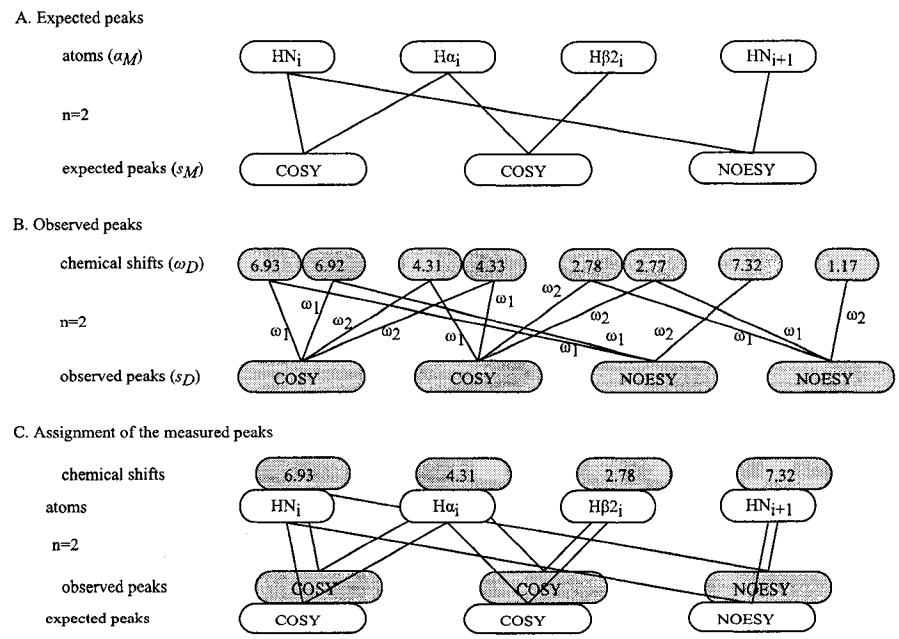


Figure 2.1: Schematic representation for 2D ($n = 2$) homonuclear NMR spectra of expected (A) and observed (B) peaks, and of the mapping used to describe possible resonance assignments (C).

GARANT uses the "mutual information" as the scoring scheme to evaluate the mapping

between the expected graph and the observed graph. The "mutual information" between a set of expected peaks/atoms M and a set of all observed peaks/chemical shifts D over a resonance assignment A is defined as

$$\begin{aligned} I_A(D;M) &= \sum_{k\in K} I_A(\alpha_D^{(k)};\alpha_M^{(k)}) \\ &= \sum_{k\in K} \log \frac{p(\alpha_D^{(k)}|\alpha_M^{(k)})}{p(\alpha_D^{(k)})} \\ &= \sum_{k\in K} \log \frac{p(\alpha_D^{(k)}|\alpha_M^{(k)})}{\sum_{l\in L_k} p(\alpha_D^{(k)}|\alpha_{M,l}^{(k)})p(\alpha_{M,l}^{(k)})}, \end{aligned}$$

where K is a set of attributes k, L_k is a set of possible values l for attribute k. In above equation, $p(\alpha_D^{(k)}|\alpha_M^{(k)})$ is the conditional probability of the value $\alpha_D^{(k)}$ observed when its expected value is $\alpha_M^{(k)}$ for attribute k, and $p(\alpha_D^{(k)})$ is *a priori* probability that $\alpha_D^{(k)}$ is observed for attribute k.

Then, the joint algorithm of a genetic algorithm and a local optimization routine is used to find the best matching, Figure 2.1 C, between two graphs, Figure 2.1 A and B. This best matching can therefore be converted to the corresponding best resonance assignment.

The combination algorithm can be briefly described as Algorithm 1, where s_{max} controls the maximum number of generations in the genetic algorithm in case that the algorithm does not converge. One disadvantage of this algorithm is that the genetic algorithm usually converges very slowly; therefore, a large number of generations has to be set to terminate the algorithm. Another disadvantage is that the complexity of the two graphs grows exponentially in the size of the protein. Some heuristic techniques are required to be applied for using this program on large proteins.

```
1 Find a local optimal assignment for the initial generation;
2 while Step t < t_max do
3 |   Generate 30 candidate assignments;
4 |   Combine to a new assignment;
5 |   if Converged then Stop loop;
6 |   else t ← t + 1;
7 end
8 Output final assignment;
```

Algorithm 1: the combination algorithm in GARANT

2.2 AutoAssign

AutoAssign (Zimmerman et al., 1997) is an expert system which combines symbolic constraint satisfaction methods with a domain-specific knowledge base for determining reso-

nance assignments from NMR spectra of proteins. This system can be run in two modes, the interactive mode or the "batch" mode. Figure 2.2 (Zimmerman et al., 1997) shows an overview of the processing sequence of AutoAssign.

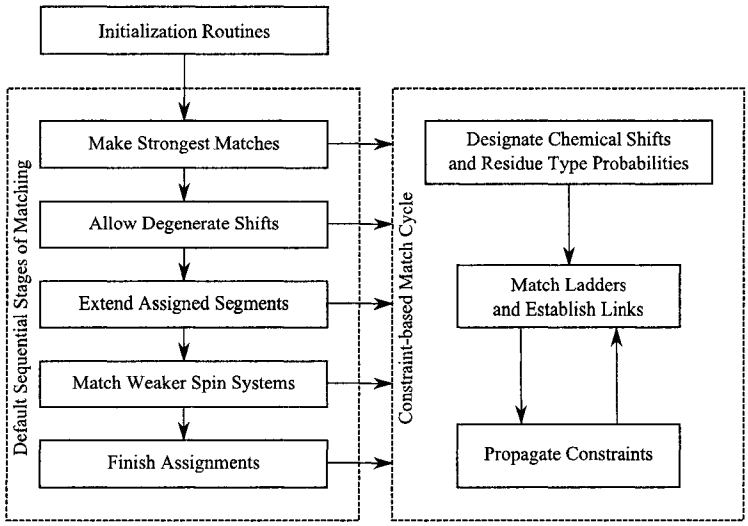


Figure 2.2: Schematic overview of AUTOASSIGNs default execution sequence

First, the input information is initialized to generate the internal representation for AutoAssign. Then a sequence of stages of "constraint-based matching" are taken place to designate chemical shifts and match the C^O-ladders (the connections between spin systems) with C^α-ladders (the mappings between spin systems and residues) for each spin system till the final assignments are found. At each stage, AutoAssign uses the constraint-based matching cycle to iteratively establish the best matches between the C^O-ladders and C^α-ladders, where the matches are the "best possible" sequential links between GSs (genetic amino acid spin-system objects derived from NMR spectral data). Here, "best" is evaluated by the scoring scheme which represents the Bayesian posterior probability $p(r|C^\alpha, C^\beta)$ of a spin system with C^α and C^β chemical shifts mapping to a residue r. And the probability is computed as

$$p(r|C^\alpha, C^\beta) = \frac{p(C^\alpha, C^\beta|r)P(r)}{\sum_r p(C^\alpha, C^\beta|r)P(r)},$$

where $p(C^\alpha, C^\beta|r)$ is the probability of observing chemical shift values C^α and C^β for residue type r, and $P(r)$ is the frequency of occurrence of residue type r in the target protein sequence.

In AutoAssign, the connectivity determination process and the string assignment process are combined together. This combination allows the two processes to validate each

other, also it increases the assignment speed by reducing the total number of possible connections. Another strategy to reduce the search space is to combine best-first search with constraint satisfaction methods (Mackworth, 1977; Fox, 1986; Nadel, 1986; Kumar, 1992). However, due to the propagation of initial errors, AutoAssign only works when errors are rare. Moreover, AutoAssign requires a careful definition of what constitutes logical inconsistencies to minimize the propagation of errors. Consequently, AutoAssign may fail due to missing data or ambiguous connections. In summary, AutoAssign requires seven to eight three-dimensional NMR spectra in order to reduce the complexity and provide meaningful assignments.

2.3 Mapper

Mapper (Güntert et al., 2000) is a semi-automatic sequence-specific NMR assignment program which uses an input of short fragments of sequentially neighboring spin systems and performs an exhaustive search for self-consistent simultaneous mappings of all these fragments onto the protein sequence. The idea can be explained in two steps.

First, consider each fragment individually and map it to all possible locations in the protein sequence. Then, use a scoring scheme to compute the evaluation value and the acceptable probability for each mapping. For the fragment F_i with length l_i, the evaluation value $\chi^2(i;k)$ of mapping it to a protein subsequence $s_{k,k+l_i-1}$, which indicates the segment from the k-th amino acid to the $(k+l_i-1)$-th amino acid in the whole protein sequence, is computed as the sum of the squared deviations of the chemical shift values in F_i from the corresponding reference chemical shift values at the position $k, \ldots, k+l_i-1$ in the protein sequence:

$$\chi^2(i;k) = \sum_{j=0}^{l_i-1} \sum_{a \in A_j(i)} \left[\frac{\omega_j^a(i) - \tilde{\omega}^a_{r(k+j)}}{\Delta\tilde{\omega}^a_{r(k+j)}} \right]^2, \tag{2.1}$$

where $A_j(i)$ denotes the set of atoms at position j in the fragment F_i, and $\omega_j^a(i)$ denotes the experimental chemical shift value for the atom $a \in A_j(i)$, and $\tilde{\omega}^a_{r(k+j)}$ and $\Delta\tilde{\omega}^a_{r(k+j)}$ are the expected chemical shift value and its standard deviation for the atom a of the residue type $r(k+j)$, respectively. This mapping can be accepted for the next step if the acceptable probability $Q(\chi^2(i;k)|\nu_i)$, where $\nu_i = \sum_{j=0}^{l_i-1} |A_j(i)|$ is the total number of atoms in the fragment F_i, is greater than a user specific threshold Q_0.

Second, an exhaustive search for simultaneous, self-consistent global mappings of all fragments is applied on the basis of the accepted individual mappings. For each global

mapping, its global evaluation value $\chi^2(Global)$ is computed as the sum of each individual mappings, and this mapping is ranked by its acceptable probability $Q(Global)$, which is calculated in the same way as individual mappings. A global mapping is considered as a "reasonable" mapping if its acceptable probability is close to 100%.

2.4 PACES

PACES (Protein sequential Assignment by Computer-assisted Exhaustive Search) (Coggins & Zhou, 2003) applies an exhaustive search on a directed graph, which represents the connectivity relationship of spin systems, to find the best non-overlapping paths and the corresponding mapping between these paths and the protein sequence segments. In this graph, each vertex represents one spin system and each directed edge represents an adjacency between two spin systems. These edges are built by the following rules. Given two spin systems,

$$\begin{aligned} s_j &= C_j^{\alpha}, C_j^{\beta}, C_j^{O}, H_j^{\alpha}, C_{j-1}^{\alpha}, C_{j-1}^{\beta}, C_{j-1}^{O}, H_{j-1}^{\alpha}, \\ s_k &= C_k^{\alpha}, C_k^{\beta}, C_k^{O}, H_k^{\alpha}, C_{k-1}^{\alpha}, C_{k-1}^{\beta}, C_{k-1}^{O}, H_{k-1}^{\alpha}, \end{aligned}$$

there is an edge from spin system s_j to spin system s_k, if

$$\begin{aligned} |C_j^{\alpha} - C_{k-1}^{\alpha}| &\leq \delta_{C^{\alpha}}, \\ |C_j^{\beta} - C_{k-1}^{\beta}| &\leq \delta_{C^{\beta}}, \\ |C_j^{O} - C_{k-1}^{O}| &\leq \delta_{C^{O}}, \\ |H_j^{\alpha} - H_{k-1}^{\alpha}| &\leq \delta_{H^{\alpha}}, \end{aligned}$$

where $\delta_{C^{\alpha}}$, $\delta_{C^{\beta}}$, $\delta_{C^{o}}$, and $\delta_{H^{\alpha}}$ are user-specified connectivity thresholds.

After the graph is built, all possible paths are enumerated. If there is a cycle in a path, the path is cut at the last visited vertex to break the cycle (see Figure 2.3). When all paths are ready, each path is examined by mapping it to every possible location on the protein sequence. A scoring scheme which contains the statistical chemical shift distribution of each amino acid type from the BioMagResBank is used to determine the possible amino acid type that a spin system can be mapped to. If only a part of the path can be mapping to a segment of protein sequence, the rest of path is cut and added back to the path pool for assignment elsewhere.

Ideally, the alignment could be straightforward when each path only contains correct connections and no two paths share the same vertices. However, in practice, due to chemical shift degeneracy, a spin system could be connected to several other spin systems, which

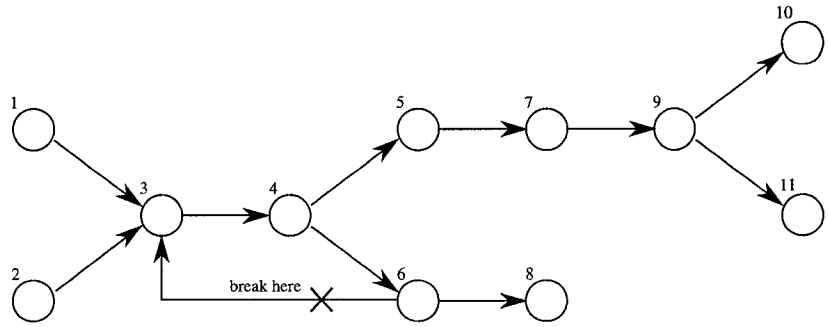


Figure 2.3: Break the cycle in a path. The line from spin system 6 to spin system 3 is cut to break the cycle.

exponentially increases the total number of all possible paths. Then, enumerating all possible paths is very difficult if not impossible. Applying an exhaustive search upon all paths becomes a both time consuming and memory consuming process. Therefore, PACES is only suitable for short sequences or when high quality data are available.

2.5 MARS

MARS (Jung & Zweckstetter, 2004) is an automatic backbone assignment program which contains five key features: (1) simultaneous optimization for both the local and the global assignment quality, (2) exhaustive search for segments containing up to five connected spin systems during linking and mapping, (3) best-first elements for both linking and mapping, (4) combination of the secondary structure, and (5) evaluation mappings by performing multiple assignment.

Figure 2.4 shows an overview of MARS assignment procedure. In the first step, MARS detects the connectivities among spin systems to build a graph. Spin systems are connected according to their experimental intra- and inter-chemical shifts. In the second step, MARS maps the connected spin systems onto the protein sequence. After all spin systems are randomly mapped to the protein sequence, MARS repeats to randomly select a spin system as a start point to refine this assignment. It checks a given number of spin systems according to the graph created in the first step until all spin systems have been selected as start points. In MARS, the number of spin systems checked are reduced from five to two in each repeat.

The main factor influencing the assignment quality of MARS is quality of chemical shift values of spin systems. For low quality NMR spectral data, the limited length of examined spin systems string may limit the performance of MARS.

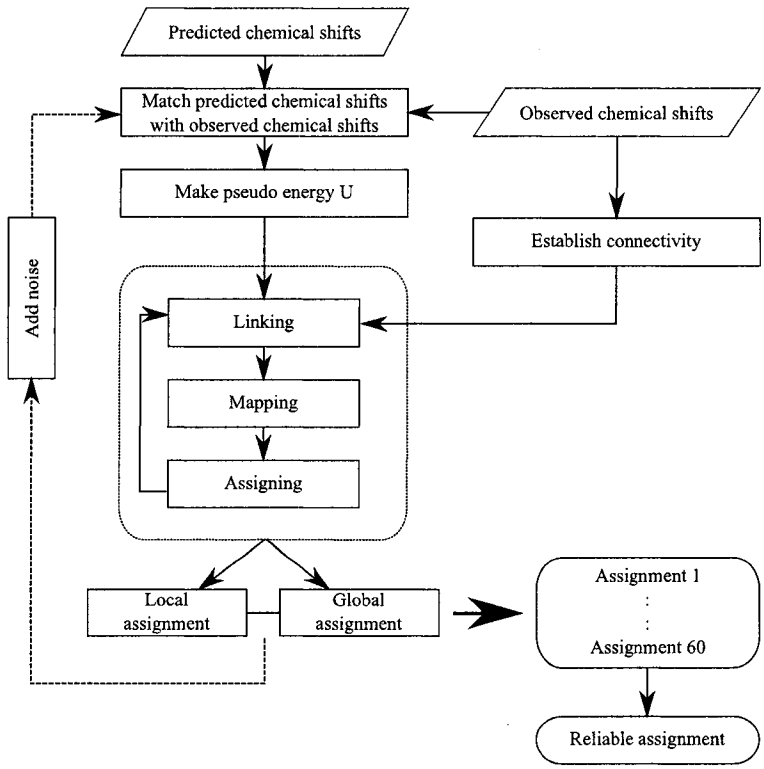


Figure 2.4: Overview of MARS assignment procedure

2.6 RIBRA

RIBRA (Relaxation and Iterative Backbone Resonance Assignment) (Wu et al., 2006) is a fully automated program for NMR resonance assignment using nearest neighbor and weighted maximum independent set algorithm. In RIBRA, all spin systems are first paired with each other and each pair is placed on all possible locations on the protein sequence according to TATAPRO II typing scheme (Atreya et al., 2002). The pair of spin systems can be viewed as a string with two spin systems. Then, a segment extension algorithm is applied on these spin system pairs to form longer strings if two pair share the same residues and have matched overlapping spin systems. After all strings are collected, RIBRA treats the string assignment problem as finding a maximum independent set of strings of spin systems to cover the target protein sequence (see Figure 2.5).

RIBRA performs very well when high quality data are available. However, when the data quality is low, the number of strings of spin systems is increased exponentially as many spin systems pairs are overlapping. Therefore, the performance of RIBRA on poor quality dataset drops rapidly. Even worse, it becomes infeasible for long protein sequence with low quality spectral data.

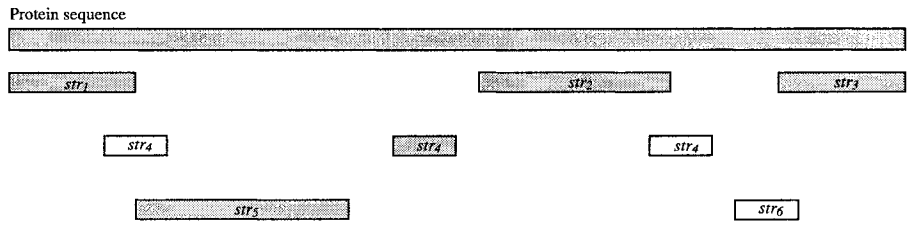


Figure 2.5: Finding independent set of strings of spin systems in RIBRA. str_1 to str_6 are six strings of spin systems. One segment can be initially mapped to multiple positions on the target protein sequence, such as string str_4. String str_1, str_2, str_3, and str_5 form a maximum independent set which covers the protein sequence.

2.7 Summary

Based on the methods used to solve sequential resonance assignment, above six systems could be divided into two groups. The first group only contains GARANT which represents the sequential resonance assignment as graph matching. It maps peaks directly to the residues and tries to find the best matching between the observed mappings and expected mappings. The disadvantage of this approach is that it cannot handle large proteins due to the complexity of groups grows exponentially in the size of proteins. The second group contains the other five systems. The method used in this group for solving string assignment can be summarized as following. First, it tries to enumerate all strings. Then, it searches all possible mapping positions for each string to find the best set of strings to cover the target protein sequence. The main drawback of this method is that the number of total strings increases exponentially when the connection complexity among spin systems increases. Therefore, this method is only suitable for string assignment when high quality spectra data are available, in this case the connections between spin systems are not too complicated. To handle large proteins with general connections between spin systems, a new method is desired for string assignment.

Chapter 3

Protein NMR String Assignment Using Integer Programming

In this chapter, an integer programming approach is demonstrated for finding the constrained maximum-weighted bipartite matching, which is the best candidate for the correct assignment between spin systems and residues.

3.1 String Assignment as bipartite matching

After peak grouping and connectivity determination, we have a set of spin systems and there are some connections between spin systems. Given the amino acid residues from the target protein, we need to map these spin systems onto the corresponding residues. At the beginning of the string assignment process, we can build initial mappings between spin systems and residues. In the initial mappings, each spin system is mapped to multiple residues, and each spin system/residue mapping is evaluated by a scoring scheme based on chemical shift values of the spin system and the type of the residue to gives the confidence of such mapping. The scoring scheme assigns a value to each mapping which indicates the confidence level of such mapping and the bigger the value, the more confident the mapping is. We can view the initial mappings as a bipartite graph (see Figure 3.1 (A)), and links between spin systems and residues are weighted by their confidences. The string assignment is to find the maximum-weighted bipartite matching (see Figure 3.1 (B)) of the graph, which represents the "best" mappings of spin systems and residues.

Although finding a maximum weighted matching is not difficult, it becomes complicated when there are connectivity constraints which have to be satisfied in the matching (see Figure 3.1 (C)), i.e., two adjacent spin systems have to be mapping to two neighboring residues. Xu et al. (Xu et al., 2002) have proved that the constrained bipartite matching is NP-hard even if all edges have the unit weight and the maximum length of strings of spin systems is 2. For a more general case, where the length of strings of spin systems could be any number from 1 to the total number of spin systems, the constrained maximum weighted bipartite matching becomes Max SNP-hard (Chen et al., Mar 2005). In the following sections, We will show how to formulate the string assignment as an integer program.

3.2 Integer Program Representation for String Assignment

3.2.1 Basic Integer Program

First of all, let's consider the simplest case without any connectivity constrains. In the simplest case, given a set of spin systems containing no false positive (fake) and false negative (missing) spin systems, the string assignment process is to find a solution in which each

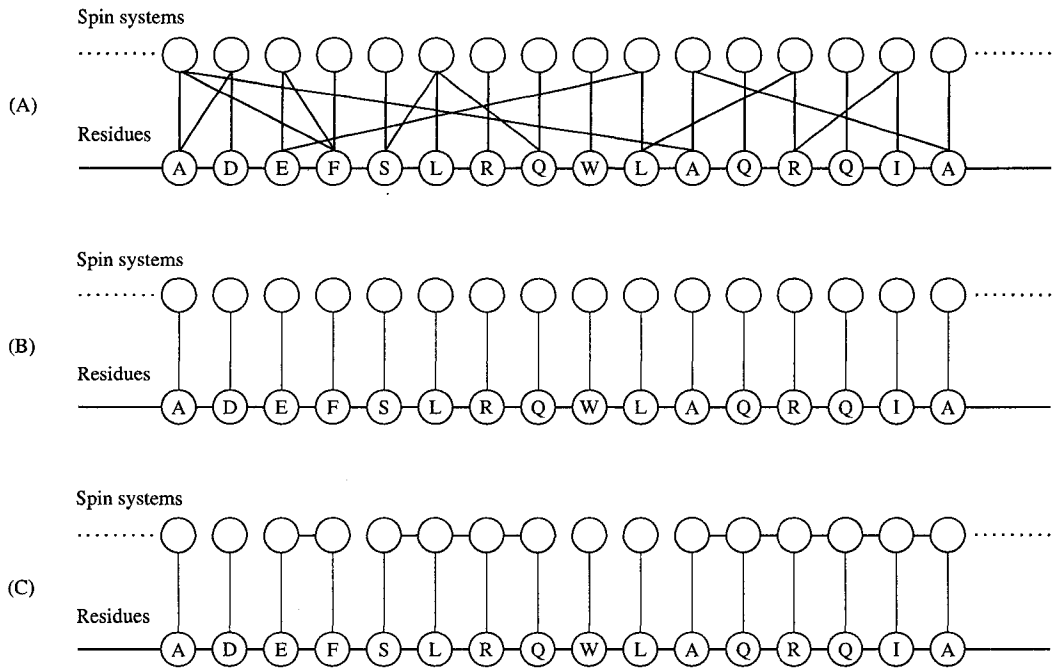


Figure 3.1: (A) initial mappings where lines between spin system and residue indicate possible mappings; (B) bipartite matching without connectivity constraints; (C) bipartite matching with connectivity constraints.

spin system has to be mapped to one and only one residue. Given weights on every mapping of spin system/residue pair, where the weight is the confidence of each mapping, this process can be modeled as an integer programming problem, or more specifically, a 0-1 integer program.

Suppose the length of the protein sequence is n, i.e., there are n residues, and we have n spin systems, we want to map these n spin systems to the n residues so that each spin system corresponds to one residue. Let the residue set be $R = \{r_j\}, 1 \leq j \leq n$, where j is the index of residue r_j in the protein sequence. Let spin system set be $S = \{s_i\}, 1 \leq i \leq n$. Given the weight matrix $W = \{w_{ij}\}, 1 \leq i, j \leq n$, where each element w_{ij} is a non-negative number which represents the confidence of mapping the spin system s_i to the residue r_j. Given the assignment matrix $X = \{x_{ij}\}, 1 \leq i, j \leq n$ and $x_{ij} \in \{0, 1\}$, where $x_{ij} = 1$ if we map the spin system s_i to the residue r_j and $x_{ij} = 0$ otherwise. Then finding the best assignment is equivalent to finding a configuration of X which maximizes $\sum_{1 \leq i,j \leq n} w_{ij} \times x_{ij}$, where $x_{ij} \in \{0, 1\}$. Because one spin system must be mapped to one and only one residue and vice versa, in the matrix X, there is only one element equals to 1 for each column and for each row. Therefore, the basic 0-1 integer program can be

formulated as Formulae (3.1 ~3.4)

$$\text{Maximize} \quad \sum_{1 \le i,j \le n} w_{ij} \times x_{ij} \tag{3.1}$$

$$\text{subject to} \quad \sum_{1 \le j \le n} x_{ij} = 1, \forall 1 \le i \le n \tag{3.2}$$

$$\sum_{1 \le i \le n} x_{ij} = 1, \forall 1 \le j \le n \tag{3.3}$$

$$x_{ij} \in \{0, 1\} \tag{3.4}$$

In practice, in the NMR experiments, there are usually two kinds of noise remaining in the spectral data. One is that an actual peak could be missing (false negative), and the other is that a visible peak could be false (false positive) (Hsu et al., 2004). Therefore, the number of observed spin systems may be less or more than the number of the amino acid residues. The false positive spin systems should not be mapped to any residue and some resides may not have the corresponding spin systems. In this case, we cannot guarantee to find the one-to-one mapping between spin systems and residues. The assignment constraints, Formulae (3.2, 3.3), have to be relaxed. Let m be the number of spin systems and n be the number of residues. We get a more general 0-1 integer program shown as Formulae (3.5, 3.6, 3.7, 3.4).

$$\text{Maximize} \quad \sum_{1 \le i \le m, 1 \le j \le n} w_{ij} \times x_{ij} \tag{3.5}$$

$$\text{subject to} \quad \sum_{1 \le j \le n} x_{ij} \le 1, \forall 1 \le i \le m \tag{3.6}$$

$$\sum_{1 \le i \le m} x_{ij} \le 1, \forall 1 \le j \le n \tag{3.7}$$

$$x_{ij} \in \{0, 1\}$$

3.2.2 Simple Connectivity Constrained Integer Program

Our experiments (Section 5.4) have shown that the assignment results from the basic integer program, Formulae (3.5, 3.6, 3.7, 3.4), are usually biologically poor, i.e., many spin systems are mapped to wrong residues. The reason is that for spin systems corresponding to the same type of residues, their spectral data are usually very similar to each other. Therefore, these spin systems could be mapped to a residue with equally high confidence. On the other hand, one spin system could be mapped to several the same type of residues also with equally high confidence.

To resolve this issue, we introduced the connectivity information into the integer program. The connectivity information is determined by the connectivity determination pro-

cess and shows the relationship between spin systems which gives the constraints for the possible mapping positions for each spin system. In general, if two spin systems are connected to each other, then they must be mapped to two neighboring residues.

To simplify the problem, let us first consider a special case in which the following connectivity constraints hold.

1. Every spin system can have at most one spin system as its immediate successor.
2. Every spin system can have at most one spin system as its immediate predecessor.
3. There is no loop for every spin system chain, i.e., no spin system has any successor which is also its predecessor.

With these constraints, the spin systems form a set of disjoint spin system strings in which no two strings share a spin system (see Figure 3.2). Also the set of strings can be viewed as a set of constraints of spin system pairs. For example, a string containing four spin systems, $s_3 s_2 s_5 s_8$, can be converted to three spin system pairs, $\langle s_3, s_2 \rangle$,$\langle s_2, s_5 \rangle$, and $\langle s_5, s_8 \rangle$.

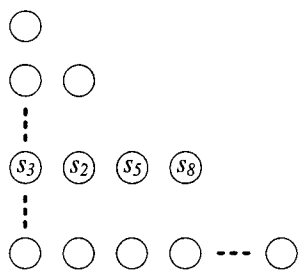


Figure 3.2: Simply connected spin systems

Let $C = \{\langle s_k, s_l \rangle\}$ be the set satisfying the above constraints, where s_k and s_l are two adjacent spin systems and s_k is the immediate predecessor of s_j. Then in the assignment result, two spin systems s_k and s_l must be mapped to two neighboring residues. i.e, if spin system s_k is mapped to residue r_j in the final assignment, spin system s_l has to be mapped to residue r_{j+1}. At the same time, if s_k is not mapped to the residue r_j, s_l must not be mapped to r_{j+1}. Therefore, we have additional constraints for the $\langle s_k, s_l \rangle$ pair (Formula 3.8). And the integer program with such limited connectivity constraints is shown as Formulae (3.5, 3.6, 3.7, 3.8, 3.4).

$$\text{Maximize} \quad \sum_{1 \le i \le m, 1 \le j \le n} w_{ij} \times x_{ij}$$

$$\text{subject to} \quad \sum_{1 \le j \le n} x_{ij} \le 1, \forall 1 \le i \le m$$

$$\sum_{1 \le i \le m} x_{ij} \le 1, \forall 1 \le j \le n$$
$$x_{kj} = x_{l,j+1}, \forall 1 \le j \le n-1 \text{ and } \langle s_k, s_l \rangle \in C \tag{3.8}$$
$$x_{ij} \in \{0, 1\}$$

3.2.3 Complex Connectivity Constrained Integer Program

Due to variability in the accuracy of measurement equipment and chemical shift degeneracy, the spectral data contains noise and errors. The connectivity determination process sometimes can only give the probability of one spin system being adjacent to another spin system, i.e., one spin system may have multiple immediate successors or multiple immediate predecessors. For instance, in Figure 3.3, the spin system s_1 has two possible immediate successors s_2 and s_4, and the spin system s_4 has two possible immediate predecessors s_1 and s_3.

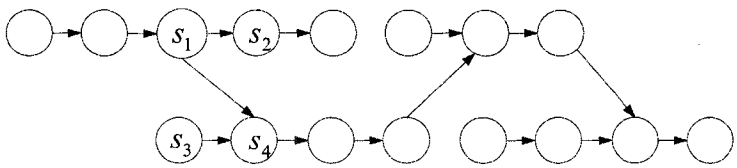


Figure 3.3: Complex connected spin systems

So we need to design an integer program to handle the more general case in which the first two connectivity constraints in Section 3.2.2 are relaxed and the connection loop is still not allowed.

First, let us define two kinds of connections, the "hard" connection and the "soft" connection. If one spin system s_p has only one immediate successor s_q and is the only one immediate predecessor for s_q, the connection between s_p and s_q is called a "hard" connection, which means that this connection has to be satisfied in the assignment if s_p or s_q is mapped to some residue. If one spin system s_p has a set of immediate successors S'_p, where $|S'_p| > 1$, each connection between s_p and its immediate successors is called a "soft" connection, which means that the connection does not have to be satisfied in the assignment. If one spin system s_q has a set of immediate predecessors S'_q, where $|S'_q| > 1$, each connection between s_q and its immediate predecessors is also a "soft" connection. However, if s_p is mapped to a residue in the assignment, one of the "soft" connections between s_p and its immediate successors has to be satisfied. Similarly, if s_q is mapped to a residue in the assignment, one of the "soft" connections between s_q and its immediate predecessors has

to be satisfied.

Therefore, the connectivity information can be partitioned into three sets. The set $C_1 = \{\langle s_k, s_l \rangle\}$ contains only the "hard" connections, the set $C_2 = \{\langle s_p, S'_p \rangle\}$ and the set $C_3 = \{\langle S'_q, s_q \rangle\}$ contain the "soft" connections. In set C_2, element $\langle s_p, S'_p \rangle$ represents all the "soft" connections between s_p and its immediate successors, and in set three, element $\langle S'_q, s_q \rangle$ represents all the "soft" connections between the immediate predecessors of s_q and s_q itself. Algorithm 2 shows the function which separates the connections. The constraint functions for these three sets of connectivities are shown as Formulae (3.9, 3.10, 3.11), respectively.

Input: Set of connections $C = \{< s_k, s_l >\}$
Output: Sets of cataloged connections C_1, C_2, and C_3

```
1  foreach Spin system s ∈ S do
2      if there are multiple connections from s then
3          S'_s ← the successors set of s;
4          C_2 ← Union(C_2, ⟨s, S'_s⟩);
5      end
6      if there are multiple connections to s then
7          S'_s ← the predecessors set of s;
8          C_3 ← Union(C_3, ⟨S'_s, s⟩);
9      end
10 end
11 Remove connections in C_2 or C_3 from C;
12 C_1 ← C;
```

Algorithm 2: Connections catalog algorithm

The integer program with the general connectivity constraints is shown as Formulae (3.5, 3.6, 3.7, 3.9, 3.10, 3.11, 3.4).

$$\text{Maximize} \quad \sum_{1 \le i \le m, 1 \le j \le n} w_{ij} \times x_{ij}$$

$$\text{subject to} \quad \sum_{1 \le j \le n} x_{ij} \le 1, \forall 1 \le i \le m$$

$$\sum_{1 \le i \le m} x_{ij} \le 1, \forall 1 \le j \le n$$

$$x_{kj} = x_{l,j+1} \; \forall 1 \le j \le n-1, \text{if } \langle s_k, s_l \rangle \in C_1 \tag{3.9}$$

$$x_{pj} \le \sum_{s_q \in S'_p} x_{q,j+1} \; \forall 1 \le j \le n-1, \text{if } \langle s_p, S'_p \rangle \in C_2 \tag{3.10}$$

$$x_{q,j+1} \le \sum_{s_p \in S'_q} x_{pj} \; \forall 1 \le j \le n-1, \text{if } \langle S'_q, s_q \rangle \in C_3 \tag{3.11}$$

$$x_{ij} \in \{0, 1\}$$

Chapter 4

PSAtip

Based on our integer program with the general connectivity constraints, Formulae (3.5, 3.6, 3.7, 3.9, 3.10, 3.11, 3.4), We have designed and developed a web application, namely, PSAtip (Protein NMR Sequential Resonance Assignment Through Integer Programming). PSAtip is designed for uses who want to rapidly get mappings of spin systems and residues for determining the protein three-dimensional structure.

4.1 System Architecture of PSAtip

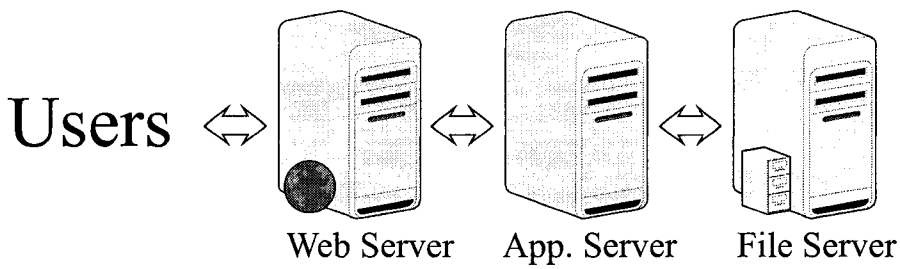


Figure 4.1: PSAtip overview

Figure 4.1 gives an overview of the system architecture of PSAtip. Users submit their jobs of string assignment to PSAtip via a web browser to the front end of PSAtip, which is a web interface that runs on a web server. The uploaded instance data from users are stored on a file server and the assignment requests are added to the request queue awaiting solving. On the application server, a daemon, the request queue monitor, monitors the request queue and starts the back end of PSAtip if there is a request awaiting in the queue. After the jobs are solved, the results of the string assignment are saved on the file server and displayed to users by the front end of PSAtip.

PSAtip contains three modules, the front end (the web interface), the request queue monitor, and the back end (the string assignment module). Among these three modules, the web interface is programmed in PHP[1] and runs on the web server; the request queue monitor and the string assignment module are programmed in Java and deployed on the application server.

4.1.1 Request Queue Monitor

The request queue monitor continuously monitors a FCFS (First Come First Serve) queue of requests and the status of the string assignment module. The string assignment module has two status: "busy" if it is solving an instance and "sleeping" otherwise. Only in "sleeping"

[1] a server-side HTML embedded scripting language. http://www.php.net/

status, the string assignment module can accept and start solving a new instance. When there is a request awaiting in the queue, the monitor module checks the status of the string assignment module first. If the assignment module is sleeping, the monitor module wakes it up to solve the request and remove this request from the queue; otherwise, it holds the request and wait till the assignment module is sleeping. The process of the request queue monitor is shown as Figure 4.2.

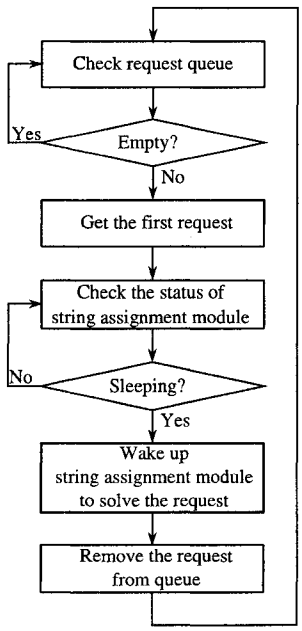


Figure 4.2: The process of request queue monitor

4.1.2 String Assignment Module

The string assignment module solves the assignment requests and prints out the results. It contains four parts, the weight determination function, the connectivity analysis function, the math engine interface, and the math engine (see Figure 4.3).

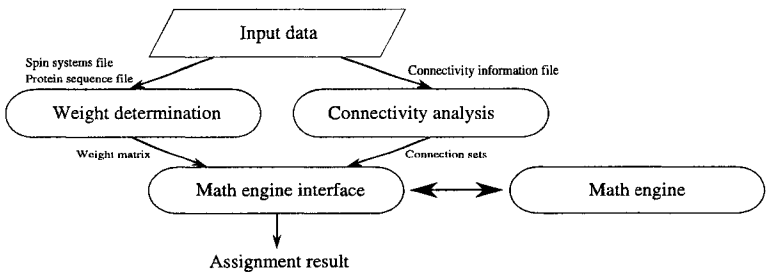


Figure 4.3: Overview of string assignment module

Weight determination function

The weight determination function computes a weight matrix for a string assignment. Entries in the weight matrix are the confidences of mapping each spin system to every residue on the target protein sequence.

The weight determination function requires a scoring scheme to calculate the confidence of each mapping based on the chemical shift values of the spin system and the residue type. There are many choices of scoring schemes that we could choose from. In order to know the appropriate scoring schemes to be used in PSAtip, we have developed a tool, scoring scheme evaluator, to compare the performance of seven scoring schemes on a standard dataset. Finally, two scoring schemes are chosen based on our experimental results (see Section 5.4).

Our experimental results also have shown that using the secondary structure information of the target protein in the weight determination function can always improve the accuracy of final assignments. For this reason, the secondary structure information is also required by the weight determination function. The secondary structure of the protein can be predicted by third party tools, such as PSIPRED[2].

Connectivity analysis function

The connectivity analysis function separates the connections in the connectivity information file into three connectivity constraint sets according to Algorithm 2. These three sets consist of the "hard" connections and the "soft" connections as defined in see Section 3.2.3.

Math engine interface

The math engine interface converts the data in the weight matrix and connectivity constraint sets into the form that can be used by the math engine according to the integer program formulae (3.5, 3.6, 3.7, 3.9, 3.10, 3.11, 3.4). Also it extracts the result from the math engine for outputting the assignment.

Math engine

The math engine is used to solve the integer program instance. There are many math engines available on the market. Some of them are free, for instance GLPK (GNU Linear Programming Kit); some of them are commercial software. Here, we adopted a commercial math engine, CPlex[3], which is well known for its high performance in solving constraint programming problems, such as linear program, integer program, and quadratic program. To solving an integer program, CPlex uses the branch and bound algorithm plus many other

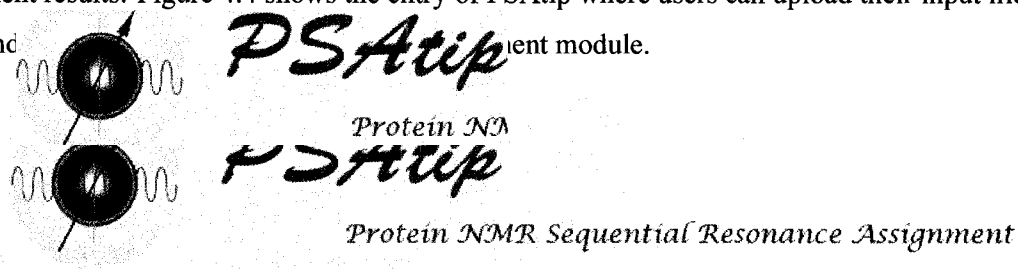
[2]http://bioinf.cs.ucl.ac.uk/psipred/

[3]A product of ILOG inc.. http://www.ilog.com/products/cplex/

techniques, such as cuts, heuristics, and a variety of branching and node selection strategies. A component library is provided to allow developers to directly integrate the power of CPlex into their applications.

4.1.3 Web interface

The web interface allows users to submit their assignment request and displays the assignment results. Figure 4.4 shows the entry of PSAtip where users can upload their input files and



Welcome to PSAtip, version 1.1, released on April 27, 2006.

1. Required Input

a. Protein sequence file	[+ Add] [View/modify] [delete]	absent
b. Secondary structure file	[+ Add] [View/modify] [delete]	absent
c. Spin systems file	[+ Add] [View/modify] [delete]	absent

2. Optional Input

a. Adjacency information file	[+ Add] [View/modify] [delete]	absent

3. Scoring Scheme Options

a. + Prediction Method	◉ Naive Bayesian*	○ Normal Distribution
b. + Secondary Structure Probability	◉ Probabilistic*	○ Non_probabilistic

Submit | [Start Over]

Figure 4.4: Entry to PSAtip

Users are required to provide at least three files for each instance (a protein sequence file, a secondary structure file, and a spin system file). Users could also provide an optional file with adjacency connectivity information to help get a better assignment. Besides the upload function which allows users to upload their input files, the web interface also offers a view/edit function to allow users to make sure that the uploaded files contains the correct information before assignment request are submitted, and a remove function to allow users to delete files that is no longer in use, e.g., the optional file with adjacency information.

After all required files are uploaded, the user can choose a scoring scheme (option 3.a in Figure 4.4) for calculating the confidence of each mapping of spin system and residue in the string assignment. If there is probability information in the secondary structure information file, the user can also specify a particular method to handle the probability information

(option 3.b in Figure 4.4).

For example, when the secondary structure information is predicted by PSIPRED, in some cases, PSIPRED can not provide the certain type of secondary structure for a residue and it gives the probability for each type of secondary structure. We have two methods to handle the cases that the probability information occurs in the secondary structure information file. We can either simply ignore the probability and only use the type of secondary structure with highest probability, which is the "Non_probabilistic" option in option 3.b, or use the probabilities when calculating the confidences of mappings, the "Probabilistic" option. Without loss generality, assume the probability that the type of the secondary structure of a residue r is α-helix, β-sheet, or coil is $p_r(\alpha)$, $p_r(\beta)$, or $p_r(c)$, respectively. The score of mapping a spin system s to a residue r

$$Score(s,r) = \sum_{t \in T} p_r(t) Score(s,r,t), \tag{4.1}$$

where T is the set of three secondary structure types and $Score(s,r,t)$ is the score of mapping spin system s to residue r with secondary structure type t. If the "Non_probabilistic" option is chosen and for a residue r, the type of secondary structure with highest probability is t_r, then $p_r(t) = 1$ if $t = t_r$ and $p_r(t) = 0$ otherwise.

After uploading all input data, the user can click the "submit" button to generate an assignment request which will be added to the request queue. When the request is solved by the string assignment module, a display function will be called to generate an HTML (HyperText Markup Language) file which contains the assignment result as well as the input information. Displaying both the input information and the assignment result at the same time helps the user to gain insights from the assignment result conveniently, instead of looking back and forth between the result and input files. Figure 4.5 shows an instance of assignment result displayed in a web browser.

4.2 Mapping Confidence

Due to the reading error on chemical shift values, spectral data are corrupted by noise. Consequently, the performance of string assignment process will be affected and the result becomes sensitive to the noise. To overcome this issue and test the robustness of the assignment, we add normal distributed independent noise[4] to the original chemical shift values, run the string assignment process many times and each time a different set of noises are

[4] in practice, a reasonable choice of noise is normal distributed noise.

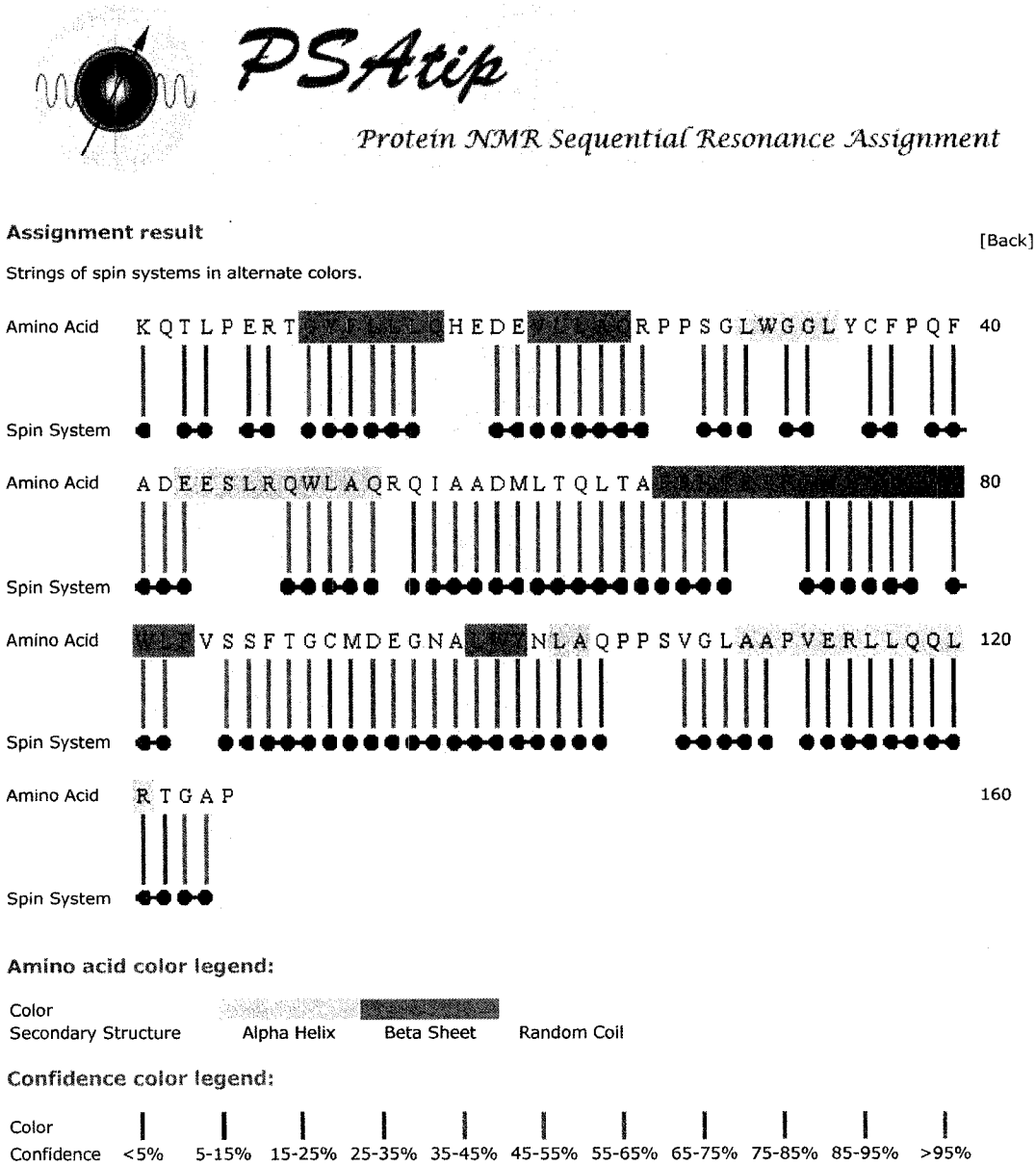


Figure 4.5: Assignment result. The protein sequence is colored according to its secondary structure. The color of lines between amino acid and spin system shows the confidence level of these mappings. Adjacency information is displayed as connected dots with same color. The chemical shift values for each spin system shows in a pop-up box when the mouse cursor is moved on one spin system.

added to the chemical shift values. After all assignments are done, the confidence of each mapping can be calculated (Equation 4.2).

$$Confidence(s_i, r_j) = \frac{\text{Number Of } < s_i, r_j > \text{ Mappings}}{\text{Total Trials}} \tag{4.2}$$

For example, spin system s_i is mapped to residue r_j in the initial assignment where there is no additional noises added, and we repeat the assignment 99 times with additional noises. In all 100 assignments, if spin system s_i is mapped to the same residue r_j 95 times, we say that this mapping has a confidence level 95%.

The information of confidence level could be very important to biologist because it tells them which mappings are more believable than others.

Chapter 5

Experiments

We developed a tool called scoring scheme evaluator which uses the same integer program in PSAtip to investigate the effectiveness of our IP approach, to compare the different scoring schemes for string assignment, and to study the impact of protein secondary structure information on scoring schemes. In our experiments, we used a standard dataset that is built from BMRB[1] (Biological Magnetic Resonance Bank) and PDB[2] (Protein Data Bank).

5.1 Standard Dataset

The dataset used for comparing scoring schemes is gathered from BMRB and PDB. In the dataset, each instance contains three kinds of information—the protein sequence, the corresponding secondary structure, and the spin systems information including chemical shift values and correct mapping positions. Each instance was chosen by applying the following criteria:

- the same protein ID is shared in both BMRB and PDB;
- the sequence similarity of protein sequences obtained from BMRB and PDB is greater than 90 %.
- the sequence length of the protein is larger than 50;
- the spin systems information contains at least H, N, C_α, C_β chemical shifts;
- the spin systems information contains no more than 50% missing chemical shifts;

In the end, 478 instances that satisfy the criteria are remained and among all instances, the longest protein sequence contains 731 residues. For all 478 instances, a pair-wise alignment on the protein sequences is applied. If the sequence similarity of two proteins is greater than 50%, one of the corresponding instance is put into the standard dataset. At the end, the standard dataset contains 161 instances and the longest protein sequence in the standard dataset contains 370 residues.

5.2 Scoring Scheme Evaluator

Scoring scheme evaluator is developed to compare the performance of different scoring schemes for string assignment because the scoring scheme plays an important role in the

[1] http://www.bmrb.wisc.edu/
[2] http://www.rcsb.org/pdb/

task of protein NMR sequential resonance assignment. It is very useful if we can evaluate the performance of a scoring scheme because a good scoring scheme could increase the accuracy of string assignment, which indirectly increases the quality of protein three-dimensional structure through NMR spectroscopy.

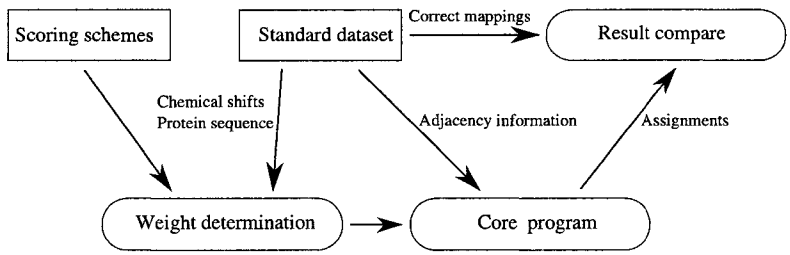


Figure 5.1: Overview of scoring scheme evaluator

Figure 5.1 shows an overview of scoring scheme evaluator. The standard dataset is used to provide required information for other functions in this tool. With a specified scoring scheme, assignments are done for all instances and the results is compared to the correct mappings to produce the overall performance for this scoring scheme. Several scoring schemes have been integrated into this tool and they can be used directly in it. If the user has a new scoring scheme to test, an API (Application Programming Interface) is provided so that the user can send the weight matrix directly to the core program. After the assignments are done, the result comparison function compares the assignments with the correct mappings in the dataset to produce the performance of the specified scoring scheme. The performance is evaluated by average precision and average recall on the standard dataset. For each assignment, precision indicates the percentage of correctly assigned amino acids over total assigned amino acids (Equation 5.1), and recall indicates the percentage of correctly assigned amino acids over amino acids with known answers (Equation 5.2):

$$Precision = \frac{\text{number of correctly assigned amino acids}}{\text{number of assigned amino acids}} \times 100\%, \quad (5.1)$$

$$Recall = \frac{\text{number of correctly assigned amino acids}}{\text{number of amino acids with known answers}} \times 100\%. \quad (5.2)$$

5.3 Effectiveness of IP Approach

We tested the string assignment over 478 instances and the experimental results have shown that the average running time for solving one instance is less than 1 second (including system I/O) on a hardware platform with a P4 3.2G CPU and 1GB memory. Among all instances, the most complex instance is a protein sequence that contains 731 residues and 90%

spin systems are connected with each other. In this case, the total number of variables and the number of constraints for the integer program are around $500,000$ each. Even for the most complex instance, the running time is less than 5 seconds. This is a big improvement comparing to other approaches, such as MARS, RIBRA, and previous works of our group. To finish one string assignment problem, other approaches could take minutes, hours, days, even weeks. For example, on a relatively smaller dataset which contains 70 instances and in which the longest protein sequence contains only 215 residues, the branch-and-bound algorithm can only solve half instances with two-day time limit per instance and the IDA* algorithm leaves 5 instance unsolved with the same time limit per instance. When the time limit is extended to 7 days, the branch-and-bound algorithm still fails to solve some instances. On the other hand, our program solves all 478 instances in the whole dataset with average running time less than 1 second for one instance.

5.4 Scoring Scheme Comparison

To see how different scoring schemes affect the performance of string assignment, seven scoring schemes, TATAPro II, PACES, AutoAssign, MARS,CASA, Normal, and HBSS (Histogram-Based Scoring Scheme (Wan, 2006)), are compared on the standard dataset described on the above with defferent percentage of spin systems connectivity. Adjacency information set is randomly generated based on a given connectivity percentage ranged from 10% to 90% with interval of 10%. The secondary structure information is excluded in this comparison because not all scoring schemes use the secondary structure information.

TATAPro II

TATAPro II, a residue typing scheme to evaluate a possible mapping between a spin system and a residue, is used by many applications, for example RIBRA. If a spin system is likely to be mapped to a residue, "1" is assigned to this spin system/residue pair; otherwise, "0" is assigned. This value is calculated based on Table 5.1 (Atreya et al., 2002).

PACES

PACES (Coggins & Zhou, 2003) also uses a typing scoring scheme which is similar to TATAPro II but with different chemical shift ranges (Table 5.2).

AutoAssign

AutoAssign evaluates the spin system/residue mapping with a probability score which is calculated as follows (Zimmerman et al., 1997). Given chemical shift C^{α} and C^{β} of a

Carbon chemical shift	Amino acid
Absence of C^β	Gly
$14 < C^\beta < 24$	Ala
$56 < C^\beta < 67$	Ser
$24 < C^\beta < 36$ and $C^\alpha < 64$	Lys, Arg, Gln, Glu, His, Trp, Cys^{red}, Val and Met
$24 < C^\beta < 36$ and $C^\alpha \geq 64$	Val
$36 < C^\beta < 52$ and $C^\alpha < 64$	Asp, Asn, Phe, Tyr, Cys^{oxd}, Ile and Leu
$36 < C^\beta < 52$ and $C^\alpha \geq 64$	Ile
—	Pro
$C^\beta > 67$	Thr

Table 5.1: TATAPRO II residue typing scheme

spin system, the probability that the spin system is mapped to an amino acid type R,

$$p(R|C^\alpha, C^\beta) = p(C^\alpha, C\beta|R) \times \frac{P(R)}{\sum_R (p(C^\alpha, C^\beta|R)P(R))}, \tag{5.3}$$

where $p(C^\alpha, C^\beta|R)$ is the probability of observing chemical shift vales C^α and C^β for R, and $P(R)$ is the frequency of occurrence of R in the protein sequence.

MARS

In MARS, a Z-score is used to guide the string assignment process. The Z-score for mapping spin system s_i to residue r_j is defined as

$$S(i,j) = \sum_{k=1}^{N_{cs}} \{\frac{\delta(i)_k^{exp} - \delta(j)_k}{\sigma_k}\}^2, \tag{5.4}$$

where $\delta(i)_k^{exp}$ is the measured chemical shift value of type k of spin system s_i, $\delta(j)_k$ is the predicted chemical shift value of type k of residue r_j, N_{cs} is the number of chemical shift types and σ_k is the standard deviation of the statistical chemical shift distribution used for calculating $\delta(j)_k$ (Jung & Zweckstetter, 2004). And $\delta(i)_k^{exp} - \delta(j)_k$ is set to zero when chemical shift of type k is missing.

CASA

CASA (Wang et al., 2005) uses the binary typing score to evaluate the mapping between chemical shift and residue. This typing score is calculated as

$$S(r,t) = \begin{cases} 1 & : \text{ if } |p - \overline{p}_t| \leq R_p \cdot \sigma_t \\ 0 & : \text{ otherwise} \end{cases}, \tag{5.5}$$

where $\overline{p}_t$ and σ_t are the mean and standard deviation of this type of chemical shift for residue r in BMRB, and $R_p = 5$ for H atom and $R_p = 4$ for other atoms.

Normal

Amino Acid		C^{α}			C^{β}			Carbonyl	
		Min	Max	^{2}H Adj.	Min	Max	^{2}H Adj.	Min	Max
A	Ala	48	57	-0.68	14	24	-1.00	171	183
C	Cys	49	66	-0.55	23	51	-0.71	166	180
D	Asp	49	59	-0.55	36	45	-0.71	170	180
E	Glu	50	62	-0.69	25	36	-0.97	170	181
F	Phe	50	65	-0.55	34	45	-0.71	170	180
G	Gly	41	49	-0.39	41	49	-0.39	167	180
H	His	49	62	-0.55	23	37	-0.71	169	180
I	Ile	53	67	-0.77	33	44	-1.28	169	181
K	Lys	50	62	-0.69	26	39	-1.11	170	182
L	Leu	49	61	-0.62	37	48	-1.26	170	181
M	Met	48	62	-0.69	25	41	-0.97	169	182
N	Asn	47	59	-0.55	34	44	-0.71	170	180
P	Pro	59	67	-0.69	26	36	-1.11	171	181
Q	Gln	50	61	-0.69	22	36	-0.97	170	180
R	Arg	49	63	-0.69	25	36	-1.11	170	181
S	Ser	51	64	-0.55	59	69	-0.71	169	180
T	Thr	55	69	-0.63	65	74	-0.81	169	179
V	Val	55	69	-0.84	28	37	-1.20	169	180
W	Trp	51	64	-0.55	24	37	-0.71	170	181
Y	Tyr	51	64	-0.66	33	45	-0.71	169	180

Table 5.2: Chemical shift ranges in PACES

GARANT (Bartels et al., 1997) uses the "mutual information" as a measure of mapping the observed chemical shift ω_D^* to the expected chemical shift $\omega(\alpha_M)$ in a resonance assignment R. And the "mutual information"

$$I_R(\omega_D^*; \alpha_M) = \log \frac{p(\omega_D^*|\alpha_M)}{p(\omega_D^*)}, \tag{5.6}$$

where $p(\omega_D^*|\alpha_M) = \mu_{\sigma(a_M)}(\omega_D^* - \omega(a_M))$ is a uniform *a priori* probability of the observed chemical shift ω_D^* and a normal distribution with mean $\omega(\alpha_M)$ and standard deviation $\sigma(a_M)$, $p(\omega_D^*) = \Delta_\omega^{-1}$ where Δ_ω is the width of the range of possible chemical shifts, and $\mu_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-1/2(x/\sigma)^2}$ is the probability density of the normal distribution with zero mean and standard deviation σ.

HBSS

HBSS uses a Naïve Bayes method to determine the score of mapping spin system to an amino acid with the secondary structure, which is

$$score = \frac{1}{4} \sum_{cs \in \{H_i, N_i, C_i^{\alpha}, C_i^{\beta}\}} \log(p(cs|aa, ss)), \tag{5.7}$$

where cs is chemical shift value, aa is the amino acid type, ss is the corresponding secondary structure, and $p(cs|aa, ss) = \frac{N(cs|aa,ss)}{N(aa,ss)}$.

Figure 5.2 summarized the assignment precision and recall on each scoring scheme. For both chart (a) and (b) in Figure 5.2, the x-axis indicates the connectivity percentages and the y-axis indicates the average accuracy (precision or recall). And the accuracy of applying different scoring schemes for string assignment is marked by different colors.

When there is no adjacency information, the accuracy of assignment totally relies on the scoring scheme. We can see that HBSS and Normal-Density scoring schemes are much better than the others. With the increasing of adjacencies, the performance margin between different scoring schemes gets smaller. The reason for the catching up performance of other scoring schemes is that a string of spin systems is considered as whole during mapping process and the possible mapping positions are evaluated by all spin systems on the string. It is more accurate than only considering individual spin system. However, even with 90% of adjacent spin systems, in which case the adjacency information highly affects the feasible assignment solution space, HBSS and Normal still perform better than others.

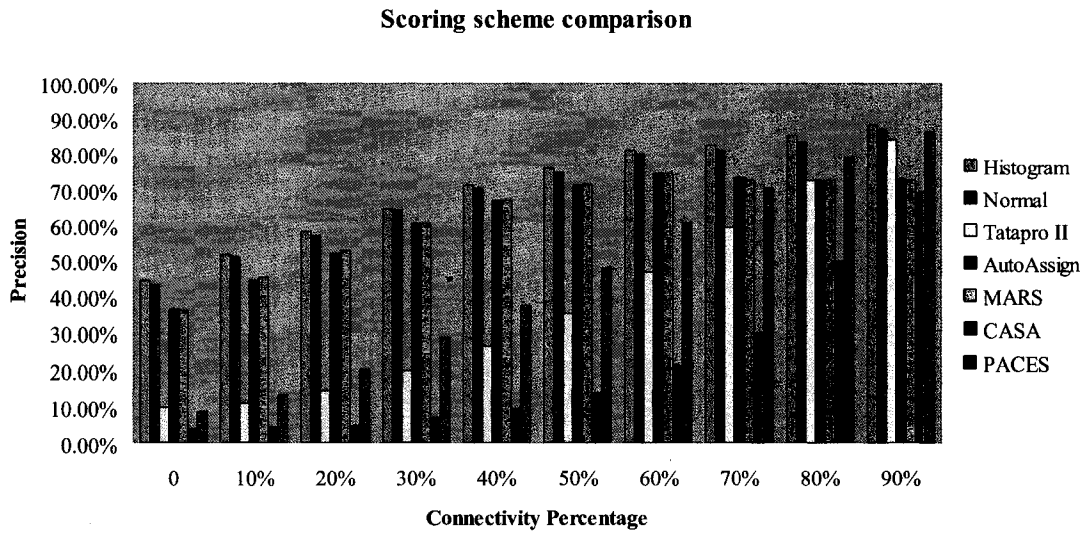
For HBSS and Normal-Density scoring schemes, their performances are very close to each other for the string assignment process. They both can perform very well and the Bayesian based HBSS outperforms Normal-Density scoring scheme when adjacency information is available. Therefore, our application, PSAtip, allows users to choose one of them for their assignment request.

5.5 Impact of Secondary Structure Information on NMR Resonance Assignment

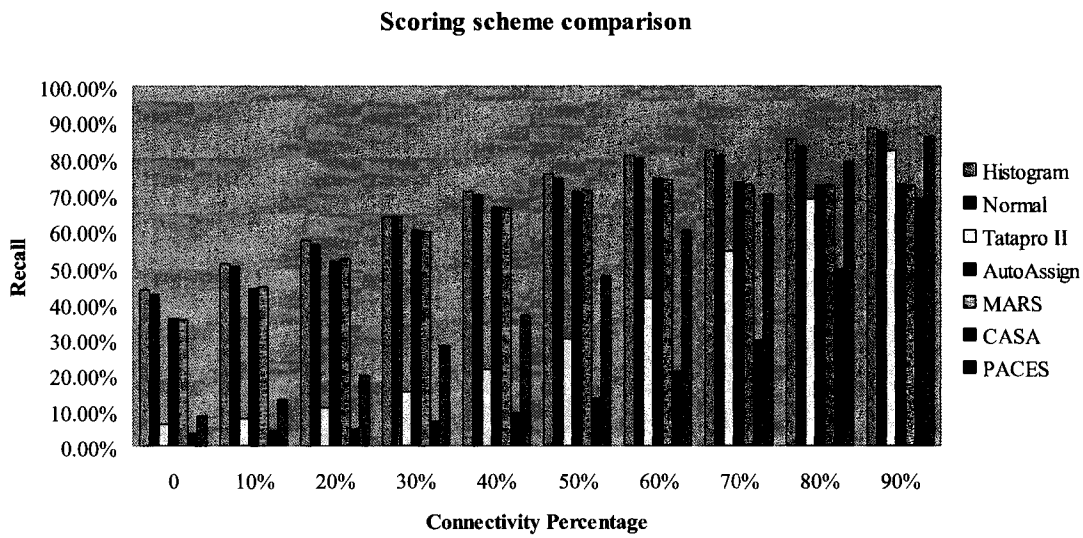
To study the impact of the secondary structure information of a protein for improving the performance of string assignment, We conducted the second group of experiments.

Two scoring schemes, HBSS and Normal, can make use of the secondary structure information. Figure 5.3 and Figure 5.4 summarized the difference of precision and recall on scoring scheme HBSS and Normal with or without using the secondary structure information, respectively. We can see that there is an average 5.59% performance gain when using the corresponding secondary structure in the string assignment. When connectivities are few, in which case the assignment is mainly determined by the chemical shifts, we can get above 10% improvement.

The secondary structure of a protein contains important information about how the protein folds in the three-dimensional space. The type of the secondary structure of a residue



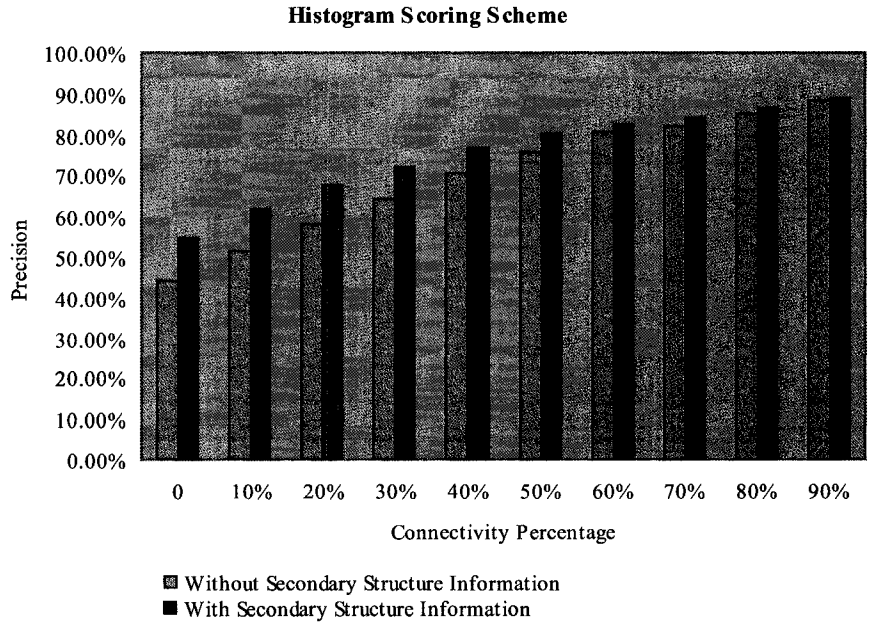
(a)



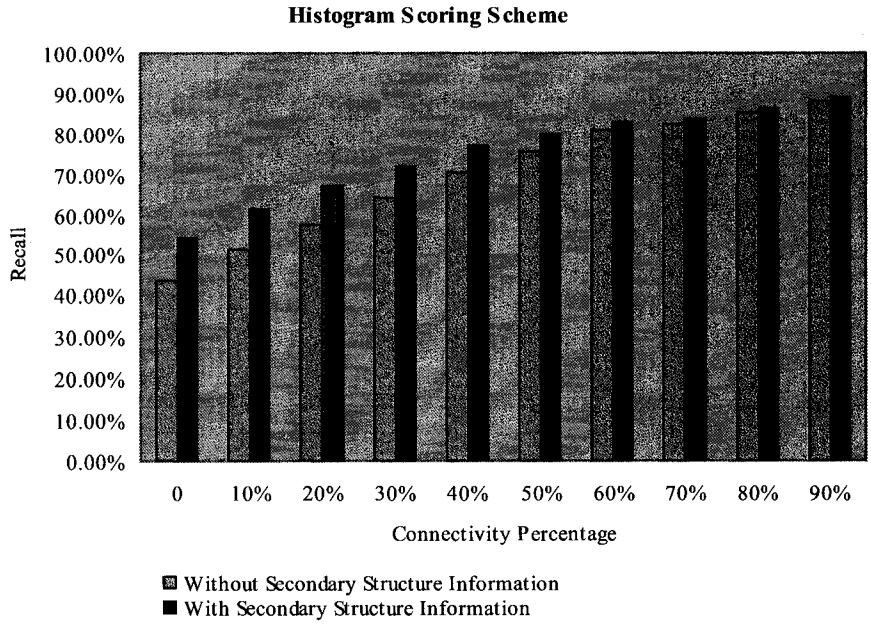
(b)

Figure 5.2: Scoring schemes comparison. (a) average precisions of seven scoring schemes upon ten different connectivity percentage settings; (b) average recall of seven scoring schemes upon ten different connectivity percentage settings.

has an influence to the observed chemical shifts from NMR spectroscopy. If we can use the secondary structure information to adjust the observed chemical shifts to neutralize this influence, we can get a better assignment. Therefore, PSAtip always requires users to provide the secondary structure information for the string assignment. Fortunately, third party tools such as PSIPRED can provide the prediction of the secondary structure for a protein if the secondary structure information is not available yet.

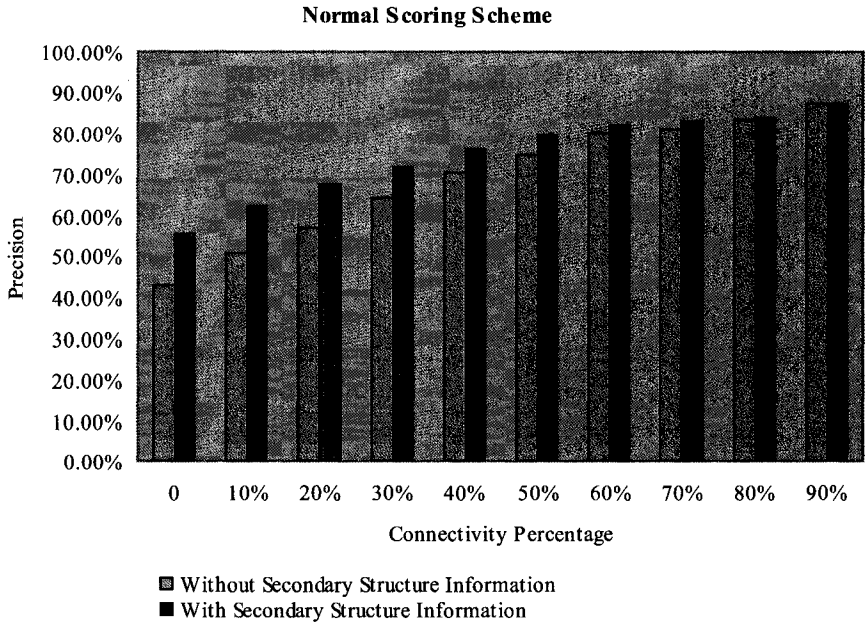


(a)

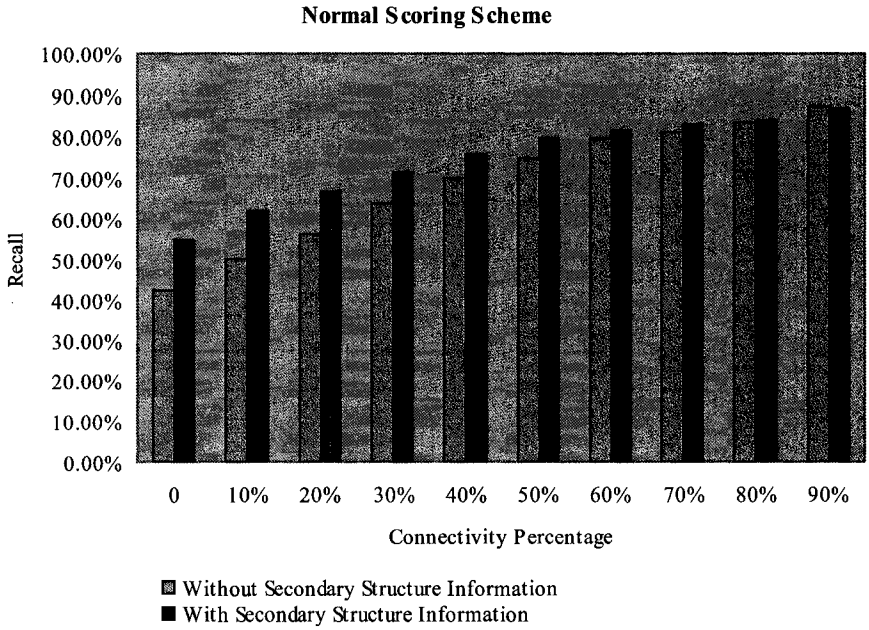


(b)

Figure 5.3: Comparison of two different scoring schemes with or without secondary structure information. (a) average precision for Histogram Based Scoring Scheme with or without secondary structure information, (b) average recall for Histogram Based Scoring Scheme with or without secondary structure information.



(a)



(b)

Figure 5.4: Comparison of two different scoring schemes with or without secondary structure information. (a) average precision for Normal Scoring Scheme with or without secondary structure information, (b) average recall for Normal Scoring Scheme with or without secondary structure information.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

String assignment is one of the key steps in the protein NMR sequential resonance assignment. Improving the performance of string assignment not only can provide high quality assignment results for the next step of protein three dimensional structure determination task, but also can be used to evaluate the methods for peak grouping and connectivity determination tasks. Many methods failed when protein sequences were long, the chemical shift data quality was low, or connectivity was complicated. In this thesis, we developed a new approach using integer programming for string assignment. Our idea could reduce the time of solving a string assignment from minutes/days to seconds. Even for a protein with 731 amino acids in which case the total number of variables and the number of constraints for the integer programming are around $500,000$ respectively, the string assignment can be solved within a few seconds. Some methods can only handle binary weights for mappings from spin systems to residues, i.e., either there is a mapping between a spin system and a residue or there is no such mapping existing. However, when the chemical shift data quality is low, it is hard to determine if a spin system can be mapped to an amino acid or not. In this case, using a real number to indicate the likelihood of such mapping existing is more suitable. Integer programming approach can easily handle this situation and flexibly adopt scoring schemes with binary value or real numbers. What's more, the amount of connections between spin systems does not affect the speed very much. Therefore, the integer programming approach for string assignment is suitable for scenarios where a high throughput application is required, such as web services, scoring scheme testing, and protein three dimensional structure determination on a high volume of data.

Not only our program runs in a fully automated fashion, it also has a great advantage in interactive scenarios because of its high speed. It usually takes other approaches a long time to solve one NMR resonance assignment and makes them very difficult to interact with human. Biologists have to wait hours, even days to get the result for one single assignment. It may take weeks for biologists to get the final assignment if they want to change some input values and repeat the assignment process after they verified the previous assignment manually. With the effectiveness of IP program, the required time can be greatly shorten. This not only saves the time, but also gives biologist more flexibility to try different adjustments during an assignment.

6.2 Future Work

The integer programming approach can handle general connectivities; however, it still has its limitations. Currently, a step of verifying acyclic connections is required. Fortunately, cyclic connections occur rarely in practice, but when it happens, breaking a circle at a wrong place may decease the assignment accuracy. A good strategy for handling cyclic connections is desired.

Although the string assignment requires the output from the other two processes, peaking grouping and connectivity determination, the assignment result can be used to refine the output of the other two processes (Wan, 2006). With the high performance integer programming approach, it is possible to make these three processes run iteratively to further improve the accuracy of final assignment. An analysis of the assignment result and input data could be made to provide the guiding information for the tasks of peaking grouping and connectivity determination.

Bibliography

Atreya, H. S., Chary, K. V. R., & Govil, G. (2002). Automated nmr assignments of proteins for high throughput structure determination: Tatapro ii. *Current Science, 83*, 1372–1376.

Bartels, C., Güntert, P., Billeter, M., & Wüthrich, K. (1997). Garant-a general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra. *Journal of Computational Chemistry, 18*, 139–149.

Chen, Z.-Z., Lin, G., Rizzi, R., Wen, J., Xu, D., Xu, Y., & Jiang, T. (Mar 2005). More reliable protein nmr peak assignment via improved 2-interval scheduling. *Journal of Computational Biology, 12*, 129–146.

Clote, P., & Backofen, R. (2000). *Computational molecular biology - an introduction.* John Wiley & Sons Ltd.

Coggins, B. E., & Zhou, P. (2003). Paces: Protein sequential assignment by computer-assisted exhaustive search. *Journal of Biomolecular NMR, 26*, 93–111.

Fox, M. S. (1986). Observations on the role of constraints in problem-solving. *The Sixth Canadian Proceedings in Artificial Intelligence.*

Güntert, P., Salzmann, M., Braun, D., & Wüthrich, K. (2000). Sequence-specific nmr assignment of proteins by global fragment mapping with the program mapper. *Journal of Biomolecular NMR, 18*, 129–137.

Hsu, W., Chang, J., Chou, W., Chen, J., Wu, K., Sung, T., Chang, C., Wu, W., & Huan, T. (2004). An iterative relaxation technique for the nmr backbone assignment problem. *Bioinformatics and Bioengineering, 2004. BIBE 2004. Proceedings. Fourth IEEE Symposium on* (pp. 89–90).

Jung, Y.-S., & Zweckstetter, M. (2004). Mars - robust automatic backbone assignment of proteins. *Journal of Biomolecular NMR, 30*, 11–23.

Kumar, V. (1992). Constraint satisfaction methods in artificial intelligence. *Artificial intelligence Magazine, Spring*, 32–44.

Mackworth, A. K. (1977). Consistency in networks of relations. *Artifici. Intell., 8*, 99–118.

Nadel, B. A. (1986). *The general consistent labeling (or constraint satisfaction) problem. technical report, dcs-tr-170* (Technical Report). Computer Science Department, Rutgers University.

Thornton, J. M., Todd, A. E., Milburn, D., Borkakoti, N., & Orengo, C. A. (2000). From structure to function: Approaches and limitations. *Nature Structural Biology, 7*, 991–994.

Wan, X. (2006). *Automated sequential resonance assignment in nmr protein structure determination.* Doctoral dissertation, University of Alberta.

Wang, J., Wang, T., Zuiderweg, E., & Crippen, G. (2005). Casa: An efficient automated assignment of protein mainchain nmr data using an ordered tree search algorithm. *Journal of Biomolecular NMR, 33*, 261–279.

Williamson, M. P., Havel, T. F., & Wüthrich, K. (1985). Solution conformation of proteinase inhibitor iia from bull seminal plasma by 1h nuclear magnetic resonance and distance geometry. *Journal of Molecular Biology, Volume 182, Issue 2*, Pages s295–315.

Wu, K.-P., Chang, J.-M., Chen, J.-B., Chang, C.-F., Wu, W.-J., Huang, T.-H., Sung, T.-Y., & Hsu, W.-L. (2006). Ribra - an error-tolerant algorithm for the nmr backbone assignment problem. *Journal of Computational Biology, 13*, 229–244.

Xu, Y., Xu, D., Kai, D., Olman, V., Razumovskaya, J., & Jiang, T. (2002). Automated assignment of backbone nmr peaks using constrained bipartite matching. *IEEE Computing in Science and Engineering, 4*, 50–62.

Zimmerman, D. E., Kulikowski, C. A., Huang, Y., Feng, W., Tashiro, M., Shimotakahara, S., Chien, C.-y., Powers, R., & Montelione, G. T. (1997). Automated analysis of protein nmr assignments using methods from artificial intelligence. *Journal of Molecular Biology, 269*, 592–610.