Journal of Bioinformatics and Computational Biology
Vol. 5, No. 2 (2007) 1–21

Imperial College Press
www.icpress.co.uk

GASA: A GRAPH-BASED AUTOMATED NMR BACKBONE RESONANCE SEQUENTIAL ASSIGNMENT PROGRAM

XIANG WAN* and GUOHUI LIN†

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

Received 30 August 2006

The success in backbone resonance sequential assignment is fundamental to three dimensional protein structure determination via Nuclear Magnetic Resonance (NMR) spectroscopy. Such a sequential assignment can roughly be partitioned into three separate steps: grouping resonance peaks in multiple spectra into spin systems, chaining the resultant spin systems into strings, and assigning these strings 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 with close-to-ideal quality, while only moderately or even poorly on most real protein datasets, where noises as well as data degeneracies occur frequently. We propose in this work to partition the sequential assignment not by physical steps, but only virtual steps, and use their outputs to cross validate each other. The novelty lies in the places, where the ambiguities at the grouping step will be resolved in finding the highly confident strings at the chaining step, and the ambiguities at the chaining step will be resolved by examining the mappings of strings at the assignment step. In this way, all ambiguities at the sequential assignment will be resolved globally and optimally. The resultant assignment program is called Graph-based Approach for Sequential Assignment (GASA), which has been compared to several recent similar developments including PACES, RANDOM, MARS, and RIBRA. The performance comparisons with these works demonstrated that GASA is 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 three dimensional (3D) protein structure determination. Although it has not 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

†To whom correspondence should be addressed.

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 3D structure. Local structural restraint extraction is mostly guided by the backbone resonance sequential assignment, which therefore is crucial to the accurate 3D 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:

(1) grouping resonance peaks from multiple spectra into spin systems,
(2) chaining the resultant spins systems into strings, and
(3) assigning the strings of spin systems to non-overlapping consecutive amino acid residues in the target protein, as illustrated in Fig. 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.[3–14] Furthermore, depending on the NMR spectra data availability, different programs may have different starting points. To name a few automated assignment programs, CBM[6] accepts the strings of spin systems and returns an optimal assignment of them to the non-overlapping peptides in the target protein; A random graph approach[10] (we abbreviate it as RANDOM in the rest of the paper), MARS[11] and GANA[12] assume the availability of spin systems and focus on chaining the spin systems into strings and their subsequent assignment; typically, RANDOM avoids exhaustive enumeration through multiple calls to Hamiltonian path/cycle generation in a randomized way, to fish for high probability strings of spin systems; MARS first searches for all possible strings of length 5 and then uses their mapping positions to identify the correct strings; GANA applies a genetic algorithm to search for an assignment. Other programs, AutoAssign,[4] PACES,[8] CASA,[13] and RIBRA,[14] accept input in spectral peak lists and apply some simple rules to group the peaks into spin systems first. Starting from the formed spin systems, AutoAssign uses a best-first search algorithm with constraint propagation to look for assignments; CASA applies a depth-first ordered tree search algorithm

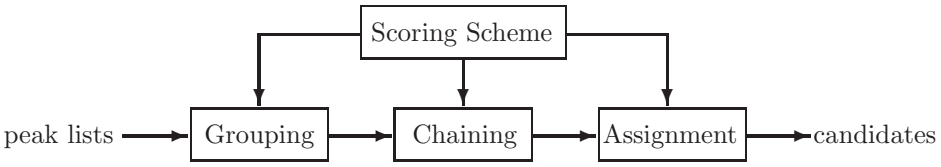


Fig. 1. The flow chart of the NMR resonance sequential assignment.

through defining a set of scores that try to resolve the ambiguities; PACES uses an exhaustive search algorithm to enumerate all possible strings that can be formed and then performs the string assignment; RIBRA applies a weighted maximum independent set algorithm to search for assignments.

The above mentioned sequential assignment programs all work well on 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 have 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 adjacencies between them can be fairly easily determined and the subsequent string assignment also becomes easy. In AutoAssign, PACES, CASA, 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.[4,8,13,14] 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 conditions and the structure complexity. Large tolerance thresholds could create too many ambiguities in resultant spin systems and consequently in the later chaining and assignment steps, leading to a dramatic decreased assignment accuracy; on the other hand, small tolerance thresholds could produce too few spin systems when the spectral data resolution is low, leading to a hardly 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 sufficient information to produce a meaningful output. However, for low resolution spectral data, the ambiguities appearing in the input of one step are very difficult to be resolved internally. Though it is possible to generate multiple outputs for manual adjustment, the inherent uncertainties in the input might cause more ambiguities in the outputs, which are taken as inputs to the downstream steps. Consequently, the whole process would fail to produce a meaningful resonance sequential assignment. However, one meaningful assignment might be possible if the outputs of downstream 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 at 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 the 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 contains the *imperfect* spin systems, in which some ambiguities need to be resolved to produce legal spin systems. In the second phase, the spin system chaining is performed to resolve 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 at the chaining step, and the ambiguities in the chaining step are resolved through examining the mappings of the resulting strings at the assignment step. Therefore, in this sense, 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 Sec. 2, we introduce the detailed steps of operations in GASA. Section 3 presents our experimental results and discussion. We conclude the paper in Sec. 4.

2. The GASA Algorithm

The input data to GASA could be a set of peak lists or, assuming the grouping has been done, a list of spin systems. In the case of a given set of peak lists, GASA first 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 the assignment for resolving the ambiguities in the imperfect spin systems and meanwhile completing the assignment. If the input is a list of spin systems, GASA skips the first phase and directly invokes the second phase to conduct the spin system chaining and the assignment.

2.1. *Phase 1: Peak filtering and grouping*

For ease of exposition and fair comparison with PACES, RANDOM, 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 well-known triple spectra containing HSQC, CBCA(CO)NH, and HNCACB spectra. Nevertheless, we point out that GASA can also accept other combinations of spectra. An HSQC spectrum contains two dimensional (2D) peaks each corresponds to a pair of chemical shifts for an amide proton and the directly attached nitrogen; An HNCACB spectrum contains 3D peaks each 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; A CBCA(CO)NH spectrum contains 3D peaks each 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

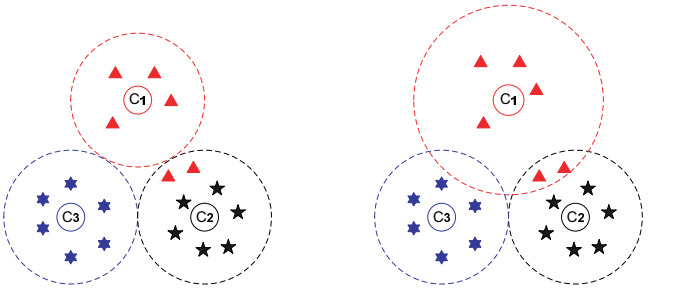
of presentation, a 3D peak containing a chemical shift of the intra-residue carbon is referred to as an *intra-residue peak*; otherwise an *inter-residue peak*. The goal of peak filtering and grouping is to identify all perfect spin systems without asking for the chemical shift tolerance thresholds. Note that the best of our knowledge, all existing peak grouping models require manually defined 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 in the following using the triple spectra as an example. Because of the high quality of the 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 on the 2D HN and N chemical shift plane. Given a center $C = (\mathrm{HN}_C, \mathrm{N}_C)$ and a peak $P = (\mathrm{HN}_P, \mathrm{N}_P, \mathrm{C}_P^{\alpha/\beta})$, the normalized Euclidean distance between them is defined as

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

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 six peaks distributed to it in total: four from the HNCACB spectrum and two from the 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 one center might turn out to be closer to the other centers. Therefore, using a set of common chemical shift tolerance thresholds for all centers would result in more troublesome centers.

Figure 2 illustrates a simple scenario where three centers present, but using the common tolerance thresholds C_1 has only four peaks associated with while C_2 has eight. In Fig. 2, 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 at center C_1 reveals that the two peaks that should belong to it but are closer to center C_2 are among the six closest peaks to center C_1, but they are the 7th and the 8th closest to center C_2. That is, by using the center specific tolerance thresholds, the spin system with center C_1 can be formed by adding these two peaks [see Fig. 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, only one perfect spin system is formed.

(b) Using the center specific tolerance thresholds, three perfect spin systems are formed.

Fig. 2. A sample scenario in the peak filtering and grouping: (a) There are three HSQC peaks as three 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 six associated peaks). (b) Using center specific tolerance thresholds, all three centers 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 the CBCA(CO)NH and HNCACB spectra to their respective closest HSQC peaks. If the HSQC peak and its associated peaks in the CBCA(CO)NH and HNCACB spectra form a perfect spin system (in this case, exactly six peaks), 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 the HSQC spectrum looks for the k closest peaks in the 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 our case of triple spectra, HSQC, HNCACB, and CBCA(CO)NH, the number of peaks in a perfect spin system is six and consequently $k = 9$. 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. It should be 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, users could specify maximal HN and N chemical shift tolerance thresholds to speed up the process, though we have noticed that the minor differences in these maximal chemical shift tolerance thresholds would not really affect the performance of this bidirectional search.

2.2. *Phase 2: Ambiguity resolving, adjacency determination, and string assignment*

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 the string assignment. In general, it is very difficult to distinguish the true peaks from the fake peaks when every imperfect spin system is examined individually. During our development, we have found that in most cases, those spin systems containing true peaks enable more confident string findings than those containing fake peaks. With this observation, we propose to extract true peaks from the imperfect spin systems through the spin system chaining and the resultant string assignment, namely, to only accept the peaks 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.[15] In the connectivity graph, one vertex corresponds to a spin system. Given two perfect spin systems $v_i = (\mathrm{HN}_i, \mathrm{N}_i, \mathrm{C}^{\alpha}_i, \mathrm{C}^{\beta}_i, \mathrm{C}^{\alpha}_{i-1}, \mathrm{C}^{\beta}_{i-1})$ and $v_j = (\mathrm{HN}_j, \mathrm{N}_j, \mathrm{C}^{\alpha}_j, \mathrm{C}^{\beta}_j, \mathrm{C}^{\alpha}_{j-1}, \mathrm{C}^{\beta}_{j-1})$, if both $|\mathrm{C}^{\alpha}_i - \mathrm{C}^{\alpha}_{j-1}| \le \delta_\alpha$ and $|\mathrm{C}^{\beta}_i - \mathrm{C}^{\beta}_{j-1}| \le \delta_\beta$ hold, then there is an edge from v_i to v_j with its weight calculated as

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

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

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

If both $|\mathrm{C}^{\alpha}_{jl} - \mathrm{C}^{\alpha}_{i-1}| \le \delta_\alpha$ and $|\mathrm{C}^{\beta}_{jk} - \mathrm{C}^{\beta}_{i-1}| \le \delta_\beta$ hold, then there is an edge from v'_j to v_i with its weight calculated as

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

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 between two imperfect spin systems.

Once the connectivity graph has been constructed, GASA proceeds essentially the same as CISA[15] 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, or *mapping score*, 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 Wan *et al.*[16] This scoring scheme is essentially a naive Bayesian scheme, which 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 $P(cs \mid aa, ss) = N(cs \mid aa, ss)/N(aa, ss)$, 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 mapping scores of all intra-residue chemical shifts in a spin system gives the spin system its mapping score to every amino acid residue in the target protein.

The edges in the connectivity graph are weighted by Eqs. (1) and (2), 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 string with the best mapping score. Another list, Complete List (CL), is used in the algorithm to save those completed strings, which are not further expandable. 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 string assignment.

2.2.1. *OL initialization*

Let G denote the constructed connectivity graph. GASA first searches for all *unambiguous* edges in G, which are the edges in G whose starting vertex has out-degree 1 and the ending vertex has in-degree 1. We note that such a process is similar to RANDOM and CASA. It then expands these edges into simple paths with a predefined length L by both tracing their starting vertices backward and their ending vertices forward. The tracing process stops when 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. All these paths stored in OL are sorted in the non-increasing

order of their mapping scores. The size of OL is fixed at S, that is, only the top S paths are kept in OL. Note that both L (= 6) and S (= 60) are set in the way to obtain the best trade-off between computing time and performance.

2.2.2. *Path growing*

In this step, GASA tries to bidirectionally expand the top ranked path stored in OL, a process similar to MARS, which enumerates all paths with length 5 and then expands each path in two directions. Note that PACES simply enumerates all possible paths from root vertices (i.e. in-degree 0) without such an expanding process; CASA has a path growing process very similar to ours but its path ranking is done using the number of mapping positions (not the mapping score) for the path. Denote this top ranked 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 (bidirectionally) expand P, and the resultant expanded paths are called the child paths of P. For every 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, if there is no space for it). If none of the child paths of P can make into OL, or P is not expandable in either direction (i.e. there is no edge incident to h, neither edge incident from t), path P is *closed* for further expanding and subsequently is moved into CL. GASA proceeds to consider the top ranked path in OL iteratively and this growing process is terminated when OL becomes empty.

2.2.3. *CL finalizing*

Let P denote the path of the highest mapping score in CL (tie is broken to the longest path, and further tie is broken arbitrarily). GASA performs the following filtering: Firstly, all paths in CL with both their lengths and their mapping scores less than 90% of the length and the score of path P are considered as of low quality compared to path P and thus discarded from further consideration. The remaining paths in CL are considered to be reliable strings. Secondly, only those edges occurring in at least 90% of the paths in CL are regarded as reliable edges. The other edges in the paths in CL are therefore removed, which might break the paths in CL into shorter ones. These resultant paths are final candidate paths to be processed in the next step.

2.2.4. *Ambiguities resolving*

GASA scans through the paths in CL for the longest one, which is taken as the confident string built in the current iteration. Nevertheless, it could be the case that the mapping position of this path in the target protein conflicts the string

mappings achieved in previous iterations. In this case, GASA respects previous string mappings and this current string/path has to be broken down by removing the spin systems that have the conflicting mapping positions. In the extreme case where every spin system in the path has a conflicting mapping position, the path is removed from the connectivity graph (for otherwise, the program would enter an infinite loop). In the other case, consequently, the spin systems that are assigned with mapping positions in this iteration might not necessarily form into a single string, but several shorter ones. Regardless, 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 used to build the spin systems and edges are considered as true peaks, while the others are considered as fake peaks, which are subsequently removed. If the remaining connectivity graph G is still non-empty, GASA proceeds to the next iteration. When GASA 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 experiments to compare with several recent works, including PACES (downloaded through http://152.16.14.71/paces/), RANDOM (source code obtained through its correspondence author), MARS (downloaded through http://www.spincore.com/nmrinfo/MARS.html), and RIBRA (web server http://bio-cluster.iis.sinica.edu.tw/ribra/index.htm). The first experiment is to compare the second phase of GASA, the *ambiguity resolving* phase, with PACES, RANDOM and MARS only. They all 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 (though PACES can accept input as a set of peak lists). 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 second experiment is to compare with RIBRA only by using the simulated datasets in RIBRA, among which each dataset corresponds to one type of spectral data noise/error. The last experiment is used for comparison with RIBRA again, but on simulated datasets that contain all types of data noises and thus are closer to real data. This experiment serves to justify the values of combining the peak grouping, the spin system chaining, and the string assignment all together.

Regarding the performance measurements, RIBRA explicitly defines two criteria, namely *precision* and *recall*. 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.[14] In this paper, we use the same criteria to facilitate the comparison.

3.1. *Experiment 1*

The dataset in Experiment 1 is simulated on the basis of 12 proteins studied in Xu *et al.*,[6] 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 (http://www.bmrb.wisc.edu) 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, except Proline, its four chemical shifts (for Glycine, which has no C^{β} atom, only three) together with C^{α} and C^{β} chemical shifts from the preceding residue formed an initial spin system. We excluded Proline residues in the simulation because in the real NMR data, there wouldn't be spin systems for Prolines since they do not have HN atoms. Next, for each initial spin system, chemical shifts for intra-residue C^{α} and C^{β} were perturbed by adding random 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 are $\delta_{\alpha} = 0.2$ ppm and $\delta_{\beta} = 0.4$ ppm, respectively.[4,8,10,11] Subsequently, the standard deviations of the random error 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 C^{α} and C^{β} tolerance thresholds (that is, $\delta_{\alpha} = 0.4$ ppm and $\delta_{\beta} = 0.8$ ppm). They are also summarized in Table 1 (the right half, having suffix 2). Obviously, Table 1 shows 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 the both sets of instances are collected in Table 2. Their assignment precision and recall on the two sets are also plotted in Figures 3 and 4. In summary, RANDOM achieved on average 50% assignment precision and recall. (We followed the exact way of determining precision and recall as described in Bailey-Kellogg *et al.*,[10] where 1000 iterations for each instance have been run.), which is roughly the same as that claimed in its original paper.[10] PACES performed better than RANDOM, but it failed on seven instances where the connectivity graphs were too complex (computer memory ran out). The collected results for PACES

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

	$\delta_\alpha = 0.2$ ppm, $\delta_\beta = 0.4$ ppm				$\delta_\alpha = 0.4$ ppm, $\delta_\beta = 0.8$ ppm			
L	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	71	20	1.26	bmr4144.2	71	77	2.06
86	bmr4579.1	82	81	1.96	bmr4579.2	82	219	3.58
89	bmr4316.1	84	118	2.61	bmr4316.2	84	309	4.62
105	bmr4288.1	93	25	1.26	bmr4288.2	93	89	1.94
112	bmr4670.1	101	24	1.12	bmr4670.2	101	100	1.79
114	bmr4929.1	109	34	1.30	bmr4929.2	109	117	2.05
115	bmr4302.1	107	18	1.16	bmr4302.2	107	87	1.80
116	bmr4353.1	97	30	1.30	bmr4353.2	97	106	2.07
158	bmr4027.1	147	71	1.48	bmr4027.2	147	252	2.70
215	bmr4318.1	190	157	1.82	bmr4318.2	190	553	3.90

on these seven instances were obtained through manually reducing the tolerance thresholds to remove a significant portion of edges from the connectivity graphs. We implemented the scheme that if PACES did not finish an instance in 8 h, 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.[8] There are at least three possible reasons for this: (1) The datasets tested in Coggins and Zhou[8] are different from ours (which, unfortunately, were unavailable to us). We have done a test on re-simulating the datasets in Coggins and Zhou,[8] according to its description, to compare PACES, RANDOM and MARS with CISA,[15] a predecessor of GASA, and the result tendency is very much the same as what we have seen here. (2) PACES is only semi-automated, in the sense that it needs manual adjustment after each 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 with manual adjustment for improved assignment. However, in the current work we were unable to do this fairly and subsequently we decided not to do so. (3) PACES is designed to take in better quality spin systems that contain in addition the carbonyl CO chemical shifts. On the current combination without the CO chemical shifts, PACES was expected to perform a bit lower, since the extra CO chemical shifts will provide extra information for resolving the adjacency ambiguities. Again, we have done a similar test on using the combination with the CO chemical shifts to compare RANDOM, PACES, and MARS with CISA,[15] and the result tendency is very much the same as what we have seen here. 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

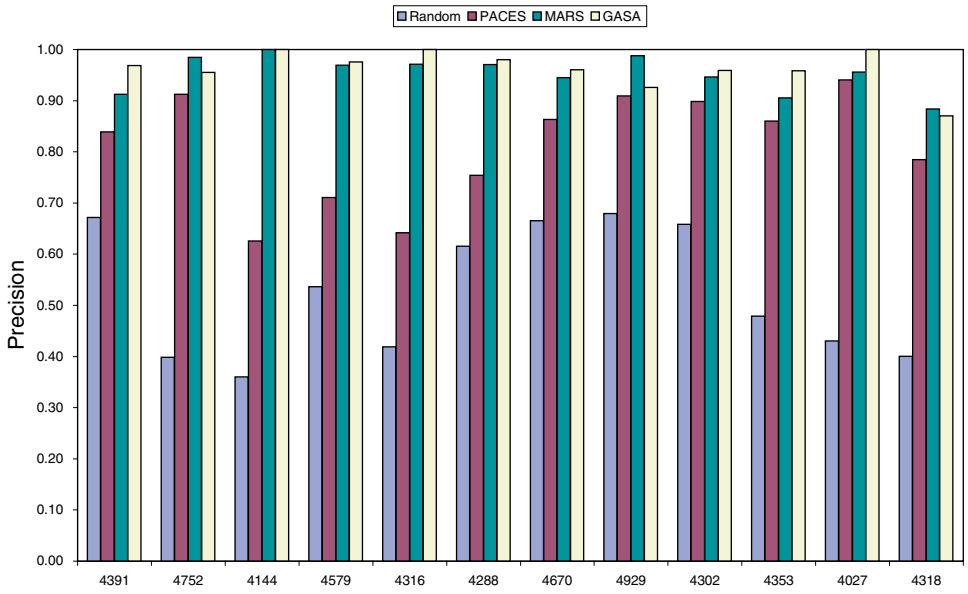
Table 2. Assignment precision (PR) and recall (RE) of RANDOM, PACES, MARS and GASA in the first experiment.

		RANDOM		PACES		MARS		GASA	
L	InstanceID	PR	RE	PR	RE	PR	RE	PR	RE
$\delta_\alpha = 0.2$ ppm, $\delta_\beta = 0.4$ ppm									
66	bmr4391.1	0.67	0.63	0.84	0.72	0.91	0.87	0.97	0.97
68	bmr4752.1	0.40	0.35	0.91	0.79	0.98	0.97	0.96	0.94
78	bmr4144.1	0.36	0.33	0.63	0.53	1.00	0.97	1.00	0.99
86	bmr4579.1	0.54	0.51	0.71*	0.62*	0.97	0.91	0.98	0.98
89	bmr4316.1	0.42	0.36	0.64*	0.40*	0.97	0.96	1.00	0.99
105	bmr4288.1	0.62	0.55	0.75	0.71	0.97	0.95	0.98	0.98
112	bmr4670.1	0.67	0.62	0.86	0.77	0.94	0.88	0.96	0.95
114	bmr4929.1	0.68	0.63	0.91	0.86	0.99	0.97	0.93	0.91
115	bmr4302.1	0.66	0.64	0.90	0.73	0.95	0.92	0.96	0.95
116	bmr4353.1	0.48	0.43	0.86	0.79	0.91	0.85	0.96	0.95
158	bmr4027.1	0.43	0.32	0.94	0.82	0.96	0.93	1.00	0.99
215	bmr4318.1	0.40	0.38	0.78*	0.54*	0.88	0.81	0.87	0.84
Avg.		0.53	0.48	0.81	0.69	0.95	0.90	0.96	0.95
$\delta_\alpha = 0.4$ ppm, $\delta_\beta = 0.8$ ppm									
66	bmr4391.2	0.58	0.55	0.82	0.69	0.86	0.85	0.91	0.91
68	bmr4752.2	0.36	0.30	0.86‡	0.74‡	0.91	0.90	0.90	0.88
78	bmr4144.2	0.33	0.31	0.45	0.38	1.00	0.97	1.00	0.99
86	bmr4579.2	0.34	0.32	0.51‡	0.43‡	0.79	0.75	0.80	0.80
89	bmr4316.2	0.35	0.30	0.29‡	0.18†	0.95	0.92	0.83	0.83
105	bmr4288.2	0.42	0.38	0.58	0.53	0.95	0.93	0.91	0.91
112	bmr4670.2	0.43	0.39	0.63	0.57	0.83	0.81	0.88	0.87
114	bmr4929.2	0.46	0.43	0.81	0.77	0.99	0.97	0.96	0.94
115	bmr4302.2	0.47	0.45	0.63	0.49	0.82	0.80	0.91	0.91
116	bmr4353.2	0.47	0.43	0.64	0.61	0.83	0.80	0.90	0.90
158	bmr4027.2	0.40	0.30	0.38	0.32	0.82	0.81	0.88	0.85
215	bmr4318.2	0.25	0.22	0.76‡	0.45‡	0.84	0.75	0.74	0.70
Avg.		0.41	0.37	0.61	0.51	0.88	0.85	0.88	0.87

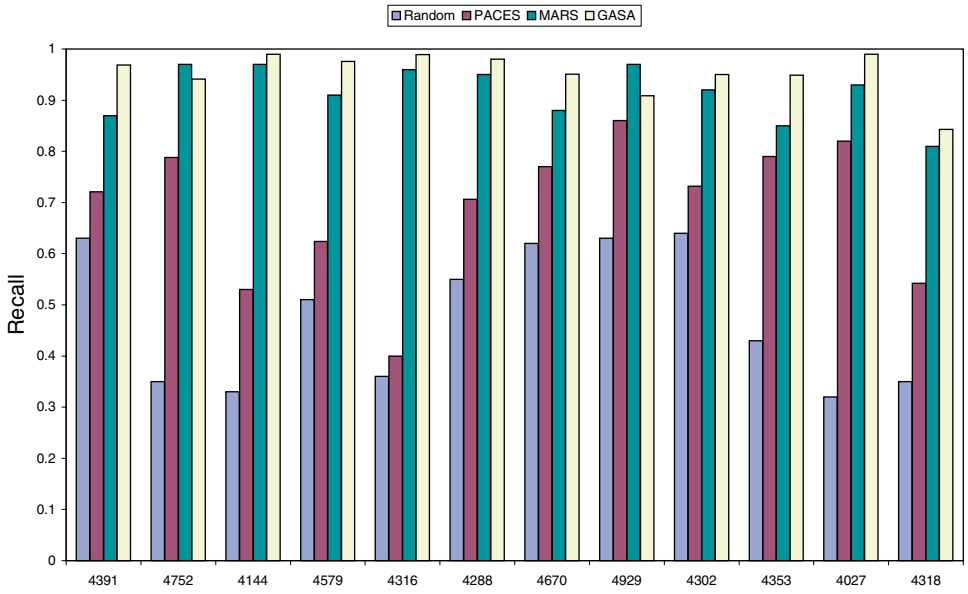
*PACES performance on these 3 datasets were obtained by reducing tolerance thresholds to $\delta_\alpha = 0.15$ ppm and $\delta_\beta = 0.3$ ppm (75%).
†PACES performance on this dataset was obtained by reducing tolerance thresholds to $\delta_\alpha = 0.3$ ppm and $\delta_\beta = 0.6$ ppm (75%).
‡PACES performance on these 3 datasets were obtained by reducing tolerance thresholds to $\delta_\alpha = 0.2$ ppm and $\delta_\beta = 0.4$ ppm (50%).

instances, which indicated that combining the spin system chaining and assignment together does more effectively resolve the adjacency ambiguities and make better assignments. Regarding the running time, RANDOM, MARS, and GASA all finished within 20 minutes (on a P4 1.8 GHz desktop) on each instance, and PACES finished within an hour on most instances but could take hours on several hard instances.

On the first set of 12 instances, the analysis of variance (ANOVA) test showed that, with $p = 1.1 \times 10^{-4}$ for precision and $p = 2.6 \times 10^{-6}$ for recall, GASA performed significantly better than PACES; with $p = 8.2 \times 10^{-11}$ for precision and

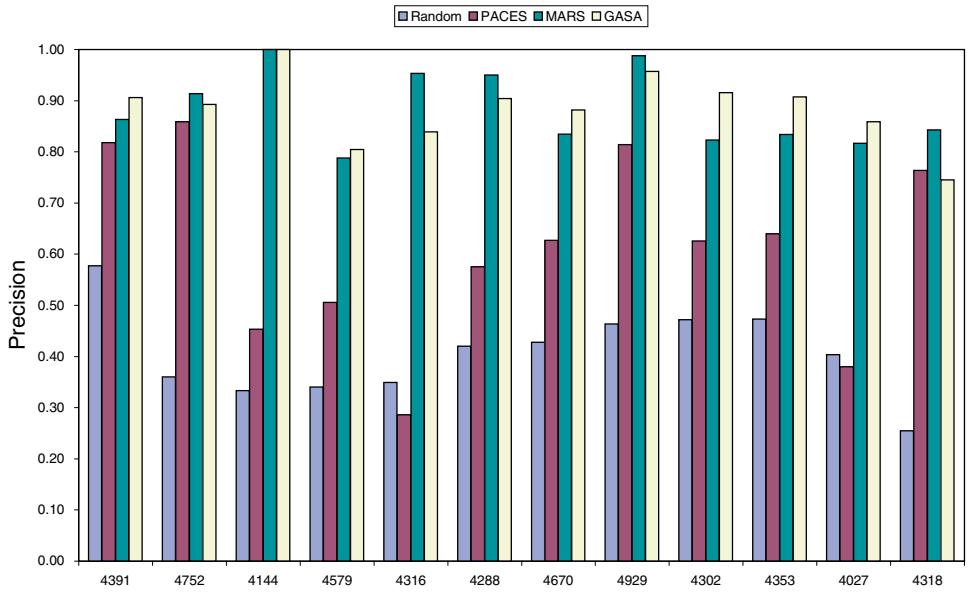


(a) Assignment precision on the 1st set of instances.

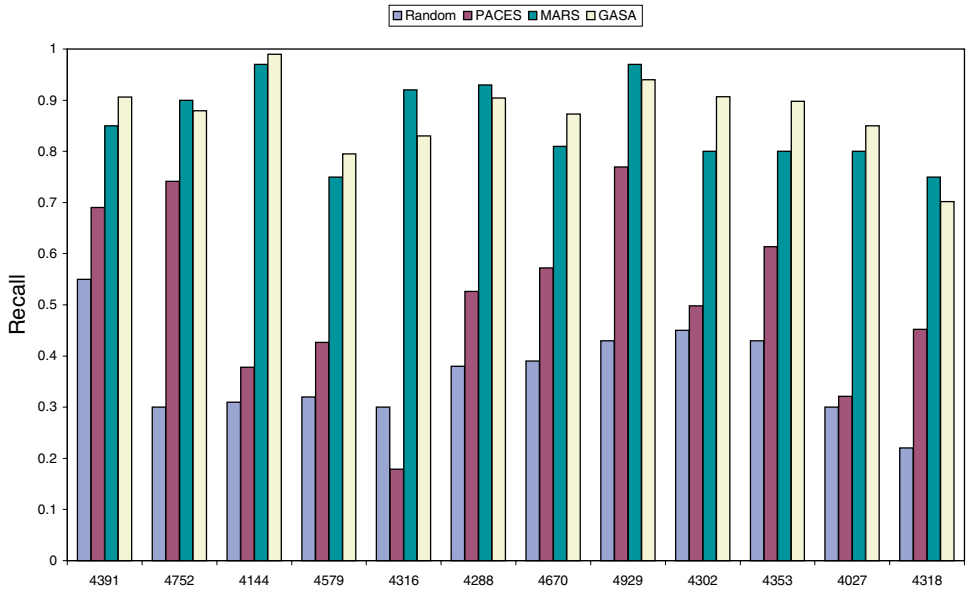


(b) Assignment recall on the 1st set of instances.

Fig. 3. Plots of assignment precision and recall for RANDOM, PACES, MARS, and GASA on the first set of instances with normal tolerance thresholds, using C^{α} and C^{β} chemical shifts for connectivity inference.



(a) Assignment precision on the 2nd set of instances.



(b) Assignment recall on the 2nd set of instances.

Fig. 4. Plots of assignment precision and recall for RANDOM, PACES, MARS, and GASA on the second set of instances with doubled tolerance thresholds, using C^{α} and C^{β} chemical shifts for connectivity inference.

$p = 4.5 \times 10^{-11}$ for recall, GASA performed significantly better than RANDOM too; On the second set of instances, with $p = 7.7 \times 10^{-5}$ for precision and $p = 1.4 \times 10^{-6}$ for recall, GASA performed significantly better than PACES as well; with $p = 4.6 \times 10^{-13}$ for precision and $p = 4.3 \times 10^{-13}$ for recall, GASA also performed significantly better than RANDOM.

3.2. *Experiment 2*

In RIBRA, five 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 data errors, and the other four datasets contain four different types of spectral data errors respectively. The *false positive* dataset is generated by respectively, adding 5% fake peaks into the perfect CBCA(CO)NH and HNCACB peak lists. The *false negative* dataset is generated by randomly removing a small portion of inter-residue peaks from the 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 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-residue peaks in the perfect HNCACB peak list. Note that each of these four datasets contains only one type of spectral data error/noise. The chemical shift perturbations are done in the same way as we did in Experiment 1, by adding to the original chemical shifts random errors that follow independent normal distributions with 0 means and constant standard deviations ($\sigma_{\mathrm{HN}} = 0.06/2.5 = 0.0024$ ppm, $\sigma_{\mathrm{N}} = 0.8/2.5 = 0.32$ ppm, $\sigma_{\alpha} = 0.2/2.5 = 0.08$ ppm, and $\sigma_{\beta} = 0.4/2.5 = 0.16$ ppm).

Table 3 collects the average performance precision and recall of RIBRA and GASA on these five datasets. As shown, there is no significant difference among the performances on the *perfect*, *false positive* and *linking error* datasets. GASA shows more robustness on the dataset with missing data while RIBRA performs better on the *grouping error* dataset. Through the detailed study, we found that all these five simulated datasets by RIBRA contain the inter-residue and intra-residue peaks with 0 C^{β} chemical shifts, simulated for Glycine residues. That is, in the

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.

	RIBRA		GASA	
Dataset	PR	RE	PR	RE
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%

RIBRA simulation, Glycine residues would have two inter-residue peaks and two intra-residue peaks in the HNCACB spectrum and the amino acid residues right after Glycine residues would have two inter-residue peaks in the CBCA(CO)NH spectrum. However, this is impossible in the real NMR spectral data. In real NMR spectral data, a huge amount of ambiguity in the sequential assignment results from Glycine residues because they correspond to various legal combinations in grouping stage which make the identification of perfect spin systems more difficult. 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 the spin systems with missing peaks. In the RIBRA simulation, therefore, grouping is considerably easier on the datasets with the "aid" of simulated 0 chemical shift values for the artificial C^{β} atom in Glycine residues. 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 was not as good as RIBRA. To verify our conjecture, we randomly selected 14 instances from the RIBRA *grouping error* dataset, with length ranging from 69 to 186, and removed all the peaks of 0 carbon chemical shift. Both RIBRA and GASA were tested on these revised instances. RIBRA achieved 87.7% precision and 72.7% recall, and GASA achieved 88.5% precision and 79.4% recall, slightly better than RIBRA. 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-residue peaks in the CBCA(CO)NH peak list, that is, no other type of data error/noise. 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 (http://bmrb.wisc.edu/) have been manually adjusted across multiple spectra. Even though HSQC is a very reliable experiment, the deposited HN and N chemical shifts in BioMagResBank are still slightly different from the measured values in the real HSQC spectra. 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 original data.

3.3. *Experiment 3*

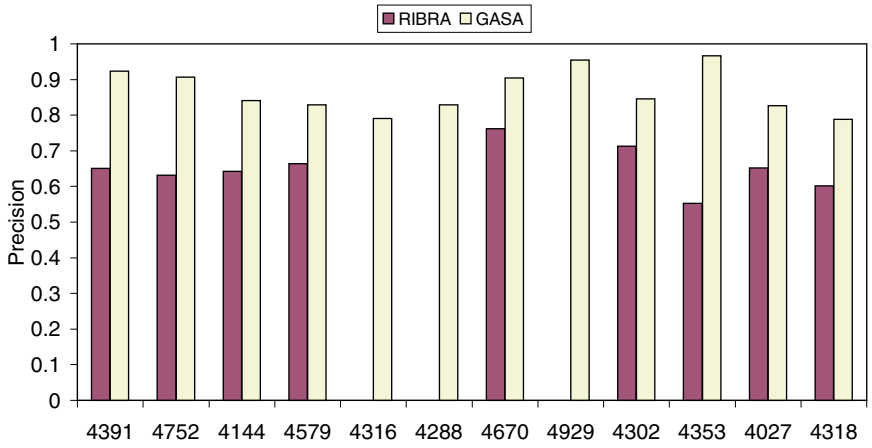
The purpose of Experiment 3 is to provide more convincing comparison results between GASA and RIBRA, based on the data simulation scheme closer to the real NMR data. For this purpose, we used the same 12 proteins in Experiment 1 and the simulation is detailed as follows. For each of these 12 proteins, 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 the HSQC peak list; its HN and N chemical shifts with C^{α} and C^{β} chemical shifts from the preceding residue formed two inter-residue peaks respectively in the 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-residue peaks and two inter-residue peaks respectively in the HNCACB peak list. Note that there is no C^β peak for Glycine in either the CBCA(CO)NH or the HNCACB peak list. Next, for each peak in the HSQC, CBCA(CO)NH, and HNCACB peak lists, the contained HN, N, C^α or C^β chemical shifts were perturbed by adding random errors that follow independent normal distributions with 0 means and constant standard deviations. We chose the same tolerance thresholds as those were used in RIBRA, which were $\delta_{\mathrm{HN}} = 0.06\,\mathrm{ppm}$, $\delta_{\mathrm{N}} = 0.8\,\mathrm{ppm}$, $\delta_\alpha = 0.2\,\mathrm{ppm}$, and $\delta_\beta = 0.4\,\mathrm{ppm}$, respectively. Subsequently, the standard deviations of the normal distributions were set to $\sigma_{\mathrm{HN}} = 0.06/2.5 = 0.0024\,\mathrm{ppm}$, $\sigma_{\mathrm{N}} = 0.8/2.5 = 0.32\,\mathrm{ppm}$, $\sigma_\alpha = 0.2/2.5 = 0.08\,\mathrm{ppm}$, and $\sigma_\beta = 0.4/2.5 = 0.16\,\mathrm{ppm}$, respectively.

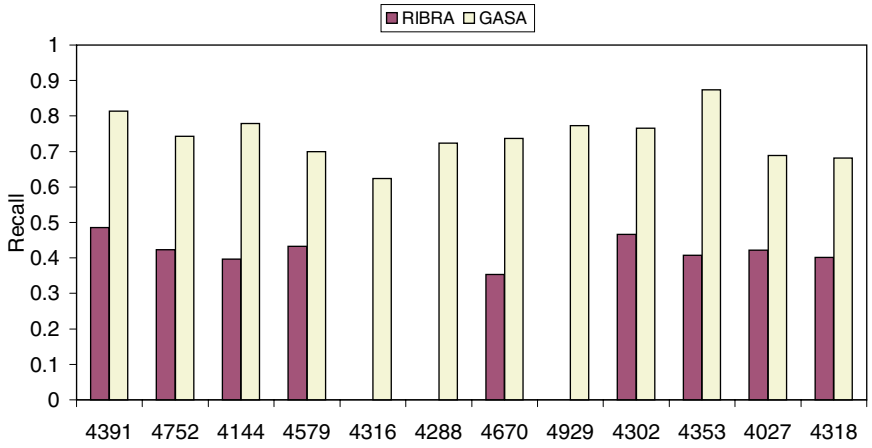
Partial information of and the performances of RIBRA and GASA on these 12 proteins are summarized in Table 4. The detailed dataset is available through the link http://www.cs.ualberta.ca/ ghlin/src/WebTools/gasa.php. From the table, we can see that GASA formed many more spin systems than RIBRA did on every instance, and from the high 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%, ANOVA $p = 3.0 \times 10^{-4}$; recall 74.18% versus 42.10%, ANOVA $p = 3.1 \times 10^{-7}$). The detailed precision and recall are also plotted in Figure 5. 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 achieved 65.23% precision and 42.1% recall on average, which are noticeably worse than what it is claimed

Table 4. Partial information of and the performance precision (PR) and recall (RE) of RIBRA and GASA on the 12 protein NMR datasets in experiment 3. "L" denotes the length of a protein, measured by the number of amino acid residues therein; "M" records the number of true spin systems that are not simulated in the dataset, including those for Prolines; "G" records the number of spin systems that were actually formed by RIBRA and GASA, respectively.

BMRB			RIBRA			GASA		
Entry	L	M	G	PR	RE	G	PR	RE
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%



(a) Assignment precision.



(b) Assignment recall.

Fig. 5. Plots of detailed assignment (a) precision and (b) recall on each of the 12 protein datasets in Experiment 3 by RIBRA and GASA.

in Wu *et al.*[14] The possible explanations for RIBRA not doing well on these 12 instances are: (1) The simulation procedure in Experiment 3 did not generate C^{β} peaks with 0 chemical shift for Glycines, which causes more ambiguities in the peak grouping, and the subsequent spin system chaining. (2) In the 12 simulated instances 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. Regarding the running time, GASA finished within minutes on each instances, while the RIBRA web server generally returned an assignment in an hour, and it could take several hours on hard instances.

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 GASA can be raw spectral peak lists or already formed spin systems. 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 PACES, RANDOM, MARS, and RIBRA showed that GASA was more effective in dealing with the NMR spectral data degeneracy and thereby provides a more promising solution to automated resonance sequential assignment.

As a byproduct, we have also proposed a spectral dataset simulation scheme that generates datasets closer to the reality. One of our future works is to formalize this simulation scheme to produce a large number of protein NMR instances for common comparative study purposes. 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 dataset in the literature that can be used for comparative studies of assignment programs from different laboratories. As a preliminary effort, these 12 simulated protein NMR instances in Experiment 3, in the form of the well-known triple spectra HSQC, HNCACB and CBCA(CO)NH, are available through the link http://www.cs.ualberta.ca/ ghlin/src/WebTools/gasa.php.

Acknowledgments

This research is supported in part by AICML, CFI, NSERC, Alberta Prion Research Institute (APRI) and PrioNet Canada. The authors would like to thank the authors of RIBRA for providing access to their datasets and for their prompt responses to our inquiries. The authors are grateful to all five reviewers (three for the LSS CSB 2006 submission), who provided many helpful comments and suggestions that improve the presentation.

References

1. Ferentz AE, Wagner G, NMR spectroscopy: a multifaceted approach to macromolecular structure, *Quarterly Rev Biophys* **33**:29–65, 2000.
2. Williamson MP, Havel TF, Wüthrich K, Solution conformation and proteinase inhibitor IIA from bull seminal plasma by proton NMR and distance geometry, *J Mol Biol* **182**:295–315, 1985.
3. Bartels C, Güntert P, Billeter M, Wüthrich K, GARANT — A general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra, *J Comput Chem* **18**:139–149, 1997.
4. Zimmerman DE, Kulikowski CA, Huang Y, Tashiro WFM, Shimotakahara S, Chien C, Powers R, Montelione GT, Automated analysis of protein NMR assignments using methods from artificial intelligence, *J Mol Biol* **269**:592–610, 1997.

5. Güntert P, Salzmann M, Braun D, Wüthrich K, Sequence-specific NMR assignment of proteins by global fragment mapping with the program Mapper, *J Biomol NMR* **18**:129–137, 2000.
6. Xu Y, Xu D, Kim D, Olman V, Razumovskaya J, Jiang T, Automated assignment of backbone NMR peaks using constrained bipartite matching, *IEEE Computing in Science & Engineering* **4**:50–62, 2002.
7. Lin G-H, Xu D, Chen ZZ, Jiang T, Wen JJ, Xu Y, Computational assignment of protein backbone NMR peaks by efficient bounding and filtering, *J Bioinfor Comput Biol* **1**:387–409, 2003.
8. Coggins BE, Zhou P, PACES: Protein sequential assignment by computer-assisted exhaustive search, *J Biomol NMR* **26**:93–111, 2003.
9. Hitchens TK, Lukin JA, Zhan Y, McCallum SA, Rule GS, MONTE: An automated Monte Carlo based approach to nuclear magnetic resonance assignment of proteins, *J Biomol NMR* **25**:1–9, 2003.
10. Bailey-Kellogg C, Chainraj S, Pandurangan G, A random graph approach to NMR sequential assignment, in *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB 2004)*, pp. 58–67, 2004.
11. Jung Y-S, Zweckstetter M, Mars — robust automatic backbone assignment of proteins, *J Biomol NMR* **30**:11–23, 2004.
12. Lin H-N, Wu K-P, Chang J-M, Sung T-Y, Hsu W-L, GANA — a genetic algorithm for NMR backbone resonance assignment, *Nucleic Acids Res* **33**:4593–4601, 2005.
13. Wang J, Wang T, Zuiderweg ERP, Crippen GM, CASA: An efficient automated assignment of protein mainchain NMR data using an ordered tree search algorithm, *J Biomol NMR* **33**:261–279, 2005.
14. Wu K-P, Chang J-M, Chen J-B, Chang C-F, Wu W-J, Huang T-H, Sung T-Y, Hsu W-L, RIBRA — an error-tolerant algorithm for the NMR backbone assignment problem, *J Computat Biol* **13**:229–244, 2006.
15. Wan X, Lin G-H, CISA: Combined NMR resonance connectivity information determination and sequential assignment, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2007 (in press).
16. Wan X, Tegos T, Lin G-H, Histogram-based scoring schemes for protein NMR resonance assignment, *J Bioinform Comput Biol* **2**:747–764, 2004.