

# A Random Graph Approach to NMR Sequential Assignment

Chris Bailey-Kellogg\*  
cbk@cs.purdue.edu

Sheetal Chainraj  
schainra@cs.purdue.edu  
Department of Computer Science  
Purdue University  
West Lafayette, IN 47907, USA

Gopal Pandurangan  
gopal@cs.purdue.edu

## ABSTRACT

Nuclear magnetic resonance (NMR) spectroscopy allows scientists to study protein structure, dynamics, and interactions in solution. A necessary first step for such applications is determining the resonance assignment, mapping spectral data to atoms and residues in the primary sequence. Automated resonance assignment algorithms rely on information regarding connectivity (e.g. through-bond atomic interactions) and amino acid type, typically using the former to determine strings of connected residues and the latter to map those strings to positions in the primary sequence. Significant ambiguity exists in both connectivity and amino acid type, and different algorithms have combined the information in two phases (find short unambiguous strings then align) or simultaneously (align while extending strings). This paper focuses on the information content available in connectivity alone, allowing for ambiguity rather than handling only unambiguous strings, and complements existing work on the information content in amino acid type.

In this paper, we develop a novel random-graph theoretic framework for algorithmic analysis of NMR sequential assignment. Our random graph model captures the structure of chemical shift degeneracy (a key source of connectivity ambiguity). We then give a simple and natural randomized algorithm for finding an optimum sequential cover. The algorithm naturally and efficiently reuses substrings while exploring connectivity choices; it overcomes local ambiguity by enforcing global consistency of all choices. We employ our random graph model to analyze our algorithm, and show that it can provably tolerate a relatively large ambiguity while still giving expected optimal performance in polynomial time. To study the algorithm's performance in practice, we tested it on experimental data sets from a variety of proteins and experimental set-ups. The algorithm was able to overcome significant noise and local ambiguity and consistently identify significant sequential fragments.

---

\* Author to whom all correspondence be addressed.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

RECOMB'04, March 27–31, 2004, San Diego, California, USA.  
Copyright 2004 ACM 1-58113-755-9/04/0003 ...\$5.00.

## Categories and Subject Descriptors

J.3 [Life and Medical Sciences]: Biology and genetics; G.3 [Probability and Statistics]: Probabilistic algorithms; F.2.2 [Analysis of Algorithms]: Nonnumerical algorithms and problems

## General Terms

Algorithms, Theory, Experimentation, Performance

## Keywords

Protein structure determination, nuclear magnetic resonance (NMR) spectroscopy, chemical shift degeneracy, automated sequential resonance assignment, random graph model, randomized algorithm, probabilistic analysis, Hamiltonian path

## 1. INTRODUCTION

Nuclear magnetic resonance (NMR) spectroscopy plays a vital role in post-genomic tasks including determination of protein structure, dynamics, and interactions. Full automation is required in order for NMR to fulfill its potential in supporting such applications at a genomic scale [18, 24]. In particular, much work [19] has focused on automated *sequential resonance assignment*, an essential step for these applications. NMR spectra operate in terms of atomic “IDs” called *chemical shifts*; the problem of resonance assignment is to determine the correspondence between these chemical shifts and the atoms in the known primary sequence. Significant progress has been made in assignment algorithms, employing approaches including best-first search [15, 26], exhaustive and heuristic search [3, 4, 5, 12], branch-and-bound [16], genetic algorithms [6], simulated annealing [9], and Monte Carlo optimization [17, 14, 13].

The primary information content in NMR data utilized by typical sequential resonance assignment algorithms can be categorized into *connectivity* information, indicating possible atomic interactions (through a small number of bonds, or through space), and *amino acid type* information, roughly characterizing the type of residue containing the atoms. Both types of information are often highly ambiguous; for example, an NMR peak could be explained by several or even tens of possible mutually-exclusive connectivities, and amino acid type information might be more or less consistent with any or all residues. Assignment algorithms typically use connectivity information to piece together short strings of connected atoms (spanning multiple residues), and use amino acid type information to identify a consistent substring in the primary sequence. These two aspects can be employed in separate “growing” and “aligning”

phases (e.g. [4, 5, 12, 16]) or in a simultaneous approach of growing aligned strings (e.g. [26] and many others). A goal essential to the wide-spread acceptance and utilization of automated assignment algorithms is the analysis of assignment algorithms in terms of the (ambiguous) connectivity and sequence information in data provided by the experimentalist.

This paper explores the role of connectivity information alone in sequential resonance assignment. Both the two-phase and simultaneous assignment approaches described above either implicitly or explicitly rely on some “good” connectivity information to help overcome the significant ambiguity in amino acid type information. For example, the two-phase MAPPER program [12] starts with unambiguously connected strings and explores possible matches against the primary sequence. In such an approach, the run-time and output quality depend on the length of the connected strings, and thus the number of permutations of them that must be explored. In fact, it might be worthwhile to allow some ambiguity in the strings grown (e.g. by independently trying multiple possibilities at connectivity choice points) in order to increase their length and thereby decrease the number of consistent alignments against the primary sequence (intuitively, each additional match provides another “bit” helping distinguish possible locations in the primary sequence [4, 5]). While the role and algorithmic impact of amino acid type information has been studied, the corresponding information content and algorithmic impact of connectivity alone has not. We demonstrate here that enforcing *global consistency of local connectivity choices* overcomes a great deal of ambiguity in growing, so that only a small number of the possible strings need to be explicitly explored for alignment. We develop a novel growing algorithm that *effectively reuses* connected strings, and we analyze the algorithm in terms of a novel random graph formalism that captures the key source of ambiguity in connectivity. Our approach is generic to different approaches to assignment and, in complementing other existing work on aligning, promises to lead to a solid algorithmic basis for resonance assignment.

## 2. GRAPH REPRESENTATION OF NMR DATA

NMR spectroscopy captures magnetic interactions between atoms as peaks in multi-dimensional spectra [10]. For example, the HSQC (heteronuclear single quantum coherence spectroscopy) experiment of Fig. 1(a) correlates bonded  $H^N-N$  atom pairs and yields a two-dimensional spectrum. Peaks in a spectrum indicate pairs of atoms that exhibit a particular interaction (amide bond in HSQC); axes are resonance frequencies, or *chemical shifts*, of the interacting atoms. Three-dimensional experiments correlate triplets of atoms connected by a few bonds; e.g. HNcoCA correlates the pair  $H^N-N$  of one residue with  $C^\alpha$  of the preceding residue, thereby capturing *sequential* interactions, while HNCA correlates  $H^N-N$  with both preceding and within-residue  $C^\alpha$ , capturing both sequential and *within-residue* interactions. Similar experiments involve interactions with  $C^\beta$ ,  $H^\alpha$ , and CO.

Individual NMR spectra provide no information on which residue locations in the protein sequence originated each peak. In order for spectral information to be useful, a scientist must determine the mapping from chemical shifts to individual atoms, the *resonance assignment*. Typically, data from a set of through-bond experiments is maintained in a data structure called a *spin system*, which references peaks by shared  $H^N-N$  shifts and contains sequential and within-residue shifts of the remaining atom types. Then assignment proceeds (Fig. 1(b)) by matching sequential and within-residue chemical shifts to determine sequential connectivity, and

aligning the connected spin systems to substrings of the primary sequence [25]. Evidence for the alignment is given by consistency of the observed chemical shifts with those expected for the corresponding residue type, based on the well-characterized effect of primary sequence on resonance [22].

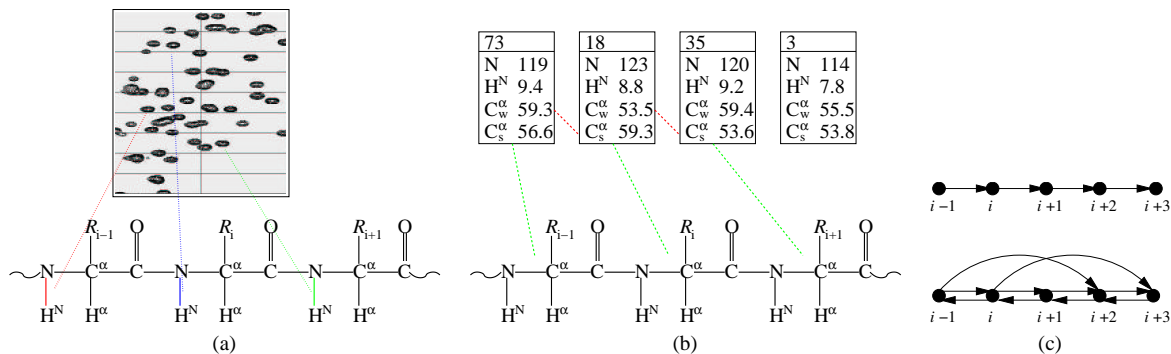
An alternative approach to assignment [23, 4, 5] uses through-space (NOESY) interactions rather than the through-bond interactions described above. For example, in an  $\alpha$ -helix, sequential residues have  $H^N-H^N$  NOE interactions, while residues  $i$  and  $i+3$  have  $H^\alpha-H^N$  NOEs. It is worth noting that in these  $^{15}N$ -only approaches, the  $H^N-H^N$  interactions are symmetric (i.e. one from  $i$  to  $i+1$  and one from  $i+1$  to  $i$ ), but the  $H^\alpha-H^N$  interactions are only from  $i$  to  $i+3$ . More complex interaction patterns arise in  $\beta$ -sheets, involving networks of cross-sheet connectivities, and sparse connectivity information (primarily sequential  $H^\alpha-H^N$ ) is also available in random coil regions. The basic approach to assignment based on NOESY still follows the generic pattern of determining connectivities and aligning connected portions against the primary sequence.

This paper focuses on the analysis of connectivity information, and is generic to the source of connectivity — NOESY or any of many possible through-bond experiments used in traditional assignment algorithms. Thus we abstract the NMR data into a graph which will be the input for our algorithm.

**DEFINITION 1 (NMR INTERACTION GRAPH).** *An NMR interaction graph  $G = (V, E)$  is a labeled, weighted, directed graph with vertices  $V$  corresponding (via an unknown mapping) to residues (or to noise, for extras), and edges  $E \subset V \times V$  such that  $e = (v_1, v_2) \in E$  iff there are observed NMR peaks that could be explained by interactions between atoms of  $v_1$  and atoms of  $v_2$ . Edges are labeled with an interaction type indicating which atoms interact in spectral peaks, and a match score indicating the quality of the chemical shift match.*

For example (see Fig. 1(c)), a correct edge for an interresidue peak from an HNCA experiment would connect the vertex for one residue (via  $H^N$  and N) with the vertex for the preceding residue (via  $C^\alpha$ ). Similarly, correct edges for interresidue peaks from an  $^{15}N$ -NOESY in an  $\alpha$ -helix region would connect the vertex for one residue (via  $H^N$  and N) with the vertex for the preceding residue (via  $H^N$ ), with the vertex for the subsequent residue (via  $H^N$ ), and with the vertex for the third residue back (via  $H^\alpha$ ). The match score in each of these cases would simply compare the chemical shift observed in the interresidue peak with the shift expected for the partner residue (from intraresidue peaks). When multiple peaks are combined in an edge (e.g. with HNCACB, using both  $C^\alpha$  and  $C^\beta$ ), scores must be appropriately combined, accounting for missing dimensions.

For the remainder, we assume that an interaction graph has been derived from the spectral data, by standard compilation of spin systems (vertices) and identification of “reasonable” matches between inter- and intra-residue chemical shifts (edges) [26]. Let  $G_* = (V_*, E_*)$  denote the interaction graph containing only physically correct vertices and edges. The graph  $G$  derived from NMR data differs substantially from  $G_*$ , due to noise and missing data; one of the goals of assignment is to identify the correct subgraph. Basic addition and deletion of vertices and edges can be treated as essentially random (e.g. a peak has some chance of falling below the noise floor, thereby deleting the corresponding edge). However, there is an interesting structure in NMR data that gives rise to a large amount of ambiguity in defining the edges from the data, and thus shows up as *correlated* noise edges. Recall that an edge is generated by matching the observed interresidue chemical shift for



**Figure 1: NMR assignment.** (a) HSQC spectrum. The  $x$ -axis indicates  $H^N$  chemical shift and the  $y$ -axis N chemical shift. A peak corresponds to a bonded  $H^N$ -N pair, but the correspondence (assignment) is unknown. (b) Assignment of spin systems. Each spin system here has within-residue N,  $H^N$ , and  $C^\alpha$ , as well as sequential  $C^\alpha$ . Matching sequential- $C^\alpha$  of one to within- $C^\alpha$  of another helps identify sequential connectivities, and aligning the connected spin systems to positions of consistent amino acid type then defines an assignment. (c) NMR interaction graphs: nodes represent residues, and edges ideal connectivities for (top) sequential backbone (e.g. based on within vs. sequential  $C^\alpha$  match as above) and (bottom) NOESY  $\alpha$ -helix.

a peak associated with one vertex, with the expected intrasidue chemical shift associated with the other vertex. The key source of ambiguity, called *chemical shift degeneracy* (Fig. 2(a)), is the fact that, after accounting for peak width and potential measurement error, multiple possible vertices have intrasidue chemical shifts “close enough” to the interresidue observation. Thus each correct edge is obscured by several or many incorrect edges. The incorrect edges are not randomly distributed, but rather correlated, since when two vertices have atoms that are similar in chemical shift, they will tend to share edges — each edge for the one will also appear for the other.

Traditional  $G_{n,p}$  random graph models [7] essentially add noise edges randomly and independently, and do not capture the correlation, due to chemical shift degeneracy, of noise edges among themselves and with correct edges. In order to develop and analyze an assignment algorithm that focuses on connectivity, we now define a simple model that does. The following focuses on edges derived by matching an interresidue chemical shift of a particular atom type  $a$  (e.g.  $H^N$ ). For edges comprised of matches with  $d$  atom types (dimensions), e.g. using both  $C^\alpha$  and  $C^\beta$ , treat each dimension independently and apply the following to add correlated noise edges:

1. Choose a permutation of  $V$ ; denote by  $\pi(v)$  the index of  $v$  in the permutation. This corresponds to sorting the vertices by chemical shift of atom type  $a$ . There is no systematic, global correlation between positions of atoms in the primary sequence or in space, and positions in the sorted list of chemical shifts. Thus the order of atoms with respect to chemical shift is well-modeled by a random permutation. Prior information on permutations (e.g. from chemical shift predictions [21]) can be incorporated when available.
2. Let  $w(v)$  be the window width around vertex  $v$ , indicating with how many other vertices  $v$  is degenerate (with respect to the single dimension being considered as atom type  $a$ ). This captures peak width and local “clustering” in chemical shift space. We note that, if  $w(v)$  is to be sampled from a distribution, care must be taken to ensure that degeneracy is symmetric (except for small perturbations capturing different peak width).
3. For each  $u \in V$ , let  $C(u)$  be the set of vertices:

$\{v \in V \mid |\pi(v) - \pi(u)| \leq w\}$ . These vertices are degenerate with  $u$  with respect to atom type  $a$ .

4. For each directed edge  $(u, v) \in E_*$ , add directed edges  $\{(u', v) \mid u' \in C(u)\}$  to  $E$ .
5. Set edge weights as a function of distance within the permutation plus noise. This captures typical scoring rules (e.g. [26, 12]) that compare absolute or squared difference in chemical shift. Note that, except for noise (reasonably modeled as Gaussian), the correct edge should match exactly and thus have the best score.

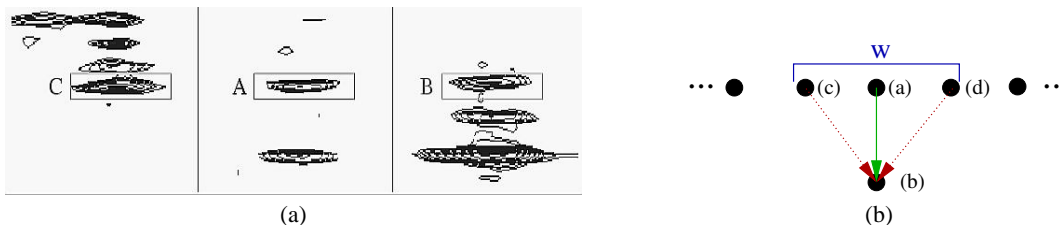
Fig. 2(b) illustrates a chemical shift degeneracy window and resulting noise edges. Note that in typical 3D NMR experiments, the ambiguity is one-sided, since the peak has 2 chemical shifts anchoring it in the residue at which the peak was observed, but only 1 interresidue chemical shift on which the match is based. It is straightforward to extend this model to allow non-zero ambiguity on the “to” side of the edge, or to allow 2 matches on the “from” side (e.g. for 4D spectra).

### 3. PATH-FINDING FOR ASSIGNMENT

As discussed in the introduction, the “growing” aspect of sequential resonance assignment uses connectivity information (edges in an NMR interaction graph) to identify chains of (possibly) connected atoms. We develop here an algorithm that pushes that aspect to its logical extreme, and seeks to find not just short unambiguous chains, but rather groups of chains that connect the entire set of vertices.

**DEFINITION 2 (SEQUENTIAL FRAGMENT).** For an NMR graph  $G = (V, E)$ , a sequential fragment  $F = (V', E')$ , where  $V' = \{v_1, v_2, \dots, v_{|F|}\} \subset V$  is a sequence of vertices and  $E' \subset \{(u, v) \in E \mid u, v \in V'\}$  is a set of edges supporting that sequential order.

This is not necessarily a path; for example, in an  $\alpha$ -helix, there might be only an  $H^N$ - $H^N$  edge from  $v_{i+1}$  to  $v_i$  (in the opposite direction from sequentiality) and there might be an  $H^\alpha$ - $H^N$  edge from  $v_i$  to  $v_{i+3}$ . However, since sequential connectivity is the motivation, we will loosely refer to it as a path. We extend the notion of match score to fragments by summing over all appropriate edges.



**Figure 2: Chemical shift degeneracy.** (a) A correct  $A \rightarrow B$  edge will also introduce an incorrect  $C \rightarrow B$  edge, since  $C$ 's chemical shift in the given dimension ( $y$ -axis) is very similar to  $A$ 's. (b) Random graph model introduces noise edges (dotted lines) to vertex (b) from vertices (c,d) in a window of width  $w$  around the correct vertex (a). The order of vertices represents sorted order by chemical shift, so that the correlation structure among edges models that in NMR data.

**DEFINITION 3 (SEQUENTIAL COVER).** For an NMR graph  $G = (V, E)$ , a sequential cover  $C = \{F_1, F_2, \dots, F_{|C|}\}$  is a set of fragments such that each vertex  $v \in V$  appears in exactly one fragment.

Some of the vertices can appear in singleton fragments. We extend the notion of match score to covers by summing over all fragments. Note that the physically correct graph  $G_*$ , with vertices ordered by residue number, corresponds to a sequential cover. The goal of NMR assignment is to find the underlying  $G_*$  given a noisy  $G$ , but the best we can hope to do is to find “good” subgraphs  $G' \subset G$  and evidence that  $G_*$  is among them. In particular, we focus here on finding sequential covers of  $G$  with near-optimal match score. Our optimization problem is then to find the sequential cover of maximum score for a given NMR graph. We note that it is also desirable to find and compare near-optimal covers as well, and discuss in Sec. 4 a practical implementation that does that.

The NP-hardness of this problem is immediate by reduction from Hamiltonian path. We focus here on a randomized algorithm that takes advantage of the problem structure to perform very well in practice (see Sec. 4). We also use our model to analyze the algorithm in various versions. The key insight is that, in searching for good covers, the best ones tend to share a lot of substructure — fragments that differ in some chunks but are largely the same. It is hard for traditional branching-based searches to take advantage of this shared structure because it can appear on many different branches; for example, multiple different paths can take a jump to a common vertex, after which they are all the same; each of these would be on a different branch. However, algorithms as developed for finding Hamiltonian paths for traditional  $G_{n,p}$  random graphs [2], only perform small rearrangements to a path in order to continue extending it. Inspired by the success of the rotation and extension algorithms for finding Hamiltonian paths, we develop a simple-to-implement randomized algorithm that exploits shared substructure in a natural way and also handles breaks and scoring.

The basic randomized algorithm is as follows (Fig. 3). Let  $C$  be a sequential cover. Denote by  $\text{succ}(u, C)$  and  $\text{pred}(u, C)$  the successor and predecessor of  $u$  in  $C$  (these can be null). Initially  $C = V$ , i.e., each vertex is a fragment by itself. We will view the algorithm in two phases. In Phase 1, we extend all *unambiguous* vertices — those with out-degree 1 to vertices with in-degree 1. At the end of Phase 1,  $C$  will be a collection of potentially non-trivial fragments. In Phase 2 we extend *ambiguous* vertices (with out-degree more than 1, or with an edge to a vertex with in-degree more than 1), choosing edges with probability proportional to their scores, and potentially swapping a chosen edge for a previously chosen edge. Note that cycles can be formed during the execution of the algorithm (when an edge connects “back” to a vertex in a

path) — thus we allow  $C$  to contain cycles. Thus at all times,  $C$  will consist of disjoint sets of (simple) paths or cycles. The algorithm is simple to implement and has two desirable properties: it naturally reuses already found paths (i.e., makes only local changes in each step) and also gives preference to edges with higher scores. Assuming that scores capture the probability of the edge being the “correct edge” (refer again to edge weights in our random graph model for a justification), the algorithm nicely balances both local and global criteria.

We first present an analysis of the algorithm on a simplified NMR interaction graph model: the random graph model assuming a random permutation and a fixed window size  $w$  generating an NMR interaction graph  $G = (V, E)$  from the correct sequential connectivities  $G_* = (V_*, E_*)$ . For now, we assume no edge weights, no breaks, and no other sources of noise (additions and deletions); the result can be generalized.

Our analysis relates the performance of the algorithm to  $w$ . We show that if  $w$  is not very large, then the algorithm will find a Hamiltonian path in polynomial time.<sup>1</sup> Actually, our analysis shows more. By giving a simple and natural *random walk* interpretation of our algorithm, we show that that the algorithm will terminate with a “good” Hamiltonian path, i.e., whose expected value is the optimum. The argument also generalizes to weighted edges and the analysis gives an upper bound on the estimated time needed to find a good path.

The following lemma and its analysis seems quite basic and can be of independent interest. The analysis relates the performance of the algorithm to a random walk.

**LEMMA 1.** Let  $G$  be a directed unweighted graph on  $n$  vertices such that each vertex has at most two outgoing edges. Assuming that  $G$  has a (directed) Hamiltonian path, the randomized algorithm will find a Hamiltonian path in expected  $O(n^4)$  steps.

**PROOF.** We first analyze the basic randomized algorithm given in Fig. 3. Let  $H$  be a Hamiltonian path in  $G$ . We focus on the algorithm finding  $H$ . At the end of Phase 1, let  $C \neq H$ , otherwise we are done. The algorithm will terminate if each vertex has the “correct” outgoing edge, i.e., the edge belongs to  $H$ . Phase 1 takes  $O(n)$  steps.

We view the algorithm as a random walk on a line numbered  $0, \dots, n-1$  as follows. In some step (one execution of the loop) of Phase 2, say  $t$ , the algorithm is stationed at number  $k$  where  $k$  is the number of correct edges. We claim that in step  $t+1$ , the algorithm can move to  $k+1$  with probability at least  $1/2$ . The reasoning is

<sup>1</sup>It is interesting to mention that an algorithm of Frieze et al. [8] finds a Hamiltonian path in a graph containing a Hamiltonian path and a large number (at least  $cn$  for some large constant  $c$ ) of random edges.

```

Given  $G = (V, E)$ 
Let initial cover  $C = V$ 
Let vertices with successors  $W = \emptyset$ 
Choose vertex  $u$  from  $V$ 

Phase 1:
Let visited vertices  $U = \emptyset$ 
While  $U \neq V$  do
  Add  $u$  to  $U$ 
  If  $u$  has single out-edge  $e = (u, v)$  and  $v$  has a single
  in-edge then
    Add  $e$  to  $C$ 
    Add  $u$  to  $W$ 
    Set  $u$  to  $v$ 
  Else Choose  $u$  from  $V - U$ 
endwhile

Phase 2:
While  $C$  is not a Hamiltonian path or cycle do
  Choose  $u$  from  $V - W$ 
  Choose an edge  $(u, v)$  with probability proportional
  to its weight
  If  $\text{pred}(v, C)$  is null then
    Join the two fragments in  $C$ 
    Add  $u$  to  $W$ 
  Else
    Create two fragments in  $C$ :  $\langle \dots u, v \dots \rangle$ ,
    and  $\langle \dots, \text{pred}(v, C) \rangle$ 
    Add  $u$  to  $W$ 
    Remove  $\text{pred}(v, C)$  from  $W$ 
endwhile

```

**Figure 3: The basic randomized algorithm for NMR interaction graph sequential cover.**

as follows. If  $u$  is the ambiguous vertex visited in this step, then the probability that the correct edge, say  $(u, v)$ , is chosen is at least  $1/2$ . Now, does this affect the probability of some other edge? The only other edge which is affected is  $(\text{pred}(v, C), v)$ . There are two cases: (a)  $\text{pred}(v, C)$  is null, in which case we have  $k + 1$  correct edges; (b) otherwise, we lose the edge  $(\text{pred}(v, C), v)$ ; but the probability that that edge is correct *conditioned* on the fact that  $(u, v)$  is correct is 0. Thus with probability at least  $1/2$  we add one correct edge. With probability at most  $1/2$ , the algorithm can move to  $k - 1$  or stay at  $k$ .

We are interested in the expected number of steps needed to reach  $n - 1$ . By setting up a difference equation (for example, see [11, pp. 73–74]), we can show that the expected number of steps needed to reach  $n - 1$  is  $O(n^2)$ . Thus, we will have all the correct edges in expected  $O(n^2)$  steps.

However, there are two pathological situations that may occur during the above random walk. The first pathological situation can occur due to the presence of outgoing edges from the last vertex in  $H$  as explained below. For the sake of argument, let  $G$  contain a unique Hamiltonian path  $H = \langle x_1, \dots, x_n \rangle$  and no Hamiltonian cycle. There is no problem if the last vertex in the path,  $x_n$ , does not have an outgoing edge. However, if  $x_n$  has some outgoing edge, then analysis breaks down when handling  $x_n$ : we have to choose an outgoing edge (assume there is only one for simplic-

ity) from  $x_n$  to some  $x_i$  ( $\neq x_1$ ). This choice then removes  $x_i$ 's predecessor (assume there is only one, i.e.,  $x_{i-1}$ , for simplicity) if present in  $C$ . Since we choose vertices to extend arbitrarily from  $V - W$ , we could choose  $x_{i-1}$  in the next step and extend it to  $x_i$ , thus removing  $x_n$  as its predecessor. Thus we can be swapping edges endlessly between  $x_n$  and  $x_{i-1}$ . The second pathological situation is more subtle: at some stage of the random walk, assume that all vertices have chosen their successors, yet we don't have a Hamiltonian path (this means that we will have a set of disjoint cycles and possibly a fragment containing the end vertex of  $H$ ). The algorithm breaks down, since we cannot continue the random walk.

A simple way to avoid the above pathological situations is to fix a start and an end vertex and then do the random walk. We won't extend the end vertex nor allow an incoming vertex to the start vertex. Since we don't *a priori* know the start and end vertices of  $H$ , we run the algorithm  $O(n^2)$  times, trying all possible pairs of vertices in parallel. Since for at least one pair of vertices the algorithm will terminate in expected  $O(n^2)$  steps, we have the result.  $\square$

The following lemma can be considered a stronger version of the above: it relaxes the bound on the condition that each vertex has at most two outgoing edges. Call a vertex *bad* if it has more than two outgoing edges. The lemma gives a bound on the number of bad vertices the algorithm can tolerate while still giving polynomial expected time performance.

**LEMMA 2.** *Let  $G$  be a directed unweighted graph on  $n$  vertices with out-degree at most  $\delta$ , where  $\delta$  is an arbitrary constant (independent of  $n$ ). Let  $G$  have at most  $O(\log n)$  bad vertices. Assuming that  $G$  has a (directed) Hamiltonian path, the randomized algorithm will find a Hamiltonian path in expected  $O(n^{4+\log(\delta-1)})$  steps.*

**Proof sketch:** The proof is similar to Lemma 1. The random walk here is as follows: on all vertices, except at most  $O(\log n)$ , the probability of advancing towards  $n - 1$  is at least  $1/2$ . For the  $O(\log n)$  bad vertices the probability of advancing is only at least  $1/\delta$  since we have at most  $\delta$  choices. The analysis is similar to the "gambler's ruin" problem with retaining barriers [11, p. 74] but with non-identical probabilities. We can set up a difference equation and show that the expected time is  $O(n^{2+\log(\delta-1)})$  (i.e., polynomial in  $n$ ) if there are at most  $O(\log n)$  points having an advancing probability of less than  $1/2$ . We incur an additional  $O(n^2)$  factor to avoid the pathological situations as in Lemma 1.  $\square$

We now focus on the simplified NMR graph model. The following theorem shows that we can get a polynomial time performance (i.e., our randomized algorithm will find a Hamiltonian path in expected polynomial (in  $n$ ) steps) even for relatively large  $w$  (i.e., substantial chemical shift degeneracy). This is shown in the limit, i.e., as  $n \rightarrow \infty$  in Part 1; Part 2 gives a high probability<sup>2</sup> result which holds for an arbitrarily large, but constant  $w$ , for any suitable  $n$ . *This gives us a theoretical explanation as to why our algorithm performs pretty well in practice.* In the following, the number of dimensions  $d$  is a constant, established by the experimental set-up. While the following arguments still hold (somewhat more weakly) when only one dimension is used, the most interesting results are obtained when there are at least two dimensions matched per edge, so that information is combined in order to overcome noise. This is true in all typical NMR protocols, as well as in the NOESY  $\alpha$ -helix case described above. For simplicity of formulas, we apply a uniform window width  $w$  that is the same in each dimension.

<sup>2</sup>With probability at least  $1 - 1/n$ .

**THEOREM 1.** *Let  $G = (V, E)$  be an NMR interaction graph of degeneracy window width  $w$  on  $n$  nodes generated by the simplified model, using  $d > 1$  ( $d$  is an integer constant independent of  $n$ ) matched dimensions for each edge. Then*

1. *If  $w = o(n^{1-(3/2d)})$ , then a Hamiltonian path can be found asymptotically almost surely in expected  $O(n^4)$  steps.*
2. *Let  $w$  be an arbitrary constant (independent of  $n$ ). Then, for a suitably large  $n$ , with high probability, a Hamiltonian path can be found in expected  $O(n^{4+\log(w-1)})$  number of steps.*

**PROOF.** 1. The probability that a noisy directed edge  $(i, j)$  exists, based on a match in one dimension, is  $O(w/n)$ , since such an edge will exist if  $i$  falls within the window of width  $w$  surrounding the true predecessor of  $j$ . Thus the probability of a fooling edge between two vertices, with matches for all  $d$  atoms, is  $O(w^d/n^d)$  (since we assume that matches in different dimensions are independent). Thus the expected number of fooling edges is  $O(w^d/n^{d-2})$ .

The probability that a vertex is bad (i.e., has more than two fooling edges) is  $O(w^{2d}/n^{2d-2})$ . Thus the expected number of bad vertices is  $O(w^{2d}/n^{2d-3})$ .

If  $w = o(n^{1-(3/2d)})$ , then by the first moment method [1],  $G$  will have no bad vertex asymptotically almost surely (a.a.s.)<sup>3</sup>. Then, a.a.s., Lemma 1 can be applied and a Hamiltonian path can be found in expected  $O(n^4)$  steps.

2. Let the random variable  $X_i$  be one if vertex  $i$  is bad, and zero otherwise. Let  $X = \sum_{i=1}^n X_i$  denote the number of bad vertices. From part 1, for a fixed constant  $w$ , the expected number of bad vertices is  $E[X] = O(1)$ . Now, each variable  $X_i$  is independent of all except at most  $dw$  other variables, because each vertex  $i$  can have at most  $dw$  neighbors (counting all the  $d$  dimensions). By partitioning  $X$  into  $dw$  disjoint sums of independent random variables and applying the Chernoff bound [20, p. 68] to each sum separately, we have for a suitably large  $n$ ,

$$\Pr(X > \log n) \leq dwe^{-O(\log^2 n)} = o(1/n)$$

Thus, with high probability we will have at most  $\log n$  bad vertices and thus Lemma 2 applies which gives the result.

□

For example, with two dimensions ( $C^\alpha + C^\beta$  in backbone protocols, or symmetric  $H^N-H^N$  interactions in NOESY  $\alpha$ -helix), if  $w = o(n^{1/4})$  then a.a.s. results are in expected  $O(n^4)$  steps, or if  $w$  is a constant, then with high probability we need expected polynomial number of steps.

The above theorem can be generalized for an arbitrary NMR graph as follows. Consider a NMR graph  $G = (V, E)$  with edge weights. For each vertex  $v$ , let's assume that the weights of the outgoing edges sum to 1, and interpret the weight of an incident outgoing edge to be the probability that it is present in the correct Hamiltonian path; this is reasonable under the scoring model. The goal is to find a Hamiltonian path with maximum edge weight. We assume that there exists a Hamiltonian path. We have a simple, weighted version of Lemma 1:

**LEMMA 3.** *Let the edges of  $G_*$  have weight at least  $1/2$ . Assuming that a Hamiltonian path exists, the randomized algorithm on  $G$  will find a Hamiltonian path whose expected weight is the optimum, in expected time  $O(n^4)$ .*

<sup>3</sup>Probability  $\rightarrow 1$  as  $n \rightarrow \infty$ .

**PROOF.** The proof for the expected time is similar to that of Lemma 1: we again choose the correct edge with probability at least  $1/2$ . Another way of viewing the proof is that the unweighted version with at most two outgoing edges per vertex is the “worst-case” scenario.

We note that when an outgoing edge is chosen for a vertex, the randomized algorithm chooses with probability proportional to its weight. Linearity of expectation gives the expected optimality of the output path. □

A natural generalization of the above lemma for arbitrary weighted NMR graphs is the following lemma which shows that the randomized algorithm indeed terminates with expected optimum, but can take time exponential in  $n$ , if  $p < 1/2$ .

**THEOREM 2.** *Let the edges of  $G_*$  have weight at least  $p$ ,  $0 < p < 1$ . Then the randomized algorithm will find a Hamiltonian path whose expected weight is the optimum, in expected  $O(n^2 (\frac{1-p}{1-2p} (\frac{1}{p} - 1)^n))$  steps.*

**PROOF.** The proof is similar to Lemma 1 and the expected time is identical to the expected time needed for completion for a “gambler’s ruin” problem with retaining barriers [11, p. 74]. □

In practice, our results show that typical data sets are much better than the worst-case scenario considered in this bound.

Finally, we note that in practice we can apply this algorithm even when there are “breaks” in the input graph, i.e., if there is not a (complete) Hamiltonian path in the input graph  $G$ . We assume that  $G$  is weakly-connected, otherwise, we run the algorithm on the individual weakly-connected components. The pathological cases are harder to fix, but arise with small probability, as our results demonstrate.

## 4. RESULTS

We tested an implementation of our random graph algorithm on a total of 10 experimental data sets: 7 complete backbone experiments provided with AUTOASSIGN [26]; and the  $\alpha$ -helical subset of the <sup>15</sup>N-edited NOESY data for the three proteins from the Bushweller lab previously tested with JIGSAW [4, 5]. In both cases, we constructed an NMR interaction graph from the spin systems previously compiled by the respective programs. The graph edges from the backbone data were formed by comparing the inter- (sequential) and intra- (within) residue chemical shifts of candidate spin systems; edge scores were computed from the difference in chemical shifts, combined over the number of dimensions matched, with a penalty for missing dimensions.<sup>4</sup> In the  $\alpha$ -helix tests, symmetric pairs of edges were constructed for the potential  $(i, i + 1)$   $H^N-H^N$  edges, to account for lack of inherent direction; scores and penalties were calculated based on a log-likelihood function of the frequency of occurrence of these edges. Thus during path-finding, an  $\alpha$ -helix graph is treated the same as a backbone graph. Directionality in a resulting cover is established by consensus of the directed  $(i, i + 3)$   $H^\alpha-H^N$  edges.

The various experimental data sets display a range of difficulty for assignment; Tab. 1 roughly characterizes contributing factors. ZDOM and NS1 are small and highly  $\alpha$ -helical, but show significant chemical shift degeneracy in the  $C^\alpha$  and  $C^\beta$  dimensions and also high overlap in the HN dimension. Apart from this, the significant problem with NS1 dataset is the 5 entirely missing spin

<sup>4</sup>While chemical shift differences are the basis for scoring in all NMR assignment, the precise scoring model is not our focus here. Details of a complete Bayesian model are too extensive for this paper and will be described in a separate publication.

systems. This was also the case with BPTI dataset (8 missing spin systems). CSP, a  $\beta$ -sheet protein, had incomplete spectra and more extra spin systems. FGF, RNASEWT, and RNASEC6 were the toughest backbone datasets because of the significantly high noise, evident from the number of noise edges in the input NMR interaction graph. Though these two datasets had no entirely missing spin systems, the difficulty was increased by the large number (30) of extra spin systems. More details are available in the AUTOASSIGN paper [26]. HUGRX and VACGRX are fairly small, closely related proteins, with a relatively large  $\alpha$ -helical content. The spectra for these two proteins have quite different noise characteristics, in terms of particular missing edges (i.e. breaks), a key source of difficulty in obtaining a complete assignment. We note that JIGSAW provided only partial assignments of the  $\alpha$ -helical regions, due to internal breaks that fragment the helices and allow much more combinatorial swapping of fragments. CBF is a larger protein, but with a smaller total amount of  $\alpha$ -helical content. It is much more readily assigned due to having only one intra-helical break.

For each dataset, the cover-finding algorithm was run for 20,000 iterations and all unique solutions having scores within a constant fraction (20%) of the highest score were collected. We characterize the quality of our algorithm on this ensemble of results. Fig. 4 illustrates the scores in the ensemble for two example datasets. Note that the reference (published) solution need not have the highest score according to our metric, so the ensemble allows us to avoid committing to a single assignment subject to parameters in a particular scoring model. The first 3 columns of Tab. 2 summarize the growing results. The set of non-redundant, high-enough-scoring solutions tends to be quite small. Further, the average number of wrong edges selected per cover is also quite small, indicating that most members of the ensemble are typically quite good.

The ensemble further allows us to elucidate the shared structure of high-quality assignments. We expect the same edges, mostly correct, to show up repeatedly over the solution ensemble. Fig. 5 illustrates that this indeed holds true, even for extremely noisy datasets. Each plot shows the frequency of *missing* the correct sequential edges in the set of covers in the ensemble. That is, a bar of height  $h$  at position  $i$  in the chart indicates that the  $(i, i + 1)$  edge was correctly found in a fraction of  $1.0 - h$  members of the ensemble. These plots show that many correct edges are conserved in the ensembles. For clarity, the bars for prolines and entirely missing spin systems are not shown (they necessarily have a height of 1.0). More detailed analysis of these plots follows.

We begin with the analysis of the toughest datasets, in order to get insight into difficulties. RNASEC6 provides significant problems (correct edge missed in over half the covers) at residues 16, 69, and 104. To a large extent, these ambiguities are due to extra spin systems — the residues in question had good edges to extras, or succeeding residues had good edges from extras. Since these extras in turn were weakly connected to other residues, once the algorithm chose a wrong edge, there was less chance to correct it. Additional ambiguities were typically due to particularly high out-degree. But in many cases the algorithm was able to overcome these ambiguities. For example, the region from 65 to 69 had an average out-degree of about 3 and an average of 2 edges to extras, and yet the search for a high-scoring *globally-consistent* cover forced the correct choice in most of the members of the ensemble. Similar reasons for ambiguity exist for RNASEC6. Here we see that in spite of the noisiness of the data, most of the correct edges appear consistently. For example, only 3 edges (from 14, 69, and 104) were missing half the time, and this was due to high-scoring out edges from those nodes to extras. Similar arguments hold for CSP, where the ambiguities at 19, 25, and 41 are due to high out-

degree (degree of 6 at 19) or good edges to extras (2 edges each at 25 and 41). The FGF results, on the other hand, demonstrate the ability of the algorithm to overcome this type of ambiguity. Spin system 110 has 9 out-edges and 117 has 17, each with two edges to extras, but the correct edges appeared in 97% and 83% of the solutions. This is also the case at residue 20 in CSP, which has out-degree of 14 but yields a correct choice in 73% of the solutions. The remaining datasets (NS1, BPTI, and ZDOM) were relatively cleaner, and the algorithm was able to largely overcome most ambiguity. One interesting case was the interleaving of two correct fragments in NS1. Because of a good scoring edge (8,17) the two correct fragments  $\langle 2 \dots 8, 9 \rangle$  and  $\langle 11 \dots 17, 18, 19 \rangle$  appeared as  $\langle 2 \dots 8, 17, 18, 19 \rangle$  and  $\langle 11 \dots 16 \rangle$  and  $\langle 9 \rangle$ .

For the  $\alpha$ -helix dataset HUGRX, the highly problematic edges occur at spin systems 94 and 95. This was seen because of the fact that the spin systems 95 and 96 had very good edges from the end-point of another helix (ending at 9). Also the region 54–58 was highly noisy, with the symmetric  $H^N$  edges between the consecutive spin systems missing. The plot for CBF shows that the ambiguity is localized at the two regions corresponding to the end of one  $\alpha$ -helix and beginning of another. For the spin systems corresponding to these regions, the data was especially noisy, and the high number of missing  $H^\alpha(i, i + 3)$  correlations added to the ambiguity. VACGRX was the noisiest of the  $\alpha$ -helix datasets with 50% of the  $H^\alpha$  correlations in the  $\alpha$ -helices missing. Furthermore, there were significantly many good edges between spin systems of different  $\alpha$ -helices, and a number of breaks. A further indication of the noisiness of the data is the fraction of possible edges provided. For example, of the expected 27  $H^\alpha$  edges possible in VACGRX, only 13 were present, along with 103 noise ones. The situation was similar with the  $H^N$  edges.

The global consistency requirement prevents any particular solution from having too many incorrect edges. Tab. 2 characterizes the average number of incorrect edges over the different test cases. There is a strong correlation between the percentage of wrong edges out and measures of the dataset difficulty. For example, when average out-degree is high, then, as shown by our theoretical analysis, the algorithm must work harder to find the optimal solution. As a practical consequence, it finds more wrong solutions during the allotted time. Substantial numbers of extras and missings had the expected impact on the quality of solutions. For example, when the 4 poorly connected vertices for residues 54–58 in HUGRX  $\alpha$ -helix were removed (in fact, JIGSAW did not identify them), the performance improved to a mean of 2.8 wrong edges. In general, the impact of extras and missings is largely localized, as Fig. 5 demonstrates.

While this paper concentrates on connectivity information, as proof-of-concept, we explored the ability of MAPPER [12] to complete the assignment, by using sequence information to align the identified fragments to the primary sequence (refer again to the discussion of growing and aligning in the introduction). MAPPER takes as input short fragments of sequentially connected residues, along with  $C^\alpha$  and  $C^\beta$  chemical shifts (indicative of amino acid type) or other data about amino acid type. MAPPER first considers each fragment individually and lists all acceptable mappings and their scores (how well each spin system matches the expectation for the mapped residue). MAPPER can, for each fragment, give no acceptable mappings, a unique mapping, or multiple scored mappings. Given the ambiguity in amino acid type information, MAPPER typically gives numerous alignments for short fragments, but the increasing number of amino acid type constraints that must be satisfied by longer fragments drastically reduce the number of consistent mappings. Thus we expect *longer and correct* fragments

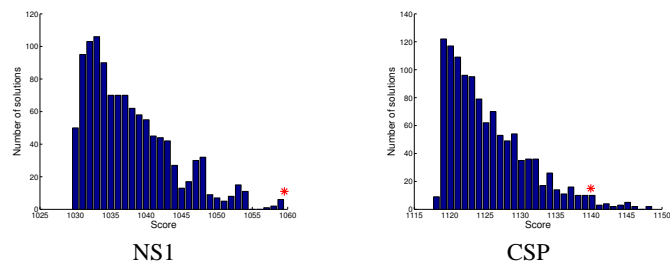


Figure 4: Example score distributions over ensembles. Reference solution scores (1059.53 for NS1 and 1139.94 for CSP) are marked.

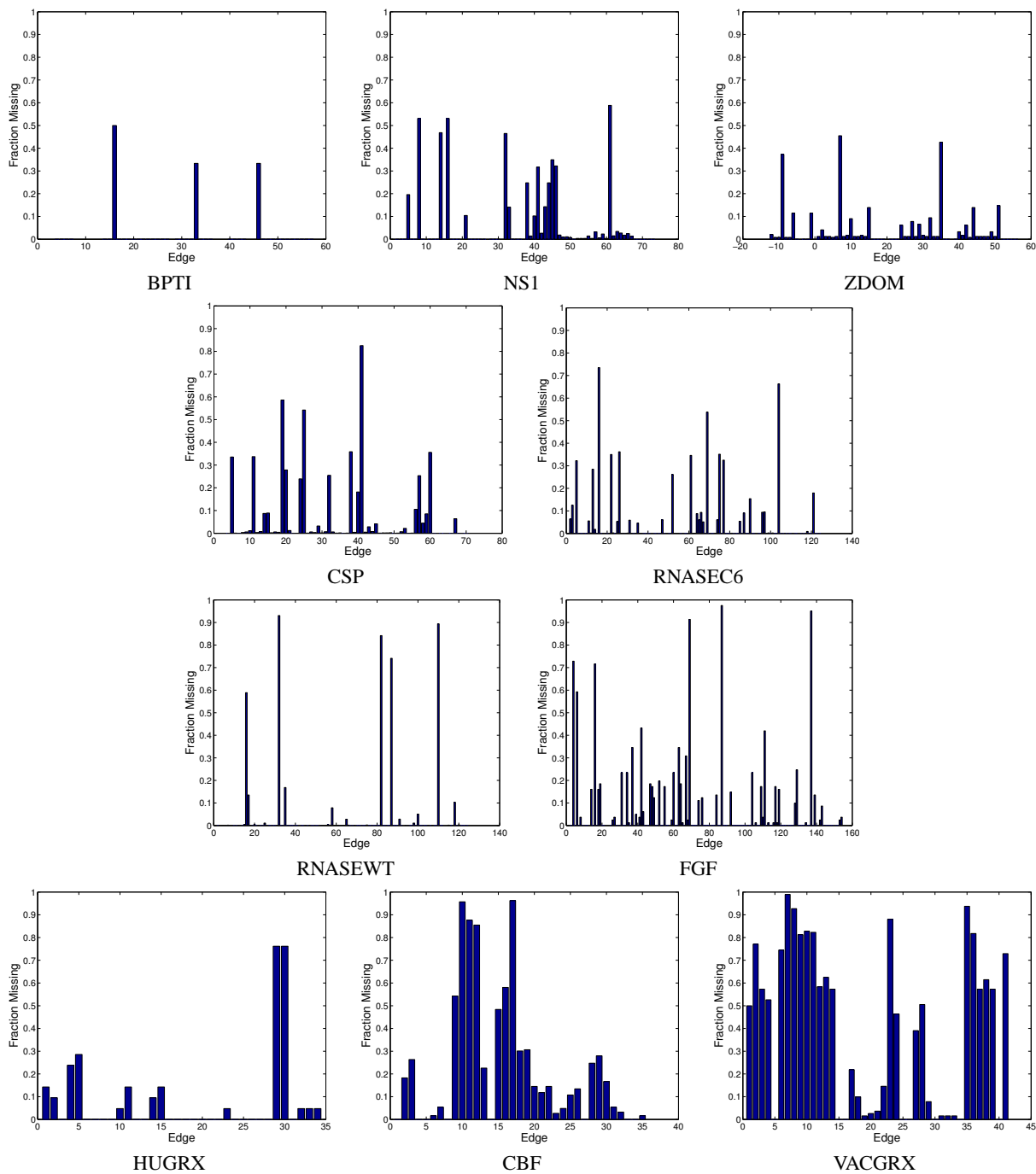


Figure 5: Frequency of *missing* the correct edge (indexed by “from” residue number), over the ensemble of solutions.

Dataset	Dimensions	Residues	Pro	Missing	Extra	Out degree	Wrong edges	Breaks
BPTI	$C^\alpha, C^\beta, H^\alpha$	58	4	8	5	0.90	4	4
NS1	$C^\alpha, C^\beta, H^\alpha$	73	1	5	1	1.30	22	0
ZDOM	$C^\alpha, C^\beta, H^\alpha$	69	3	1	1	1.50	38	0
CSP	$C^\alpha, C^\beta, H^\alpha$	68	2	0	4	1.72	50	0
RNASEC6	$C^\alpha, C^\beta, CO, H^\alpha$	124	4	0	30	2.07	193	1
RNASEWT	$C^\alpha, C^\beta, CO, H^\alpha$	124	4	0	30	2.19	209	12
FGF	$C^\alpha, C^\beta, CO, H^\alpha$	154	9	2	3	2.90	338	0
CBF $\alpha$ -helix	$H^N, H^\alpha$	36	NA	0	0	4.16	114	1
HUGRX $\alpha$ -helix	$H^N, H^\alpha$	38	NA	0	0	5.00	160	8
VACGRX $\alpha$ -helix	$H^N, H^\alpha$	42	NA	0	0	3.67	112	4

**Table 1: Summary of datasets, including factors making assignment difficult. The top 7 use traditional backbone experiments, while the bottom three use  $^{15}\text{N}$ -NOESY (only residues in  $\alpha$ -helices are considered in our tests). Node characteristics are summarized by the numbers of prolines and entirely missing spin systems, which cause breaks in a cover; and the number of extra spin systems (i.e. not corresponding to any residue), which can mislead the assignment by filling in for correct or missing spin systems. Edge characteristics are summarized by the mean out degree over the entire graph, including both correct and noise edges; the total number of wrong edges; and the number of missing edges (breaks), not counting those due to missing spin systems and prolines. All datasets also have some extra and missing peaks within individual spin systems; these aren’t detailed here, as they only locally affect the scoring of an edge.**

Dataset	Ensemble size	Wrong grown edges	Wrong mapped edges	Assigned residues
BPTI	6	1.00	0.5	34.66
NS1	816	4.60	1.29	28.3
ZDOM	126	2.90	0.7	43
CSP	1169	4.39	1.5	41.3
RNASEC6	8852	7.28	0.95	93.88
RNASEWT	3327	11.39	1.5	50.08
FGF	81	18.27	2.02	44.95
CBF $\alpha$ -helix	111	5.53	NA	NA
HUGRX $\alpha$ -helix	19	9.40	NA	NA
VACGRX $\alpha$ -helix	785	16.60	NA	NA

**Table 2: Growing and mapping results. Our algorithm generates a number of solutions (20,000) and keeps an ensemble of acceptable solutions with scores near (within 20% of) the best found. Results are averaged over these ensembles. Error rate is summarized by the mean number of wrong edges per cover. Growing alone yields a surprisingly good error rate, due to the requirement of global consistency of a cover. A simple application of MAPPER then filters the set of fragments, keeping only long enough ones that unambiguously map to the primary sequence. The error rate drops, and a large number of the residues are unambiguously assigned.**

to have unique mappings and incorrect fragments to have none or poor-scoring mappings. Based on this reasoning, along with our claim that we expect the sequential covers in the ensemble to contain long chunks of correct fragments, we apply MAPPER to each sufficiently-long fragment in the ensemble, and we keep those fragments that have a unique mapping. We expect the union of the long, mapped fragments from a solution to provide a reasonable global assignment.

We applied this basic heuristic to the output ensembles from each backbone dataset (we did not attempt to apply MAPPER to the particularly weak amino acid type information provided by JIGSAW from  $^{15}\text{N}$ -TOCSY data). Tab. 2 shows that the combined approach — growing ambiguous fragments with our algorithm and mapping/filtering them with MAPPER — effectively assigns a large number of residues with a low error rate. There were no false negatives (i.e. MAPPER correctly aligned correct fragments). The false positive rate is reduced by at least half compared to growing alone. It is still possible to map incorrect edges to the primary sequence, particularly when an extra spin system with several missing peaks serves as a “wild card” replacing the correct spin system or a missing spin system. The average number of residues correctly mapped

was reasonably large, more than 50% of the alignable length (actual length minus number of missing, including Pro) except for NS1 (42%) and FGF (32%). Results in Tab. 2 are obtained with a minimum fragment length of 6; when that threshold is raised to 8, yielding even more constraint, the error rate drops to 0. Naturally, the average number of assigned residues drops, too, but the experimentalist might consider this a trade-off worth making. This approach would lend itself naturally to further heuristics, including iterative growing/aligning combining information across the ensemble, considering global alignments or ambiguous alignments, etc. However, our focus here is on the study of the information content available from sequential connectivities, and we believe this simple heuristic provides sufficient evidence to justify the analysis of connectivity information separate from mapping information.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we developed a novel random-graph theoretic framework for algorithmic analysis of NMR sequential assignment. We then gave a simple and natural randomized algorithm for finding an optimum sequential cover, and presented a probabilistic analysis of the algorithm under our random graph framework. Our analysis

shows that our algorithm can tolerate a relatively large ambiguity while still giving polynomial time expected performance. To study the algorithm's performance in practice, we tested it on experimental data sets from a variety of proteins and experimental set-ups. The algorithm was able to overcome significant noise and local ambiguity and consistently identify significant sequential fragments. These preliminary results are encouraging, and our next goal, for which we already conducted preliminary tests, is to more tightly integrate the connectivity phase (solved by our algorithm) and the aligning phase (solved by MAPPER, for example). In this process, we also plan to incorporate rigorous statistical scoring techniques, in order to support inference about the results. Finally, we plan to extend the search to higher-order connectivities, as in the  $\beta$ -sheet patterns utilized by JIGSAW.

## Acknowledgment

This work benefited greatly from discussions with Olga Vitek, Dept. of Statistics, Purdue University, on uncertainty in NMR data, as well as her assistance in compiling and properly scoring the input NMR interaction graphs. We would like to thank Drs. Gaetano Montelione and Hunter Moseley of the Center for Advanced Biotechnology and Medicine, Rutgers University, for providing access to peak list files and the AutoPeak and AutoAssign programs, and Dr. John H. Bushweller, University of Virginia, for providing the NOESY data sets. This work was supported in part by a grant to CBK from the National Science Foundation (IIS-0237654).

## 6. REFERENCES

- [1] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley, 2000.
- [2] D. Angluin and L. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18:155–193, 1979.
- [3] H. S. Atreya, S. C. Sahu, K. V. R. Chary, and G. Govil. A tracked approach for automated NMR assignments in proteins (TATAPRO). *J. Biomol. NMR*, 17:125–136, 2000.
- [4] C. Bailey-Kellogg, A. Widge, J. J. Kelley III, M. J. Berardi, J. H. Bushweller, and B. R. Donald. The NOESY Jigsaw: automated protein secondary structure and main-chain assignment from sparse, unassigned NMR data. In *The 4th Int'l Conf. on Computational Molecular Biology (RECOMB)*, pages 33–44, April 2000.
- [5] C. Bailey-Kellogg, A. Widge, J. J. Kelley III, M. J. Berardi, J. H. Bushweller, and B. R. Donald. The NOESY Jigsaw: automated protein secondary structure and main-chain assignment from sparse, unassigned NMR data. *J. Comp. Biol.*, 7:537–558, 2000.
- [6] C. Bartels, P. Güntert, M. Billeter, and K. Wüthrich. GARANT – a general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra. *J. Comp. Chem.*, 18(1):139–149, 1997.
- [7] B. Bollobas. *Random Graphs*. Cambridge University Press, 2001.
- [8] A. Broder, A. Frieze, and E. Shamir. Finding hidden hamilton cycles. *Random Structures and Algorithms*, 5:395–410, 1994.
- [9] N. E. G. Buchler, E. P. R. Zuiderweg, H. Wang, and R. A. Goldstein. Protein heteronuclear NMR assignments using mean-field simulated annealing. *J. Mol. Resonance*, 125:34–42, 1997.
- [10] J. Cavanagh, W. J. Fairbrother, A. G. Palmer III, and N. J. Skelton. *Protein NMR Spectroscopy*. Academic Press, 1996.
- [11] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, second edition, 1992.
- [12] P. Güntert, M. Saltzmann, D. Braun, and K. Wüthrich. Sequence-specific NMR assignment of proteins by global fragment mapping with program Mapper. *J. Biomol. NMR*, 17:129–137, 2000.
- [13] T. K. Hitchens, J. A. Lukin, Y. Zhan, S. A. McCallum, and G. S. Rule. MONTE: An automated Monte Carlo based approach to nuclear magnetic resonance assignment of proteins. *J. Biomol. NMR*, 25:1–9, 2003.
- [14] M. Leutner, R. M. Gschwind, J. Liermann, C. Schwarz, G. Gemmecker, and H. Kessler. Automated backbone assignment of labeled proteins using the threshold accepting algorithm. *J. Biomol. NMR*, 11:31–43, 1998.
- [15] K.-B. Li and B. C. Sanctuary. Automated resonance assignment of proteins using heteronuclear 3D NMR. 1. Backbone spin systems extraction and creation of polypeptides. *J. Chem. Info. and Comp. Sci.*, 37:359–366, 1997.
- [16] G. Lin, D. Xu, Z.-Z. Chen, T. Jiang, and Y. Xu. A branch-and-bound algorithm for assignment of protein backbone NMR peaks. In *First IEEE Bioinformatics Conference*, pages 165–174, 2002.
- [17] J. A. Lukin, A. P. Gove, S. N. Talukdar, and C. Ho. Automated probabilistic method for assigning backbone resonances of ( $^{13}\text{C}$ ,  $^{15}\text{N}$ )-labeled proteins. *J. Biomol. NMR*, 9:151–166, 1997.
- [18] G. T. Montelione, D. Zheng, Y. J. Huang, K. Gunsalus, and T. Szyperski. Protein NMR spectroscopy in structural genomics. *Nature America*, 2000. Suppl.
- [19] H. N. B. Moseley and G. T. Montelione. Automated analysis of NMR assignments and structures for proteins. *Curr. Opin. Struct. Biol.*, 9:635–642, 1999.
- [20] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [21] S. Neal, A.M. Nip, H. Zhang, and D.S. Wishart. Rapid and accurate calculation of protein  $1\text{h}$ ,  $13\text{c}$  and  $15\text{n}$  chemical shifts. *Journal of Biomolecular NMR*, 26:215–240, 2003.
- [22] B. R. Seavey, E. A. Farr, W. M. Westler, and J. Markley. A relational database for sequence-specific protein NMR data. *J. Biomol. NMR*, 1:217–236, 1991. <http://www.bmrb.wisc.edu>.
- [23] D.L. Di Stefano and A.J. Wand. Two-dimensional  $^1\text{H}$  NMR study of human ubiquitin: a main-chain directed assignment and structure analysis. *Biochemistry*, 26:7272–7281, 1987.
- [24] R. C. Stevens, S. Yokoyama, and I. A. Wilson. Global efforts in structural genomics. *Science*, 294(5540):89–92, 2001.
- [25] K. Wüthrich. *NMR of Proteins and Nucleic Acids*. John Wiley & Sons, 1986.
- [26] D.E. Zimmerman, C.A. Kulikowski, Y. Huang, W. Feng, M. Tashiro, S. S. Shimotakahara, C. Chien, R. Powers, and G. T. Montelione. Automated analysis of protein NMR assignments using methods from artificial intelligence. *J. Mol. Biol.*, 269:592–610, 1997.