

1 Journal of Bioinformatics and Computational Biology
 2 Vol. 5, No. 2 (2007) 1–21
 3 © Imperial College Press



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

7 XIANG WAN* and GUOHUI LIN†

8 *Department of Computing Science, University of Alberta*
 9 *Edmonton, Alberta T6G 2E8, Canada*

10 *xiangwan@cs.ualberta.ca

†ghlin@cs.ualberta.ca

11 Received 30 August 2006

12
 13 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.

14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31 *Keywords:* Protein NMR backbone resonance sequential assignment; chemical shift; spin system; connectivity graph.

32
 33 **1. Introduction**

34 Nuclear Magnetic Resonance (NMR) spectroscopy has been increasingly used for
 35 three dimensional (3D) protein structure determination. Although it has not been
 36 able to achieve the same accuracy as X-ray crystallography, enormous technological
 37 advances have brought NMR to the forefront of structural biology¹ since the publication of the first complete solution structure of a protein (bull seminal trypsin

†To whom correspondence should be addressed.

2 *X. Wan & G. Lin*

1 inhibitor) determined by NMR in 1985.² The underlined mathematical principle for
 2 protein NMR structure determination is to employ NMR spectroscopy to obtain
 3 local structural restraints such as the distances between hydrogen atoms and the
 4 ranges of dihedral angles, and then to calculate the 3D structure. Local structural
 5 restraint extraction is mostly guided by the backbone resonance sequential assign-
 6 ment, which therefore is crucial to the accurate 3D structure calculation. The reso-
 7 nance sequential assignment is to map the identified resonance peaks from multiple
 8 NMR spectra to their corresponding nuclei in the target protein, where every peak
 9 captures a nuclear magnetic interaction among a set of nuclei and its coordinates
 10 are the chemical shift values of the interacting nuclei. Normally, such an assignment
 11 procedure is roughly partitioned into three main steps:

- 12 (1) grouping resonance peaks from multiple spectra into spin systems,
- 13 (2) chaining the resultant spins systems into strings, and
- 14 (3) assigning the strings of spin systems to non-overlapping consecutive amino acid
 15 residues in the target protein, as illustrated in Fig. 1, where the scoring scheme
 16 quantifies the residual signature information of the peaks and spin systems.

17 Separately dealing with these three steps has been adopted in many existing
 18 assignment programs.³⁻¹⁴ Furthermore, depending on the NMR spectra data avail-
 19 ability, different programs may have different starting points. To name a few auto-
 20 mated assignment programs, CBM⁶ accepts the strings of spin systems and returns
 21 an optimal assignment of them to the non-overlapping peptides in the target pro-
 22 tein; A random graph approach¹⁰ (we abbreviate it as RANDOM in the rest of the
 23 paper), MARS¹¹ and GANA¹² assume the availability of spin systems and focus on
 24 chaining the spin systems into strings and their subsequent assignment; typically,
 25 RANDOM avoids exhaustive enumeration through multiple calls to Hamiltonian
 26 path/cycle generation in a randomized way, to fish for high probability strings of
 27 spin systems; MARS first searches for all possible strings of length 5 and then uses
 28 their mapping positions to identify the correct strings; GANA applies a genetic
 29 algorithm to search for an assignment. Other programs, AutoAssign,⁴ PACES,⁸
 30 CASA,¹³ and RIBRA,¹⁴ accept input in spectral peak lists and apply some sim-
 31 ple rules to group the peaks into spin systems first. Starting from the formed spin
 32 systems, AutoAssign uses a best-first search algorithm with constraint propagation
 33 to look for assignments; CASA applies a depth-first ordered tree search algorithm

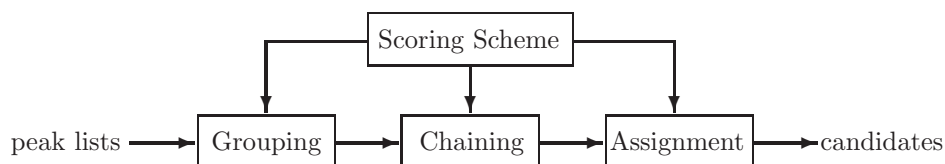


Fig. 1. The flow chart of the NMR resonance sequential assignment.

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

5 The above mentioned sequential assignment programs all work well on high
6 quality NMR data, but most of them remain unsatisfactory in practice and even
7 fail when the spectral data is of low resolution. Through a thorough investigation,
8 we have identified that the bottleneck of automated sequential assignment is reso-
9 nance peak grouping. Essentially, a good grouping output gives well organized high
10 quality spin systems, for which the adjacencies between them can be fairly easily
11 determined and the subsequent string assignment also becomes easy. In AutoAssign,
12 PACES, CASA, and RIBRA, the grouping is done through a binary decision model
13 that considers the HSQC peaks as anchor peaks and subsequently maps the peaks
14 from other spectra to these anchor peaks.^{4,8,13,14} For such a mapping, the HN and
15 N chemical shift values in the other peaks are required to fall within the pre-specified
16 HN and N chemical shift tolerance thresholds of the anchor peaks. However, this
17 binary-decision model in the peak grouping inevitably suffers from its sensitivity to
18 the tolerance thresholds. In practice, from one protein dataset to another, chemical
19 shift thresholds vary due to the experimental conditions and the structure com-
20 plexity. Large tolerance thresholds could create too many ambiguities in resultant
21 spin systems and consequently in the later chaining and assignment steps, leading
22 to a dramatic decreased assignment accuracy; on the other hand, small tolerance
23 thresholds could produce too few spin systems when the spectral data resolution is
low, leading to a hardly useful assignment.

25 Secondly, we found that in the traditional three-step procedure, which is the
26 basis of many automated sequential assignment programs, each step is separately
27 executed, without consideration of inter-step effects. Basically, the input to each
28 step is assumed to contain sufficient information to produce a meaningful output.
29 However, for low resolution spectral data, the ambiguities appearing in the input
30 of one step are very difficult to be resolved internally. Though it is possible to
31 generate multiple outputs for manual adjustment, the inherent uncertainties in the
32 input might cause more ambiguities in the outputs, which are taken as inputs to
33 the downstream steps. Consequently, the whole process would fail to produce a
34 meaningful resonance sequential assignment. However, one meaningful assignment
35 might be possible if the outputs of downstream steps are used to validate the input
to the current step.

37 In this paper, we propose a *two-phase Graph-based Approach for Sequential*
38 *Assignment* (GASA) that uses the spin system chaining results to validate the
39 peak grouping and uses the string assignment results to validate the spin system
40 chaining. Therefore, GASA not only addresses the chemical shift tolerance threshold
41 issue at the grouping step but also presents a new model to automate the sequential
42 assignment. In more details, we propose a two-way nearest neighbor search approach
43 in the first phase to eliminate the requirement of user-specified HN and N chemical

4 *X. Wan & G. Lin*

1 shift tolerance thresholds. The output of the first phase consists of two lists of
2 spin systems. One list contains the *perfect* spin systems, which are regarded as
3 of high quality, and the other contains the *imperfect* spin systems, in which some
4 ambiguities need to be resolved to produce legal spin systems. In the second phase,
5 the spin system chaining is performed to resolve the ambiguities contained in the
6 imperfect spin systems and the string assignment step is included as a subroutine to
7 identify the confident strings. In other words, the ambiguities in the imperfect spin
8 systems are resolved through finding the highly confident strings at the chaining
9 step, and the ambiguities in the chaining step are resolved through examining the
10 mappings of the resulting strings at the assignment step. Therefore, in this sense,
11 GASA does not separate the sequential assignment into physical steps but only
12 virtual steps, and all ambiguities in the whole assignment process are resolved
13 globally and optimally.

14 The rest of the paper is organized as follows: In Sec. 2, we introduce the detailed
15 steps of operations in GASA. Section 3 presents our experimental results and dis-
16 cussion. We conclude the paper in Sec. 4.

17 **2. The GASA Algorithm**

18 The input data to GASA could be a set of peak lists or, assuming the grouping
19 has been done, a list of spin systems. In the case of a given set of peak lists,
20 GASA first conducts a bidirectional nearest neighbor search to generate the perfect
21 spin systems and the imperfect spin systems with ambiguities. It then invokes the
22 second phase, which applies a heuristic search guided by the quality of the string
23 mapping to the target protein, to perform the chaining and the assignment for
24 resolving the ambiguities in the imperfect spin systems and meanwhile completing
25 the assignment. If the input is a list of spin systems, GASA skips the first phase
26 and directly invokes the second phase to conduct the spin system chaining and the
27 assignment.

28 **2.1. Phase 1: Peak filtering and grouping**

29 For ease of exposition and fair comparison with PACES, RANDOM, MARS, and
30 RIBRA, we assume the availability of spectral peaks containing chemical shifts for
31 C^α and C^β , and the HSQC peak list. One typical example would be the well-known
32 triple spectra containing HSQC, CBCA(CO)NH, and HNCACB spectra. Never-
33 theless, we point out that GASA can also accept other combinations of spectra.
34 An HSQC spectrum contains two dimensional (2D) peaks each corresponds to a
35 pair of chemical shifts for an amide proton and the directly attached nitrogen;
36 An HNCACB spectrum contains 3D peaks each is a triple of chemical shifts for
37 a nitrogen, the directly adjacent amide proton, and a carbon alpha/beta from the
38 same or the preceding amino acid residue; A CBCA(CO)NH spectrum contains 3D
39 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

1 of presentation, a 3D peak containing a chemical shift of the intra-residue carbon
 2 is referred to as an *intra-residue peak*; otherwise an *inter-residue peak*. The goal of
 3 peak filtering and grouping is to identify all perfect spin systems without asking
 4 for the chemical shift tolerance thresholds. Note that the best of our knowledge,
 5 all existing peak grouping models require manually defined chemical shift tolerance
 6 thresholds in order to decide whether two resonance peaks should be grouped into
 7 the same spin system or not. Consequently, different tolerance thresholds clearly
 8 produce different sets of possible spin systems, and for the low resolution spec-
 9 tral data, a minor change of tolerance thresholds would lead to huge difference
 10 in the formed spin systems and subsequently the final sequential assignment. In
 11 fact, the proper tolerance thresholds are normally dataset dependent and how to
 12 choose them is a very challenging issue in the automated resonance assignment. We
 13 propose to use the nearest neighbor approach, detailed in the following using the
 14 triple spectra as an example. Because of the high quality of the HSQC spectrum,
 15 the peaks in HSQC are considered as centers, and every peak in CBCA(CO)NH
 16 and HNCACB is distributed to the closest center using the normalized Euclidean
 17 distance on the 2D HN and N chemical shift plane. Given a center $C = (HN_C, N_C)$
 18 and a peak $P = (HN_P, N_P, C_P^{\alpha/\beta})$, the normalized Euclidean distance between them
 19 is defined as

$$D = \sqrt{\left(\frac{HN_P - HN_C}{\sigma_{HN}}\right)^2 + \left(\frac{N_P - N_C}{\sigma_N}\right)^2},$$

21 where σ_{HN} and σ_N are the standard deviations of HN and N chemical shifts that
 22 are collected from BioMagResBank (<http://www.bmr.b.wisc.edu>).

23 In the ideal case, each center should have six peaks distributed to it in total: four
 24 from the HNCACB spectrum and two from the CBCA(CO)NH spectrum. However,
 25 due to the chemical shift degeneracy, some centers may have less than 6 or even 0
 26 peaks. The reasons for this is that the peaks should be associated with one center
 27 might turn out to be closer to the other centers. Therefore, using a set of common
 28 chemical shift tolerance thresholds for all centers would result in more troublesome
 29 centers.

30 Figure 2 illustrates a simple scenario where three centers present, but using the
 31 common tolerance thresholds C_1 has only four peaks associated with while C_2 has
 32 eight. In Fig. 2, using the common tolerance thresholds, only one perfect spin system
 33 with center C_3 is formed, because the two peaks that should belong to center C_1 are
 34 closer to center C_2 , which create ambiguities in both spin systems. Nevertheless, a
 35 closer look at center C_1 reveals that the two peaks that should belong to it but are
 36 closer to center C_2 are among the six closest peaks to center C_1 , but they are the
 37 7th and the 8th closest to center C_2 . That is, by using the center specific tolerance
 38 thresholds, the spin system with center C_1 can be formed by adding these two
 39 peaks [see Fig. 2(b)]; similarly, using the center specific tolerance thresholds, the
 spin system with center C_2 becomes another perfect spin system.

6 X. Wan & G. Lin

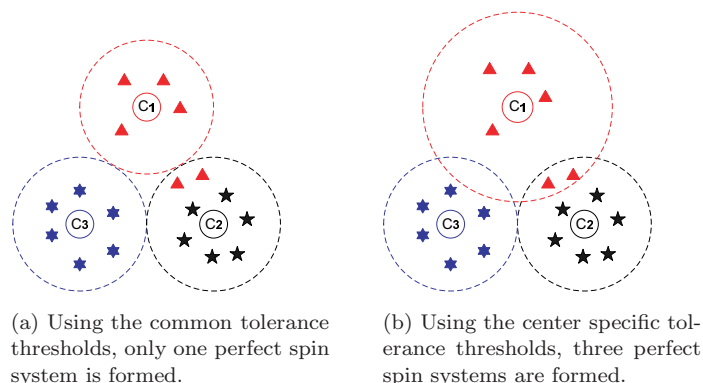


Fig. 2. A sample scenario in the peak filtering and grouping: (a) There are three HSQC peaks as three centers C_1, C_2, C_3 . Every peak is distributed to the closest center, measured by the normalized Euclidean distance. Using the common tolerance thresholds, only C_3 forms a perfect spin system (with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

1 We designed a bidirectional nearest neighbor model, which essentially applies
 2 the center specific tolerance thresholds, to have two steps of operations: *residing*
 3 and *inviting*. In the residing step, we associated each peak in the CBCA(CO)NH
 4 and HNCACB spectra to their respective closest HSQC peaks. If the HSQC peak
 5 and its associated peaks in the CBCA(CO)NH and HNCACB spectra form a per-
 6 fect spin system (in this case, exactly six peaks), then the resultant spin system is
 7 inserted into the list of perfect spin systems. These already associated peaks are
 8 then removed from the nearest neighbor model for further consideration. In the
 9 inviting step, each remaining peak in the HSQC spectrum looks for the k closest
 10 peaks in the CBCA(CO)NH and HNCACB spectra, and if a perfect spin system
 11 can be formed using some of these k peaks, then the spin system is formed and the
 12 associated peaks are removed. The parameter k is related to the number of peaks
 13 contained in a perfect spin system, which is known ahead of resonance assign-
 14 ment. A typical value of k is set as 1.5 times the number of peaks in a perfect
 15 spin system. In our case of triple spectra, HSQC, HNCACB, and CBCA(CO)NH,
 16 the number of peaks in a perfect spin system is six and consequently $k = 9$.
 17 The aforementioned two steps will be iteratively executed until no more perfect
 18 spin systems can be found and two lists of spin systems, perfect and imperfect,
 19 are constructed. It should be note that this bidirectional nearest neighbor model
 20 essentially applies the center specific tolerance thresholds, and thus it does not
 21 require any chemical shift tolerance thresholds. Nonetheless, users could specifiy
 22 maximal HN and N chemical shift tolerance thresholds to speed up the process,
 23 though we have noticed that the minor differences in these maximal chemical shift
 24 tolerance thresholds would not really affect the performance of this bidirectional
 25 search.

1 **2.2. Phase 2: Ambiguity resolving, adjacency determination, and**
 2 **string assignment**

3 The goal of resolving is to identify the true peaks contained in the imperfect spin
 4 systems and then to conduct the spin system chaining and the string assignment.
 5 In general, it is very difficult to distinguish the true peaks from the fake peaks when
 6 every imperfect spin system is examined individually. During our development, we
 7 have found that in most cases, those spin systems containing true peaks enable more
 8 confident string findings than those containing fake peaks. With this observation,
 9 we propose to extract true peaks from the imperfect spin systems through the spin
 10 system chaining and the resultant string assignment, namely, to only accept the
 11 peaks that result in spin systems having highly confident mapping positions in the
 12 target protein.

13 The relationships between spin systems are formulated into a connectivity
 14 graph, similar to what we have proposed in another sequential assignment pro-
 15 gram CISA.¹⁵ In the connectivity graph, one vertex corresponds to a spin system.
 16 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 $v_j = (\text{HN}_j,$
 17 $\text{N}_j, \text{C}_j^\alpha, \text{C}_j^\beta, \text{C}_{j-1}^\alpha, \text{C}_{j-1}^\beta)$, if both $|\text{C}_i^\alpha - \text{C}_{j-1}^\alpha| \leq \delta_\alpha$ and $|\text{C}_i^\beta - \text{C}_{j-1}^\beta| \leq \delta_\beta$ hold,
 18 then there is an edge from v_i to v_j with its weight calculated as

$$19 \quad \frac{1}{2} \left(\frac{|\text{C}_i^\alpha - \text{C}_{j-1}^\alpha|}{\delta_\alpha} + \frac{|\text{C}_i^\beta - \text{C}_{j-1}^\beta|}{\delta_\beta} \right). \quad (1)$$

20 In Eq. (1), both δ_α and δ_β are pre-determined chemical shift tolerance thresh-
 21 olds, which are typically set to 0.2 ppm and 0.4 ppm, respectively, though minor
 22 adjustments are sometimes necessary to ensure a sufficient number of connectivi-
 23 ties. 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 one
 24 imperfect spin system $v_j = (\text{HN}_j, \text{N}_j, \text{C}_{j1}^\alpha, \text{C}_{j2}^\alpha, \dots, \text{C}_{jm}^\alpha, \text{C}_{j1}^\beta, \text{C}_{j2}^\beta, \dots, \text{C}_{jn}^\beta)$, we
 25 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 $l, p \in [1, m]$
 26 and $k, q \in [1, n]$. Those carbon chemical shifts with subscription l, k represent the
 27 intra-residue chemical shifts and those with subscription p, q represent the inter-
 28 residue chemical shifts. Subsequently, if both $|\text{C}_i^\alpha - \text{C}_{jp}^\alpha| \leq \delta_\alpha$ and $|\text{C}_i^\beta - \text{C}_{jq}^\beta| \leq \delta_\beta$
 29 hold, then there is an edge from v_i to v'_j with its weight calculated as

$$30 \quad \frac{1}{2} \left(\frac{|\text{C}_i^\alpha - \text{C}_{jp}^\alpha|}{\delta_\alpha} + \frac{|\text{C}_i^\beta - \text{C}_{jq}^\beta|}{\delta_\beta} \right). \quad (2)$$

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

$$33 \quad \frac{1}{2} \left(\frac{|\text{C}_{jl}^\alpha - \text{C}_{i-1}^\alpha|}{\delta_\alpha} + \frac{|\text{C}_{jk}^\beta - \text{C}_{i-1}^\beta|}{\delta_\beta} \right).$$

34 Note that it is possible that there are multiple edges between one perfect spin
 35 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.

1 Once the connectivity graph has been constructed, GASA proceeds essentially
2 the same as CISA¹⁵ to apply a local heuristic search algorithm, guided by the
3 mapping quality of the generated string of spin systems in the target protein. Given
4 a string, its mapping quality in the target protein, or *mapping score*, is measured
5 by the average likelihood of spin systems at the best mapping position for the
6 string, where the likelihood of a spin system at a position is estimated by the
7 histogram-based scoring scheme developed in Wan *et al.*¹⁶ This scoring scheme is
8 essentially a naive Bayesian scheme, which uses the chemical shift values collected in
9 BioMagResBank (<http://www.bmrb.wisc.edu>) as prior distributions and estimates
10 for every observed chemical shift value the probability that it is associated with an
11 amino acid residue residing in certain secondary structure. More precisely, for every
12 type of chemical shift, there is a tolerance window of length ϵ . For an observed
13 chemical shift value cs , the number of chemical shift values in BioMagResBank
14 that fall in the range $(cs - \epsilon, cs + \epsilon)$, denoted as $N(cs | aa, ss)$, is counted for
15 every combination of amino acid type aa and secondary structure type ss . The
16 probability is then computed as $P(cs | aa, ss) = N(cs | aa, ss)/N(aa, ss)$, where
17 $N(aa, ss)$ is the total number of the same kind of chemical shift values collected
18 in BioMagResBank. The scoring scheme then takes the absolute logarithm of the
19 probability as the mapping score. Summing up the individual mapping scores of
20 all intra-residue chemical shifts in a spin system gives the spin system its mapping
21 score to every amino acid residue in the target protein.

22 The edges in the connectivity graph are weighted by Eqs. (1) and (2), and they
23 are used to order the edges coming out of the ending spin system in the current
24 string to provide the candidate spin systems for the current string to grow to. It has
25 been observed that a sufficiently long string itself is able to detect the succeeding
26 spin system by taking advantage of the discerning power of the scoring scheme.
27 In each iteration of GASA, the search algorithm starts with an Open List (OL)
28 of strings and seeks to expand the string with the best mapping score. Another
29 list, Complete List (CL), is used in the algorithm to save those completed strings,
30 which are not further expandable. In the following, we briefly describe the GASA
31 algorithm for resolving the ambiguities in imperfect spin systems through the spin
32 system chaining into strings and the subsequent string assignment.

33 2.2.1. *OL initialization*

34 Let G denote the constructed connectivity graph. GASA first searches for all *unam-*
35 *biguous* edges in G , which are the edges in G whose starting vertex has out-degree
36 1 and the ending vertex has in-degree 1. We note that such a process is similar to
37 RANDOM and CASA. It then expands these edges into simple paths with a pre-
38 defined length L by both tracing their starting vertices backward and their ending
39 vertices forward. The tracing process stops when either of the following conditions
40 is satisfied: (1) The newly reached vertices are already in the paths; (2) The length
41 of each path reaches L . All these paths stored in OL are sorted in the non-increasing

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

2.2.2. Path growing

5 In this step, GASA tries to bidirectionally expand the top ranked path stored in
6 OL, a process similar to MARS, which enumerates all paths with length 5 and
7 then expands each path in two directions. Note that PACES simply enumerates
8 all possible paths from root vertices (i.e. in-degree 0) without such an expanding
9 process; CASA has a path growing process very similar to ours but its path ranking
10 is done using the number of mapping positions (not the mapping score) for the path.
11 Denote this top ranked path as P , the starting vertex in P as h and the ending vertex
12 in P as t . All the directed edges incident to h and incident from t are considered
13 as candidate edges to potentially (bidirectionally) expand P , and the resultant
14 expanded paths are called the child paths of P . For every child path, GASA finds
15 its best mapping position in the target protein and calculates its mapping score.
16 If the mapping score is higher than that of some path already stored in OL, then
17 this child path makes into OL (and accordingly the path with the least mapping
18 score is removed from OL, if there is no space for it). If none of the child paths of P
19 can make into OL, or P is not expandable in either direction (i.e. there is no edge
20 incident to h , neither edge incident from t), path P is *closed* for further expanding
21 and subsequently is moved into CL. GASA proceeds to consider the top ranked
22 path in OL iteratively and this growing process is terminated when OL becomes
23 empty.

2.2.3. CL finalizing

25 Let P denote the path of the highest mapping score in CL (tie is broken to the
26 longest path, and further tie is broken arbitrarily). GASA performs the following
27 filtering: Firstly, all paths in CL with both their lengths and their mapping scores
28 less than 90% of the length and the score of path P are considered as of low
29 quality compared to path P and thus discarded from further consideration. The
30 remaining paths in CL are considered to be reliable strings. Secondly, only those
31 edges occurring in at least 90% of the paths in CL are regarded as reliable edges.
32 The other edges in the paths in CL are therefore removed, which might break the
33 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

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

10 *X. Wan & G. Lin*

1 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
3 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
5 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
7 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
9 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
11 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
13 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
15 their mapping positions are reported as the output assignment.

2.3. Implementation

17 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
19 or as a whole package through the corresponding author.

3. Experimental Results

21 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/>),
23 RANDOM (source code obtained through its correspondence author), MARS (downloaded through <http://www.spincore.com/nmrinfo/MARS.html>),
25 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
27 *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
29 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 inter-
31 esting since the experimental results will show the validity of combining the spin
system chaining with the resultant string assignment in order to resolve the ambigu-
33 ities 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
35 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
37 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
39 the string assignment all together.

1 Regarding the performance measurements, RIBRA explicitly defines two criteria, namely *precision* and *recall*. In particular, *precision* is defined as the percentage
3 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
5 that should be assigned spin systems, respectively.¹⁴ In this paper, we use the same criteria to facilitate the comparison.

7 3.1. Experiment 1

The dataset in Experiment 1 is simulated on the basis of 12 proteins studied in
9 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
11 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
13 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
15 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
17 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
19 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
21 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
23 average out-degree of the vertices in the connectivity graph.

All four programs — RANDOM, PACES, MARS, and GASA — were called
25 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
27 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
29
31
33
35
37
39
41

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.

L	$\delta_\alpha = 0.2 \text{ ppm}, \delta_\beta = 0.4 \text{ ppm}$				$\delta_\alpha = 0.4 \text{ ppm}, \delta_\beta = 0.8 \text{ ppm}$			
	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

1 on these seven instances were obtained through manually reducing the tolerance
2 thresholds to remove a significant portion of edges from the connectivity graphs.
3 We implemented the scheme that if PACES did not finish an instance in 8 h, then
4 the tolerance thresholds would be reduced by 25%, for example, from $\delta_\alpha = 0.2 \text{ ppm}$
5 to $\delta_\alpha = 0.15 \text{ ppm}$. We remark that the performance of PACES in this experiment
6 is a bit lower than that is claimed in its original paper.⁸ There are at least three
7 possible reasons for this: (1) The datasets tested in Coggins and Zhou⁸ are different
8 from ours (which, unfortunately, were unavailable to us). We have done a test on
9 re-simulating the datasets in Coggins and Zhou,⁸ according to its description, to
10 compare PACES, RANDOM and MARS with CISA,¹⁵ a predecessor of GASA, and
11 the result tendency is very much the same as what we have seen here. (2) PACES
12 is only semi-automated, in the sense that it needs manual adjustment after each
13 iteration to iteratively improve the assignment. In this experiment, PACES was
14 taken as fully automated and it was run for only one iteration. One could run it
15 several iterations with manual adjustment for improved assignment. However, in
16 the current work we were unable to do this fairly and subsequently we decided not
17 to do so. (3) PACES is designed to take in better quality spin systems that contain
18 in addition the carbonyl CO chemical shifts. On the current combination without
19 the CO chemical shifts, PACES was expected to perform a bit lower, since the
20 extra CO chemical shifts will provide extra information for resolving the adjacency
21 ambiguities. Again, we have done a similar test on using the combination with the
22 CO chemical shifts to compare RANDOM, PACES, and MARS with CISA,¹⁵ and
23 the result tendency is very much the same as what we have seen here. MARS and
24 GASA performed equally very well. They both outperformed PACES and RAN-
25 DOM 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.

L	InstanceID	RANDOM		PACES		MARS		GASA	
		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

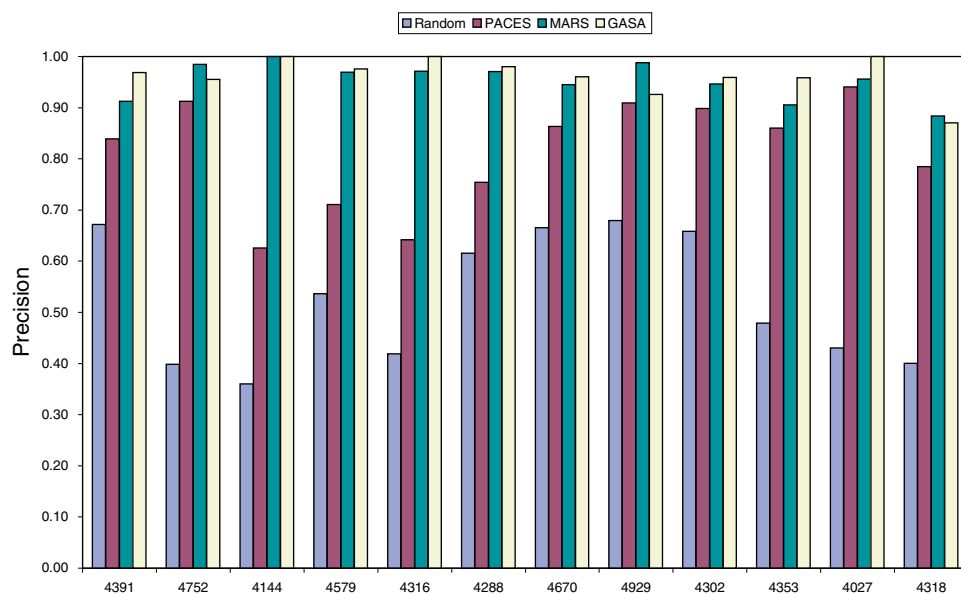
*PACES performance on these 3 datasets were obtained by reducing tolerance thresholds to $\delta_\alpha = 0.15$ ppm and $\delta_\beta = 0.3$ ppm (75%).

[†]PACES performance on this dataset was obtained by reducing tolerance thresholds to $\delta_\alpha = 0.3$ ppm and $\delta_\beta = 0.6$ ppm (75%).

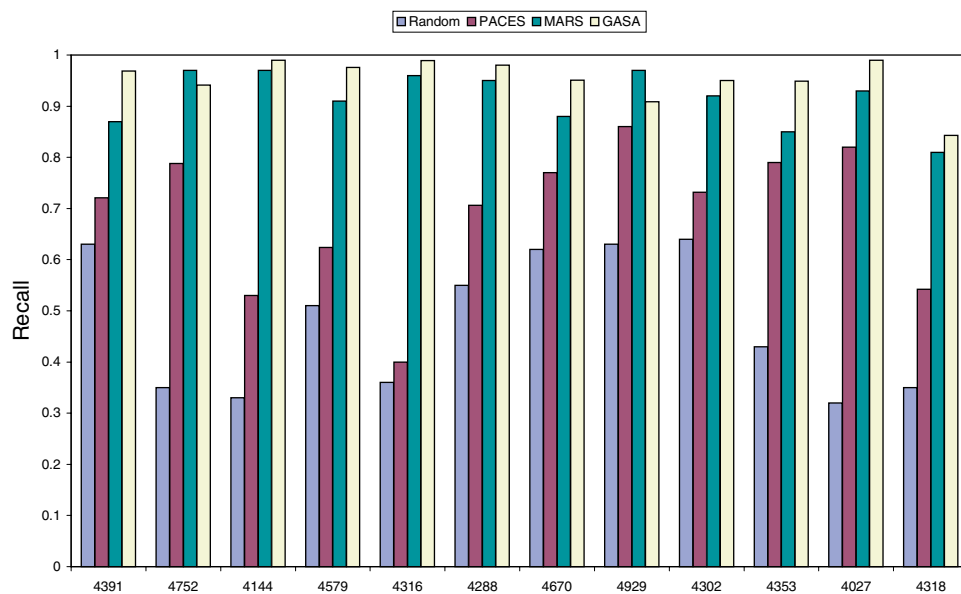
[‡]PACES performance on these 3 datasets were obtained by reducing tolerance thresholds to $\delta_\alpha = 0.2$ ppm and $\delta_\beta = 0.4$ ppm (50%).

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

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

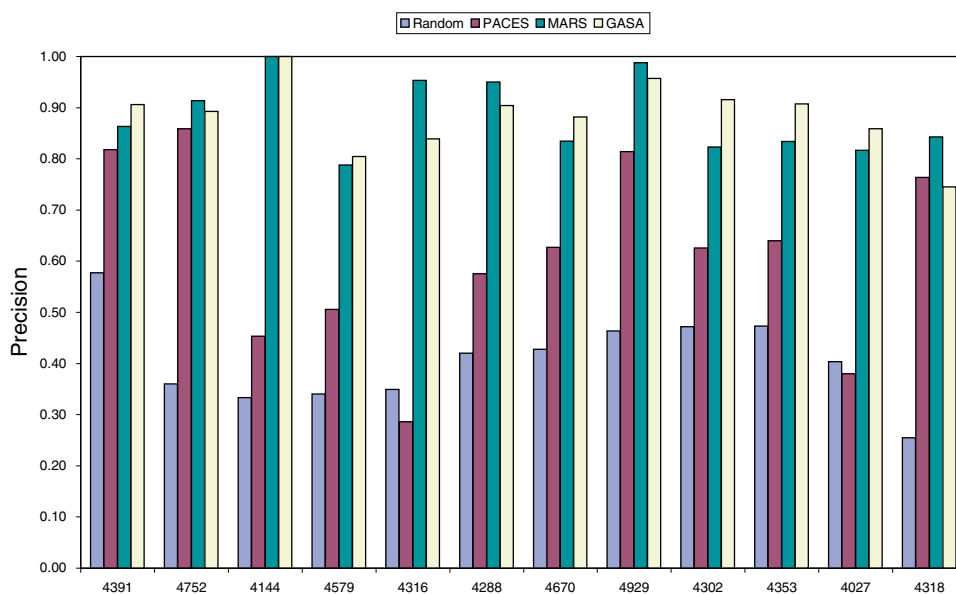
14 *X. Wan & G. Lin*

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

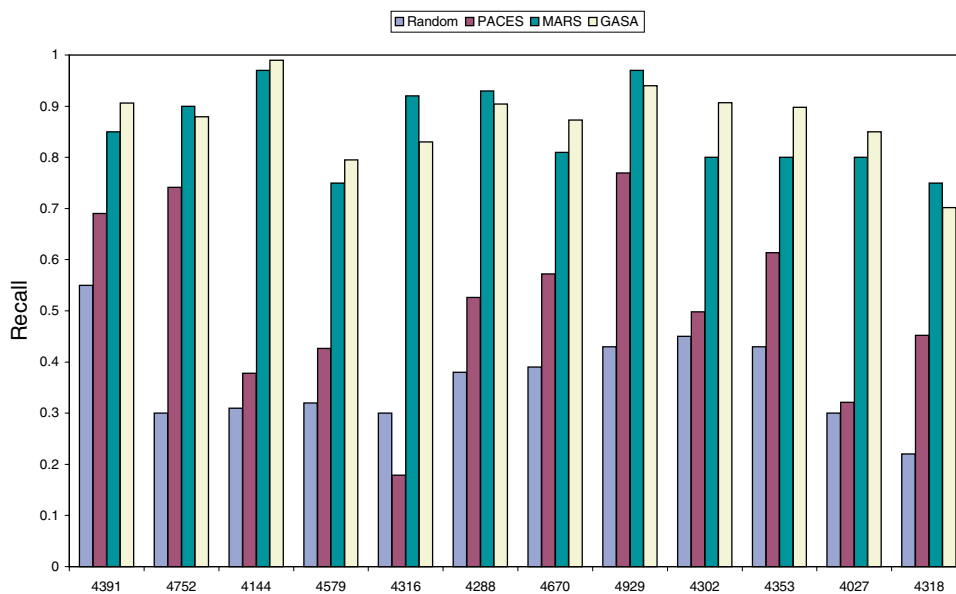


(b) Assignment recall on the 1st set of instances.

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.

16 X. Wan & G. Lin

1 $p = 4.5 \times 10^{-11}$ for recall, GASA performed significantly better than RANDOM
 2 too; On the second set of instances, with $p = 7.7 \times 10^{-5}$ for precision and
 3 $p = 1.4 \times 10^{-6}$ for recall, GASA performed significantly better than PACES as
 4 well; with $p = 4.6 \times 10^{-13}$ for precision and $p = 4.3 \times 10^{-13}$ for recall, GASA also
 5 performed significantly better than RANDOM.

3.2. Experiment 2

7 In RIBRA, five different datasets were simulated from the data entries deposited in
 8 BioMagResBank. Among them, one is *perfect* dataset, which is simulated from
 9 BioMagResBank without adding any data errors, and the other four datasets
 10 contain four different types of spectral data errors respectively. The *false positive*
 11 dataset is generated by respectively, adding 5% fake peaks into the perfect
 12 CBCA(CO)NH and HNCACB peak lists. The *false negative* dataset is generated
 13 by randomly removing a small portion of inter-residue peaks from the perfect
 14 CBCA(CO)NH and HNCACB peak lists. The *grouping error* dataset is generated
 15 by adding HN, N, C^α and C^β perturbations into peaks in the perfect CBCA(CO)NH
 16 peak list. The *linking error* dataset is generated by adding C^α and C^β perturbations
 17 into inter-residue peaks in the perfect HNCACB peak list. Note that each of these
 18 four datasets contains only one type of spectral data error/noise. The chemical shift
 19 perturbations are done in the same way as we did in Experiment 1, by adding to the
 20 original chemical shifts random errors that follow independent normal distributions
 21 with 0 means and constant standard deviations ($\sigma_{\text{HN}} = 0.06/2.5 = 0.0024$ ppm,
 22 $\sigma_{\text{N}} = 0.8/2.5 = 0.32$ ppm, $\sigma_\alpha = 0.2/2.5 = 0.08$ ppm, and $\sigma_\beta = 0.4/2.5 = 0.16$ ppm).

23 Table 3 collects the average performance precision and recall of RIBRA and
 24 GASA on these five datasets. As shown, there is no significant difference among
 25 the performances on the *perfect*, *false positive* and *linking error* datasets. GASA
 26 shows more robustness on the dataset with missing data while RIBRA performs
 27 better on the *grouping error* dataset. Through the detailed study, we found that all
 28 these five simulated datasets by RIBRA contain the inter-residue and intra-residue
 29 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.

Dataset	RIBRA		GASA	
	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%	88.57%	91.12%	81.27%
	(87.7%)	(72.7%)	(88.5%)	(79.4%)
Linking error	96.28%	89.15%	96.17%	89.74%
Average	97.33%	87.95%	95.84%	89.14%

1 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
3 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
5 spectral data, a huge amount of ambiguity in the sequential assignment results
from Glycine residues because they correspond to various legal combinations in
7 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
9 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
11 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
13 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
15 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
17 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
19 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
21 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
23 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
25 because the chemical shifts deposited in BioMagResBank (<http://bmrw.wisc.edu/>)
have been manually adjusted across multiple spectra. Even though HSQC is a very
27 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
29 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.

31 **3.3. Experiment 3**

The purpose of Experiment 3 is to provide more convincing comparison results
33 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
35 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,
37 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
39 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
41 and N chemical shifts with its own C^α and C^β chemical shifts and with C^α and

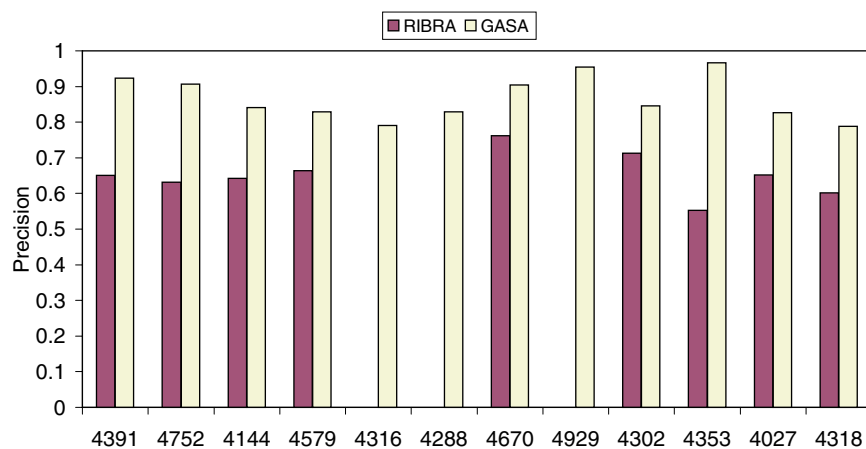
18 *X. Wan & G. Lin*

1 C^β chemical shifts from the preceding residue formed two intra-residue peaks and
 2 two inter-residue peaks respectively in the HNCACB peak list. Note that there is
 3 no C^β peak for Glycine in either the CBCA(CO)NH or the HNCACB peak list.
 4 Next, for each peak in the HSQC, CBCA(CO)NH, and HNCACB peak lists, the
 5 contained HN, N, C^α or C^β chemical shifts were perturbed by adding random errors
 6 that follow independent normal distributions with 0 means and constant standard
 7 deviations. We chose the same tolerance thresholds as those were used in RIBRA,
 8 which were $\delta_{\text{HN}} = 0.06$ ppm, $\delta_{\text{N}} = 0.8$ ppm, $\delta_{\alpha} = 0.2$ ppm, and $\delta_{\beta} = 0.4$ ppm,
 9 respectively. Subsequently, the standard deviations of the normal distributions were
 10 set to $\sigma_{\text{HN}} = 0.06/2.5 = 0.0024$ ppm, $\sigma_{\text{N}} = 0.8/2.5 = 0.32$ ppm, $\sigma_{\alpha} = 0.2/2.5 =$
 11 0.08 ppm, and $\sigma_{\beta} = 0.4/2.5 = 0.16$ ppm, respectively.

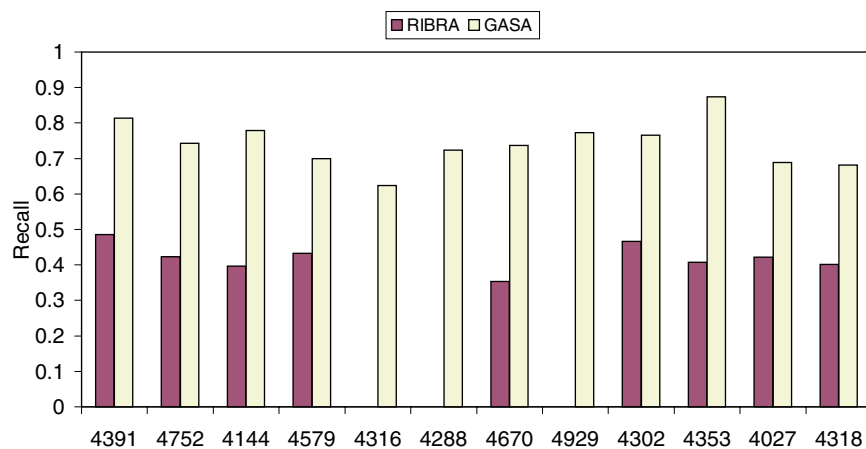
12 Partial information of and the performances of RIBRA and GASA on these 12
 13 proteins are summarized in Table 4. The detailed dataset is available through the
 14 link <http://www.cs.ualberta.ca/~ghlin/src/WebTools/gasa.php>. From the table, we
 15 can see that GASA formed many more spin systems than RIBRA did on every
 16 instance, and from the high assignment precision we can conclude that most of
 17 these spin systems are true spin systems. On average, GASA performed significantly
 18 better than RIBRA (precision 86.72% versus 65.23%, ANOVA $p = 3.0 \times 10^{-4}$;
 19 recall 74.18% versus 42.10%, ANOVA $p = 3.1 \times 10^{-7}$). The detailed precision
 20 and recall are also plotted in Figure 5. In summary, GASA outperformed RIBRA
 21 in all instances and RIBRA failed to solve three instances, which are **bmr4316**,
 22 **bmr4288** and **bmr4929**. As shown in Table 4, RIBRA achieved 65.23% precision
 23 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 Entry	L	M	RIBRA			GASA		
			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.

1 in Wu *et al.*¹⁴ The possible explanations for RIBRA not doing well on these 12
 2 instances are: (1) The simulation procedure in Experiment 3 did not generate C^β
 3 peaks with 0 chemical shift for Glycines, which causes more ambiguities in the
 4 peak grouping, and the subsequent spin system chaining. (2) In the 12 simulated
 5 instances in Experiment 3, the chemical shifts in every peak in all HSQC, HNCACB,
 6 and CBCA(CO)NH peak lists were perturbed with random reading errors, which
 7 generated more uncertainties in every step of operation in the sequential assignment.
 8 Regarding the running time, GASA finished within minutes on each instances, while
 9 the RIBRA web server generally returned an assignment in an hour, and it could
 take several hours on hard instances.

20 X. Wan & G. Lin

1 4. Conclusions

3 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
5 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
7 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
9 are resolved globally and optimally. The extensive comparison experiments with
several recent works including PACES, RANDOM, MARS, and RIBRA showed
11 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.

13 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
15 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 BioMa-
17 gResBank 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
19 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-
21 known triple spectra HSQC, HNCACB and CBCA(CO)NH, are available through
the link <http://www.cs.ualberta.ca/~ghlin/src/WebTools/gasa.php>.

23 Acknowledgments

25 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
27 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
29 improve the presentation.

References

- 31 1. Ferentz AE, Wagner G, NMR spectroscopy: a multifaceted approach to macromolec-
ular structure, *Quarterly Rev Biophys* **33**:29–65, 2000.
- 33 2. Williamson MP, Havel TF, Wüthrich K, Solution conformation and proteinase
inhibitor IIA from bull seminal plasma by proton NMR and distance geometry,
35 *J Mol Biol* **182**:295–315, 1985.
- 37 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.
- 39 4. Zimmerman DE, Kulikowski CA, Huang Y, Tashiro WFM, Shimotakahara S, Chien C,
Powers R, Montelione GT, Automated analysis of protein NMR assignments using
41 methods from artificial intelligence, *J Mol Biol* **269**:592–610, 1997.

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