A GRAPH-BASED AUTOMATED NMR BACKBONE RESONANCE SEQUENTIAL ASSIGNMENT

Xiang Wan and Guohui Lin*

Department of Computing Science, University of Alberta
Edmonton, Alberta T6G 2E8, Canada
* Email: ghlin@cs.ualberta.ca

The success in backbone resonance sequential assignment is fundamental to protein three dimensional structure determination via NMR spectroscopy. Such a sequential assignment can roughly be partitioned into three separate steps, which are grouping resonance peaks in multiple spectra into spin systems, chaining the resultant spin systems into strings, and assigning strings of spin systems to non-overlapping consecutive amino acid residues in the target protein. Separately dealing with these three steps has been adopted in many existing assignment programs, and it works well on protein NMR data that is close to ideal quality, while only moderately or even poorly on most real protein datasets, where noises as well as data degeneracy occur frequently. We propose in this work to partition the sequential assignment not into physical steps, but only virtual steps, and use their outputs to cross validate each other. The novelty lies in the places where the ambiguities in the grouping step will be resolved in finding the highly confident strings in the chaining step, and the ambiguities in the chaining step will be resolved by examining the mappings of strings in the assignment step. In such a way, all ambiguities in the sequential assignment will be resolved globally and optimally. The resultant assignment program is called GASA, which was compared to several recent similar developments RIBRA, MARS, PACES and a random graph approach. The performance comparisons with these works demonstrated that GASA might be more promising for practical use.

Keywords: Protein NMR backbone resonance sequential assignment, chemical shift, spin system, connectivity graph.

1. INTRODUCTION

Nuclear Magnetic Resonance (NMR) spectroscopy has been increasingly used for protein three-dimensional structure determination. Although it hasn't been able to achieve the same accuracy as X-ray crystallography, enormous technological advances have brought NMR to the forefront of structural biology [1] since the publication of the first complete solution structure of a protein (bull seminal trypsin inhibitor) determined by NMR in 1985 [2]. The underlined mathematical principle for protein NMR structure determination is to employ NMR spectroscopy to obtain local structural restraints such as the distances between hydrogen atoms and the ranges of dihedral angles, and then to calculate the three dimensional structure. Local structural restraint extraction is mostly guided by the backbone resonance sequential assignment, which therefore is crucial to the accurate three dimensional structure calculation. The resonance sequential assignment is to map the identified resonance peaks from multiple NMR spectra to their corresponding nuclei in the target protein, where every peak captures a nuclear magnetic interaction among a set of nuclei and its coordinates are the chemical shift values of the interacting nuclei. Normally, such an assignment procedure is roughly partitioned into three main steps, which are grouping resonance peaks from multiple spectra into spin systems, chaining the resultant spins systems into strings, and assigning the strings of spin systems to non-overlapping consecutive amino acid residues in the target protein, as illustrated in Figure 1, where the scoring scheme quantifies the residual signature information of the peaks and spin systems.

Separately dealing with these three steps has been adopted in many existing assignment programs [3–10]. Furthermore, depending the NMR spectra data availability, different programs may have different starting points. To name a few automated assignment programs, PACES [6], a random graph approach [8] (we abbreviate it as RANDOM in the rest of the paper) and MARS [10] assume the availability of spin systems and focus on chaining the spin systems and their subsequent assignment; AutoAssign [3] and RIBRA [9] can start with the multiple spectral peak lists and automate the whole sequential

---

Scoring

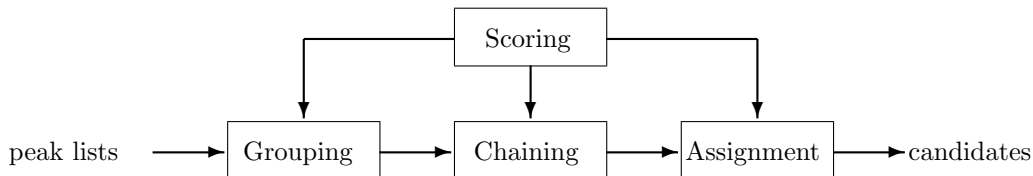peak lists → Grouping → Chaining → Assignment → candidates

Fig. 1. The flow chart of the NMR resonance sequential assignment.

assignment process. In terms of computational techniques, PACES uses exhaustive search algorithms to enumerate all possible strings and then performs the string assignment; RANDOM [8] avoids exhaustive enumeration through multiple calls to Hamiltonian path/cycle generation in a randomized way; MARS [10] first searches all possible strings of length 5 and then uses their mapping positions to filter out the correct strings; AutoAssign [3] uses a best-first search algorithm with constraint propagation to look for assignments; RIBRA [9] applies a weighted maximum independent set algorithm for assignments.

The above mentioned sequential assignment programs all work well on the high quality NMR data, but most of them remain unsatisfactory in practice and even fail when the spectral data is of low resolution. Through a thorough investigation, we identified that the bottleneck of automated sequential assignment is resonance peak grouping. Essentially, a good grouping output gives well organized high quality spin systems, for which the correct strings can be fairly easily determined and the subsequent string assignment also becomes easy. In AutoAssign and RIBRA, the grouping is done through a binary decision model that considers the HSQC peaks as anchor peaks and subsequently maps the peaks from other spectra to these anchor peaks. For such a mapping, the HN and N chemical shift values in the other peaks are required to fall within the pre-specified HN and N chemical shift tolerance thresholds of the anchor peaks. However, this binary-decision model in the peak grouping inevitably suffers from its sensitivity to the tolerance thresholds. In practice, from one protein dataset to another, chemical shift thresholds vary due to the experimental condition and the structure complexity. Large tolerance thresholds could create too many ambiguities in resultant spin systems and consequently in the later chaining and assignment, leading to a dramatic decrease of assign-

ment accuracy; On the other hand, small tolerance thresholds would produce too few spin systems when the spectral data resolution is low, hardly leading to a useful assignment.

Secondly, we found that in the traditional three-step procedure, which is the basis of many automated sequential assignment programs, each step is separately executed, without consideration of inter-step effects. Basically, the input to each step is assumed to contain enough information to produce meaningful output. However, for the low resolution spectral data, the ambiguities appearing in the input of one step seem very hard to be resolved internally. Though it is possible to generate multiple sets of outputs, the contained uncertainties in one input might cause more ambiguities in the outputs, which are taken as inputs to the succeeding steps. Consequently, the whole process would fail to produce a meaningful resonance sequential assignment, which might be possible if the outputs of succeeding steps are used to validate the input to the current step.

In this paper, we propose a *two-phase Graph-based Approach for Sequential Assignment* (GASA) that uses the spin system chaining results to validate the peak grouping and uses the string assignment results to validate the spin system chaining. Therefore, GASA not only addresses the chemical shift tolerance threshold issue in the grouping step but also presents a new model to automate the sequential assignment. In more details, we propose a two-way nearest neighbor search approach in the first phase to eliminate the requirement of user-specified HN and N chemical shift tolerance thresholds. The output of first phase consists of two lists of spin systems. One list contains the *perfect* spin systems, which are regarded as of high quality, and the other the *imperfect* spin systems, in which some ambiguities have to be resolved to produce legal spin systems. In the second phase, the spin system chaining is performed to re-

solve the ambiguities contained in the imperfect spin systems and the string assignment step is included as a subroutine to identify the confident strings. In other words, the ambiguities in the imperfect spin systems are resolved through finding the highly confident strings in the chaining step, and the ambiguities in the chaining step are resolved through examining the mappings of strings in the assignment step. Therefore, GASA does not separate the sequential assignment into physical steps but only virtual steps, and all ambiguities in the whole assignment process are resolved globally and optimally.

The rest of the paper is organized as follows. In Section 2, we introduce the detailed steps of operations in GASA. Section 3 presents our experimental results and discussion. We conclude the paper in Section 4.

2. THE GASA ALGORITHM

The input data to GASA could be a set of peak lists or, assuming the grouping is done, a list of spin systems. In the case of a given list of spin systems, GASA skips the first phase and directly invokes the second phase to conduct the spin system chaining and the assignment. In the other case, GASA firstly conducts a bidirectional nearest neighbor search to generate the perfect spin systems and the imperfect spin systems with ambiguities. It then invokes the second phase which applies a heuristic search, guided by the quality of the string mapping to the target protein, to perform the chaining and assignment for resolving the ambiguities in the imperfect spin systems and meanwhile complete the assignment.

2.1. Phase 1: Filtering

For ease of exposition and fair comparison with RANDOM, PACES, MARS and RIBRA, we assume the availability of spectral peaks containing chemical shifts for C^α and C^β, and the HSQC peak list. One typical example would be the triple spectra containing HSQC, CBCA(CO)NH and HNCACB. Nevertheless, GASA can accept other combinations of spectra. An HSQC spectrum contains 2D peaks each of which corresponds to a pair of chemical shifts for an amide proton and the directly attached nitrogen; An HNCACB spectrum contains 3D peaks each of which is a triple of chemical shifts for a nitrogen, the directly adjacent amide proton, and a carbon

alpha/beta from the same or the preceding amino acid residue; An CBCA(CO)NH spectrum contains 3D peaks each of which is a triple of chemical shifts for a nitrogen, the directly adjacent amide proton, and a carbon alpha/beta from the preceding amino acid residue. For ease of presentation, a 3D peak containing a chemical shift of the intra-residue carbon alpha is referred to as an *intra-peak*; otherwise an *inter-peak*. The goal of filtering is to identify all perfect spin systems without asking for the chemical shift tolerance thresholds. Note that to the best of our knowledge, all existing peak grouping models require the manually set chemical shift tolerance thresholds in order to decide whether two resonance peaks should be grouped into the same spin system or not. Consequently, different tolerance thresholds clearly produce different sets of possible spin systems, and for the low resolution spectral data, a minor change of tolerance thresholds would lead to huge difference in the formed spin systems and subsequently the final sequential assignment. In fact, the proper tolerance thresholds are normally dataset dependent and how to choose them is a very challenging issue in the automated resonance assignment. We propose to use the nearest neighbor approach, detailed as follows using the triple spectra as an example. Due to the high quality of HSQC spectrum, the peaks in HSQC are considered as centers, and every peak in CBCA(CO)NH and HNCACB is distributed to the closest center, using the normalized Euclidean distance. Given a center $C = (\text{HN}_C, \text{N}_C)$ and a peak $P = (\text{HN}_P, \text{N}_P), C_P^{\alpha/\beta})$, the normalized Euclidean distance between them is defined as

$$D = \sqrt{\left(\frac{\text{HN}_P - \text{HN}_C}{\sigma_{\text{HN}}}\right)^2 + \left(\frac{\text{N}_P - \text{N}_C}{\sigma_{\text{N}}}\right)^2}, \quad (1)$$

where σ_{HN} and σ_{N} are the standard deviations of HN and N chemical shifts that are collected from BioMagResBank (`http://www.bmrb.wisc.edu`).

In the ideal case, each center should have 6 peaks distributed to it in total, 4 from HNCACB spectrum and 2 from CBCA(CO)NH spectrum. However, due to the chemical shift degeneracy, some centers may have less than 6 or even 0 peaks. The reasons for this is that the peaks should be associated with these centers might turn out closer to other centers. Therefore, using a set of common chemical shift tolerance thresholds results in more troublesome centers.

Figure 2 illustrates a simple scenario where 3

centers present, but using the common tolerance thresholds C_1 has only 4 peaks associated while C_2 has 8. In Figure 2(a), using the common tolerance thresholds, only one perfect spin system with center C_3 is formed because the two peaks that should belong to center C_1 are closer to center C_2, which create ambiguities in both spin systems. Nevertheless, a closer look that center C_1 reveals that the two peaks that should belong to it but are closer to center C_2 are among the 6 most closest peaks. That is, using the center specific tolerance thresholds, the spin system with center C_1 can be formed by adding these two peaks (see Figure 2(b)); Similarly, using the center specific tolerance thresholds, the spin system with center C_2 becomes another perfect spin system.
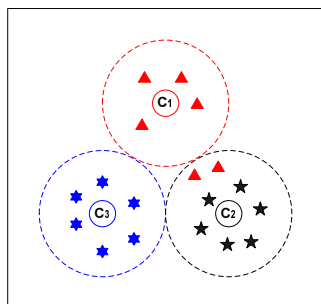
(a) Using the common tolerance thresholds.
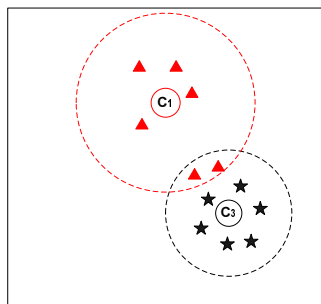
(b) Using the center specific tolerance thresholds.

Fig. 2. A sample scenario in the peak grouping: (a) There are 3 HSQC peaks as 3 centers C_1, C_2, C_3. Every peak is distributed to the closest center, measured by the normalized Euclidean distance. Using the common tolerance thresholds, only C_3 forms a perfect spin system (with exactly 6 associated peaks). (b) Using center specific tolerance thresholds, both C_1 and C_2 find their 6 closest peaks to form perfect spin systems, respectively.

We designed a bidirectional nearest neighbor model, which essentially applies the center specific tolerance thresholds, to have two steps of operations: *Residing* and *Inviting*. In the Residing step, we associated each peak in CBCA(CO)NH and HN-CACB spectra to their respective closest HSQC peak. If the HSQC peak and its associated peaks in CBCA(CO)NH and HNCACB spectra form a perfect spin system, then the resultant spin system is inserted into the list of perfect spin systems. These already associated peaks are then removed from the nearest neighbor model for further consideration. In the Inviting step, each remaining peak in HSQC spectrum looks for the k closest peaks in CBCA(CO)NH and HNCACB spectra, and if a perfect spin system can be formed using some of these k peaks, then the spin system is formed and the associated peaks are removed. The parameter k is related to the number of peaks contained in a perfect spin system, which is known ahead of resonance assignment. A typical value of k is set as 1.5 times the number of peaks in a perfect spin system. In the triple spectra case (HSQC, HN-CACB and CBCA(CO)NH), $k = 9$. The aforementioned two steps will be iteratively executed until no more perfect spin systems can be found and two lists of spin systems, perfect and imperfect, are constructed. Note that this bidirectional nearest neighbor model essentially applies the center specific tolerance thresholds, and thus it does not require any chemical shift tolerance thresholds. Nonetheless, the user could specify maximal HN and N chemical shift tolerance thresholds to speed up the process, though we have noticed that minor differences in these maximal chemical shift tolerance thresholds would not really affect the performance of this bidirectional search.

2.2. Phase 2: Resolving

The goal of *Resolving* is to identify the true peaks contained in the imperfect spin systems and then to conduct the spin system chaining and string assignment. In general, it is very difficult to distinguish the true peaks from the fake peaks when every imperfect spin system is individually examined. During our development, we have found that in most cases, those spin systems containing true peaks enable more confident string finding than those containing fake

peaks. With this observation, we propose to extract true peaks from the imperfect spin systems through spin system chaining and the resultant string assignment, namely, to accept those that result in spin systems having highly confident mapping positions in the target protein.

The relationships between spin systems are formulated into a *connectivity graph* similar to what we have proposed in another sequential assignment program CISA [11]. In the connectivity graph, one vertex corresponds to a spin system. Given two perfect spin systems $v_i = (\text{HN}_i,\ \text{N}_i,\ \text{C}_i^\alpha,\ \text{C}_i^\beta,\ \text{C}_{i-1}^\alpha,\ \text{C}_{i-1}^\beta)$ and $v_j = (\text{HN}_j,\ \text{N}_j,\ \text{C}_j^\alpha,\ \text{C}_j^\beta,\ \text{C}_{j-1}^\alpha,\ \text{C}_{j-1}^\beta)$, if both $|\text{C}_i^\alpha - \text{C}_{j-1}^\alpha| \le \delta_\alpha$ and $|\text{C}_i^\beta - \text{C}_{j-1}^\beta| \le \delta_\beta$ hold, then there is an edge from v_i to v_j with its weight calculated as

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

In Equation (2), both δ_α and δ_β are pre-determined chemical shift tolerance thresholds, which are typically set to 0.2ppm and 0.4ppm, respectively, though minor adjustments are sometimes necessary to ensure a sufficient number of connectivities. Given one perfect spin system $v_i = (\text{HN}_i,\ \text{N}_i,\ \text{C}_i^\alpha,\ \text{C}_i^\beta,\ \text{C}_{i-1}^\alpha,$ $\text{C}_{i-1}^\beta)$ and another imperfect spin system $v_j = (\text{HN}_j,$ $\text{N}_j,\ \text{C}_{j1}^\alpha,\ \text{C}_{j2}^\alpha,\ \cdots,\ \text{C}_{jm}^\alpha,\ \text{C}_{j1}^\beta,\ \text{C}_{j2}^\beta,\ \cdots,\ \text{C}_{jn}^\beta)$, we check each legal combination $v_j' = (\text{HN}_j,\ \text{N}_j,\ \text{C}_{jl}^\alpha,\ \text{C}_{jk}^\beta,\ \text{C}_{jp}^\alpha,$ $\text{C}_{jq}^\beta)$ where $l,k \in [1,m]$ and $p,q \in [1,n]$. Those carbon chemical shifts with subscription l, k represent the intra-residue chemical shifts and those with subscription p, q represent the inter-residue chemical shifts. Subsequently, if both $|\text{C}_i^\alpha - \text{C}_{jp}^\alpha| \le \delta_\alpha$ and $|\text{C}_i^\beta - \text{C}_{jq}^\beta| \le \delta_\beta$ hold, then there is an edge from v_i to v_j' with its weight calculated as

$$\frac{1}{2}\left(\frac{|\text{C}_i^\alpha - \text{C}_{jp}^\alpha|}{\delta_\alpha} + \frac{|\text{C}_i^\beta - \text{C}_{jq}^\beta|}{\delta_\beta}\right). \qquad (3)$$

If both $|\text{C}_{jl}^\alpha - \text{C}_i^\alpha| \le \delta_\alpha$ and $|\text{C}_{jk}^\beta - \text{C}_i^\beta| \le \delta_\beta$ hold, then there is an edge from v_j' to v_i with its weight calculated as

$$\frac{1}{2}\left(\frac{|\text{C}_{jl}^\alpha - \text{C}_i^\alpha|}{\delta_\alpha} + \frac{|\text{C}_{jk}^\beta - \text{C}_i^\beta|}{\delta_\beta}\right). \qquad (4)$$

Note that it is possible that there are multiple edges between one perfect spin system and one imperfect spin system, but at most one of them could be true.

In GASA, no connection is allowed for two imperfect spin systems.

Once the connectivity graph has been constructed, GASA proceeds essentially the same as CISA [11] to apply a local heuristic search algorithm, guided by the mapping quality of the generated string of spin systems in the target protein. Given a string, its mapping quality in the target protein is measured by the average likelihood of spin systems at the best mapping position for the string, where the likelihood of a spin system at a position is estimated by the histogram-based scoring scheme developed in [12]. This scoring scheme is essentially a naive Bayesian learning, and it uses the chemical shift values collected in BioMagResBank (http://www.bmrb.wisc.edu) as prior distributions and estimates for every observed chemical shift value the probability that it is associated with an amino acid residue residing in certain secondary structure. More precisely, for every type of chemical shift, there is a tolerance window of length ϵ. For an observed chemical shift value cs, the number of chemical shift values in BioMagResBank that fall in the range $(cs - \epsilon, cs + \epsilon)$, denoted as $N(cs \mid aa, ss)$, is counted for every combination of amino acid type aa and secondary structure type ss. The probability is then computed as $P(cs \mid aa, ss) = \dfrac{N(cs \mid aa, ss)}{N(aa, ss)}$, where $N(aa, ss)$ is the total number of the same kind of chemical shift values collected in BioMagResBank. The scoring scheme then takes the absolute logarithm of the probability as the mapping score. Summing up the individual intra-residue chemical shift mapping scores in a spin system gives for the spin system its mapping score to every amino acid residue in the target protein.

Therefore, the edges in the connectivity graph are weighted by the scoring scheme, and they are used to order the edges coming out of the ending spin system in the current string to provide the candidate spin systems for the current string to grow to. It has been observed that a sufficiently long string itself is able to detect the succeeding spin system by taking advantage of the discerning power of the scoring scheme. In each iteration of GASA, the search algorithm starts with an *Open List* (OL) of strings and seeks to expand the one with the best mapping score. Another list, *Complete List* (CL), is used in the algorithm to save those completed strings. In the

following, we briefly describe the GASA algorithm for resolving the ambiguities in imperfect spin systems through the spin system chaining into strings and the subsequent assignment.

OL Initialization: Let G denote the constructed connectivity graph. GASA firstly searches for all *unambiguous* edges in G, which are edges in G such that its starting vertex has out-degree 1 and its ending vertex has in-degree 1. It then expands these edges into simple paths with a pre-defined length L by both tracing their starting vertices backward and their ending vertices forward. The tracing stops if either of the following conditions is satisfied. (1) The newly reached vertices are already in the paths; (2) The length of each path reaches L. These paths are stored in OL in the non-increasing order of their mapping scores. The size of OL is fixed at S and thus only the top S paths are kept in OL. Note that both L and S are set in the way to obtain the best trade-off between the computing time and the performance.

Path Growing: In this step, GASA tries to bidirectionally expand the top ranked path stored in OL. Denote this path as P, the starting vertex in P as h and the ending vertex in P as t. All the directed edges incident to h and incident from t are considered as candidate edges to potentially expand P, and the resultant expanded paths are called *child paths* of P. For every potential child path, GASA finds its best mapping position in the target protein and calculates its mapping score. If the mapping score is higher than that of some path already stored in OL, then this child path makes into OL (and accordingly the path with the least mapping score is removed from OL). When none of the potential child paths of P is actually added into OL, or P is not expandable in either direction (that is, there is no edge incident to h, nor edge incident from t), path P is closed for further expanding and subsequently is added into CL. GASA proceeds to consider the top ranked path in OL iteratively and this growing process is terminated when OL becomes empty.

CL Finalizing: Let P denote the path of the highest mapping score in CL (tie is broken to the longest path). GASA performs the following filtering: Firstly, all paths in CL with both their lengths and their scores less than 90% of the length and the

score of path P are discarded from further consideration. These paths are considered as of low quality compared to path P. All the remaining paths are considered to be reliable strings. Next, only those edges occurring in at least 90% of the paths in CL are regarded as reliable ones. The other edges in the paths are therefore removed, which might break the paths into shorter ones. These resultant paths are final candidate paths.

Ambiguities Resolving: GASA scans through the paths in CL for the longest one, which is the confident string built in the current iteration. Nevertheless, it could be that the mapping position in the target protein for this string conflicts mappings in the previous iteration. In this case, GASA respects previous mappings and the current string has to be broken down by removing the spin systems that have the conflicts. Consequently, the spin systems assigned in this iteration might not necessarily form into a single string. These assigned spin systems are then removed from the connectivity graph G, as well as those edges incident to and from them. Additionally, for the imperfect spin systems that are assigned in the current iteration, those peaks that are used to build the spin systems and edges are considered as true peaks, while the others are considered as fake peaks subsequently removed. If the remaining connectivity graph G is still non-empty, GASA proceeds to the next iteration. When it terminates, all the assigned spin systems and their mapping positions are reported as the output assignment.

2.3. Implementation

All components in GASA 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 evaluated the performance of GASA through three comparison experiments with several recent works, including RANDOM, PACES, MARS and RI-BRA. We note that there is another recent work GANA [13] that uses a genetic algorithm to automatically perform backbone resonance assignment with a high degree of precision and recall, which how-

ever due to time constraint we would not be able to make comparison with in the current work. The first experiment is to compare GASA with RANDOM, PACES and MARS only, all of which work well when assuming the availability of spin systems and their original design focuses are on chaining the spin systems into strings and the subsequent string assignment. Such a comparison is interesting since the experimental results will show the validity of combining the spin system chaining with the resultant string assignment in order to resolve the ambiguities in the adjacencies between spin systems. The other two experiments are used for comparison with RIBRA only to judge the value of combining peak grouping into spin systems, spin system chaining, and string assignment all together.

RIBRA explicitly defines two criteria, namely *precision* and *recall*, to measure its performance. In particular, *precision* is defined as the percentage of correctly assigned amino acids among all the assigned amino acids, and *recall* is defined as the percentage of correctly assigned amino acids among the amino acids that should be assigned spin systems, respectively [9]. In this work, we use the same criteria in the second and the third experiments to facilitate the comparison. For the first experiment on the availability of spin systems, where the datasets are simulated such that there is no fake spin system, the performance of an assignment program is measured by the assignment *accuracy*, which is defined as the percentage of correctly assigned spin systems among all the simulated spin systems. (In fact, in this case, accuracy = precision = recall.)

3.1. Experiment 1

The dataset in Experiment 1 is simulated on the basis of 12 proteins in [14], whose lengths range from 66 to 215. The dataset 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 the amide proton HN, the directly attached nitrogen N, the carbon alpha C^α, and the carbon beta C^β. For each amino acid residue, its four chemical shifts together with C^α and C^β chemical shifts from the preceding residue formed the initial spin system. Next, for each such initial spin system, chemical shifts for intra-residue C^α and C^β were perturbed by adding to them random errors that fol-

low 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 $\delta_\alpha = 0.2$ppm and $\delta_\beta = 0.4$ppm, respectively [3, 6, 8, 10]. Subsequently, the standard deviations of the normal distributions were set to $0.2/2.5 = 0.08$ppm and $0.4/2.5 = 0.16$ppm, respectively. The achieved spin system is called a final spin system. These 12 instances, with suffix 1, are summarized in Table 1 (the left half).

In order to test the robustness of all four programs, we generated another set of 12 instances through doubling the tolerance thresholds (that is, $\delta_\alpha = 0.4$ppm and $\delta_\beta = 0.8$ppm). They, having suffix 2, are also summarized in Table 1 (the right half). Obviously, Table 1 tells that instances in the second set are much harder than the corresponding ones in the first set, where the complexity of an instance can be measured by the average out-degree of the vertices in the connectivity graph.

All four programs — RANDOM, PACES, MARS and GASA — were called to run on both sets of instances. The performance results of RANDOM, PACES, MARS and GASA on both sets of instances are collected in Table 2. Their assignment accuracies on two sets are also plotted in Figure 3. In summary, RANDOM achieved on average 50% assignment accuracy (We followed the exact way of determining accuracy as described in [8], where 1000 iterations for each instance have been run.), which is roughly the same as that claimed in its original paper [8]. PACES performed better than RANDOM, but it failed on seven instances where the connectivity graphs were too complex (computer memory ran out, see Discussion for more information). The collected results for PACES on these seven instances were obtained through manually reducing the tolerance thresholds to remove a significant portion of edges from the connectivity graph. We implemented the scheme that if PACES didn't finish an instance in 8 hours, then the tolerance thresholds would be reduced by 25%, for example, from $\delta_\alpha = 0.2$ppm to $\delta_\alpha = 0.15$ppm. We remark that the performance of PACES in this experiment is a bit lower than that is claimed in its original paper [6]. There are at least three reasons for this: (1) The datasets tested in [6] are different from ours. We have done a test on using the datasets in [6] to compare RANDOM, PACES, MARS and CISA, a predecessor of GASA [11], and the result tendency

Table 1. Two sets of instances, each having 12 ones, in the first experiment: '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 (Length −1), and '#WE' records the number of wrong edges, respectively; 'Avg.OD' records the average out-degree of the connectivity graph.

Length	$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm				$\delta_\alpha = 0.4$ppm, $\delta_\beta = 0.8$ppm			
	InstanceID	#CE	#WE	Avg.OD	InstanceID	#CE	#WE	Avg.OD
66	bmr4391.1	65	20	1.29	bmr4391.2	65	51	1.76
68	bmr4752.1	67	43	1.62	bmr4752.2	67	118	2.72
78	bmr4144.1	77	30	1.37	bmr4144.2	77	86	2.09
86	bmr4579.1	85	82	1.94	bmr4579.2	85	221	3.56
89	bmr4316.1	88	168	2.88	bmr4316.2	88	349	4.91
105	bmr4288.1	104	45	1.42	bmr4288.2	104	139	2.34
112	bmr4670.1	111	35	1.30	bmr4670.2	111	109	1.96
114	bmr4929.1	113	41	1.35	bmr4929.2	113	128	2.11
115	bmr4302.1	112	44	1.38	bmr4302.2	112	166	2.46
116	bmr4353.1	114	47	1.40	bmr4353.2	114	139	2.29
158	bmr4027.1	157	85	1.53	bmr4027.2	157	224	3.04
215	bmr4318.1	206	191	1.85	bmr4318.2	206	652	3.99

Table 2. Assignment accuracies of RANDOM, PACES, MARS and GASA in the first experiment. *PACES performance on these 3 datasets were obtained by reducing tolerance thresholds to $\delta_\alpha = 0.15$ppm and $\delta_\beta = 0.3$ppm (75%). †PACES performance on this dataset was obtained by reducing tolerance thresholds to $\delta_\alpha = 0.3$ppm and $\delta_\beta = 0.6$ppm (75%). ‡PACES performance on these 3 datasets were obtained by reducing tolerance thresholds to $\delta_\alpha = 0.2$ppm and $\delta_\beta = 0.4$ppm (50%).

Length	$\delta_\alpha = 0.2$ppm, $\delta_\beta = 0.4$ppm				$\delta_\alpha = 0.4$ppm, $\delta_\beta = 0.8$ppm			
	InstanceID	RANDOM	PACES	MARS	GASA	InstanceID	RANDOM	PACES	MARS	GASA
66	bmr4391.1	0.67	0.70	0.87	0.89	bmr4391.2	0.56	0.67	0.80	0.92
68	bmr4752.1	0.37	0.77	0.97	0.97	bmr4752.2	0.32	0.72‡	0.90	0.87
78	bmr4144.1	0.40	0.51	0.97	0.99	bmr4144.2	0.33	0.35	0.97	0.99
86	bmr4579.1	0.52	0.60*	0.85	0.79	bmr4579.2	0.34	0.41‡	0.71	0.61
89	bmr4316.1	0.37	0.38*	0.96	0.99	bmr4316.2	0.29	0.17†	0.92	0.94
105	bmr4288.1	0.56	0.63	0.95	0.98	bmr4288.2	0.49	0.47	0.93	0.92
112	bmr4670.1	0.62	0.70	0.80	0.83	bmr4670.2	0.44	0.52	0.70	0.61
114	bmr4929.1	0.66	0.83	0.97	0.97	bmr4929.2	0.48	0.74	0.97	0.99
115	bmr4302.1	0.65	0.69	0.92	0.95	bmr4302.2	0.49	0.47	0.82	0.90
116	bmr4353.1	0.48	0.67	0.80	0.90	bmr4353.2	0.45	0.52	0.73	0.87
158	bmr4027.1	0.32	0.77	0.93	0.99	bmr4027.2	0.30	0.30	0.81	0.76
215	bmr4318.1	0.38	0.48*	0.80	0.81	bmr4318.2	0.22	0.40‡	0.62	0.57
Avg.		0.50	0.64	0.90	0.92		0.39	0.48	0.82	0.83

is very much the same as what we have seen in this experiment. (2) PACES is only semi-automated, in the sense that it needs manual adjustment after one iteration to iteratively improve the assignment. In this experiment, PACES was taken as fully automated and it was run for only one iteration. One could run it several iterations for improved assignment. However, in the current work we were unable to manually adjust fairly well and we decided not to do so. (3) PACES is designed to take in better spin systems containing in addition carbonyl chemical shifts. With the current combination PACES was expected to perform a bit lower, since the extra CO chemical shifts provide extra information for resolving ambiguities. Again, we have done a similar test

on using the combination (HN, N, C^α, C^β, CO) of chemical shifts in [6] to compare RANDOM, PACES, MARS and CISA [11], and the result tendency is very much the same as what we have seen in this experiment. MARS and GASA performed equally very well. They both outperformed PACES and RANDOM in all instances, and even more significantly on the second set of more difficult instances, which indicates that combining the chaining and assignment together does effectively resolve the ambiguities and then make better assignments.
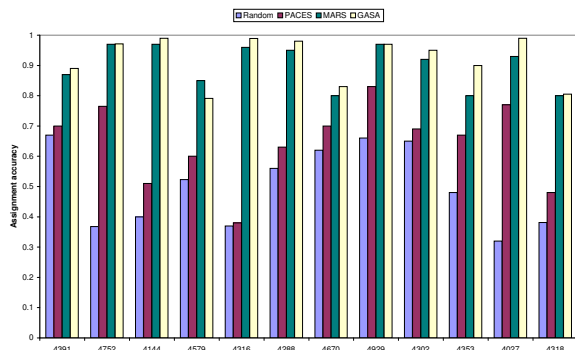
(a) Assignment accuracies on the 1st set of instances.
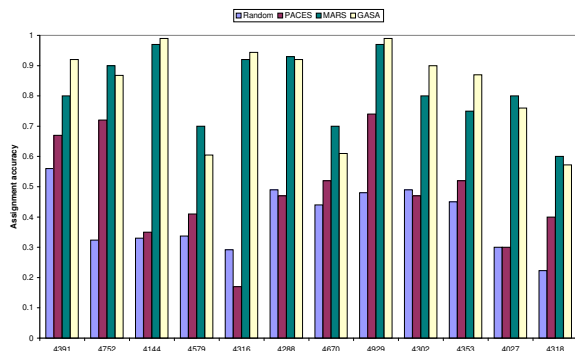
(b) Assignment accuracies on the 2nd set of instances.

Fig. 3. Plots of assignment accuracies for RANDOM, PACES, MARS and GASA on two sets of instances with different tolerance thresholds, using C^α and C^β chemical shifts for connectivity inference.

3.2. Experiment 2

In RIBRA, 5 sets of different datasets were simulated from the data entries deposited in BioMagResBank. Among them, one is *perfect* dataset, which is simulated from BioMagResBank without adding any errors, and the other four datasets contain four different types of errors respectively. The *false positive* dataset is generated by respectively adding 5% carbon fake peaks into perfect CBCA(CO)NH and HNCACB peak lists. The *false negative* dataset is generated by randomly removing a small portion of inter carbon peaks from perfect CBCA(CO)NH and HNCACB peak lists. The *grouping error* dataset is generated by adding HN, N, C^α and C^β perturbations into inter peaks in the perfect CBCA(CO)NH peak list. The *linking error* dataset is generated by

adding C^α and C^β perturbations into inter peaks in the perfect HNCACB peak list.

Table 3 collects the average performances of RI-BRA and GASA on these 5 sets of datasets. As shown, there is no significant difference among the performances on the *perfect, false positive* and *link error* datasets. GASA shows more robustness on the dataset with missing data while RIBRA performs better on the *grouping error* dataset. Through the detailed investigation, we found that these 5 sets of datasets contain the C^β inter and intra peaks with 0 C^β chemical shifts for Glycine, indicating that in the RIBRA simulation, Glycine would have two inter peaks and two intra peaks in HNCACB and the amino acid residues after Glycine would have two inter peaks in CBCA(CO)NH. However, this is not the case in the real NMR spectral data. In fact, a huge amount of ambiguity in the sequential assignment results from Glycine because it produces various legal combinations in grouping and thus making the identification of perfect spin systems harder. For example, the spin systems containing 3, 4 and 5 peaks have the same chance to be perfect spin systems as those containing 6 peaks and meanwhile they could be considered as the spin systems with missing peaks. Therefore, grouping is much easier on the datasets with the simulated C^β peaks for Glycine. Since GASA is designed to deal with the real spectral data, in which there are no peaks with 0 carbon chemical shifts, the performance of GASA on the *grouping error* dataset is not as good as RIBRA. To verify our thoughts, we randomly selected 14 proteins among the *grouping error* dataset, with length ranging from 69 to 186, and removed all the peaks of 0 C^β chemical shift. Both RIBRA and GASA were tested on them. RIBRA achieved 87.7% precision and 72.7% recall, and GASA achieved 88.5% precision and 79.4% recall, slightly better. It is noticed that in the construction of *grouping error* dataset, RIBRA kept the perfect HSQC and HNCACB peak lists untouched and only added some perturbations to the inter peaks in the CBCA(CO)NH peak list. We believe that to simulate a real NMR spectral dataset, perturbing chemical shifts in all simulated peaks is necessary and would be closer to the reality because the chemical shifts deposited in BioMagResBank have been manually adjusted across multiple spectra. Even though HSQC is a very reliable experiment, the deposited HN and N chemical

Table 3. Comparison results for RIBRA and GASA in Experiment 2. Percentages in parentheses were obtained on 14 randomly chosen proteins with C^β peaks for Glycine removed.

Dataset	RIBRA		GASA	
	Precision	Recall	Precision	Recall
Perfect	98.28%	92.33%	98.24%	93.44%
False positive	98.28%	92.35%	97.33%	92.24%
False negative	95.61%	77.36%	96.34%	89.0%
Grouping error	98.16% (87.7%)	88.57% (72.7%)	91.12% (88.5%)	81.27% (79.4%)
Linking error	96.28%	89.15%	96.17%	89.74%
Average	97.33%	87.95%	95.84%	89.14%

shifts in BioMagResBank are still slightly different from the measured values in the real HSQC spectra (`http://bmrb.wisc.edu/`). In the next Experiment 3, we chose not to simulate C^β peaks for Glycine and to perturb every piece of chemical shift in the data.

3.3. Experiment 3

The purpose of Experiment 3 is to provide more convincing comparison results between GASA and RIBRA, based on the better data simulation. For this purpose, we used the same 12 proteins in Experiment 1 and the simulation 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 HN, N, C^α, and C^β. For each amino acid residue in the protein, except Proline, its HN and N chemical shifts formed a peak in HSQC peak list; its HN and N chemical shifts with C^α and C^β chemical shifts from the preceding residue formed two inter peaks respectively in CBCA(CO)NH peak list; and its HN and N chemical shifts with its own C^α and C^β chemical shifts and with C^α and C^β chemical shifts from the preceding residue formed two intra peaks and two inter peaks respectively in HNCACB peak list. Note that there is no C^β peak for Glycine in either CBCA(CO)NH or HNCACB peak list. Next, for each peak in HSQC, CBCA(CO)NH and HNCACB peak lists, the contained HN, N, C^α or C^β chemical shifts were perturbed by adding to them random errors that follow independent normal distributions with 0 means and constant standard deviations. We chose the same tolerance thresholds as those used in RIBRA, which were $\delta_{HN} = 0.06$ppm for HN, $\delta_N = 0.8$ppm for N, $\delta_\alpha = 0.2$ppm for C^α, and $\delta_\beta = 0.4$ppm for C^β, respectively. Subsequently, the standard deviations of the normal distributions were set to $0.06/2.5 = 0.0024$ppm, $0.8/2.5 = 0.32$ppm,

$0.2/2.5 = 0.08$ppm, and $0.4/2.5 = 0.16$ppm, respectively.

Partial information of and the performances of RIBRA and GASA on these 12 proteins are summarized in Table 4. The detailed datasets are available through link `http://www.cs.ualberta.ca/~ghlin/src/WebTools/gasa.php`. From the table, we can see that GASA formed many more spin systems than RIBRA did on every dataset, and from the assignment precision we can conclude that most of these spin systems are true spin systems. On average, GASA performed significantly better than RIBRA (precision 86.72% versus 65.23%, recall 74.18% versus 42.10%). The detailed precision and recall are also plotted in Figure 4. In summary, GASA outperformed RIBRA in all instances and RIBRA failed to solve three instances, which are `bmr4316`, `bmr4288` and `bmr4929`. As shown in Table 4, RIBRA only achieved 65.23% precision and 42.1% recall on average, which are noticeably worse than what it is claimed in [9]. The possible explanations for RIBRA not doing well on these 12 instances are: (1) The simulation procedure in Experiment 3 didn't generate C^β peaks with 0 chemical shift for Glycines, which causes more ambiguities in the peak grouping, and subsequent spin system chaining. (2) In the 12 simulated datasets in Experiment 3, the chemical shifts in every peak in all HSQC, HNCACB and CBCA(CO)NH peak lists were perturbed with random reading errors, which generated more uncertainties in every step of operation in the sequential assignment.

4. CONCLUSIONS

In this paper, we proposed a novel two-stage graph-based algorithm called GASA for protein NMR backbone resonance sequential assignment. The input to

Table 4. Partial information of and the performance of RIBRA and GASA on the 12 protein NMR datasets in Experiment 3. 'Length' denotes the length of a protein, measured by the number of amino acid residues therein; 'Missing' records the number of true spin systems that are not simulated in the dataset, including those for Prolines; 'Grouped' records the number of spin systems that were actually formed by RIBRA and GASA, respectively.

BMRB Entry	Length	Missing	RIBRA			GASA		
			Grouped	Precision	Recall	Grouped	Precision	recall
bmr4391	66	7	44	65.12%	48.54%	52	92.32%	81.41%
bmr4752	68	2	44	63.12%	42.33%	54	90.71%	74.22%
bmr4144	78	10	42	64.25%	39.68%	63	84.12%	77.93%
bmr4579	86	3	54	66.34%	43.22%	70	82.92%	69.93%
bmr4316	89	4	N/A	N/A	N/A	67	79.11%	62.37%
bmr4288	105	9	N/A	N/A	N/A	84	82.91%	72.32%
bmr4670	112	10	47	76.23%	35.35%	83	90.44%	73.65%
bmr4929	114	4	N/A	N/A	N/A	89	95.51%	77.32%
bmr4302	115	8	70	71.35%	46.67%	97	84.52%	76.61%
bmr4353	116	18	72	55.24%	40.75%	89	96.62%	87.38%
bmr4027	158	10	96	65.23%	42.15%	123	82.64%	68.92%
bmr4318	215	24	127	60.22%	40.17%	165	78.81%	68.13%
Average				65.23%	42.1%		86.72%	74.18%

GASA can be spin systems or raw spectral peak lists. GASA is based on an assignment model that separates the whole assignment process only into virtual steps and uses the outputs from these virtual steps to cross validate each other. The novelty of GASA lies in the places where all ambiguities in the assignment process are resolved globally and optimally. The extensive comparison experiments with several recent works including RANDOM, PACES, MARS and RIBRA showed that GASA is more effective in dealing with the NMR spectral data degeneracy and thereby provides a more promising solution to automated resonance sequential assignment.

We have also proposed a spectral dataset simulation method that generates datasets closer to the reality. One of our future works is to formalize this simulation method to produce a large number of protein NMR datasets for common comparison purpose. One of the reasons for doing this is that, though BioMagResBank as a repository has collected all known protein NMR data, somehow there is no benchmark testing datasets in the literature. As a preliminary effort, the 12 simulated protein NMR datasets, in the format of triple spectra HSQC, HNCACB and CBCA(CO)NH, are available through link `http://www.cs.ualberta.ca/~ghlin/src/WebTools/gasa.php`.
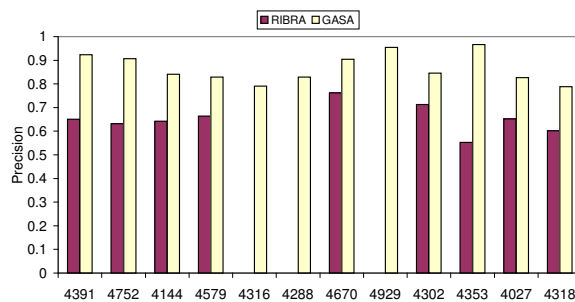
(a) Assignment precision.
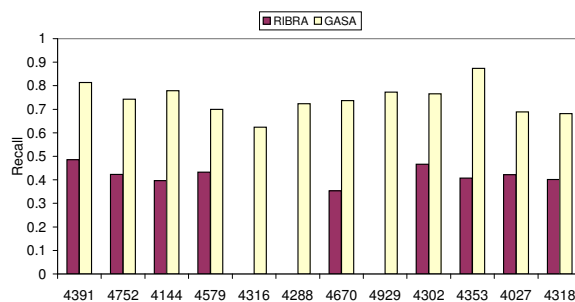
(b) Assignment recall.

Fig. 4. Plots of detailed assignment (a) precision and (b) recall on each of the 12 protein datasets in Experiment 3 by RIBRA and GASA.

ACKNOWLEDGMENTS

References

1. A. E. Ferentz and G. Wagner. NMR spectroscopy: a multifaceted approach to macromolecular structure. *Quarterly Review Biophysics*, 33:29–65, 2000.

2. M. P. Williamson, T. F. Havel, and K. Wüthrich. Solution conformation and proteinase inhibitor IIA from bull seminal plasma by proton NMR and distance geometry. *Journal of Molecular Biology*, 182:295–315, 1985.

3. 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. *Journal of Molecular Biology*, 269:592–610, 1997.

4. P. Güntert, M. Salzmann, D. Braun, and K. Wüthrich. Sequence-specific NMR assignment of proteins by global fragment mapping with the program Mapper. *Journal of Biomolecular NMR*, 18:129–137, 2000.

5. G.-H. Lin, D. Xu, Z. Z. Chen, T. Jiang, J. J. Wen, and Y. Xu. An efficient branch-and-bound algorithm for the assignment of protein backbone NMR peaks. In *Proceedings of the First IEEE Computer Society Bioinformatics Conference (CSB 2002)*, pages 165–174. IEEE Computer Society Press, 2002.

6. B. E. Coggins and P. Zhou. PACES: Protein sequential assignment by computer-assisted exhaustive search. *Journal of Biomolecular NMR*, 26:93–111, 2003.

7. 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. *Journal of Biomolecular NMR*, 25:1–9, 2003.

8. C. Bailey-Kellogg, S. Chainraj, and G. Pandurangan. A random graph approach to NMR sequential assignment. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2004)*, pages 58–67, 2004.

9. K.-P. Wu, J.-M. Chang, J.-B. Chen, C.-F. Chang, W.-J. Wu, T.-H. Huang, T.-Y. Sung, and W.-L. Hsu. RIBRA – an error-tolerant algorithm for the NMR backbone assignment problem. In *Proceedings of the 9th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2005)*, pages 103–117, 2005.

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

11. X. Wan and G.-H. Lin. CISA: Combined NMR resonance connectivity information determination and sequential assignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2005. Submitted.

12. X. Wan, T. Tegos, and G.-H. Lin. Histogram-based scoring schemes for protein NMR resonance assignment. *Journal of Bioinformatics and Computational Biology*, 2:747–764, 2004.

13. H.-N. Lin, K.-P. Wu, J.-M. Chang, T.-Y. Sung, and W.-L. Hsu. GANA – a genetic algorithm for NMR backbone resonance assignment. *Nucleic Acids Research*, 33:4593–4601, 2005.

14. 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 & Engineering*, 4:50–62, 2002.