Integer Programming Model for Automated Structure-based NMR Assignment

Richard Jang*, Xin Gao*, and Ming Li**

David R. Cheriton School of Computer Science, University of Waterloo,
Waterloo, Ontario, Canada N2L 6P7

Technical Report CS-2009-32

Abstract. We introduce the "Automated Structure-based Assignment" problem: Given a reference 3D structure, a protein sequence, and its NMR spectra, automatically interpret the NMR spectra and do backbone resonance assignment. We then propose a solution to solve this problem. The core of the solution is a novel integer linear programming model, which is a general framework for many versions of the structure-based assignment problem. As a proof of concept, our system has generated an automatic assignment on a real protein TM1112 with 91% recall and 99% precision, starting from scratch. When we restrict ourselves to the special case where perfect peak lists are given, we are able to compare our results with existing results in the field. In particular, we reduced the assignment error of Xiong-Pandurangan-Bailey-Kellogg's method by 5 folds on average, with over a thousand fold speed up. Our system also achieves 91% assignment accuracy on real experimental data for Ubiquitin. These results have direct practical implications. For example, in the protein design process, a protein is modified slightly and its structure is again measured by NMR experiments. Our method automates this process, saving time on tedious peak-picking and resonance assignment. As another example, when there is a homologous protein with known structure, our method increases the assignment accuracy and hence enables automated NMR structure determination.

* The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

** All correspondence should be addressed to mli@uwaterloo.ca.

1 Introduction

The NMR resonance assignment problem has been extensively studied for twenty years [1–19]. Traditional resonance assignment methods depend mainly on the primary amino acid sequence and carbon connectivity information extracted from triple resonance experiments. These methods are referred to as sequence-based assignment methods [1–9]. Although the number of known protein structures is increasing rapidly, the number of new protein folds being discovered is slowing down [20, 21]. Thus, it can be expected that most proteins have homologs in the Protein Data Bank (PDB). Homologs typically have similar 3D structure, which can provide information to guide assignment. The known structure is used as a template, where the contact patterns from its atoms are compared to the experimental evidence to accelerate assignment on the target. Therefore, several structure-based assignment methods have been propose in the past decade [10,11,17–19,22–29].

The Nuclear Vector Replacement (NVR) approach [13, 26] used ^{15}N-HSQC spectra, residual dipolar couplings (RDC), sparse d_{NN} NOEs, amide exchange rates, and no triple resonance data for assignment. The problem was cast as a maximum bipartite matching problem, which they solved in polynomial time. Using close structural templates, they achieved an accuracy of over 99%. Their work was extended to handle more distant templates using normal mode analysis to obtain an ensemble of template structures [18]. However, in NMR labs, RDC experiments are not as commonly used as 3D NOESY experiments. For assignment using 3D NOESY data, Xiong *et al.* developed a branch-and-bound algorithm [28], which they later improved to a randomized algorithm [17], which shall be referred to as the contact replacement (CR) method. The CR method was demonstrated to tolerate 1-2Å structural variation, 250-600% noise, and 10-40% missing contact edges. It achieved an assignment accuracy of above 80% in α-helices, 70% in β-sheets, and 60% in loops. To our knowledge, it is the most error tolerant structure-based assignment method in terms of the noise level. The data consisted of 2D ^{15}N-HSQC, 3D ^{15}N TOCSY-HSQC, 3D ^{15}N NOESY-HSQC, and $^{3}J_{HNH\alpha}$ coupling constants derived from 3D HNHA. The problem was cast as a subgraph matching problem, where one graph consisted of the contacts in the known protein structure, and the other consisted of the NOESY cross peaks (NOEs) that connected spin system pairs. In general, the mapping of NOESY peaks to specific contacts is ambiguous due to experimental errors, missing peaks, and false peaks. Although the graph problem that was solved was NP-hard, Xiong *et al.* proved that under their noise model, the problem could be solved in polynomial time with high probability. In [19], they cast the problem similarly to [17], however, their method required unambigious NOEs from 4D NOESY experiments.

In NMR studies, NMR spectra are often examined by visual inspection, where the cross peaks get picked by inspection, or by automatic methods but then checked by the scientist. The peaks get accumulated in a list of peaks, and this list can change during the study as errors and inconsistencies are discovered. For the purpose of exposition, we define *manual peak lists* as peaks obtained by such a process. Until now, all of the previously proposed structure-based assignment methods require such peak lists. For example, the inputs for the CR method require manually picked peaks from four NMR spectra (^{15}N-HSQC, ^{15}N-edited NOESY, TOCSY, and HNHA) [17]; and the inputs for Langmead *et al.*'s method require manually picked peaks from ^{15}N-HSQC , and distance and orientation restraints derived from manually picked peaks from RDC, ^{15}N NOESY-HSQC and $H-D$ exchange *HSQC* [26]. Therefore, the structure-based assignment problem can be formally defined as follows:

Structure-Based Assignment Problem (SBA): Given homologous structure(s) of a target protein, and manually picked peak lists of the necessary spectra, assign the chemical shifts to the backbone atoms of the target protein.

In NMR labs, the peak picking step and the resonance assignment step are usually combined into an integrative process, so when given homologous structure(s), it does not take much extra work for NMR spectroscopists to do the assignment than to solely pick the peaks. Thus, it is difficult for the NMR community to benefit from the SBA methods that assume the availability of manual peak lists. In applications involving multiple NMR assignments, such as protein-ligand binding, time is much more important than cost. A fast

system that is able to take various types of input spectra is needed. Therefore, in this paper, we propose the Automated Structure-Based Assignment Problem (ASBA), which is formally defined as follows:

Automated Structure-Based Assignment Problem (ASBA): Given homologous structure(s) of a target protein, and *the necessary NMR spectra* only, assign the chemical shifts to the backbone atoms of the target protein.

We propose a system to solve the ASBA problem, the core of which is a novel and general integer linear programming (ILP) model, with combinations of an automatic peak picking method, PICKY [30], and an automatic triple resonance experiments-based assignment method, IPASS [9]. Both PICKY and IPASS were recently developed in our lab as automatic NMR tools. Our system first calls PICKY to conduct automatic peak picking. However, PICKY peak lists are still far from perfect, which makes the assignment problem difficult. When the necessary spectra are available, our system directly performs assignment with the ILP model. Otherwise, it first calls IPASS to fix an initial assignment, and then the ILP model to improve it. Nevertheless, the entire process is fully automatic.

We then show that the ILP model can take various NMR data as input, such as ^{15}N-labelled and ^{13}C-labelled (if available) peaks generated by PICKY or other peak picking methods, and can optimize various types of objectives that compare spin systems to residues and pairs of spin systems to pairs of residues. We further develop an iterative process that fixes reliable assignment fragments and calls the ILP model. To the best of our knowledge, this is the first attempt on fully automated structure-based assignment that starts directly from the original NMR spectra. The proposed method is tested on the real protein TM1112. With the noise level at 1200% at 4 Å distance cutoff for contacts, and without the necessary spectra for amino acid typing and secondary structure typing, our system achieves 91% recall and 99% precision[1]. Finally, we show that our ILP model can naturally be applied to solve the SBA problem. We first tested the ILP model on 9 proteins from the data set used by the CR method [17]. Our method, on average, has 5 times fewer incorrect assignments than the CR method, and is over a 1000 times faster. We then tested the ILP model on real experimental data for Ubiquitin, where we achieved an assignment accuracy of 91%.

2 Method

The main difference between an SBA method and an ASBA method is that the ASBA method must be able to do the assignment on imperfect peak lists generated by automatic peak picking methods. In our system, we used PICKY to do the peak picking. PICKY was tested in [30] on 32 raw spectra and achieved an average recall and precision of 88% and 74%, respectively. In this section, we propose a general integer linear programming model for the ASBA problem. The general framework is described first and then implementation and experimental details are described later.

2.1 General ILP Model for the ASBA Problem

We use the graph representation from the SBA CR method [17] to present the ASBA problem after the peaks are picked by PICKY. However, our ILP is general in that it does not depend on the specific scoring function or the type of NMR data that the CR method used. Integer programming has been applied to the maximum clique problem [31], to which graph matching can be reduced [32]. Our formulation models finding the common edge subgraph that maximizes the score. The common subgraph can be disconnected, and vertices in either graph can be unassigned. Unlike subgraph isomorphism, we match the presence of edges, but not the absence since the interaction graph tends to have many more edges than the contact graph. Figure 1 illustrates the graph problem.

[1] Let C be the number of correct assignments, R be the number of assignable residues, and S the number of residue assignments made by the method. Then Recall is C\ R and Precision is C\ S. By accuracy, we mean precision.

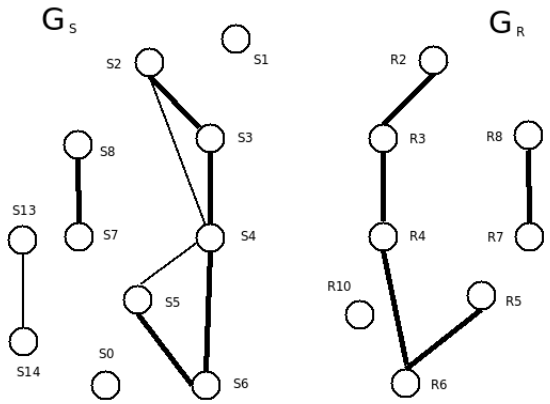

Fig. 1. The maximum common edge subgraph between graphs G_s and G_R. The vertices that are mapped have the same number. Edges that are matched are highlighted.

Definitions

Contact Graph: Each residue in the template protein is represented by a vertex labelled with residue-related features, such as the amino acid type and the secondary structure type. An edge is created between a pair of amino acids if there is a contact according to a given distance threshold.

Interaction Graph: Each spin system, which consists of the backbone N, H^N, H^α, and side chain chemical shifts of a pseudo-residue, is represented by a vertex that is labelled with the same class of features as the residues in the contact graph. An edge is created between a pair of spin systems if there is NMR evidence that support that they are close in Euclidean space, such as NOESY cross peaks or carbon connectivity from ^{13}C-labelled triple resonance experiments. A match score can be computed for each edge based on the type of data used.

Define V_{cg}, V_{ig} to be the set of vertices in the contact graph and interaction graph, respectively. Define E_{cg}, E_{ig} to be the set of edges in the contact and interaction graph, respectively.

Input Data

$m(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)})$ The edge match score between spin systems k and l and residues i and j. The score is assumed to be non-negative.

$m(v_{cg(i)}v_{ig(k)})$ The type match score between amino acid i and spin system k. The score is assumed to be non-negative

$E_{ig}(v_{cg(i),}v_{cg(j)})$ The set of edges in the interaction graph that are *compatible* with edge $v_{cg(i)}v_{cg(j)} \in E_{cg}$. An edge $v_{ig(k)}v_{ig(l)} \in E_{ig}$ is compatible with $v_{cg(i)}v_{cg(j)}$ if their features match, such as the vertices match on amino acid type, and the pair of vertices from one graph and the pair from the other graph match on contact type.

C_{cg} The set of all vertices $v_{cg(i)}$ in the contact graph such that there exist at least one edge $v_{cg(i)}v_{cg(j)}$ that is compatible with at least one edge in the interaction graph.

C_{ig} The set of all vertices $v_{ig(k)}$ in the interaction graph such that there exists at least one edge $v_{ig(k)}v_{ig(l)}$ that is compatible with at least one edge in the contact graph.

Decision Variables

$X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)})$ A binary variable. It equals to 1 if spin system $v_{ig(k)}$ is assigned to amino acid $v_{cg(i)}$, and spin system $v_{ig(l)}$ is assigned to amino acid $v_{cg(j)}$; and 0 otherwise. This variable represents an edge match between the graphs.

$X(v_{cg(i)}v_{ig(k)})$ A binary variable. It equals to 1 if spin system $v_{ig(k)}$ is assigned to the amino acid $v_{cg(i)}$; and 0 otherwise. This variable represents a vertex match.

Formulation

$$\max_{X} \begin{pmatrix} \sum_{v_{cg(i)} \in C_{cg},\ v_{ig(k)} \in C_{ig}} m(v_{cg(i)}v_{ig(k)})X(v_{cg(i)}v_{ig(k)}) + \\ \sum_{v_{cg(i)}v_{cg(j)} \in E_{cg}} \sum_{\substack{v_{ig(k)}v_{ig(l)} \in \\ E_{ig}(v_{cg(i)}v_{cg(j)})}} m(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)})X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) \end{pmatrix} \tag{1}$$

subject to

$$\sum_{v_{ig(k)}} X(v_{cg(i)}v_{ig(k)}) \leq 1 \quad \forall v_{cg(i)} \in V_{cg} \tag{2}$$

$$\sum_{v_{cg(i)}} X(v_{cg(i)}v_{ig(k)}) \leq 1 \quad \forall v_{ig(k)} \in V_{ig} \tag{3}$$

$$\sum_{\substack{v_{ig(l)}\ s.t. \\ v_{ig(k)}v_{ig(l)} \in E_{ig}(v_{cg(i),}v_{cg(j)})}} X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) \leq X(v_{cg(i)}v_{ig(k)}) \quad \forall v_{cg(i)}v_{cg(j)} \in E_{cg},\ \forall v_{ig(k)} \in C_{ig} \tag{4}$$

$$X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) \in \{0,1\} \tag{5}$$

$$X(v_{cg(i)}v_{ig(k)}) \in \{0,1\} \tag{6}$$

Discussion Equation 1, the objective function, expresses the total edge and type match score of the assignment. The first summation is over all vertices that are involved in at least one edge match. The second summation is over all edges that match. We generate only the variables involved in at least one edge match. We do not assign vertices that are isolated, unless the vertices can be unambiguously assigned, such as being the only ones with a particular type. Constraint 2 ensures that each amino acid is assigned to at most one spin system. Constraint 3 ensures that each spin system is assigned to at most one amino acid. Constraint 4 ensures if $X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) = 1$, then $X(v_{cg(i)}v_{ig(k)}) = 1$ and $X(v_{cg(j)}v_{ig(l)}) = 1$. Since we allow missing edges, if $X(v_{cg(i)}v_{ig(k)}) = 1$ and $X(v_{cg(j)}v_{ig(l)}) = 1$, the left hand side of Constraint 4 can be zero. However, since edge match scores are always non-negative and we are maximizing the score; if there is a match, we will have some $X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) = 1$. The final two constraints ensure that the decision variables are binary.

2.2 Generalizing the ILP Model

Our ILP model embodies a general framework for many versions of the ASBA, the SBA, and closely-related problems.

Scoring Functions The ILP model can optimize various objectives that can be expressed as a sum of terms comparing spin systems and residues, such as matching amino acid and secondary structure type, or comparing chemical shifts from their predicted values; and terms comparing pairs of spin systems to pairs of residues, such as matching on the type of connectivity or contact. Weights can be added to edge match variables that correspond to long range beta sheet contacts and local H^{α} and H^{N}contacts in alpha helices, so that greater weight, or different weights can be given to well-defined secondary structure regions in the template protein

Fixing Assignment Fragments By fixing specific vertex match variables to 1, we can fix assignments between particular amino acids and spin systems. This allows resonance assignment to be performed given a partial or an *apriori* assignment.

Hamiltonian Path Version The CR method focused on finding common Hamiltonian path fragments in the graphs to be matched. We can incorporate this by adding the constraint

$$\sum_{\substack{v_{cg(i)}v_{cg(j)} \in \\ E_{cg}\, s.t.\, |i-j|=1}} \; \sum_{\substack{v_{ig(k)}v_{ig(l)} \in \\ E_{ig}(v_{cg(i)}, v_{cg(j)})}} X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) \geq n-m \tag{7}$$

where the first sum is over all vertices of adjacent amino acids. n is the number of amino acids minus one, and m is the maximum allowable number of missing edges along the path. Alternatively or additionally, a weighted version of the left-hand side of the above constraint could be added to the objective, so that we optimize a weighted version of the score and the Hamiltonian path length.

Assigning NOESY Cross Peaks The ILP model can be extended to NOE assignment to identify which NOESY peak corresponds to exactly which contact. Therefore, it is possible to perform resonance and NOE assignment simultaneously. To enforce that each NOESY peak corresponds to at most one interaction, for each NOESY peak p, we have

$$\sum_{v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}} X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}, p) \leq 1 \tag{8}$$

where we have defined a new binary variable $X\left(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}, p\right)$ corresponding to an edge match with an interaction that is explained by NOESY peak p. To tie this variable to the other variables, we have, $\forall X\left(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}\right)$,

$$X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) \leq \sum_{p} X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}, p) \tag{9}$$

$$|p| \cdot X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}) \geq \sum_{p} X(v_{cg(i)}v_{ig(k)}v_{cg(j)}v_{ig(l)}, p) \tag{10}$$

where $|p|$ is the number of interaction graph edges containing an interaction that can possibly be attributed to NOESY peak p. Constraint 9 ensures that if there is an edge match, the match is due to at least one NOESY peak. Constraint 10 ensures that if there are NOESY peaks explaining the match, the corresponding edge variable will get selected.

Generating Multiple Assignments The sequential algorithm, introduced by Greisdorfer et al. [33] and generalized to more than two solutions in [34], can be used to generate solutions that are within a certain percentage of the optimal solution and have maximum diversity as measured by a diversity measure, such as average pairwise hamming distance. The one tree algorithm can also be used [34]. Multiple assignments can be generated for use with consensus methods.

3 Results

3.1 Implemention Details

For comparison with the previously reported SBA methods, in the contact graph, each vertex contains the amino acid type and the secondary structure type of an amino acid in the template protein. Each edge is

labelled by all pairs of directed proton-proton interaction types. We consider only two types of interactions, H^{α} and H^{N}, and H^{N} and H^{N}. Since H^{α} and H^{N} is not symmetric, the labels have the direction. In the interaction graph, each spin system is associated with possible amino acid types and secondary structure types. Amino acid type predictions can be obtained from the RESCUE software [35], and secondary structure type predictions can be obtained from $^{3}J_{HNH\alpha}$ coupling constants [36]. An edge is created between a pair of spin systems if there is a NOESY peak (^{15}N, ^{1}H, H^{N}), where the ^{15}N, H^{N}matches the backbone N, H^{N}chemical shift of one spin system and the ^{1}H matches the backbone H^{N}or H^{α} of the other spin system. Edges are labelled similarly to the contact graph with the addition of an edge match score for each NOE. The match score is defined as $erfc(\frac{|\Delta e|}{0.02\times\sqrt{2}})$ as used in [17], where $erfc$ is the complementary error function and $|\Delta e|$ is the chemical shift difference between ^{1}H and the matching H^{N}or H^{α}.

The input to RESCUE consists of the proton chemical shifts of each spin system including those from the side chain. This was the only time side chain proton chemical shift information was used; although in the future, we may consider using side chain contact information. Of the 10 possible amino classes returned by RESCUE, we found that using all classes with positive reliability score rather than the highest scoring class improved assignment accuracy, so we used all such classes.

To solve the ILP model, we used the solver in the commercial optimization package ILOG CPLEX® version 9.130.

3.2 Iterative Approach for Handling Typing Errors

We develop an iterative approach of the ILP model to recover from typing errors. Figure 2 gives a flowchart of the approach. We first solve the ILP model with the set of compatible edges constrained to vertices where both the amino acid and secondary structure types match. From this assignment, we identify individual reliable amino acid to spin system assignments to fix. We use contact information to determine reliability. We consider, for each amino acid, the percentage of its incident contact graph edges that are matched in the assignment. For the first criteria, at least 50% of the incident edges of each amino acid must be matched for its assignment to be fixed. The ILP model is then solved with these fixed assignments without the hard constraint that the amino acid and secondary structure types must match. The assignments for the non-fixed amino acids may have changed from the previous step, which may influence the fixed status of the currently fixed assignments, so we keep iterating and identifying new fixed assignments until the set of fixed assignments does not change or after a maximum number of iterations. Not shown in the diagram is a step where we try to further allow fixed assignments to change in order to escape local maxima. This is done by solving an integer program without fixed assignments, but restricting assignments to a $i\pm 4$ window about the current assignment for each amino acid i. After there is no change in the fixed assignments, we tighten the fixed assignment criteria, so that there exist edge matches to at least one sequential amino acid neighbor. This is later tightened to two sequential neighbors. For the final criteria, we use structural constraints, where for each alpha helical amino acid, there must be two local contacts ($i\pm 5$) that are matched in addition to the ones to the sequential neighbors, and for each beta sheet amino acid, there must be a matched contact to another amino acid in a different beta sheet. Finally, the highest scoring assignment is returned.

With no type matching constraints, the problem size would be too large without using fixed assignments. Moreover, using too strict a criteria early on for fixing assignments may result in a large problem size. Due to erroneous assignments, individual correct assignments may fail strict criteria, so we start with relaxed tolerances and then gradually tighten. This idea is analogous to the idea of decreasing the temperature gradually in the simulated annealing optimization protocol. Since we are solving multiple ILP models, we cannot guarantee the enumeration of the top N solutions. Nevertheless, we find that generating multiple solutions, improving each one, and then taking the best scoring one at the end produces a better final assignment. The generation of multiple solutions can be started at the initial ILP step or at subsequent ILP steps. In the latter case, previous assignments could be supplied to CPLEX as an initial feasible solution to speed up the optimization. Multiple solutions could also be generated from the final assignment by fixing assignments and then running the sequential or one tree algorithm. This allows the examination of the possible assignments

for the non-fixed amino acids. We applied the sequential algorithm described in [33] to obtain multiple solutions. The ILP model can also be used to generate the consensus assignment from a set of assignments by using it to solve the bipartite matching problem.

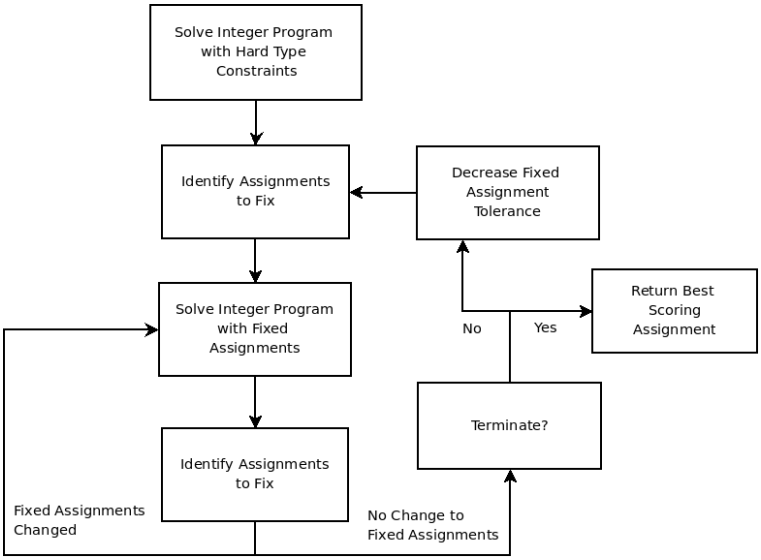

Fig. 2. Iterative Integer Programming with Fixed Assignments.

3.3 Probabilistic Scoring Model for Identifying Fixed Assignments

The criteria described in Section 3.2 is rather ad-hoc and does not take into account the frequency of amino acid, secondary structure, and interaction type. These effects mask the contact patterns, and therefore need to removed. Although the criteria we used worked, quantifying the reliability of each spin system to residue assignment is desirable. We propose a probabilistic scoring function that is analogous to the knowledge-based scoring functions used in protein folding [37] in that the score of a reference state is subtracted from the assignment score. Given an assignment $\overrightarrow{A}$ for a set of amino acids and spin systems, to score an individual assignment between amino acid aa_a and spin system ss_b, we consider two models, where one model is the reference state. In model 1, each assigned spin system is a true, but noisy representation of the atoms of the corresponding amino acid. Model 1 incorporates edge match score probabilities, estimated amino acid type prediction accuracies, estimated secondary structure type prediction accuracies, percentage of contacts missing in the NMR data, and the notise ratio. We call this model the residue-spin system correspondance (RSC) model. In model 2, the reference state model, the assignments happened by chance rather than having any special association. Pairs of vertices are selected from each graph at random and then assigned. Their corresponding edges are also assigned if there are interaction matches. If one graph has more vertices, then some vertices in that graph will get unassigned. Vertices in the graph with fewer vertices can be unassigned with a small user-defined probability ε_v. We call model 2, the background assignment (BA) model.

Notation

- Let $A(aa_a, ss_b)$ represent amino acid aa_a assigned to spin system ss_b. The direction of assignment does not matter. Denote $A(aa_a, nil)$ as aa_a is unassigned. Denote $A(nil, ss_b)$ as ss_b is unassigned.
- Let $E(aa_a)$ be the set of incident directed contact edges from aa_a, corresponding to the directed proton-proton interactions. Let $E(ss_b)$ be the set of incident directed NOE edges from ss_b.

- Let $V^-(aa_a)$ be the set of incident vertices of aa_a excluding aa_a itself. Let $V^-(ss_b)$ be the set of incident vertices of ss_b excluding ss_b itself.
- Let $A(V^-(aa_a))$ be the set of assignments for the vertices adjacent to aa_a.
- Let $A(E(aa_a))$ be the set of assignments for the edges incident to aa_a that are induced by the assignments $A(V^-(aa_a)) \cup A(aa_a, ss_b)$. An edge $e_{a,i,s}$ between aa_a and $aa_i \in V^-(aa_a)$ of type s is assigned to $e_{b,j,t}$ between ss_b and $ss_j \in V^-(ss_b)$ if aa_a is assigned to ss_b and aa_i is assigned to ss_j and s equals t. If there is more than one match of each interaction type, the edges are assigned in descending order of match score. An edge cannot be assigned more than once. Edges can be unassigned if there are no matches. Denote $A(e_{a,i,t})$ as the assignment for the edge of type t from aa_a to aa_i . Similarly for $A(e_{b,j,t})$ for ss_b to ss_j. Denote $A(e_{a,i,t}, e_{b,j,t})$ as the assignment between edge $e_{a,i,t}$ and $e_{b,j,t}$. Denote $A(e_{a,i,t}, nil)$, where $e_{a,i,t} \in E(aa_a)$, as being unassigned. Denote $A(nil, e_{b,j,t})$, where $e_{b,j,t} \in E(ss_b)$, as being unassigned.
- Let $A(star(aa_a))$ be $A(aa_a, ss_b) \cup A(V^-(aa_a)) \cup A(E(aa_a))$. Similarly for $A(star(ss_b))$.
- Let nr_d be the noise ratio equal to the number of directed NOE edges in the interaction graph divided by the number directed contact edges in the contact graph. Let nr_{ud} be the undirected noise ratio equal to the number of undirected NOE edges in the interaction graph divided by the number undirected contact edges in the contact graph, where each undirected edge consists of all the directed edges between a given pair of vertices in the corresponding graph. In general, both noise ratios are greater than 1.
- Let $\#aa$ be the number amino acids in the contact graph, and $\#ss$ be the number of spin systems in the interaction graph.
- Let $aaType(aa)$ be the amino acid type of aa. Let $ssType(aa)$ be the secondary structure type of aa.
- Let $aaClass(ss)$ be the predicted amino acid class for spin system ss, such as one of the ten mutally exclusive classes returned by the RESCUE software. Let $ssClass(ss)$ be the predicted secondary structure class for spin system ss. The classes are assumed to be mutually exclusive.
- Let $\#class(ss)$ be the number spin systems with the same amino acid and secondary structure class as spin system ss; analogously for $\#type(aa)$ for amino acid aa.
- Let $\#e_t^{ig}$ be the number of directed interaction graph edges with type t. Let $\#e_d^{ig}$ be the total number of directed interaction graph edges, and $\#e_u^{ig}$ be the total number of undirected edges. Similarly for $\#e_t^{cg}$, $\#e_d^{cg}$, and $\#e_u^{cg}$.
- Let $\#e_{a,i,t}^{>ig}$ be the number of undirected interaction graph edges containing fewer interactions of type t than the contact graph edge $e_{a,i}$. Let $\#e_{b,j,t}^{>cg}$ be the number of undirected contact graph edges containing fewer interactions of type t than the interaction graph edge $e_{b,j}$.

Scoring Function Given an assignment $\overrightarrow{A}$, for an assignment between amino acid aa_a and spin system ss_b, we compute $P(model\,|\,A(star(aa_a)), A(star(ss_b)))$ for each model. Assuming $P(model\,1) = P(model\,2)$,

$$P(model\,|\,A(star(aa_a)), A(star(ss_b)) = \frac{P(A(star(aa_a)), A(star(ss_b))\,|\,model)}{\sum_m P(A(star(aa_a)), A(star(ss_b))\,|\,model = m)} \tag{11}$$

$$P(model\,1\,|\,A(star(aa_a)), A(star(ss_b)) = \frac{R}{1+R} \tag{12}$$

where $R = \frac{P(A(star(aa_a)), A(star(ss_b))\,|\,model\,1)}{P(A(star(aa_a)), A(star(ss_b))\,|\,model\,2)}$

$$P(A(star(aa_a)), A(star(ss_b))\,|\,model) = P(A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), A(E(aa_a), A(E(ss_b)\,|\,model) \tag{13}$$

$$
\begin{aligned}
P(A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), A(E(aa_a)A(E(ss_b)\,|\,model) = \\
P(A(E(aa_a)), A(E(ss_b)\,|\,A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), model) \times \\
P(A(V^-(aa_a), A(V^-(ss_b)\,|\,A(aa_a, ss_b), model) \times \\
P(A(aa_a, ss_b)\,|\,model)
\end{aligned} \tag{14}
$$

For fixing assignments, we can rank each individual assignment by their score, which is taken to be $log\,(P(model\,1\,|\,A(star(aa_a)), A(star(ss_b)))$, and then fix the assignments that score above a specific threshold. This threshold can increase after each iteration. The parameters for each model is based on the following cases. The probabilities for each of these cases are given in Table 1. For brevity, the conditioned space of the probability is omitted if it is clear from the context that it is present.

For $P(A(aa_a, ss_b)) = P(aa_a) \times P(ss_b\,|\,aa_a)$, $P(aa_a)$ will be the same in both models, so we consider only $P(ss_b\,|\,aa_a)$.

For $aa_i \in V^-(aa_a)$, $ss_j \in V^-(ss_b)$, and $A(aa_i, ss_j)$

$$
P(A(E(aa_a)), A(E(ss_b)\,|\,A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), model) = \prod_a P(e_{aa,i,t}) \tag{15}
$$

$$
\begin{aligned}
P(A(E(aa_a)), A(E(ss_b)\,|\,A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), model) = \\
\times \prod_t P(A(e_{a,i,t}, e_{b,j,t})\,|\,A(aa_a, ss_b), A(aa_i, aa_j), model) \\
\times \prod_s P(A(e_{a,i,s}, nil)\,|\,A(aa_a, ss_b), A(aa_i, aa_j), model) \\
\times \prod_r P(A(nil, e_{b,j,r})\,|\,A(aa_a, ss_b), A(aa_i, aa_j), model)
\end{aligned} \tag{16}
$$

$$
\begin{aligned}
P(A(V^-(aa_a)), A(V^-(ss_b))\,|\,A(aa_a, ss_b), model) = \prod_{i,j} P(A(aa_i, ss_j)\,|\,A(aa_a, ss_b), model) \\
\times \prod_k P(A(aa_k, nil)\,|\,A(aa_a, ss_b), model) \\
\times \prod_l P(A(nil, ss_l)\,|\,A(aa_a, ss_b), model)
\end{aligned} \tag{17}
$$

For $P(A(aa_i, ss_j)\,|\,A(aa_a, ss_b), model) = P(aa_i) \times P(ss_j\,|\,aa_i)$, $P(aa_i)$ will be the same in both models, so we consider only $P(ss_j\,|\,aa_i)$.

For $P(A(e_{a,i,t}, e_{b,j,t})\,|\,A(aa_a, ss_b), A(aa_i, aa_j), model) = P(e_{a,i,t}) \times P(e_{b,j,t}\,|\,e_{a,i,t}), P(e_{a,i,t})$ will be the same in both models, so we consider only $P(e_{b,j,t}\,|\,e_{a,i,t})$.

For $P(A(e_{a,i,t}, nil)\,|\,A(aa_a, ss_b), A(aa_i, aa_j), model) = P(e_{a,i,t}) \times P(A(e_{a,i,t}, nil)\,|\,e_{a,i,t})$, $P(e_{a,i,t})$ will be the same in both models, so we consider only $P(A(e_{a,i,t}, nil)\,|\,e_{a,i,t})$.

For $P(A(nil, e_{b,j,t})\,|\,A(aa_a, ss_b), A(aa_i, aa_j), model) = P(e_{b,j,t}) \times P(A(nil, e_{b,j,t})\,|\,e_{b,j,t})$, $P(e_{b,j,t})$ will be the same in both models, so we consider only $P(A(nil, e_{b,j,t})\,|\,e_{b,j,t})$.

For $aa_i \in V^-(aa_a)$, $ss_j \notin V^-(ss_b)$, and $A(aa_i, ss_j)$; or $aa_i \in V^-(aa_a)$, $A(aa_i, nil)$, there are no edges incident to ss_b for matching the contact graph edge $e_{a,i}$, so we have

$$
P(A(V^-(aa_a)), A(V^-(ss_b))\,|\,A(aa_a, ss_b), model) = \prod_i P(A(aa_i, nil)\,|\,A(aa_a, ss_b), model) \tag{18}
$$

$$P(A(E(aa_a)), A(E(ss_b)\,|\,A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), model) = \prod_s P(A(e_{a,i,s}, nil)) = 1 \qquad (19)$$

For $P(A(aa_i, nil)\,|\,A(aa_a, ss_b), model) = P(aa_i) \times P(A(aa_i, nil)\,|\,aa_i)$, $P(aa_i)$ will be the same in both models, so we consider only $P(A(aa_i, nil)\,|\,aa_i)$.

For $aa_i \notin V^-(aa_a)$, $ss_j \in V^-(ss_b)$,and $A(aa_i, ss_j)$; or $ss_j \in V^-(ss_b)$, $A(nil, ss_j)$, there are no edges incident to aa_a for matching the interaction graph edge $e_{b,j}$, so we have

$$P(A(V^-(aa_a)), A(V^-(ss_b))\,|\,A(aa_a, ss_b), model) = \prod_j P(A(nil, ss_j)\,|\,A(aa_a, ss_b), model) \qquad (20)$$

$$P(A(E(aa_a)), A(E(ss_b)\,|\,A(aa_a, ss_b), A(V^-(aa_a), A(V^-(ss_b), model) = \prod_s P(A(nil, e_{b,j,s})) = 1 \qquad (21)$$

For $P(A(nil, ss_j)\,|\,A(aa_a, ss_b), model) = P(ss_j) \times P(A(nil, ss_j)\,|\,ss_j)$, $P(ss_j)$ will be the same in both models, so we consider only $P(A(nil, ss_j)\,|\,ss_j)$.

Table 1 gives the probabilities for each case in each model.

Table 1. Probabilistic scoring model for identifying reliable assignments.

Cases	Residue-Spin System Correspondance Model	Background Assignment Model
$P(ss_b\,\|\,aa_a)$ for $P(A(aa,ss))$	$P(match_{at}) \times P(match_{st})$ if both types match $P(match_{at}) \times P(mismatch_{st})$ if only the amino acid types match $P(mismatch_{at}) \times P(match_{st})$ if only the secondary structure types match $P(mismatch_{at}) \times P(mismatch_{st})$ if neither types match The match and mismatch probabilities are user-defined	$\frac{\#class(ss)}{\#ss}$
$aa_i \in V^-(aa_a)$, $ss_j \in V^-(ss_b)$, $A(aa_i, ss_j)$, $A(e_{a,i,t}, e_{b,j,t})$	$P(e_{b,j,t}\,\|\,e_{a,i,t}) = erfc(\frac{\|\Delta e\|}{0.02\times\sqrt{2}})$ as used in [17] For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$	$P(e_{b,j,t}\,\|\,e_{a,i,t}) = \frac{\#e_t^{ig}}{\#e^{ig}}$ For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$
$aa_i \in V^-(aa_a)$, $ss_j \in V^-(ss_b)$, $A(aa_i, ss_j)$, $A(e_{a,i,t}, nil)$	$P(A(e_{a,i,t}, nil)\,\|\,e_{a,i,t}) = \begin{cases} 0.21 & \text{if distance} \le 3\mathring{A} \\ 0.41 & \text{if distance} \le 4\mathring{A} \\ 0.38 & \text{otherwise} \end{cases}$ For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$	$P(A(e_{a,i,t}, nil)\,\|\,e_{a,i,t}) = \frac{\#e_{a,i,t}^{>ig}}{\#e_u^{ig}}$ For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$
$aa_i \in V^-(aa_a)$, $ss_j \in V^-(ss_b)$, $A(aa_i, ss_j)$, $A(nil, e_{b,j,t})$	$P(A(nil, e_{b,j,t})) = (1 - \frac{1}{nr_d})$ For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$	$P(A(nil, e_{b,j,t})) = \frac{\#e_{b,j,t}^{>cg}}{\#e_u^{cg}}$ For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$
$aa_i \in V^-(aa_a)$, $ss_j \notin V^-(ss_b)$, $A(aa_i, ss_j)$	$P(ss_j\,\|\,aa_i) = 0.1 \times min(P(ss_b\,\|\,aa_a))$, which serves as a penalty	For $P(ss_j\,\|\,aa_i)$, see case above for $P(ss_b\,\|\,aa_a)$
$aa_i \in V^-(aa_a)$, $A(aa_i, nil)$	$P(A(aa_i, nil)\,\|\,aa_i) = 0.1 \times min(P(ss_b\,\|\,aa_a))$, which serves as a penalty. Otherwise, a missing spin system probability can be used if it can be estimated.	$P(A(aa_i, nil)\,\|\,aa_i) = \begin{cases} \frac{\#aa-\#ss}{\#aa} & \text{if } \#aa > \#ss \\ \varepsilon_v & else \end{cases}$
$aa_i \notin V^-(aa_a)$, $ss_j \in V^-(ss_b)$, $A(aa_i, ss_j)$	$P(A(aa_i, ss_j)\,\|\,ss_j) = 1 - \frac{1}{nr_u}$	$P(aa_i\,\|\,ss_j) = \frac{type(ss_i)}{\#aa}$
$ss_j \in V^-(ss_b)$, $A(nil, ss_j)$	$P(A(nil, ss_j)\,\|\,ss_j) = 1 - \frac{1}{nr_u}$	$P(A(nil, ss_j)\,\|\,ss_j) = \begin{cases} \frac{\#ss-\#aa}{\#ss} & \text{if } \#ss > \#aa \\ \varepsilon_v & else \end{cases}$

Note that the probabilities can be weighted to consider differently edge matches for adjacent amino acids, matches for local alpha helix contacts, and matches for long range beta sheet contacts.

3.4 One Experiment on the ASBA Problem

For proof of concept, with the data we have in hand, we tested our system on the target protein TM1112, a protein from *Thermotoga maritima* [38]. TM1112 has 89 residues, with 17 residues in α-helix regions, 58 in β-sheets, and 14 in loops. The NMR data for TM1112 was provided by the Arrowsmith Lab at the University of Toronto. Since TM1112 contains 5 prolines, the manual resonance assignment done by Arrowsmith Lab has 84 residues assigned.

We applied PICKY to automatically pick peaks from the ^{15}N-HSQC, ^{15}N-edited NOESY, HCCONH-TOCSY spectra of TM1112. However, we were unable to do amino acid typing with HCCONH-TOCSY because it gave the chemical shifts of the previous rather than the current residue's side chain protons. We were also unable to do secondary structure prediction because HNHA specta was not available. We modified the contact graph to solve the TOCSY problem, so that contacts for H_i^{α} to H_j^{N} were represented by edges from H_i^{α} to H_{j+1}^{N}. The noise ratio (number of NOE edges per contact graph edge) was 12.0. We used the x-ray structure (PDB id 1O5U) to form the contact graph. With no typing and secondary structure information, due to the large problem size, we could not directly use our ILP to do the assignment, so we gave it the assignment from IPASS [9], which was used as a fixed assignment. IPASS is an automatic triple resonance experiment-based resonance assignment method that was recently developed by our lab. The IPASS assignment achieved a recall value of 84% and precision of 97% on TM1112.

After assignment by our ILP model, we were able to correctly assign 76 residues out 83 [2], which gave a recall of 91% and precision of 92%. Since NMR spectroscopists prefer reliability over the total number of assigned residues, we tried to reduce the false positive rate by peforming consensus on the assignments generated by our iterative process. We discarded amino acids with no clear assignment as the majority (less than 50% of the assignments). The process was repeated without these amino acids until no amino acids could be discarded. This yielded 76 out of 77 correct assignments, which gave a recall of 91% but precision of 99%. This even corrected one wrong assignment made by IPASS involving an amino acid that had no contact information missing in the NMR data. The majority of the amino acids discarded were missing a large number of NOEs.

Although the power of our ILP model is not fully demonstrated in this test case due to the availability of NMR spectra, this process highlights the ability of our ILP model to explore an assignment given some *apriori* assignment. More importantly, it highlights the possibility of fully automatic structure-based assignment. To our knowledge, this is the first attempt on ASBA which does assignment directly from the NMR spectra.

3.5 Experiments on SBA Problems

Although our goal is to solve the ASBA problem, our ILP model can naturally be applied to solve the SBA problem. To demonstrate this, we tested the performance of the ILP model on the synthetic data set used by the CR method. It consisted of 9 proteins. We were provided data for the proteins 1KA5, 1EGO, 1G6J, 1SGO, and 1YYC from the authors. The data for the other 4 proteins were simulated according to the simulation method described in [28]. Table 2 describes the test set and gives the noise level and the percentage of contact graph edges missing in the interaction graph.

Results with Correct Typing The authors of the CR method also provided us with their program, so we tested it on all 9 proteins. However, the program that was provided did not allow the input of amino acid and secondary structure type predictions, so we could only perform the comparison assuming correct amino acid and correct secondary structure typing.

We followed the experimental approach used by the CR method, and tested each protein using each model in the protein's PDB file as the template. To control noise, our method automatically increased the distance cutoff (in the contact graph) at 0.25Å increments until the noise level was under 8. This gives an improvement

[2] 83 is the maximum number of assignable residues taking into account 5 prolines and the N-terminal residue.

Table 2. Test set. From left to right: target protein/template protein, number of residues in the template (total/helix/sheet/loop); number of spin systems (total/helix/sheet/loop); number of prolines; noise (number NOE edges per contact); distance threshold for defining a contact; (top) percentage of contacts missing in the NMR data ($\leq$ 3Å, $\leq$ 4 Å, $\leq$dist. cutoff), (bottom) percentage of contacts missing by secondary structure type (helix/sheet/loop); average pairwise RMSD of each model structure in the template (total/helix/sheet/loop), or the RMSD between the target and template for 1KTE. Proteins are denoted by their PDB identifier

Target/ Template	No. Residues	No. Spin Sys	No. PRO	Noise (x)	Dist. Cutoff (Å)	Missing (%) by Dist. (by Struct.)	RMSD (Å)
1KA5/1KA5	88/40/23/25	85/39/23/23	1	5.5/5.6/5.9/5.3	4.0	10/29 (20/21/22)	0.2/0.2/0.1/0.2
1EGO/1EGO	85/40/19/26	81/40/19/22	3	5.6/5.4/5.8/6.3	4.0	13/29 (22/26/19)	1.6/1.4/0.9/2.3
1EGO/1KTE	85/33/23/29	81/40/19/22	3	2.9/3.3/2.2/3.1	5.0	17/38/59 (42/54/36)	2.2
1G6J/1G6J	76/18/22/36	72/18/22/32	3	4.4/3.5/5.1/4.8	4.0	22/44 (31/32/35)	1.1/0.6/0.4/1.5
1SGO/1SGO	139/46/28/65	136/46/28/61	3	5.5/4.7/4.0/7.4	4.5	23/40/60 (38/49/40)	10.9/7.3/5.5/14.1
1YYC/1YYC	174/36/72/66	158/36/70/52	10	6.6/5.2/7.5/7.3	4.25	24/43/57 (35/38/40)	4.0/2.5/1.6/6.0
2NBT/2NBT	66/-/16/50	60/-/16/48	5	3.4/-/3.6/3.3	4.0	25/46 (-/22/40)	3.4/-/1.7/3.8
1RYJ/1RYJ	70/9/27/34	67/9/27/31	2	3.1/2.0/3.1/3.8	4.0	16/41 (33/29/25)	1.5/1.0/0.9/1.9
2FB7/2FB7	80/-/32/48	73/-/32/41	7	3.1/-/3.0/3.2	4.0	24/43 (-/30/36)	5.4/-/2.0/6.8
1P4W/1P4W	87/66/-/21	82/65/-/17	3	5.5/5.3/-/6.7	4.0	20/37 (28/-/40)	1.1/0.7/-/1.9

over using a hard 4.0 Å cutoff. We used the same distance thresholds on the CR program. Table 3 compares our method with the CR method, where the first row of each entry gives our results, while the row below gives the CR's. On 8 of the 9 proteins, our average accuracy on the entire protein is better. In fact, the ILP model achieves an average accuracy of 97.1%, whereas the CR method has 86.0% accuracy, which means the ILP model assigns 4.8 times fewer wrong residues. We also noticed that the ILP model significantly outperforms the CR method on both β-sheet and loop regions. This may be due to the fact that our method can maximize the score better as shown in Table 3. In many instances, the score was higher than the score of the correct assignment, which indicates that maximizing contact matches alone may not necessarily give the correct assignment. For 2NBT, where 40% of loop contacts are missing, we did slightly worse, but the score was greater than the score of the correct assignment; similarly for helix residues in 1RYJ. In general, since amino acids in helices tend to have contacts with other amino acids nearby, in many of our tests, we observed that missing edges and typing errors produced local errors in helices. For 1RYJ, the accuracy for helices using a $(i \pm 2)$ window, *i.e.*, allowing a spin system to be assigned within two residues away from the correct residue, was 100%. The run time of our program was significantly faster. The average runtime of the ILP model is only 1.9 seconds per model, which is 1,127 times faster than the CR program. One thing to note is that the CR program was written in Python, and ours was written in Java, and we used CPLEX. All tests were performed on our servers, consisting of Pentium 4 1.4Ghz, 4 GB RAM machines. Our program is single threaded.

Results with Predicted Amino Acid Typing Perfect amino acid typing cannot easily be achieved in reality, although some amino acids, such as Ser, Gly, Thr, Ala, can be identified based on N, H^{α} and H^{β} shifts. Thus, we further tested the ILP model with predicted typing information. We ran RESCUE Version 1 [35] on the experimental proton chemical shifts from the protein's entry in the Biological Magnetic Resonance Bank (BMRB) [39] for the 5 proteins that we received from them. For these tests, the secondary structure type was assumed to be correct. Since we could not provide type predictions to the CR program, we show

Table 3. Comparison between the ILP model and the Contact Replacement Method with correct amino acid and secondary structure typing. For each protein, the first row gives our results, while the second row gives the CR's. From left to right: target protein; average accuracy over all the models(total/helix/sheet/loop); accuracy ranges (total/helix/sheet/loop); number of times the assignment score was greater than, less than, or equal to the score of the correct assignment; and the CPU time per model.

Target	Avg Acc. (%)	Acc. Range (%)	Times Score >, <, = Ref	CPU Time (sec)
1KA5	100/100/100/100	100/100/100	0, 0, 16	2
	94/100/76/100	98-93/91-74/100	0, 16, 0	804
1EGO	98/100/100/93	100-97/100/100/100-90	15, 0, 5	1
	96/96/100/93	100-92/100-90/100/100-79	4, 12, 4	708
1G6J	97/100/100/94	100-95/100/100/100-90	25, 2, 5	1
	91/100/ 87/88	97-89/100/100-86/100-85	0, 32, 0	756
1SGO	96/97/100/94	100-86/100-95/100/100-70	13, 3, 4	3
	80/95/95/62	88-71/100-87/100-86/76-45	0, 20, 0	4,302
1YYC	97/99/96/98	100-93/100/100-91/100-92	17, 0, 3	4
	72/92/62/72	76-67/100-89/69-53/79-64	0, 20, 0	5,292
2NBT	91/-/98/88	96-85/-/100-93/95-79	10, 0 , 0	1
	92/-/95/90	100-88/-/100-88/96-82	1, 9, 0	2,328
1RYJ	97/98/96/96	97-94/100-88/96/96-93	20, 0, 0	1
	82/100/70/86	82-75/100/70/88-72	0, 20, 0	918
2FB7	96/-/97/96	100-91/-/100-93/100-90	7, 0, 3	1
	92/-/94/90	95-88/-/100-94/95-83	0, 10, 0	1,566
1P4W	99/100/-/97	100-97/100/-/100-88	4, 0, 16	3
	77/77/-/77	91-63/91-63/-/90-58	0, 20, 0	3,612
Average	97/99/99/96	-	-	2
	86/94/85/84	-	-	2254

only our results in Table 4. For comparison, we included the results of using amino acid type matching as hard constraints in our program. In general, using soft constraints resulted in higher accuracy than using hard ones. For 1G6J, the amino acid typing accuracy was high, so the improvement was minimal. For 1YYC, the improvement was significant even though the typing accuracy was low. The accuracy, however, varied substantially depending on the model used as the template. Nevertheless, the template with the best score yielded at accuracy of 89.9%, which increases to 94.1% when considering an $(i \pm 2)$ window. This indicates that using multiple templates, such as those generated by normal mode analysis [18], may improve accuracy. For these tests, we did not use the structural constraints criteria for fixing assignments.

Table 4. Assignment accuracy with predicted amino acid typing and correct secondary structure typing. From left to right: target protein; average accuracy with hard typing; average accuracy of soft typing over all the models (total/helix/sheet/loop); accuracy ranges for soft typing (total/helix/sheet/loop); amino acid typing accuracy; number of times the assignment score was greater than, less than, or equal to the score of the correct assignment; and the CPU time per model. Values in parenthesis give the accuracy within an $i \pm 2$ window.

Target	Avg Acc Hard (%)	Avg Acc (%)	Range Acc (%)	A.A. Typing Acc (%)	Times Score >, <, = Ref	CPU Time (sec)
1KA5	86	100/100/100/100	100/100/100/100	89	0, 0, 16	30
1EGO	86	94/92 (99)/100/94	100-91/100-87/100/100-90	90	15, 3, 2	22
1G6J	92	94/100/93/91	97-87/100/100-90/100-78	96	7, 25, 0	3
1SGO	82	92/90 (100)/95/93	96-87/100-84/100-82/96-83	92	7, 13, 0	180
1YYC	59	77/86 (92)/81/66	94-68/100-58/100-52/90-50	79	0, 20 , 0	504

Results with Predicted Amino Acid and Predicted Secondary Structure Typing $^3J_{HNH\alpha}$ coupling constants are used to predict secondary structures as described in [36]. The standard method is similar to the

following: if the coupling value fell between 2.5 and 5.5, the spin system was predicted as helix. If the value was between 8 and 11.5, the spin system was predicted as beta sheet; otherwise, it was predicted as loop. From a test set of the following BMRB entries with accession numbers 4267, 4071, 2151, 4458, 4376, 4136, 4784, 4347, 4163, 4297, plus ubiquitin experimental values from the literature [40], we obtained an average typing accuracy of 60% with a range of 50-69%. This will likely be too low for resonance assignment, so we classified coupling constants into classes consisting of two secondary structure types, which dramatically increased the average accuracy at the cost of increased problem size. For values less than 6.5, we classified it as helix and loop; otherwise we classified it as beta sheet and loop. With this, we obtained an average accuracy of 92% with a range of 82-100%.

We introduced secondary structure class typing errors with probability 0.88 yielding the typing accuracies in Table 5. For the convenience of time, we tested each target using only the first model in the template. The noise ratio and percentage of missing NOEs is similar to the average values in Table 2. For 1KA5, the accuracy did not change from the previous test. For 1EGO, the accuracy actually improved because we used structural constraints for fixing assignments. For 1EGO, we also tested it with the template 1KTE obtained from LOMETS [41], a consensus protein threading server for protein structure prediction. The sequence identity was 26.5%. Since 1EGO and 1KTE had secondary structures of different length, when solving the initial integer program, we found that allowing the secondary structure type to differ at the secondary structure boundaries gave better results. The larger 1SGO struggled to maximize the score, but the accuracy was still much higher than the hard typing case. For 1YYC, its large size combined with its low amino acid typing accuracy, produced poor quality fixed assignments, but there was a large improvement over the hard typing case.

Table 5. Assignment accuracy with predicted amino acid and secondary structure typing. From left to right: target protein; template protein; accuracy with hard typing; accuracy of the best scoring model (total/helix/sheet/loop); amino acid typing accuracy; secondary structure typing accuracy; percentage difference in score of the best scoring assignment compared to the correct one (+ means score of our assignment was higher); and the CPU time per model. Values in parenthesis gives the accuracy in a $(i \pm 2)$ window.

Target	Template	Acc Hard (%)	Acc Best Score (%)	A.A. Typing Acc (%)	S.S. Typing Acc (%)	Diff Ref Score (%)	CPU Time
1KA5	1KA5	72	100/100/100/100	89	91	0	4 hr
1EGO	1EGO	65	97/95 (100)/100/100	90	85	-1.5%	1 h
1EGO	1KTE	67	92/90 (100)/89/100	90	85	+0.4%	1.7 hr
1SGO	1SGO	63	88/82 (91)/96/88	92	87	-3.0%	10.5 hr
1G6J	1G6J	75	91/100/86/90	96	90	+0.5%	32 min
1YYC	1YYC	40	70/91/71/53	79	91	-3.1%	46 hr

3.6 Results on Ubiquitin with Manual Peaks

For ubiquitin, we obtained real ^{15}N HSQC, ^{15}N TOCSY-HSQC, and ^{15}N NOESY-HSQC data from Richard Harris's The Ubiquitin Resource Page [42]. We picked the peaks manually by inspecting the spectra with SPARKY [43]. Ubiquitin has 76 residues and 3 prolines. The noise level is 4.6 while the missing edge percentage is 28.3%. HSQC peaks without an H^{α} chemical shift were identified as noise. For amino acid typing, RESCUE performed poorly, giving an accuracy of 68.6%. We performed the typing manually using the expected number of proton shifts and their expected range of values. Manual typing gave an accuracy of 90%, where the average number of possible amino acid types per spin system was 3.3 with a range of 1 to 8. We used the results of manual typing for assignment. We used experimental $^{3}J_{HNH\alpha}$ coupling constants from the literature [40]. Eight spin systems did not have J-coupling values, so their predicted class included all three secondary structure types. The best scoring assignment had accuracy 87.1%, with 64.3% on α-helix (85.7% with $i \pm 2$ window), 95.7% on β-sheet, and 90.0% on loops. Although the accuracy for helix residues

was low, many of the errors were due to a +1 assignment shift error due to the HSQC peak of a nearby amino acid not being present in the NMR data. We also obtained a consensus assignment by generating 10 solutions from the best scoring assignment with fixed assignments meeting the structure constraint criteria, and then taking the most frequently assigned spin system as the assignment. Consensus gave an accuracy of 91% with 78% for helices (92% $i \pm 2$) and the other types unchanged.

4 Conclusion

We have proposed the Automated Structure-Based Assignment problem (ASBA) and developed the first system to solve this problem. The core novel component of this system is a general ILP model, which was demonstrated to be robust and flexible. The entire system was tested, fully automatically, on a real protein TM1112. The ILP model was also extensively compared with a top SBA method and demonstrated significant improvements in both accuracy and speed.

Acknowledgment

We would like to thank Xiong, Pandurangan, Bailey-Kellogg for providing us with their program and the test data for 5 proteins. We are grateful to Thorsten Dieckmann for thoughtful discussions. NMR spectra for TM1112 were generated as part of the US NIH Protein Structure initiative and kindly provided by A. Gutmanas and C. Arrowsmith.

This work is partially supported by NSERC Grant OGP0046506, China's MOST 863 Grant 2008AA02Z313, Canada Research Chair program, MITACS, an NSERC Collaborative Grant, Premier's Discovery Award, SHARCNET, and the Cheriton Scholarship.

References

1. 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(4):592–610, 1997.
2. 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(2):129–137, 2000.
3. B.E. Coggins and P. Zhou. PACES: Protein sequential assignment by computer-assisted exhaustive search. *Journal of Biomolecular NMR*, 26(2):93–111, 2003.
4. H.N. Moseley, G. Sahota, and G.T. Montelione. Assignment validation software suite for the evaluation and presentation of protein resonance assignment data. *Journal of Biomolecular NMR*, 28(4):341–355, 2004.
5. Y. Jung and M. Zweckstetter. MARS – robust automatic backbone assignment of proteins. *Journal of Biomolecular NMR*, 30(1):11–23, 2004.
6. A. Grishaev, C. A. Steren, B. Wu, A. Pineda-Lucena, C. Arrowsmith, and M. Llinas. Abacus, a direct method for protein nmr structure computation via assembly of fragments. *Proteins*, 61(1):36–43, Oct 2005.
7. K. Wu, J. Chang, J. Chen, C. Chang, W. Wu, T. Huang, T. Sung, and W. Hsu. RIBRA–an error-tolerant algorithm for the NMR backbone assignment problem. *Journal of Computational Biology*, 13(2):229–244, 2006.
8. A. Lemak, C.A. Steren, C.H. Arrowsmith, and Miguel Llinás. Sequence specific resonance assignment via Multicanonical Monte Carlo search using an ABACUS approach. *Journal of Biomolecular NMR*, 41(1):29–41, 2008.
9. 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. *Technical Report CS-2009-16, David R. Cheriton School of Computer Science, University of Waterloo, ON*, 2009. http://www.cs.uwaterloo.ca/research/tr/2009/.
10. 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.
11. J. Hus, J.J. Prompers, and R. Brüschweiler. Assignment strategy for proteins with known structure. *Journal of Magnetic Resonance*, 157(1):119–123, 2002.
12. J. Meiler and D. Baker. Rapid protein fold determination using unassigned NMR data. *Proc Natl Acad Sci U S A*, 100(26):15404–15409, 2003.
13. C.J. Langmead, A. Yan, R. Lilien, L. Wang, and B.R. Donald. A polynomial-time nuclear vector replacement algorithm for automated NMR resonance assignments. *Journal of Computational Biology*, 11(2-3):277–298, 2004.

14. Y. Jung and M. Zweckstetter. Backbone assignment of proteins with known structure using residual dipolar couplings. *Journal of Biomolecular NMR*, 30(1):25–35, 2004.
15. G. Pintacuda, M.A. Keniry, T. Huber, A.Y. Park, N.E. Dixon, and G.Otting. Fast structure-based assignment of 15N HSQC spectra of selectively 15N-labeled paramagnetic proteins. *Journal of the American Chemical Society*, 126(9):2963–2970, 2004.
16. H. Kamisetty, C. Bailey-Kellogg, and G. Pandurangan. An efficient randomized algorithm for contact-based NMR backbone resonance assignment. *Bioinformatics*, 22(2):172–180, 2006.
17. F. Xiong, G. Pandurangan, and C. Bailey-Kellogg. Contact replacement for NMR resonance assignment. *Bioinformatics*, 24(13):i205–i213, 2008.
18. M.S. Apaydin, V. Conitzer, and B.R. Donald. Structure-based protein NMR assignments using native structural ensembles. *Journal of Biomolecular NMR*, 40(4):263–276, 2008.
19. D. Stratmann, C. Heijenoort, and E. Guittet. NOEnet–use of NOE networks for NMR resonance assignment of proteins with known 3D structure. *Bioinformatics*, 25(4):474–481, 2009.
20. 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.
21. 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. W. Gronwald, L. Willard, T. Jellard, R.F. Boyko, K. Rajarathnam, D.S. Wishart, F.D. Sönnichsen, and B.D. Sykes. CAMRA: chemical shift based computer aided protein NMR assignments. *Journal of Biomolecular NMR*, 12(3):395–405, 1998.
23. C. Bailey-Kellogg, A. Widge, J. Kelly, J. Brushweller, and B.R. Donald. The NOESY jigsaw: automated protein secondary structure and main-chain assignment from sparse, unassigned NMR data. *Journal of Computational Biology*, 7:537–558, 2000.
24. P. Pristovsek, H. Rüterjans, and R. Jerala. Semiautomatic sequence-specific assignment of proteins based on the tertiary structure - the program st2nmr. *Journal of Computational Chemistry*, 23:335–340, 2002.
25. M.A. Erdmann and G.S. Rule. Rapid protein structure detection and assignment using residual dipolar couplings. Technical Report CMU-CS-02-195, School of Computer Science, Carnegie Mellon University, 2002.
26. C.J. Langmead and B.R. Donald. An expectation/maximization nuclear vector replacement algorithm for automated NMR resonance assignments. *Journal of Biomolecular NMR*, 29(2):111–138, 2004.
27. P. Pristovsek and L. Franzoni. Stereospecific assignments of protein NMR resonances based on the tertiary structure and 2D/3D NOE data. *Journal of Computational Chemistry*, 27(6):791–797, 2006.
28. F. Xiong and C. Bailey-Kellogg. A hierarchical grow-and-match algorithm for backbone resonance assignments given 3D structure. In *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering*, pages 403–410, 2007.
29. F. Fiorito, T. Herrmann, F.F. Damberger, and K. Wüthrich. Automated amino acid side-chain NMR assignment of proteins using 13C- and 15N-resolved 3D [1H, 1H]-NOESY. *Journal of Biomolecular NMR*, 42(1):23–33, 2008.
30. 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.
31. I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*, pages 1–74. Kluwer Academic Publishers, 1999.
32. J.W. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16(7):521–533, 2002.
33. P. Greistorfer, A. Lokketangen, S. Vob, and D. Woodruff. Experiments concerning sequential versus simultaneous maximization of objective function and distance. *Journal of Heuristics*, 14(6):613–625, 2008.
34. E. Danna, M. Fenelon, Z. Gu, and R. Wunderling. Generating multiple solutions for mixed integer programming problems. *Integer Programming and Combinatorial Optimization*, pages 280–294, 2007.
35. J. L. Pons and M. A. Delsuc. RESCUE: an artificial neural network tool for the NMR spectral assignment of proteins. *Journal of Biomolecular NMR*, 15(1):15–26, 1999.
36. K. Wüthrich. *NMR of Proteins and Nucleic Acids*. John Wiley & Sons, New York, 1986.
37. Jeffrey Skolnick. In quest of an empirical potential for protein structure prediction. *Curr Opin Struct Biol*, 16(2):166–171, Apr 2006.
38. Y. Xia, A. Yee, A. Semesi, and C.H. Arrowsmith. Solution structure of hypothetical protein TM1112. *PDB Database*, 2002.
39. E.L. Ulrich, H. Akutsu, J.F. Doreleijers, Y. Harano, Y.E. Ioannidis, J. Lin, M. Livny, S. Mading, D. Maziuk, Z. Miller, E. Nakatani, C.F. Schulte, D.E. Tolmie, R.K. Wenger, H. Yao, and J.L. Markley. BioMagResBank. *Nucleic Acids Research*, 36(Database issue):D402–D408, 2008.
40. A.C. Wang and A. Bax. Determination of the backbone dihedral angles phi in human ubiquitin from reparametrized empirical Karplus equations. *Journal of the American Chemical Society*, 118(10):2483–2494, 1996.
41. S. Wu and Y. Zhang. LOMETS: a local meta-threading-server for protein structure prediction. *Nucleic Acids Researcj*, 35(10):3375–3382, 2007.
42. R. Harris. The Ubiquitin NMR Resource Page. http://www.biochem.ucl.ac.uk/bsm/nmr/ubq/index.html.
43. T. D. Goddard and D. G. Kneller. Sparky 3. University of California, San Francisco.