CISA: Combined NMR Resonance Connectivity Information Determination and Sequential Assignment

Xiang Wan and Guohui Lin

Abstract—A nearly complete sequential resonance assignment is a key factor leading to successful protein structure determination via NMR spectroscopy. Assuming the availability of a set of NMR spectral peak lists, most of the existing assignment algorithms first use the differences between chemical shift values for common nuclei across multiple spectra to provide the evidence that some pairs of peaks should be assigned to sequentially adjacent amino acid residues in the target protein. They then use these connectivities as constraints to produce a sequential assignment. At various levels of success, these algorithms typically generate a large number of potential connectivity constraints and it grows exponentially as the quality of spectral data decreases. A key observation used in our sequential assignment program, CISA, is that chemical shift residual signature information can be used to improve the connectivity determination and, thus, dramatically decrease the number of predicted connectivity constraints. Fewer connectivity constraints lead to fewer ambiguities in the sequential assignment. Extensive simulation studies on several large test data sets demonstrated that CISA is efficient and effective compared to the three most recently proposed sequential resonance assignment programs, RANDOM, PACES, and MARS.

Index Terms—NMR sequential resonance assignment, spin system, spin system sequential connectivity, spin system residual signature, spin system assignment.

1 Introduction

Protein functions are largely determined by the three-dimensional structure that the protein folds into. Besides computer-aided structure prediction through homology modeling and threading, Nuclear Magnetic Resonance (NMR) spectroscopy and X-ray crystallography are the key experimental technologies for structure determination. Within NMR spectroscopy, a nearly complete sequential assignment of resonance peaks is crucial to successful structure determination and structure dynamics study because minor errors in the assignment might lead to huge structural gaps. The sequential resonance assignment for small proteins can be easy, but becomes complicated and time-consuming for large proteins. Despite many great efforts in the development of automated sequential assignment methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], the assignment remains unsatisfactory in general. Even worse, most of these methods typically fail to output a structurally meaningful assignment when the quality of the protein spectral data is low. Nonetheless, key observations accumulated throughout these development efforts have been mostly agreed upon by NMR spectroscopists and should be taken advantage of in the next generation of automated assignment program development. Our current work could be regarded as one such further effort. In this work, we propose combining the chemical shift residual signature information into connectivity determination to improve the sequential resonance assignment.

Assuming familiarity with protein NMR spectroscopy [15], we briefly describe the sequential resonance assignment problem from a computational point of view. The input data for the assignment are the target protein sequence and a set of NMR spectral resonance peak lists, each of which is identified from an NMR spectrum. One resonance peak is essentially a vector of chemical shifts that records a nuclear magnetic interaction among (normally two or three) nuclei separated by a small number of covalent bonds in the target protein. For example, an HSQC peak list contains 2D peaks, each of which is a pair of chemical shifts for an amide proton and the directly attached nitrogen; an HNCA peak list contains 3D peaks, each of which is a triple of chemical shifts for a nitrogen, the directly adjacent amide proton, and a carbon alpha from the same or the preceding amino acid residue. For ease of presentation, an HNCA peak containing a chemical shift of the intraresidue carbon alpha is referred to as an *intraresidue peak*; otherwise, it is referred to as an *interresidue peak*. The goal of sequential resonance assignment is to map all of these resonance peaks to their corresponding nuclei. By this mapping, the nuclei in the target protein will be labeled with chemical shift values which enable the local structural restraint extraction for the three-dimensional structure calculation.

The underlying principle for the sequential resonance assignment is that the chemical shift values for any specific nucleus are the same across multiple spectra. However,

- *The authors are with the Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada.*
E-mail: {xiangwan, ghlin}@cs.ualberta.ca.

Manuscript received 14 Dec. 2005; revised 8 May 2006; accepted 6 Nov. 2006; published online 22 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0144-1205.
Digital Object Identifier no. 10.1109/TCBB.2007.1047.

1545-5963/07/$25.00 © 2007 IEEE Published by the IEEE CS, CI, and EMB Societies & the ACM

taking the experimental errors into consideration, these values are typically not the same but only fall in a very narrow range (in other words, within some tolerance threshold). Therefore, for every observed chemical shift value, identifying its corresponding nucleus in the target protein is nontrivial. Fortunately, with the availability of a large amount of spectral data for proteins of known 3D structures, such as the BioMagResBank (http://www.bmrb.wisc.edu), the chemical shift distribution for each nucleus (referenced by the atom and the amino acid residue type) can be statistically collected. Subsequently, for the observed chemical shift value, the likelihood that its corresponding nucleus resides in a certain type of amino acid residue can be estimated. Such a likelihood is referred to as the residual signature information of the chemical shift. For a resonance peak, its residual signature information is defined as the sum of the residual signature information of individual chemical shifts therein. Furthermore, since several resonance peaks share their chemical shifts for the amide proton and the directly adjacent nitrogen, they can be grouped together to form a super-vector of chemical shifts, which is referred to as a *spin system*. Similarly, a spin system has its residual signature information defined as the sum of the residual signature information of all individual chemical shifts therein. One of the most recent efforts on effectively quantifying spin system residual signature information to produce a *scoring scheme* for assigning spin systems to amino acid residues was presented in [16] and is employed in this work.

A perfect scoring scheme is expected to always assign a highest probability of each spin system mapping to its corresponding amino acid residue, while significantly lowering probabilities of mapping to other residues in the target protein. In practice, however, scoring schemes have limited strength due to a number of facts. For example, Phenylalanine and Tyrosine residues always have very close chemical shift values for amide proton, nitrogen, carbon alpha, carbon beta, hydrogen alpha, and carbonyl nuclei. The scoring schemes based solely on chemical shifts would not be able to quantify all differences perfectly. Besides this, multiple copies of one type of amino acid residue in the target protein also complicate the assignment process by producing a large number of "equivalent" assignments achieving the objective optimum. This occurs because scoring schemes usually can only differentiate amino acid residue types, but not the residual sequential positions in the target protein. Therefore, to achieve nearly complete assignments, more knowledge is needed.

The second piece of information that enables the success of sequential resonance assignment is the connectivity between spin systems, to be detailed in the following. Recall that interresidue peaks contain chemical shifts for an amide proton and the directly adjacent nitrogen and the carbon alpha in the preceding residue. In other words, the carbon alpha (and the carbon beta in our experiments) chemical shifts appear as intraresidue in one spin system and appear as interresidue in another spin system. This observation tells us that, if these two spin systems are correctly identified, we would be able to draw the conclusion that they must be assigned to two sequentially adjacent amino acid residues in the target protein. Such a two-spin system is called a *connectivity pair*. In the literature, there are many proposals on how to take advantage of the connectivity information in the sequential resonance assignment, such as [1], [17], [4], [18], [7], [19], [9], [10], [13], [14], and many others.

Among the existing sequential resonance assignment programs that use connectivity information, some of them assume deterministic connectivity information [1], [17], [4], [7], [19]. That is, in these programs, connectivities are used as hard constraints on the feasible assignments and, consequently, every spin system can start at most one connectivity pair and end at most one connectivity pair. Spin systems that are chained together through connectivity pairs form into a (directed) path and a maximal path of this type is referred to as a *string* of spin systems. Essentially, this deterministic connectivity information gives rise to a set of disjoint strings and some isolated spin systems (referred to as *singletons*, which might be regarded as strings of length 1). Subsequently, the sequential assignment seeks an optimal mapping of the strings to their corresponding peptide segments in the target protein. Note that the mapped peptide segments in the target protein must be nonoverlapping and the optimum is with respect to the spin system residual signature information. A mathematical model, the *Constrained Bipartite Matching* (CBM), which describes such a global optimization problem, is presented in [8]. Note that, in CBM, when all connectivities are identified, all spin systems will be chained into a single string for which the sequential assignment becomes trivial. In practice, connectivity identification is nontrivial and the general CBM problem is NP-hard [8]. A number of computational methods, such as simulated annealing [17], generic algorithm [1], branch and bound [19], deterministic search [7], and heuristic search [20], have been adopted to tackle this optimization problem.

There are other models proposed for the sequential assignment [18], [9], [10], [12], [14], [13] in which the connectivity information is determined along the way to assignment, typically by using the differences between chemical shift values for common nuclei. As a result, mostly due to the noise and chemical shift degeneracy, connectivity is no longer a binary decision but is probabilistic and one spin system could start more than one connectivity pair (many in general) and could end more than one connectivity pair. This method of connectivity determination essentially gives rise to a *spin system connectivity graph* in which vertices represent spin systems, directed edges represent connectivity pairs, and edge weights represent the probabilities associated with the connectivities. The three most recent representatives are PACES [9], MARS [14], and a random-graph theoretic framework (which we abbreviate as RANDOM) [13]. They all examine how connectivity can be identified from spin systems and then used to both greatly speed up the assignment process and to improve the assignment accuracy at the same time. Essentially, after the connectivity graph is constructed, PACES enumerates all of the paths in the graph, RANDOM generates multiple Hamiltonian paths in a randomized way using edge probabilities proportional to their weights, and MARS examines all paths of a fixed length. A final set of nonconflicting paths is picked as identified connectivities. All three of these programs are run for many iterations to output the best assignment. It is known that, because of data degeneracy, the probability estimation for a connectivity using chemical shift differences might not be accurate. PACES, RANDOM, and MARS all employ tolerance thresholds to cut off unlikely connectivities from the connectivity graph. This process not only narrows down the search space of connectivity, but also simplifies the connectivity graph to make path enumeration/generation feasible. It is worth pointing out that both PACES and MARS enumerate all paths in the connectivity graph without using the edge weights and the enumeration might not be feasible if the graph is not sparse enough (see Section 3 for more information). RANDOM avoids exhaustive enumeration through multiple calls to Hamiltonian path/cycle generation in a randomized way, where the probability of including an edge is set proportional to the edge weight. RANDOM could recover the correct connectivities with a high probability, theoretically, but the correct connectivities are not necessarily all present in one Hamiltonian path unless an extremely large number of paths is generated (this, however, could be even more expensive than exhaustive enumeration). PACES returns the identified connectivities together with the assignment using the TATAPRO residue typing scheme [6]; RANDOM calls to Mapper [7] to return an assignment; MARS returns the identified connectivities together with the assignment using its own z-score-based scoring scheme.

In this paper, we propose performing the Connectivity Information determination and Sequential Assignment simultaneously (acronym CISA, pronounced as "kiss-a") by combining the chemical shift (or spin system) residual signature information into the connectivity determination. The main distinction between CISA and the existing programs PACES, RANDOM, and MARS is the use of spin system residual signature information to progressively grow and validate the paths in the connectivity graph. In this way, a large number of connectivities are filtered out and, thus, would not be examined due to the low quality of their resulting assignments. Consequently, the paths found by CISA might not necessarily be maximal paths in the connectivity graph, but they all have outstanding mapping positions in the target protein. The advantages of this growing-and-assignment proposal, in terms of both efficiency and effectiveness, are demonstrated in the rest of the paper. In Section 2, we present in detail the steps of operations in CISA. Section 3 presents our extensive experimental results. In Section 4, we further discuss our experimental results, as well as the advantages and potential disadvantages of CISA.

2 Methods

2.1 Input and Preprocessing

The inputs to CISA are the target protein sequence and a set of its NMR peak lists. CISA does not require any specific NMR spectra as long as they are sufficient for the sequential assignment purpose. For ease of exposition and to make a fair comparison with PACES, RANDOM, and MARS, we assume the availability of spectral peaks containing chemical shifts for carbon alpha, carbon beta, and carbonyl, besides the HSQC peak list. In the first step, peaks from multiple spectra are anchored at their HSQC peaks to form spin systems, using user-defined chemical shift tolerance thresholds for the amide proton and the directly adjacent nitrogen chemical shifts. The default values for these two thresholds are 0.025 ppm and 0.35 ppm [4], respectively. However, the users may adjust them according to the quality of the given spectral data. One guideline for setting the thresholds is to ensure that the correct combinations of peaks for a particular spin system are always generated, even in the case of severe amide degeneracy. The output of this step is a set of spin systems. Note that the number of spin systems might not be equal to, and is usually less than, the number of residues in the target protein since spin systems for some amino acid residues, for example, Proline, might not be observed (*missing spin systems*). A typical resultant spin system is in the form of a multidimensional vector of chemical shifts $(\mathrm{H}_i^{\mathrm{N}}, \mathrm{N}_i, \mathrm{C}_i^{\alpha}, \mathrm{C}_i^{\beta}, \mathrm{C}_i, \mathrm{C}_{i-1}^{\alpha}, \mathrm{C}_{i-1}^{\beta}, \mathrm{C}_{i-1})$ in which the indices indicate that, if this spin system is mapped to the $i\mathrm{th}$ residue in the target protein, then the last three chemical shifts correspond to the nuclei in its preceding residue. Note that some of the entries might be missing.

In the second step, a scoring scheme is employed to weight the mapping between each spin system and every amino acid residue in the target protein. In the current implementation, the naive Bayesian scoring scheme developed in [16] is adopted, where PsiPred [21] is applied to predict the secondary structure for the residue. Roughly speaking, for each chemical shift in the spin system, the chemical shift values for the corresponding nucleus in the residue are retrieved from the BioMagResBank (http://www.bmrb.wisc.edu) and used as a prior distribution to estimate the probability that the observed chemical shift is associated with the residue residing in the predicted secondary structure. More precisely, there is an error window ϵ and, for the observed chemical shift value cs, we count the number of chemical shift values in the BioMagResBank that fall in the range $(cs - \epsilon, cs + \epsilon)$ for this combination of residue type aa and secondary structure type ss. Denote this number as $N(cs \mid aa, ss)$. The probability is computed as $P(cs \mid aa, ss) = \frac{N(cs \mid aa,ss)}{N(aa,ss)}$, where $N(aa, ss)$ is the total number of the chemical shift values collected in the BioMagResBank. It then takes the absolute logarithm of the probability as the residual signature information. Summing up the individual intraresidue chemical shift residual signature gives the weight for the mapping between the spin system and the amino acid residue in the target protein. Before moving on to the next stage of connectivity determination, each spin system is treated as a singleton.

2.2 Connectivity Graph

The connectivity relationships between spin systems are formulated into an edge-weighted directed graph, referred to as a *connectivity graph*. For every spin system, there is a vertex in the graph (in the rest of the paper, vertex and spin system are used interchangeably). Here, we describe the use of C^{α} and C^{β} chemical shift differences to determine the connectivities between spin systems. In our experiments, we have used another combination which contains C^{α}, C^{β}, and C chemical shift differences. Besides these two, other combinations of chemical shifts are also possible and their connectivity graphs can be similarly built. For two spin systems

$$v_i = (\mathrm{H}_i^{\mathrm{N}}, \mathrm{N}_i, \mathrm{C}_i^{\alpha}, \mathrm{C}_i^{\beta}, \mathrm{C}_{i-1}^{\alpha}, \mathrm{C}_{i-1}^{\beta})$$

and

$$v_j = (\mathrm{H}_j^{\mathrm{N}}, \mathrm{N}_j, \mathrm{C}_j^{\alpha}, \mathrm{C}_j^{\beta}, \mathrm{C}_{j-1}^{\alpha}, \mathrm{C}_{j-1}^{\beta}),$$

if both $|\mathrm{C}_i^{\alpha} - \mathrm{C}_{j-1}^{\alpha}| \leq \delta_{\alpha}$ and $|\mathrm{C}_i^{\beta} - \mathrm{C}_{j-1}^{\beta}| \leq \delta_{\beta}$ hold, then v_i can map to the preceding residue to the one v_j maps to and, therefore, there is an edge from v_i to v_j with its weight calculated as

$$\frac{1}{2}\left(\frac{|\mathrm{C}_i^{\alpha} - \mathrm{C}_{j-1}^{\alpha}|}{\delta_{\alpha}} + \frac{|\mathrm{C}_i^{\beta} - \mathrm{C}_{j-1}^{\beta}|}{\delta_{\beta}}\right). \qquad (1)$$

Similarly, if both $|\mathrm{C}_j^{\alpha} - \mathrm{C}_{i-1}^{\alpha}| \leq \delta_{\alpha}$ and $|\mathrm{C}_j^{\beta} - \mathrm{C}_{i-1}^{\beta}| \leq \delta_{\beta}$ hold, then there is an edge from v_j to v_i with its weight calculated as

$$\frac{1}{2}\left(\frac{|\mathrm{C}_j^{\alpha} - \mathrm{C}_{i-1}^{\alpha}|}{\delta_{\alpha}} + \frac{|\mathrm{C}_j^{\beta} - \mathrm{C}_{i-1}^{\beta}|}{\delta_{\beta}}\right).$$

Here, both δ_{α} and δ_{β} are predetermined chemical shift tolerance thresholds, which are typically set to 0.2 ppm and 0.4 ppm [9], [13], [14], though minor adjustments are sometimes necessary to ensure a sufficient number of connectivities. If neither case occurs, then there is no connectivity between v_i and v_j. We note that (1) is not necessarily the only weighting function. In fact, some other functions on the chemical shift differences could be adopted to weight the edges, for example, the one as suggested in RANDOM.

In the other combination that contains C^{α}, C^{β}, and C chemical shift differences, it is required that at least two out of the following three conditions hold: $|\mathrm{C}_i^{\alpha} - \mathrm{C}_{j-1}^{\alpha}| \leq \delta_{\alpha}$, $|\mathrm{C}_i^{\beta} - \mathrm{C}_{j-1}^{\beta}| \leq \delta_{\beta}$, and $|\mathrm{C}_i - \mathrm{C}_{j-1}| \leq \delta$, and the weight of the edge from v_i to v_j is evaluated analogously as in (1).

After all pairs of vertices (spin systems) have been examined, we finish the construction of the connectivity graph. It is worth pointing out that some correct connectivities might not be present in the connectivity graph (*false negatives*), while some wrong ones might be present (*false positives*). We use #CE to denote the number of correct edges in the connectivity graph and #WE to denote the number of wrong edges. Both of these quantities tell, to some extent, how good the tolerance thresholds are. In the perfect case, #CE is one less than the number of spin systems subtracted by the number of strings and #WE is 0. For every vertex in the connectivity graph, the number of edges coming out is called its *out-degree*. The average out-degree of the connectivity graph, denoted as Ave.OD, captures the complexity (or the density) of the connectivity graph.

2.3 String Growing

With the connectivity graph constructed, PACES, RANDOM, and MARS all proceed to enumerate the paths in the graph, with or without using the edge weights. These paths are then evaluated by examining their mapping positions in the target protein. We choose a different approach to determine the connectivities and perform the assignment, that is, to grow a path only when the path has a good mapping position in the target protein, which is evaluated by the naive Bayesian scoring scheme: For the path under consideration, all edges coming out of the ending spin system are sorted in a nonincreasing order of their weights. For the edge at the head of the order, the temporary extended path (called a *child path*) is formed and its best mapping position on the target protein can be found (via a linear search). The mapping score of this child path, defined as the average mapping score of all the spin systems in the path, is calculated and compared with the mapping scores of all existing paths (including the parent path) to decide whether to accept the extension or not. It is observed that a sufficiently long path is able to detect its succeeding spin system by taking advantage of the discerning power of the naive Bayesian scoring scheme [16]. Therefore, it is expected that using mapping scores to filter the path extensions would give rise to much fewer potential paths for further consideration and eventually avoid exhaustive search, such as is done in PACES and MARS. On the other hand, such path growing accompanied by mapping position verification would also help avoid the blind search such as is done in RANDOM, where the quality of a Hamiltonian path is assessed only toward the end (through a call to Mapper).

```
INPUT:    Spin system connectivity graph $G$ and
              the scoring scheme
OUTPUT:   Determined connectivities and the resultant
              string assignment
      While $G \neq \emptyset$
1.        Initialize OL to fill in simple paths in non-
              increasing average mapping scores;
2.        Grow the path at the head of the order, if
              OL is non-empty;
3.        Remove from CL those paths of length and
              score less than 90% of the top one;
4.1.      Remove edges from paths in CL, if they
              occur in less than 90% of the paths;
4.2.      Report the top path $P$ in CL as the string
              in this iteration;
5.1.      Find the mapping position for $P$, or parts
              of $P$;
5.2.      Remove the spin systems that have been
              assigned to residues from $G$;
      Output the paths and their mapping positions.
```

Fig. 1. A high-level description of steps in CISA.

In more detail, CISA constructs in every iteration one most reliable string out of the connectivity graph through the five steps described below (cf., the high-level description of the steps of operations in CISA in Fig. 1). CISA terminates when the connectivity graph becomes empty and returns the constructed strings, together with their mapping positions in the target protein. In each iteration, CISA starts with an *Open List* (OL) of paths and seeks to expand the one with the best mapping score. The OL can allocate a maximum of S paths (in our experiments, $S = 60$) and, thus, lower quality paths are unlikely to be included. The detailed value set for S (and the value for L defined in the following OL Initialization stage) depends on the computer memory size. In our case, the experiments were done on a typical desktop with 1 Gbyte RAM. We found that S can be chosen from a value in the range [40, 80] (L can be chosen from a value in the range [4, 8]) without affecting the performance significantly. We chose to go with the median value of 60 (6 for L, respectively). The subsequently generated child paths are appended to OL if their mapping scores are high enough and there is room in OL or if their mapping scores are higher than that of some existing path in OL. Another list, the *Complete List* (CL), is kept in CISA to save those paths that cannot be expanded further. At the time when OL becomes empty, high quality paths with their mapping positions are extracted out of CL, where the conflicts are resolved in a greedy fashion, to be detailed in the following.

OL Initialization. Let G denote the connectivity graph. The vertices with in-degree 0 are referred to as *root* vertices. Starting from a root vertex, a breadth first search (BFS) is applied to find all simple paths with a predefined length L (in our experiments, $L = 6$). Back edges associated with the BFS tree are not included in any path. As a result, some found paths might be shorter than L. For every path, the algorithm finds its best mapping position in the target protein and calculates its mapping score. If there is room in OL or its mapping score is higher than that of some path already stored in OL, then it is added (and, in the latter case, the path with the lowest mapping score is removed from OL). The search is done for all root vertices in G. The paths stored in OL are sorted in the nonincreasing order of their mapping scores.

Path Growing. In this step, CISA tries to grow the top ranked path stored in OL. Denote this path as P and remove it from OL. Denote the ending spin system in P as s. Sort the edges coming out of s in G in the nonincreasing order of their weights. Note that these edges should not be back edges with respect to the BFS tree associated with path P. Consider the first edge in the order, say, (s, t), that indicates a possible extension of path P to t. For this potential child path, CISA finds its best mapping position in the target protein and calculates the mapping score. If the mapping score is greater than 90 percent of the score of path P, the child path is considered as an extension. If there is room in OL or its mapping score is higher than that of some path already stored in OL, then the child path is added to OL (and, in the latter case, the path with the least mapping score is removed from OL). Edge (s, t) is removed from the order and CISA continues to consider the edge at the head of the order. If the mapping score is less than 90 percent of the score of path P, there is no more extension for path P to be considered even though there are still edges in the order or even if there is room in OL. Finally, if there is no child path of P that has been added to OL, P is *nonexpandable* and is added to CL. CISA proceeds to consider the top ranked path in OL iteratively and the growing process is done when OL becomes empty.

CL Filtering. Let P denote the path of the highest mapping score (the tie is broken to the longest path) in CL. Other paths in CL with both length and score less than 90 percent of the length and score of path P are discarded from further consideration. The remaining paths are considered to contain reliable connectivities and will be examined further.

Connectivity Filtering. The paths in CL might not necessarily be compatible with each other. Note that two long correct paths are very unlikely to be chained together by a wrong edge to form a longer path since, in any case, at least one of them must be incorrectly mapped to a position in the target protein and such a mapping would bring down the mapping score significantly. Therefore, one naive conclusion could be made that incorrect connectivities would predominantly be close to the ends of the path, while internal connectivities of the paths are largely correct. Our extensive simulation study confirmed this conclusion, which leads to a corollary that correct connectivities must occur on the paths much more frequently than the wrong connectivities. It follows that only those edges occurring in at least 90 percent of the paths in CL are chosen as reliable connectivities and that the other edges are removed from further consideration. These removed edges are returned to the connectivity graph for the next iteration consideration. We remark that the occurrence frequency threshold can be adjusted by users, but we found 90 percent to be the most appropriate throughout our experiments.

The Most Reliable String Finding. At this last stage, the paths in CL are regarded to contain only reliable connectivities. The longest one of them is the string found in this iteration. Denote this path as P. The best mapping position for path P on the target protein is also known. If this mapping conflicts with the mappings of strings determined in the previous iterations, then P has to be broken down by cutting off those conflicting spin systems. Only the longest nonconflicting portion of P, again denoted as P, becomes the string found in this iteration. The spin systems on P are removed from the connectivity graph G, as well as the edges incident to/from them. Note that if there is no spin system from P assigned, then P is discarded from G. If the remaining connectivity graph is still nonempty, CISA proceeds to the next iteration. Otherwise, it terminates and reports the assignment, i.e., the strings it found and their mapping positions in the target protein.

2.4 Implementation

All components of CISA are written in the C/C++ programming language and can be compiled on both Linux and Windows systems. They can be obtained separately or as a whole package through the corresponding author.

3 Experimental Results

We have designed four experiments to test CISA and to compare it with RANDOM and MARS. These four experiments are designed so as to reflect different aspects of CISA and to make fair comparisons. Note that we originally included PACES in our comparison, but, due to the fact that manual adjustments are required in PACES in every iteration in order to obtain the final assignment [9] and we were not able to conduct such manual adjustments to push PACES to the maximal performance, we chose not to report PACES results by us in this paper.

The first experiment is on 12 protein data sets, each of which was simulated from real protein NMR data deposited in the BioMagResBank (http://www.bmrb.wisc.edu). These 12 proteins do not have solved structures and, thus, would not bias the chemical shift signature information [8], [16]. In these data sets, chemical shifts for amide proton H^N, the directly attached nitrogen N, carbon alpha C^α, and carbon beta C^β are included. The use of this typical combination of chemical shifts is to make fair comparisons with RANDOM, which is designed to use the same combination of chemical shifts (in fact, the combination is used by Mapper [7] and RANDOM calls Mapper for the final assignment). MARS was also run on the data sets for extra comparison. Note that MARS does not specify any preference on the chemical shift combinations.

The second experiment is on 10 out of the 12 protein data sets used in the first experiment, where two were left out because they do not have carbonyl C chemical shifts. That is, in these 10 protein data sets, carbonyl C chemical shifts are included in addition to the four types of chemical shifts used in the first experiment. The purpose of this consideration is to make comparisons with MARS on the use of carbonyl C chemical shifts. Both CISA and MARS are expected to perform better on this combination of five types of chemical shifts.

The protein data sets used in the first two experiments are from our previous study on the CBM problem. The third experiment includes 22 proteins tested by PACES [9] and one real data set, Zdom, that we obtained from the example data sets provided by the AutoAssign [4] package. Unfortunately, we were not able to obtain for each protein the exact simulated data set tested by PACES in [9]. Instead, the included data sets were simulated from the corresponding protein entries in the BioMagResBank according to the exact simulation procedure described in [9] with exactly the same parameter setting. All of these data sets are used to make fair comparison with MARS again, but we included the performance of multirun PACES on the simulated data sets in [9] for reference. One should note that there are several more data sets tested by PACES in [9] which do not require simulations and subsequently were excluded from the third experiment (note again that we do not have access to these nonsimulated data sets). Note also that most of the included proteins have been tested by MARS.

The fourth experiment is to demonstrate the computational speed of CISA and its overall performance. For this purpose, all eligible protein entries deposited in the BioMagResBank, which must have chemical shifts for H^N, N, C^α, and C^β for most of the residues, were simulated and tested. We have collected a total of 360 such protein data sets and note that, since it was essentially impossible to run PACES, RANDOM, and MARS on them, only the performance of CISA was collected.

Note that both RANDOM and MARS were run for several iterations on each protein data set in their original papers. In our experiments, they were run for the default numbers of iterations as stated in their original papers, respectively. MARS returns an assignment in which each mapping between a spin system and a residue is associated with either high, medium, or low reliability. We treated all of them as reliable and, thus, the reported performance of MARS could be an overestimation since, in general, only high and medium reliability mappings by MARS are considered confident.

We measure the performance of an assignment program using *precision* and *recall*. The *precision* is defined as the percentage of correctly assigned amino acids among all of the assigned amino acids and the *recall* is defined as the percentage of correctly assigned amino acids among the amino acids that should be assigned spin systems, respectively.

3.1 Experiment 1

In [8], 14 proteins were carefully chosen to form data sets for simulation study on the proposed CBM model for sequential assignments with deterministic adjacencies. These proteins do not have solved atomic structures and were not used to derive the naive Bayesian scoring scheme adopted in our experiments. Among these proteins, bmr4309 and bmr4393 data entries in the BioMagResBank do not contain carbon beta chemical shifts and, thus, cannot be used for our simulation purposes. As a result, only 12 of them were included in our data sets, whose lengths range from 66 to 215.

The data set construction is detailed as follows: For each of these 12 proteins, we extracted its data entry from BioMagResBank to obtain all the chemical shift values for all H^N, N, C^α, and C^β. For each amino acid residue, except for Proline and Glycine, the corresponding four chemical shifts together with C^α and C^β chemical shifts from the preceding residue formed the initial spin system. We excluded Proline residues in the simulation because, in the real NMR data, there would not be spin systems for Prolines since they do not have a H^N atom. Next, for each initial spin system, chemical shifts for intraresidue C^α and C^β were perturbed by adding to them randomized errors that follow independent normal distributions with 0 means and constant standard deviations. We adopted the widely accepted tolerance thresholds for C^α and C^β chemical shifts, which were $\delta_\alpha = 0.2\mathrm{ppm}$ and $\delta_\beta = 0.4\mathrm{ppm}$, respectively [4], [9], [13], [14]. Subsequently, the standard deviations of the normal distributions were set to $0.2/2.5 = 0.08\mathrm{ppm}$ and $0.4/2.5 = 0.16\mathrm{ppm}$, respectively. These thus perturbed spin systems are the final spin systems that form the data set. The 12 instances, with suffix 1, are summarized in Table 1. In order to test the robustness of all four programs, we generated another set of 12 instances through doubling the tolerance thresholds (that is, $\delta_\alpha = 0.4\mathrm{ppm}$ and $\delta_\beta = 0.8\mathrm{ppm}$). They, having suffix 2, are also summarized in Table 1. Obviously, instances in the second set are expected to be harder than the corresponding ones in the first set, for example, indicated by the average out-degrees of the vertices in their connectivity graphs.

All three programs—RANDOM, MARS, and CISA—were called to run on both sets of instances. Note that, in order to run CISA, the intraresidue chemical shift values in a final spin system were used to set up a score for mapping the spin system to every amino acid residue in the target protein, according to the naive Bayesian scoring scheme [16]. The detailed assignment precision and recall by RANDOM, MARS, and CISA are collected in Table 2, and

TABLE 1
Twenty-Four Instances for the First Experiment

	$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm				$\delta_\alpha = 0.4$ppm, $\delta_\beta = 0.8$ppm			
Length	InstanceID	#CE	#WE	Avg.OD	InstanceID	#CE	#WE	Avg.OD
66	bmr4391.1	63	20	1.30	bmr4391.2	63	46	1.72
68	bmr4752.1	65	43	1.64	bmr4752.2	65	120	2.80
78	bmr4144.1	67	20	1.26	bmr4144.2	67	77	2.06
86	bmr4579.1	81	81	1.96	bmr4579.2	81	219	3.58
89	bmr4316.1	82	118	2.61	bmr4316.2	82	309	4.62
105	bmr4288.1	86	25	1.26	bmr4288.2	86	89	1.94
112	bmr4670.1	100	24	1.12	bmr4670.2	100	100	1.79
114	bmr4929.1	108	34	1.30	bmr4929.2	108	117	2.05
115	bmr4302.1	103	18	1.16	bmr4302.2	103	87	1.80
116	bmr4353.1	91	30	1.30	bmr4353.2	91	106	2.07
158	bmr4027.1	141	71	1.48	bmr4027.2	141	252	2.70
215	bmr4318.1	179	157	1.82	bmr4318.2	179	553	3.90

"Length" denotes the length of a protein, measured by the number of amino acid residues therein, "#CE" records the number of correct edges in the connectivity graph, which ideally should be equal to the number of available spin systems subtracted by the number of strings, and "#WE" records the number of wrong edges, respectively, "Avg.OD" records the average out-degree of the connectivity graph.

TABLE 2
Assignment Precision and Recall of RANDOM, MARS, and CISA in the First Experiment

		$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm					
		RANDOM		MARS		CISA	
Length	InstanceID	PR	RE	PR	RE	PR	RE
66	bmr4391.1	0.67 (21)	0.63	0.91 (6)	0.87	0.97 (2)	0.97
68	bmr4752.1	0.40 (40)	0.35	0.98 (1)	0.97	0.96 (3)	0.94
78	bmr4144.1	0.36 (47)	0.33	1.00	0.97	1.00	0.99
86	bmr4579.1	0.54 (39)	0.51	0.97 (3)	0.91	0.98 (2)	0.98
89	bmr4316.1	0.42 (50)	0.36	0.97 (2)	0.96	1.00	0.99
105	bmr4288.1	0.62 (37)	0.55	0.97 (3)	0.95	0.98 (2)	0.98
112	bmr4670.1	0.67 (37)	0.62	0.94 (6)	0.88	0.96 (4)	0.95
114	bmr4929.1	0.68 (36)	0.63	0.99 (1)	0.97	0.93 (8)	0.91
115	bmr4302.1	0.66 (38)	0.64	0.95 (6)	0.92	0.96 (5)	0.95
116	bmr4353.1	0.48 (55)	0.43	0.91 (10)	0.85	0.96 (4)	0.95
158	bmr4027.1	0.43 (85)	0.32	0.96 (7)	0.93	1.00	0.99
215	bmr4318.1	0.40 (122)	0.38	0.88 (24)	0.81	0.87 (26)	0.84
Avg.		0.53	0.48	0.95	0.90	0.96	0.95

		$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm					
		RANDOM		MARS		CISA	
Length	InstanceID	PR	RE	PR	RE	PR	RE
66	bmr4391.2	0.58 (27)	0.55	0.86 (9)	0.85	0.91 (6)	0.91
68	bmr4752.2	0.36 (43)	0.30	0.91 (6)	0.90	0.90 (7)	0.88
78	bmr4144.2	0.33 (49)	0.31	1.00	0.97	1.00	0.99
86	bmr4579.2	0.34 (55)	0.32	0.79 (18)	0.75	0.80 (16)	0.80
89	bmr4316.2	0.35 (56)	0.30	0.95 (4)	0.92	0.83 (14)	0.83
105	bmr4288.2	0.42 (55)	0.38	0.95 (5)	0.93	0.91 (9)	0.91
112	bmr4670.2	0.43 (63)	0.39	0.83 (18)	0.81	0.88 (13)	0.87
114	bmr4929.2	0.46 (60)	0.43	0.99 (1)	0.97	0.96 (5)	0.94
115	bmr4302.2	0.47 (58)	0.45	0.82 (19)	0.80	0.91 (9)	0.91
116	bmr4353.2	0.47 (56)	0.43	0.83 (18)	0.80	0.90 (10)	0.90
158	bmr4027.2	0.40 (89)	0.30	0.82 (28)	0.81	0.88 (21)	0.85
215	bmr4318.2	0.25 (151)	0.22	0.84 (32)	0.75	0.74 (52)	0.70
Avg.		0.41	0.37	0.88	0.85	0.88	0.87

The numbers in the parentheses record the wrong assignments by RANDOM, MARS, and CISA, respectively.

plotted in Fig. 2. In summary, RANDOM achieved, on average, 50 percent and 40 percent assignment precision and recall on the first and the second sets of data sets, respectively, where we followed the exact way of testing as described in [13], in which 1,000 iterations for each instance have been run. This performance of RANDOM matches what is claimed in the original paper [13]. We treated all output mappings by MARS as confident, not just those of high and medium reliability. Therefore, the reported performance of MARS could be an overestimation. Nevertheless, one can still see that CISA performed a little bit better than MARS, though not on every protein data set

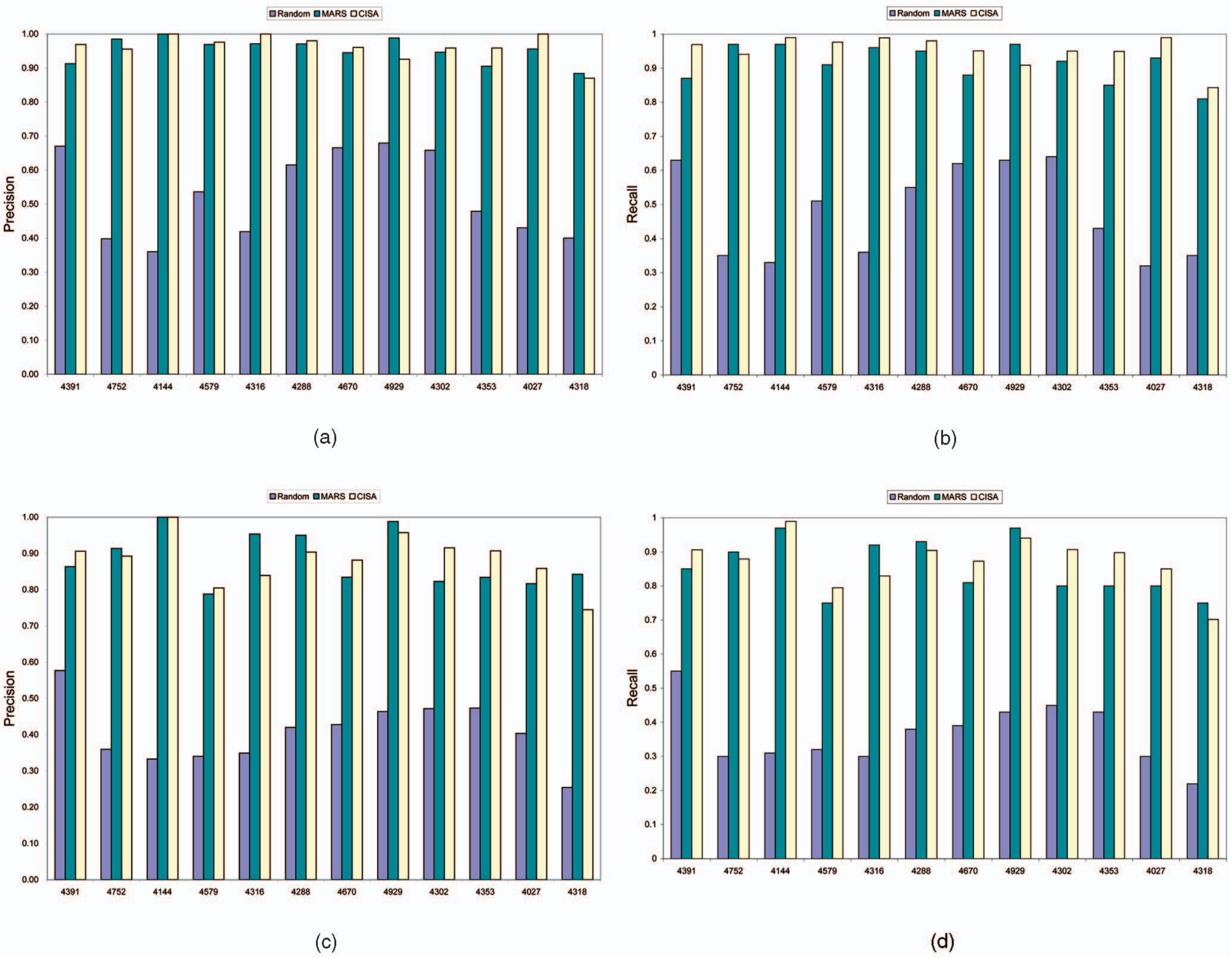


Fig. 2. Plots of assignment precision and recall for RANDOM, MARS, and CISA on two sets of instances with different tolerance thresholds, using C^{α} and C^{β} chemical shifts for connectivity inference. (a) Assignment precision and recall on the first set of 12 instances in the first experiment. (b) Assignment precision and recall on the second set of 12 instances in the first experiment.

(Table 2). We have also run PACES on all of these data sets, each for one iteration. The performance of this one-iteration run was worse than both MARS and CISA. But, since PACES was designed to take in spectral data, including carbonyl C chemical shifts, and we were not able to manually adjust the intermediate assignments for several iterations, one-iteration run of PACES would certainly underestimate the performance of multirun PACES and, therefore, we chose not to report the detailed performance.

3.2 Experiment 2

The instances used in the second experiment are the same set of proteins used in the first experiment, excluding bmr4391 and bmr4316 because their data entries do not have carbonyl C chemical shifts. We use this experiment to compare MARS and CISA. For each residue, five chemical shifts, H^{N}, N, C^{α}, C^{β}, and carbonyl C, were included. Similarly to the data set generation in the first experiment, a spin system here additionally includes the chemical shifts for the intraresidue carbonyl C and for the carbonyl C in the preceding residue. C^{α}, C^{β}, and carbonyl C chemical shift values were used to infer the connectivities. The tolerance threshold for carbonyl C chemical shift was set at $\delta = 0.15\mathrm{ppm}$ and, subsequently, the standard deviation in the error distribution was set at $0.15/2.5 = 0.06\mathrm{ppm}$. For the same reason as in the first experiment, we also generated another set of more difficult instances to test the robustness of all programs through doubling the tolerance thresholds. These two sets of 20 instances are summarized in Table 3.

The detailed assignment precision and recall of MARS and CISA on both sets of instances are collected in Table 4 and plotted in Fig. 3. Again, the reported performance of MARS could be an overestimation since we treated all output mappings by MARS as confident. Nevertheless, one can still see that CISA performed a little bit better than MARS, though not on every protein data set (Table 4). We have also run PACES on all of these data sets, each for one iteration, and we have observed the same tendencies as in the first experiment.

3.3 Experiment 3

Our third experiment was specifically for the fair comparison with MARS and additionally with PACES. This time we chose to use the data sets tested in [9], most of which

TABLE 3
Two Sets of 20 Instances for the Second Experiment

	$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm, $\delta = 0.15$ppm				$\delta_\alpha = 0.4$ppm, $\delta_\beta = 0.8$ppm, $\delta = 0.30$ppm			
Length	InstanceID	#CE	#WE	Avg.OD	InstanceID	#CE	#WE	Avg.OD
68	bmr4752.1	65	15	1.21	bmr4752.2	65	95	2.60
78	bmr4144.1	67	3	1.03	bmr4144.2	67	45	1.61
86	bmr4579.1	81	53	1.63	bmr4579.2	81	188	3.25
105	bmr4288.1	86	1	1.01	bmr4288.2	86	32	1.33
112	bmr4670.1	100	7	1.06	bmr4670.2	100	39	1.37
114	bmr4929.1	108	8	1.06	bmr4929.2	108	60	1.54
115	bmr4302.1	103	4	1.03	bmr4302.2	103	46	1.54
116	bmr4353.1	91	10	1.09	bmr4353.2	91	37	1.38
158	bmr4027.1	141	11	1.06	bmr4027.2	141	91	1.57
215	bmr4318.1	179	25	1.13	bmr4318.2	179	214	2.12

TABLE 4
Assignment Precision and Recall of MARS and CISA in the Second Experiment

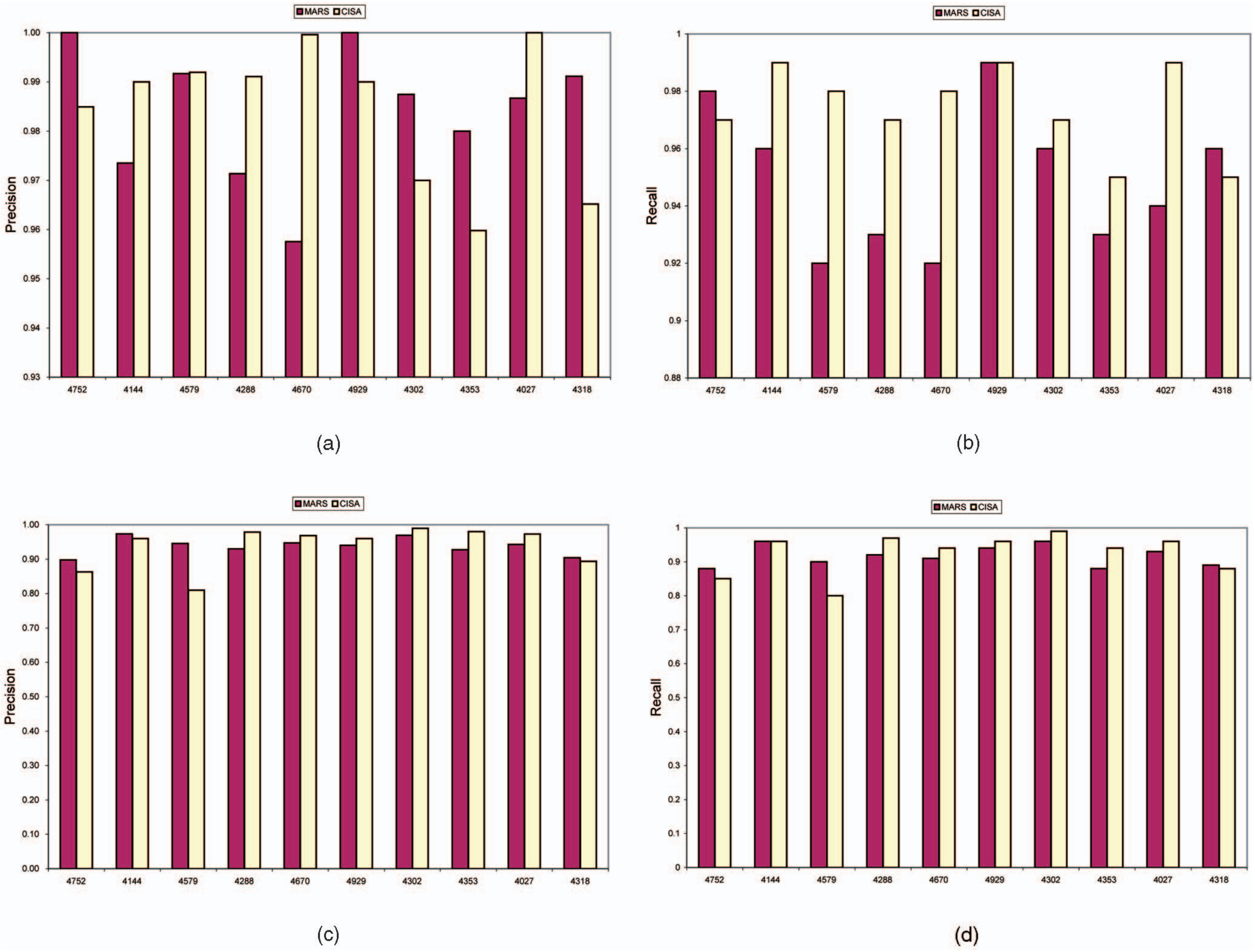
	$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm, $\delta = 0.15$ppm					$\delta_\alpha = 0.4$ppm, $\delta_\beta = 0.8$ppm, $\delta = 0.30$ppm				
		MARS		CISA			MARS		CISA	
Length	InstanceID	PR	RE	PR	RE	InstanceID	PR	RE	PR	RE
68	bmr4752.1	1.00	0.98	0.98 (1)	0.97	bmr4752.2	0.90 (7)	0.88	0.85 (9)	0.85
78	bmr4144.1	0.97 (2)	0.96	0.99 (1)	0.99	bmr4144.2	0.97 (2)	0.96	0.96 (3)	0.96
86	bmr4579.1	0.99 (1)	0.92	0.99 (1)	0.98	bmr4579.2	0.95 (5)	0.90	0.83 (16)	0.80
105	bmr4288.1	0.97 (3)	0.93	0.99 (1)	0.97	bmr4288.2	0.93 (7)	0.92	0.97 (2)	0.97
112	bmr4670.1	0.96 (5)	0.92	1.00	0.98	bmr4670.2	0.95 (6)	0.91	0.97 (3)	0.94
114	bmr4929.1	1.00	0.99	0.99 (1)	0.99	bmr4929.2	0.94 (7)	0.94	0.96 (4)	0.96
115	bmr4302.1	0.99 (1)	0.96	0.97 (3)	0.97	bmr4302.2	0.98 (3)	0.96	0.99 (1)	0.99
116	bmr4353.1	0.98 (2)	0.93	0.96 (4)	0.95	bmr4353.2	0.93 (8)	0.88	0.98 (2)	0.94
158	bmr4027.1	0.99 (2)	0.94	1.00	0.99	bmr4027.2	0.93 (9)	0.93	0.97 (4)	0.96
215	bmr4318.1	0.99 (2)	0.96	0.97 (7)	0.95	bmr4318.2	0.90 (19)	0.89	0.89 (22)	0.87
Avg.		0.98	0.95	0.98	0.98		0.94	0.92	0.94	0.93

The numbers in parentheses record the wrong assignments by RANDOM, MARS, and CISA, respectively.

have also been tested by MARS. We followed the exact simulation procedure that uses three interresidue chemical shifts C^α, C^β, and carbonyl C for connectivity graph construction (tolerance thresholds were $\delta_\alpha = 0.2\text{ppm}$, $\delta_\beta = 0.4\text{ppm}$, and $\delta = 0.15\text{ppm}$). However, note that we did our own simulation because of the unavailability of the simulated data sets from [9]. Our simulated data sets were very close to the corresponding data sets in [9] in terms of the number of missing spin systems. Overall, in these data sets, the percentage of missing spin systems (false negatives) ranged from 3 percent to 39 percent. We note that the existence of missing spin systems challenged the robustness of CISA in many ways. Note also that there was one real instance Zdom included in this experiment, which we obtained from AutoAssign package, and it does not need simulation. Again, the detailed assignment accuracies of MARS and CISA on these 23 instances are collected in Table 5, and plotted in Fig. 4. In addition to these, we also collected the performance of PACES on the corresponding simulated instances included in [9] for reference (PACES was run on Zdom for a single run and its performance was much inferior compared with that on the other instances). In summary, CISA performed slightly better than MARS; PACES with multiple iterations on the simulated data sets included in [9] were the best among the three, typically on precision. That is, PACES didn't make wrong assignments when multiple runs were executed. We have also tested a single run of PACES on our simulated data sets. Again due to the fact that we did not know how to manually adjust to execute multiple iterations to push PACES to the maximal performance, the one-iteration run performance was often much inferior to both MARS and CISA (the overall precision was 83.5 percent and the overall recall was 73.6 percent). Nonetheless, we expect that, with the expertise in manual adjustment, PACES should perform equally well on our simulated data sets. We also expect that, with similar manual adjustments, CISA could perform at least as well as PACES as CISA also accepts manual adjustments. The performance of PACES, MARS, and CISA on the last data set Zdom might show some such hint. Additionally, on the third group of the most difficult data sets, CISA performed noticeably better than MARS (precision 83.0 percent versus 77.3 percent, recall 72.7 percent versus 61.8 percent).

3.4 Experiment 4

The fourth experiment was designed to show the computational efficiency of CISA and its overall performance in terms of assignment precision. To this purpose, we simulated all eligible protein entries deposited in the BioMagResBank using the default tolerance thresholds. We chose to use the chemical shift combination (H^N, N, C^α, C^β) and, consequently, the eligible proteins are those that contain all four of these types of chemical shifts (though they might be obtained from different combinations of NMR spectra). The default tolerance thresholds for C^α and C^β are 0.2 ppm and 0.4 ppm, respectively. To screen out



(a)

(b)

(c)

(d)

Fig. 3. Plots of assignment precision and recall for MARS and CISA on two sets of instances with different tolerance thresholds, using C^{α}, C^{β}, and carbonyl C chemical shifts for connectivity inference. (a) Assignment precision and recall on the first set of 10 instances in the second experiment. (b) Assignment precision and recall on the first set of 10 instances in the second experiment.

some highly degenerate protein entries, we set up a 5 minute time limit for CISA on each protein. That is, if CISA could not terminate the assignment for one protein in 5 minutes, then the protein entry was discarded. We remark that 5 minutes was long enough since, for most of the proteins on which CISA terminated, it terminated within seconds. One interesting discovery is that we found that some proteins have significant resolution differences within their spectral profiles, for example, bmr4402 (cf., Experiment 3) has one half of high resolution but the other half of very low resolution. Through setting up the time limit, CISA was able to detect the low resolution proteins of about 20 kDa in size.

In summary, CISA was able to finish the assignments for 360 proteins in total. The length of these proteins ranges from 58 to 198, and the assignment precision from 0.62 to 1.00. The average assignment precision is 0.903, which is fairly consistent with the results in the first three experiments. For these 360 proteins, the assignment precision versus the length of the protein is plotted in Fig. 5, where each diamond represents an instance. From the plot, we would be able to claim that CISA is insensitive to the size of proteins.

4 Conclusions and Discussion

On a normal desktop with a 1.6 GHz AMD-2000 processor and a 1 Gbyte RAM, for the instances in the first two experiments, the overall running time of CISA ranges from a few seconds to 4 hours (and most of them were done in less than 20 minutes). For the instances in the third experiment, the overall running time of CISA never exceeds 30 minutes. Across all of the experiments, we found that CISA spent a large portion (about 50 percent) of the time in finding the first string. We also observed that, for all instances, after three to four iterations, CISA found the best string in a straightforward way. In other words, CISA running time was mostly consumed in its first three to four iterations. One possible way to speed up CISA in the first string finding could be to use only high probability edges in the connectivity graph. This is currently under investigation. On the other hand, so far, only the average mapping score is used in CISA for the purpose of search space pruning. Some other measurements or their combinations could be incorporated for both efficiency and quality considerations.

CISA uses a number of parameters that have been tuned ahead of time through extensive simulation study (and to

TABLE 5
Assignment Precision and Recall of MARS and CISA in the Third Experiment on Simulated Data Sets for Proteins from [9], Using the Exact Data Set Generation Method as Described in [9] and a Real Data Set `Zdom` Obtained from the AutoAssign Package [4], and the Assignment Precision and Recall of PACES on the Corresponding Simulated Data Sets Included in [9], where the Number of Iterations for PACES Ranged from 1 to 5 and, on Average, Was 2.5

			PACES		MARS		CISA	
Length	InstanceID	#SpinSystems	PR	RE	PR	RE	PR	RE
731	bmr5471	654	1.000	0.960	0.984 (10)	0.956	0.970 (20)	0.945
370	bmr4354	330	1.000	0.940	0.972 (9)	0.948	0.994 (2)	0.979
288	bmr5316	265	1.000	1.000	0.965 (9)	0.911	0.985 (4)	0.940
266	bmr5468	240	1.000	0.980	0.960 (10)	0.923	0.975 (6)	0.938
262	bmr4384	221	1.000	0.940	0.980 (4)	0.900	0.990 (2)	0.950
260	bmr4022	242	1.000	0.930	0.991 (2)	0.930	0.991 (2)	0.959
232	bmr4102	212	1.000	0.950	0.966 (7)	0.933	1.000	0.991
221	bmr4844	198	1.000	0.990	0.969 (6)	0.926	0.979 (4)	0.939
217	bmr4836	206	1.000	0.970	0.978 (5)	0.942	0.985 (3)	0.961
189	bmr4834	166	1.000	0.990	0.980 (3)	0.947	0.969 (5)	0.934
133	bmr4094	129	1.000	1.000	1.000	1.000	1.000	1.000
130	bmr5142	127	1.000	1.000	0.987 (2)	0.941	1.000	0.992
128	bmr4444	106	1.000	1.000	0.947 (6)	0.922	1.000	0.991
124	bmr4032	119	1.000	1.000	1.000	0.990	1.000	0.990
Group 1 Avg.			**1.000**	**0.975**	**0.977**	**0.940**	**0.988**	**0.965**
214	bmr4152	197	1.000	0.970	0.915 (17)	0.791	0.840 (31)	0.772
105	bmr4402	(126–230) 93	1.000	0.980	0.952 (4)	0.880	0.930 (7)	0.860
139	bmr4082	132	1.000	0.990	0.961 (5)	0.932	0.984 (2)	0.924
81	bmr4721	74	1.000	0.950	0.856 (11)	0.810	0.959 (3)	0.933
68	bmr4769	67	1.000	0.880	0.969 (2)	0.940	0.969 (2)	0.956
Group 2 Avg.			**1.000**	**0.954**	**0.931**	**0.870**	**0.937**	**0.889**
227	bmr4457	162	1.000	0.170	0.746 (41)	0.419	0.847 (25)	0.575
192	bmr4341	117	1.000	0.510	0.893 (13)	0.854	0.895 (12)	0.872
110	bmr4136	105	1.000	0.820	0.771 (24)	0.676	0.753 (26)	0.724
71	Zdom	65	0.511 (32)	0.338	0.680 (21)	0.523	0.827 (11)	0.738
Group 3 Avg.			**0.878**	**0.460**	**0.773**	**0.618**	**0.830**	**0.727**
Overall Avg.			**0.983**	**0.723**	**0.931**	**0.867**	**0.950**	**0.907**

Tolerance thresholds are $\delta_\alpha = 0.2\text{ppm}$, $\delta_\beta = 0.4\text{ppm}$, and $\delta = 0.15\text{ppm}$. #SpinSystems records the number of available spin systems for an instance. The data sets are partitioned into three groups. In the first group, data sets all have C^α, C^β, and carbonyl C chemical shifts of high quality. In the second group, data sets all have C^α, C^β, and carbonyl C chemical shifts, but of low quality. In the third group, data sets have only C^α and C^β chemical shifts of various quality. The numbers in the parentheses record the wrong assignments by RANDOM, MARS, and CISA, respectively.

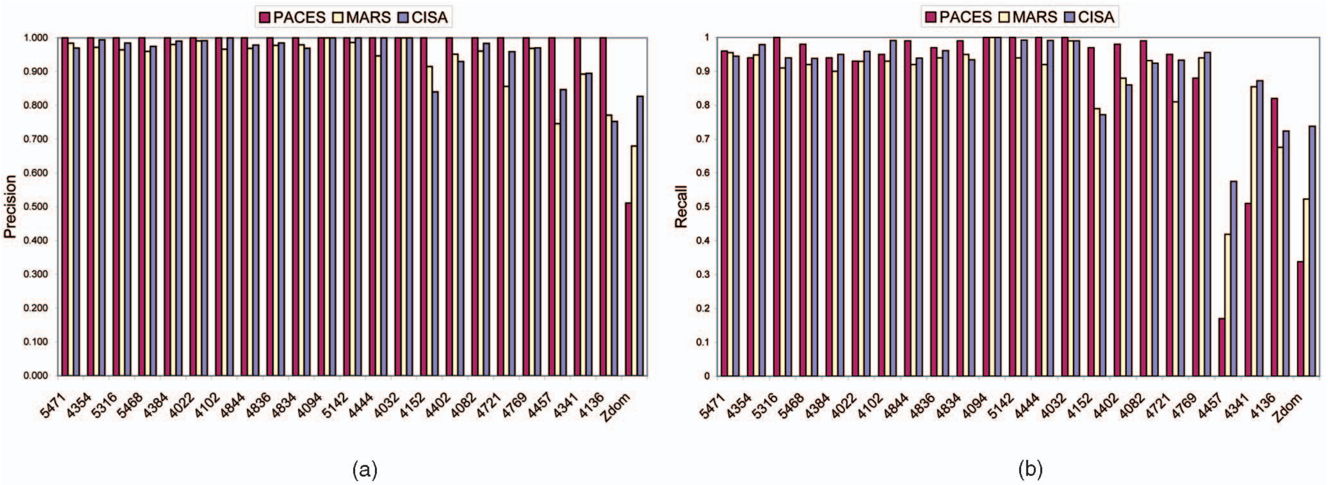


Fig. 4. Plots of assignment precision and recall for MARS and CISA on the simulated data sets for proteins from [9], using the exact data set generation method as described in [9] and a real data set `Zdom` obtained from the AutoAssign package [4], and the plots of assignment precision and recall for PACES on the corresponding simulated data sets included in [9].

map to the computer specifications). It would certainly be better if CISA had an automatic mechanism to detect the complexity of the connectivity graph and thus to use different parameter settings for instances at different complexity levels. We are currently adding this feature into CISA and testing the reliability.

Through CISA, we seem to have successfully combined the spin system residual signature information into the path growing in the connectivity graph, which prunes the search

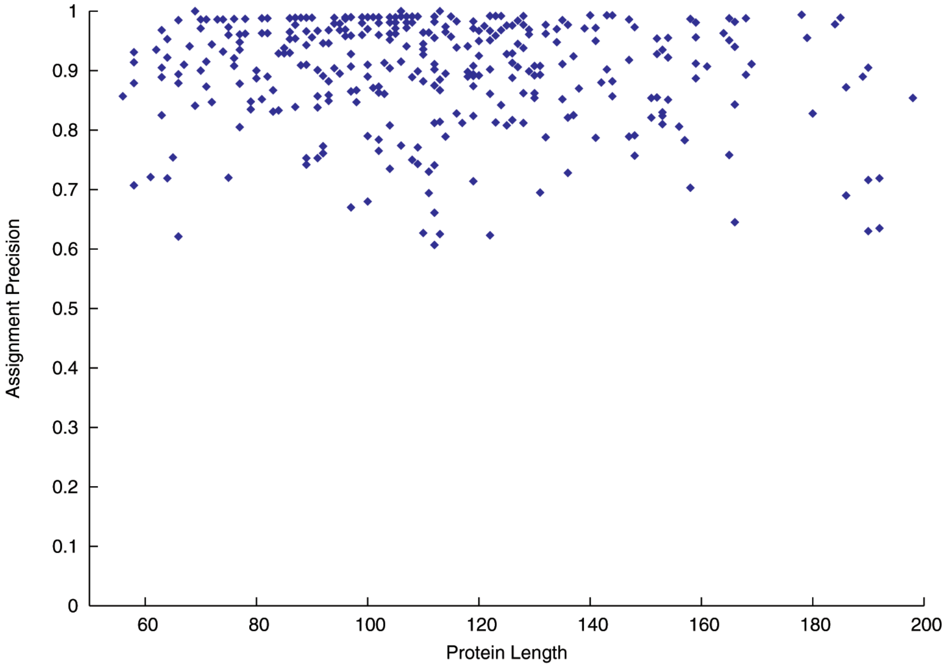


Fig. 5. Plots of assignment precision for CISA on the simulated data sets for 360 proteins from the BioMagResBank, where each diamond represents one instance indexed by its length.

space more effectively compared to PACES and MARS. However, in the current version of CISA, the weights of edges are only used to order the child paths. Taking the idea in RANDOM that uses edge weights as edge selection probabilities, we believe that some better usage of edge weights into the mapping score evaluation for a growing path would help to more effectively quantify the quality of the growing path. We have tried some simple linear functions on the edge weights and the mapping scores of paths that turned out not to serve satisfactorily. We are currently investigating more combinations.

Our last comment on the possible disadvantage of the current version of CISA is very similar to that of RANDOM, where wrong edges included during the OL initialization might continue to stay in and, thus, would lead to erroneous final assignments. Although this is very unlikely to happen according to our extensive simulation study, we feel that some mechanism might need to be set up to shuffle low mapping score paths to be considered once every few iterations during the path growing step.

Acknowledgments

Both authors would like to thank the three referees and Brett Poulin for many critical comments and suggestions which improved the exposition of the manuscript. They are grateful for the research support from AICML, APRI, CFI, NSERC, PENCE, and PrioNet Canada.

References

[1] C. Bartels, P. Güntert, M. Billeter, and K. Wüthrich, "GARANT—A General Algorithm for Resonance Assignment of Multidimensional Nuclear Magnetic Resonance Spectra," *J. Computational Chemistry,* vol. 18, pp. 139-149, 1997.

[2] J.A. Lukin, A.P. Gove, S.N. Talukdar, and C. Ho, "Automated Probabilistic Method for Assigning Backbone Resonances of (13C, 15N)-Labeled Proteins," *J. Biomolecular NMR,* vol. 9, pp. 151-166, 1997.

[3] K.B. Li and B.C. Sanctuary, "Automated Resonance Assignment of Protein Using Heteronuclear 3D NMR.1. Backbone Spin Systems Extraction and Creation of Polypeptides," *J. Chemical Information and Computational Science,* vol. 37, pp. 359-366, 1997.

[4] D.E. Zimmerman, C.A. Kulikowski, Y. Huang, W.F.M. Tashiro, S. Shimotakahara, C. Chien, R. Powers, and G.T. Montelione, "Automated Analysis of Protein NMR Assignments Using Methods from Artificial Intelligence," *J. Molecular Biology,* vol. 269, pp. 592-610, 1997.

[5] H.N.B. Moseley and G.T. Montelione, "Automated Analysis of NMR Assignments and Structures for Proteins," *Current Opinion in Structural Biology,* vol. 9, pp. 635-642, 1999.

[6] H.S. Atreya, S.C. Sahu, K.V.R. Chary, and G. Govil, "A Tracked Approach for Automated NMR Assignments in Proteins (TATAPRO)," *J. Biomolecular NMR,* vol. 17, pp. 125-136, 2000.

[7] P. Güntert, M. Salzmann, D. Braun, and K. Wüthrich, "Sequence-Specific NMR Assignment of Proteins by Global Fragment Mapping with the Program Mapper," *J. Biomolecular NMR,* vol. 18, pp. 129-137, 2000.

[8] Y. Xu, D. Xu, D. Kim, V. Olman, J. Razumovskaya, and T. Jiang, "Automated Assignment of Backbone NMR Peaks Using Constrained Bipartite Matching," *IEEE Computing in Science & Eng.,* vol. 4, pp. 50-62, 2002.

[9] B.E. Coggins and P. Zhou, "PACES: Protein Sequential Assignment by Computer-Assisted Exhaustive Search," *J. Biomolecular NMR,* vol. 26, pp. 93-111, 2003.

[10] T.K. Hitchens, J.A. Lukin, Y. Zhan, S.A. McCallum, and G.S. Rule, "MONTE: An Automated Monte Carlo Based Approach to Nuclear Magnetic Resonance Assignment of Proteins," *J. Biomolecular NMR,* vol. 25, pp. 1-9, 2003.

[11] C.J. Langmead, A. Yan, R. Lilien, L. Wang, and B.R. Donald, "A Polynomial-Time Nuclear Vector Replacement Algorithm for Automated NMR Resonance Assignment," *Proc. Seventh Ann. Int'l Conf. Research in Computational Molecular Biology (RECOMB '03),* pp. 1-12, 2003.

[12] C.M. Slupsky, R.F. Boyko, V.K. Booth, and B.D. Sykes, "SMARTNOTEBOOK: A Semi-Automated Approach to Protein Sequential NMR Resonance Assignments," *J. Biomolecular NMR,* vol. 27, pp. 313-321, 2003.

[13] C. Bailey-Kellogg, S. Chainraj, and G. Pandurangan, "A Random Graph Approach to NMR Sequential Assignment," *Proc. Eighth Ann. Int'l Conf. Research in Computational Molecular Biology (RECOMB '04),* pp. 58-67, 2004.

[14] Y.-S. Jung and M. Zweckstetter, "Mars—Robust Automatic Backbone Assignment of Proteins," *J. Biomolecular NMR,* vol. 30, pp. 11-23, 2004.

[15] J. Cavanagh, W. Fairbrother, A.G. Palmer, and N.J. Skelton, *Protein NMR Spectroscopy: Principles and Practice.* Academic Press, 1996.

[16] X. Wan, T. Tegos, and G.-H. Lin, "Histogram-Based Scoring Schemes for Protein NMR Resonance Assignment," *J. Bioinformatics and Computational Biology,* vol. 2, pp. 747-764, 2004.

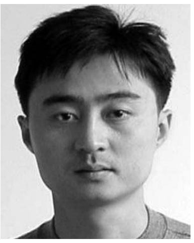
[17] N.E.G. Buchler, E.P.R. Zuiderweg, H. Wang, and R.A. Goldstein, "Protein Heteronuclear NMR Assignments Using Mean-Field Simulated Annealing," *J. Magnetic Resonance,* vol. 125, pp. 34-42, 1997.

[18] C. Bailey-Kellogg, A. Widge, J.J. Kelley III, M.J. Berardi, J.H. Bushweller, and B.R. Donald, "The NOESY Jigsaw: Automated Protein Secondary Structure and Main-Main Assignment from Sparse, Unassigned NMR Data," *J. Computational Biology,* vol. 7, pp. 537-558, 2000.

[19] G.-H. Lin, D. Xu, Z.Z. Chen, T. Jiang, J.J. Wen, and Y. Xu, "Computational Assignment of Protein Backbone NMR Peaks by Efficient Bounding and Filtering," *J. Bioinformatics and Computational Biology,* vol. 1, pp. 387-409, 2003.

[20] G.-H. Lin, T. Tegos, and Z.-Z. Chen, "Heuristic Search in NMR Resonance Peak Assignment," *J. Bioinformatics and Computational Biology,* vol. 3, pp. 1331-1350, 2005.

[21] L.J. McGuffin, K. Bryson, and D.T. Jones, "The PSIPRED Protein Structure Prediction Server," *Bioinformatics,* vol. 16, pp. 404-405, 2000.



Xiang Wan received the BS degree in information systems from Renmin University, China, in 1994, and the MS and PhD degrees in computing science from the University of Alberta, Canada, in 2002 and 2006, respectively. Currently, he is a postdoctoral researcher at the UBC Bioinformatics Center. His research interests include computational biology, machine learning, and metadata mining.

Guohui Lin received the PhD degree in theoretical computer science from the Chinese Academy of Sciences in 1998. He joined the University of Alberta, Canada, as an assistant professor of computing science in July 2001. His research interests include bioinformatics and computational biology and his recent work focuses on algorithmic developments for protein structure determination and comparison, whole genome phylogenetic analysis, RNA structure prediction and comparison, and microarray data analysis. He is a member of the ACM, the IEEE, and the IEEE Computer Society.