BIOINFORMATICS

Vol. 00 no. 00
Pages 1–8

An Efficient Randomized Algorithm for Contact-Based NMR Backbone Resonance Assignment

Hetunandan Kamichetty [a], Chris Bailey-Kellogg [b*], Gopal Pandurangan [a*]

[a] Department of Computer Science, Purdue University, West Lafayette, IN 47907. [b] Department of Computer Science, Dartmouth College, Hanover, NH 03755.

ABSTRACT

Motivation: Backbone resonance assignment is a critical bottleneck in studies of protein structure, dynamics, and interactions by nuclear magnetic resonance (NMR) spectroscopy. A minimalist approach to assignment, which we call "contact-based," seeks to dramatically reduce experimental time and expense by replacing the standard suite of through-bond experiments with the through-space (NOESY) experiment. In the contact-based approach, spectral data are represented in a graph with vertices for putative residues (of unknown relation to the primary sequence) and edges for hypothesized NOESY interactions, such that observed spectral peaks could be explained if the residues were "close enough." Due to experimental ambiguity, several incorrect edges can be hypothesized for each spectral peak. An assignment is derived by identifying consistent patterns of edges (e.g., for α-helices and β-sheets) within a graph, and mapping the vertices to the primary sequence. The key algorithmic challenge is to be able to uncover these patterns even when they are obscured by significant noise.

Results: This paper develops, analyzes, and applies a novel algorithm for the identification of polytopes representing consistent patterns of edges in a corrupted NOESY graph. Our randomized algorithm aggregates simplices into polytopes and fixes inconsistencies with simple local modifications, called rotations, that maintain most of the structure already uncovered. In characterizing the effects of experimental noise, we employ an NMR-specific random graph model in proving that our algorithm gives optimal performance in expected polynomial time, even when the input graph is significantly corrupted. We confirm this analysis in simulation studies with graphs corrupted by up to 500 percent noise. Finally, we demonstrate the practical application of the algorithm on several experimental β-sheet data sets. Our approach is able to eliminate a large majority of noise edges and uncover large consistent sets of interactions.

Availability: Our algorithm has been implemented in platform-independent Python code. The software can be freely obtained for academic use by request from the authors.

Contact: cbk@cs.dartmouth.edu; gopal@cs.purdue.edu

1 INTRODUCTION

Nuclear Magnetic Resonance (NMR) spectroscopy is a key experimental technique for studying protein structure, dynamics, and interactions in physiological conditions. Efforts in *structural genomics* seek to conduct these studies at a massive scale (Montelione et al., 2000; Stevens et al., 2001), but are rate-limited by the necessity of interpreting the spectral data provided by NMR experiments. One such interpretation bottleneck is that of determining the *backbone resonance assignment*, which specifies the mapping from backbone atoms to their seemingly arbitrary but relatively unique identities in the spectra (Fig. 1). While substantial progress has been made in traditional approaches to backbone resonance assignment (e.g., Zimmerman et al., 1997; Moseley and Montelione, 1999; Lin et al., 2002; Vitek et al., 2004, 2005), some investigators are pursuing alternative *minimalist* approaches that seek to reduce the experimental complexity, circumvent traditional barriers to interpretation, and open the door to higher-throughput, lower-cost structural studies (e.g., Stefano and Wand, 1987; Nelson et al., 1991; Bailey-Kellogg et al., 2000; Erdmann and Rule, 2002; Grishaev and Llinás, 2002; Langmead and Donald, 2004).

This paper pursues an algorithmic basis for a minimalist backbone assignment protocol, which we call *contact-based assignment*, that is centered on the through-space NOESY (nuclear Overhauser enhancement spectroscopy) experiment (Stefano and Wand, 1987; Nelson et al., 1991; Bailey-Kellogg et al., 2000; Erdmann and Rule, 2002) rather than the standard suite of sequential backbone experiments. Contact-based assignment turns "upside down" the standard approach of first deriving a backbone assignment and using it in assigning the NOESY and solving for the structure. It instead analyzes unassigned backbone NOESY data in order to derive the backbone resonance assignment. One successful contact-based approach, Jigsaw (Bailey-Kellogg et al., 2000), focused on assignment within secondary structure elements, which produce regular patterns in graph representations of NOESY data (Fig. 2) due to biophysical and geometric constraints. Identified pattern instances can subsequently be aligned against the primary sequence, according to spectral "fingerprints" of amino acid type, thereby producing an assignment. Jigsaw needed data from only four straightforward NMR experiments that require only days of spectrometer time and relatively cheap and easy ^{15}N-labeled (rather than ^{13}C-^{15}N-labeled) protein. It was successfully applied to experimental spectra for three different proteins.

While contact-based assignment has displayed much potential, work so far has not demonstrated how to deal with the combinatorial explosion due to experimental ambiguity—there are several mutually-exclusive edges potentially explaining each spectral peak, and we want to identify relatively large consistent sets (α-helices and β-sheets). This paper thus develops and analyzes an algorithm that efficiently uncovers consistent edge patterns in the presence of substantial noise. Consistent sets of edges are represented as simplices, which are aggregated into polytopes that explain the experimental data and are consistent with biophysical and geometric constraints. While the identification of large polytopes in general

*to whom correspondence should be addressed.

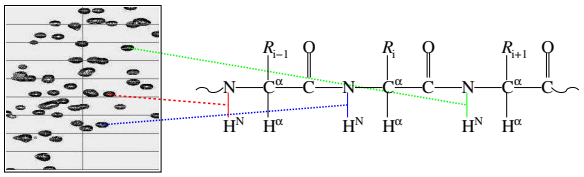

Fig. 1. HSQC spectrum. The axes indicate H^N and N chemical shifts, so that a peak corresponds to a bonded H^N–N atom pair with those shifts. However, the correspondence (assignment) between chemical shifts and atoms is unknown.

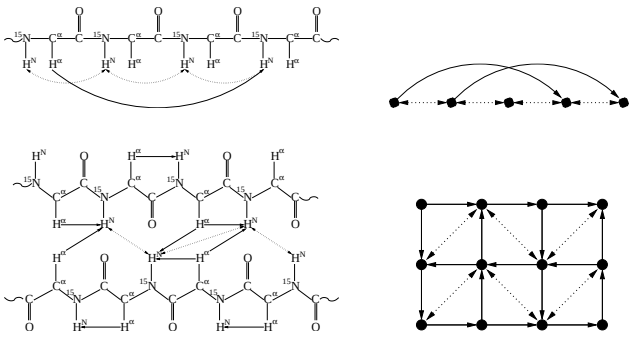

Fig. 2. NOESY interaction graphs: (top) α-helix and (bottom) β-sheet; (left) actual atomic interactions and (right) graph representation. Due to the regular geometry of these secondary structure elements, an ^{15}N-NOESY detects regular patterns of interactions ("contacts") between H^N atoms and both H^N (dotted line) and H^α (solid line) partners. These atomic-level interactions can be collected into a graph in which vertices represent residues and edges represent hypothesized H^N-H^N and H^α-H^N interactions between the respective atoms of the residues. These graphs show perfect patterns, but graphs for experimental NMR data contain many extra incorrect edges and some missings.

is NP-hard, as it involves subgraph isomorphism, our algorithm employs a key insight to work effectively: *in searching for good structures, the best ones tend to share a lot of substructure*. In branching-based searches (Bailey-Kellogg et al., 2000), such shared substructure can appear on many different branches (e.g., multiple different polytopes each adding a common simplex and growing from that simplex in the same fashion). Thus, instead of backtracking upon discovering an inconsistency, our algorithm employs a much more efficient local fix with a generalized simplex rotation step that moves a simple or small number of simplices to a new context, while leaving most of the structure intact. This *reuse*-based approach avoids wasteful undoing and redoing and tends to produce large, correct polytopes. This algorithm significantly generalizes our earlier approach to finding paths for traditional sequential assignment (Bailey-Kellogg et al., 2005), in order to handle higher-order patterns like those appropriate for β-sheets (Fig. 2).

Contribution: Our algorithm represents the first approach that gives optimal performance in expected polynomial time for the problem of uncovering secondary structures (and thereby backbone assignment) in significantly corrupted NOESY graphs. In addition to analyzing its expected behavior, we demonstrate and characterize its effectiveness in simulation studies with up to 500 percent noise, and experimental datasets for β-sheet regions of several proteins.

2 MATERIALS AND METHODS

We first summarize the underlying graph representation of the input NMR data, along with a random graph model that allows us to reason about the algorithm and conduct simulation studies. We then present our randomized simplex rotation algorithm, discussing the general mechanism and instantiating for the β-sheet case.

2.1 Graph Representation of NMR Data

NMR spectra capture magnetic interactions between atoms as peaks in $\mathbb{R}^2$ or $\mathbb{R}^3$, where each dimension indicates the coordinates (resonance frequencies, in units called *chemical shifts*) of one of the interacting atoms (see again Fig. 1). In the ^{15}N-edited spectra employed here, each peak is generated by interactions between an ^{15}N and H^N in a residue, and possibly an additional ^{1}H atom in the same or another residue. We assume that the spectra have been processed and the peaks picked, thereby yielding lists of peak maxima and intensities.

Following Bailey-Kellogg et al. (2000, 2005), we represent the input NMR spectra in a labeled, weighted, directed graph $G = (V, E)$, called a *NOESY interaction graph* (see Fig. 2). The vertices correspond (via an unknown mapping) to residues (or to noise, for extras). An edge $e = (v_1, v_2) \in E$ represents a possible explanation — interaction between atoms of v_1 and v_2 — for a NOESY peak. A vertex is labeled with a secondary structure type, either α or β. An edge is labeled with an *interaction type*, either H^N or H^α, and a *match score*, estimating the confidence in the edge as an explanation for the peak.

A NOESY interaction graph is constructed from peak lists for four ^{15}N-edited spectra as follows. *(Vertices)* The HSQC spectrum establishes vertices by identifying a set of chemical shift pairs for bonded H^N and ^{15}N. There is one such pair per residue, ignoring side-chain amide groups. The HNHA and TOCSY spectra augment a vertex by identifying the chemical shift of its H^α by an interaction with the anchor H^N-^{15}N. The HNHA further allows labeling vertices by hypothesized secondary structure type: the *J coupling constant*, $^3J_{H^N H^\alpha}$, is correlated with the ϕ bond angle of a residue and thus is characteristically different for α-helices and β-sheets. *(Edges)* The NOESY spectrum establishes edges by indicating possible interactions among N-H^N-^{1}H triplets. An edge is placed between two vertices when the N and H^N chemical shifts of a NOESY peak match the those of the first vertex (by reference to the HSQC) and the ^{1}H chemical shift matches that of either the H^N (by reference to the HSQC) or H^α (by reference to the HNHA and TOCSY) of the second vertex. The match type labels the interaction as either H^N or H^α accordingly. The match score evaluates the degree of confidence in the match, according to the similarity in chemical shift. It is worth noting that in the ^{15}N-only approach, the H^N interactions are symmetric, but the H^α interactions are only in one direction. For simplicity, we assume here that symmetric edges have been merged into a single edge with a joint score.

Chemical shift degeneracy is a key source of noise corrupting a NOESY interaction graph. Uncertainty in the measured chemical shifts of the protons leads to ambiguity in matches, and thus the construction of spurious noise edges. These noise edges are not randomly distributed, but in fact can be modeled by a random graph model that properly captures their correlation structure (Bailey-Kellogg et al., 2005). In particular, note that when two vertices have atoms that are fairly similar in chemical shift, by the above

construction they will tend to share edges, since a proton chemical shift matching the one (within some tolerance) will also likely match the other. This relationship can be modeled by treating atoms as being sorted in chemical shift order, i.e., in some random permutation π according to their magnetic environments as probed by the experiment. We then consider as ambiguous all atoms within some "window" of size w around a particular atom. Noise edges for an edge (u, v) are introduced for each u' in the window of width w around $\pi(u)$. Thus, there are w noise edges for every correct edge. Note that this definition of w is slightly different from that of Bailey-Kellogg et al. (2005), where it describes half the width of the window. Typical scoring rules (e.g., Zimmerman et al., 1997; Güntert et al., 2000; Vitek et al., 2004, 2005) compare absolute or squared difference in chemical shift. Except for noise (reasonably modeled as Gaussian), the correct edge should match exactly and have the best score. Thus with respect to the permutation model, scores are a function of distance within the permutation, perturbed by a zero mean Gaussian distribution.

In practice there are several other sources of noise, including missing and extra vertices and incorrect secondary structure labels. In order to maintain our focus on the problem of dealing with significant numbers of noise edges, we ignore here these other sources of noise and their effects on the assignment algorithm, and save such analysis for future research.

2.2 Randomized Simplex Rotation Algorithm

A NOESY interaction graph contains geometric information (Fig. 2). Due to the rapid fall-off of the nuclear Overhauser effect with distance (proportional to $1/r^6$), detected interactions are between protons at most 6 Å apart. Thus NOESY peaks correspond to "contacts" in a protein structure. The regularity of secondary structure elements, by geometric and biophysical constraints, produces regular patterns in a NOESY interaction graph (Fig. 2). The patterns can naturally be represented with simplices aggregated into polytopes (Erdmann and Rule, 2002), as follows.

A *k-simplex* is defined by a set of $k + 1$ residues that are pairwise adjacent in the input graph; here, we consider points (k=0), segments (1), triangles (2), and tetrahedra (3). A k-simplex has $(k - 1)$-simplices as its *proper faces*; the *faces* of the simplex include its proper faces, their proper faces, and so forth. For example, the proper faces of a triangle are segments and the proper faces of the segments are points; the faces of the triangle include both the segments and the points. Two k-simplices whose intersection is a $(k - 1)$-simplex are *adjacent*. A set of k-simplices that are transitively adjacent define a *k-polytope*. For a polytope to be well-formed, an intersection between any two simplices must be a face of each, and each $(k - 1)$-simplex must be a proper face of at most two k-simplices. For example, in a β-sheet, two triangles can only share one edge and each edge can appear in at most two triangles.

With this simplex-based representation, we formulate the assignment problem as follows:

- *Input*: A NOESY interaction graph $G = (V, E)$ and all simplices $\mathcal{S}$ formed from edges in E.
- *Output*: An optimal consistent set of k-polytopes formed from a subset of the input k-simplices.
- *Consistency*: Polytopes do not intersect; and they obey any additional constraints (e.g., canonical β-sheet structure as in Fig. 2).

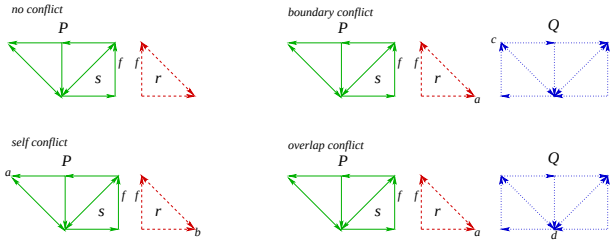

Fig. 3. Polytope growth in 2D: adding simplex (triangle) r to polytope P via shared edge f, in the presence of polytope Q. Several cases can result: no conflicts (f is the only intersection), a self conflict ($a = b$), a boundary conflict ($a = c$), or an overlap conflict ($a = d$). A conflict can be resolved by rotating a simplex to its new context and performing local modifications to the polytope(s).

- *Optimality*: Minimal score for the polytope set, defined as the total score of edges, plus a penalty for each vertex not in any polytope.

We focus on simplices and polytopes of a single dimension (homogeneous); the discussion section mentions possible extensions.

Our algorithm maintains a set $\mathcal{P}$ of consistent polytopes, as well as a free set $\mathcal{S}$ of simplices that don't appear in any of the current polytopes. At each iteration, the algorithm either initiates a new polytope or attempts to grow from a polytope $P \in \mathcal{P}$ that has a simplex s with a free proper face f. Growth adds to P a simplex r from the free set that is adjacent to s on f. Fig. 3 illustrates the cases. If the only intersection between r and $\mathcal{P}$ is f, then we simply add f to P. If r intersects P in some other (not necessarily proper) face, or r intersects some other polytope $Q \in \mathcal{P}$, then there is a *conflict*. When faced with a conflict, we can fix up the polytope(s) by moving a simplex out of its current context into a new one, and performing any other modifications necessary to restore consistency (next paragraph). In deference to the terminology employed by randomized algorithms for finding Hamiltonian paths (Angluin and Valiant, 1979), we call such simplex moves *rotations*. Another way to handle a conflict is simply to decide not to grow with the chosen simplex. The choice between keeping the original vs. growing and rotating is made randomly with probabilities proportional to the scores of the polytopes. Thus the algorithm naturally encourages improving the current interpretation but may occasionally be more destructive in order to change direction.

To handle a boundary conflict, the two polytopes are merged by way of the new simplex. To ensure that the resulting polytope is well-formed, we may need to add another "bridging" simplex to complete the simplex adjacency relationship. In Fig. 3, such a bridging simplex would share the diagonal edge with r and link the top strand of P with the bottom strand of Q (flipping Q upside down). To handle a self conflict or overlap conflict, all simplices intersecting the added simplex are removed. If this leads to a polytope that is not well-formed (e.g., the removal leaves a hole), then the polytope is broken into sub-polytopes by removing additional simplices. If the removed simplices are contiguous, we create new polytopes from them directly, rather than adding them back to the free pool $\mathcal{S}$. While it is possible that a rotation step might cause the loss of a large number of triangles, in practice (see the results section) we found that the we never lost more than 6 correct triangles.

We also show in the analysis section that the algorithm will still make progress if the probability of such an event occurring is low.

The generality of our grow/rotate mechanism allows identification of arbitrary regular polytopes. In fact, the randomized sequential cover algorithm of Bailey-Kellogg et al. (2005) can be seen as a special case of this general framework. That specialization is sufficient to uncover α-helices in a noisy NOESY graph; while there are non-sequential $(i, i+3)$ edges in an α-helix, the main structure is that of a sequential path, and the "jumps" can be used simply to provide extra support for a path. Thus for the rest of this paper we will focus on the β-sheet case (i.e., 2D), although our method and analysis can be generalized to other polytopes in finding 3D structures (see the discussion section).

Algorithm 1 provides pseudocode for the β-sheet instantiation. The algorithm repeats the basic growing/rotating process until either all the data are explained (i.e., each peak is represented by one edge in $\mathcal{P}$) or until a maximum number of iterations is reached. A significant implementation detail is that, in order to efficiently identify intersections upon simplex growth, we maintain vertex coordinates in 2D (this can be done uniquely up to rigid-body motions). This requires updating coordinates of an entire polytope upon merging, but we found in practice this to be a negligible cost in time. In addition, we enforce adherence to the canonical β-sheet pattern (see Fig. 2); the discussion section describes possible relaxation of this rule.

Data: NOESY interaction graph $G = (V, E)$ and all triangles $\mathcal{S}$ formed from edges in E
Result: Set $\mathcal{P}$ of polytopes (triangulations)
initialize $\mathcal{P}$ with $\emptyset$;
repeat
  choose at random a vertex $v \in V$, such that either $v \notin \mathcal{P}$ (unvisited) or v has at least one edge appearing in only one triangle in a polytope $P \in \mathcal{P}$;
  if *v is unvisited* **then**
    choose a triangle $r \in \mathcal{S}$ with v as a vertex, with probability proportional to its score;
  else
    choose an edge e incident on v which appears in only one triangle s in P;
    choose a triangle $r \in \mathcal{S} - \{s\}$ that also has edge e, with probability proportional to its score;
  end
  if *r conflicts with triangles C in $\mathcal{P}$* **then**
    randomly choose between r and C with probability proportional to r's score and the sum of scores of C;
    if *choose r* **then**
      remove triangles C from their polytopes (breaking the polytopes if necessary);
      $\mathcal{S} \leftarrow \mathcal{S} \bigcup C$;
      add r and its vertices and edges to P (merging polytopes if necessary);
    end
  else
    if *v is not in a polytope* **then**
      create a new polytope P containing v;
    end
    add r and its vertices and edges to v's polytope P (merging polytopes if necessary);
    $\mathcal{P} \leftarrow \mathcal{P} \bigcup \{P\}$;
  end
until *$\mathcal{P}$ explains all data, or MAXITER iterations performed*;
return $\mathcal{P}$;

Algorithm 1: Randomized simplex rotation algorithm.

3 RESULTS

In order to gain insights into the behavior of our algorithm, we study it from three different perspectives—theory, simulation, and experiment. We first analyze the amount of noise our algorithm can tolerate under an NMR-specific random graph model that captures the structure of chemical shift degeneracy, the key source of edge ambiguity. We show that our algorithm can tolerate a large amount of ambiguity while still giving optimal performance in expected polynomial time. We then employ a set of simulation studies, in which we vary the degree of ambiguity, to show that the algorithm can recover nearly an entire structure correctly in the presence of significant noise. We use insights from these simulations to develop a method for obtaining a posterior confidence score for edges, and show how to use this in handling missing edges. Finally, we present our results on experimental data sets, showing that our approach is able to eliminate a large majority of noise edges and uncover large consistent sets of interactions.

3.1 Algorithm Analysis

To gain insight into the performance of our algorithm, we analyze it in a simplified setting under a few assumptions. We assume that the correct graph is a actually a β-sheet torus—each triangle is in the interior and has a full complement of neighbors. Thus we need not list all the special cases that arise due to boundary effects. We note that the vertices have a constant maximum degree Δ, due to packing constraints. We assume a scoring model for the edges such that the expected score for a correct edge is larger than that for a noise edge (as is the case under our Gaussian model). Thus, if we try to grow from a correct edge in a polytope, we choose a correct triangle with probability $> 1/2$, since with high probability (i.e., $1 - o(1)$) there is at most one wrong triangle per correct edge (cf. Lemma 1). Finally, we assume that there is some function f such that any rotation removes k correct triangles with probability at most $f(k)$. This function is determined by the quality of scores: the better the quality of scores, the smaller the probability of breaking correct triangles. We can precisely characterize this probability with respect to a particular scoring model (e.g., Vitek et al., 2004, 2005), but the exact form is not crucial to our analysis. In practice we found $f(k) = 0$, for all $k \geq 6$. We will denote by $M = \sum_{k=1}^{\infty} k f(k)$ and $V = (\sum_{k=1}^{\infty} k^2 f(k)) - M^2$, the mean and variance of the number of correct triangles removed in an iteration. As we have observed in practice, we assume that M and V are bounded by some fixed constant (independent of the size of the graph).

We account for the correlated noise structure of NMR interaction graphs by employing the random graph model of Bailey-Kellogg et al. (2005), described in Section 2.1. Given a window width w defining the amount of chemical shift degeneracy, a correct graph is corrupted by generating for each edge a set of noise edges replacing the original from-node with vertices within distance w of it in a random vertex permutation (modeling chemical shift order).

We now proceed to characterize the algorithm's performance on a corrupted version of a graph with n correct triangles, in terms of the noise parameter w. We show that, if $w = o(n^{1/4})$, then the algorithm makes progress towards finding the n-triangle polytope and is expected to find it in polynomial time (as in a "gambler's ruin" problem (Grimmett and Stirzaker, 1992)). Thus the algorithm can tolerate a relatively large amount of noise.

We model the progress of the algorithm, over its iterations, as a random walk on a line from 0 to n, with position i indicating that the current polytope has i of the correct triangles. Since we have a β-sheet torus, we can assume without loss of generality that in each iteration, the algorithm chooses to grow from a particular triangle via one of its edges. We also assume, without loss of generality, that it starts from a correct edge in the very first iteration. We call the state of the algorithm at each iteration *progressive* if the chosen triangle is correct, and *regressive* otherwise. To show that the algorithm makes progress, we bound the of number of correct triangles that can be removed in any state.

LEMMA 1. *Suppose $w = o(n^{1/4})$. Given an edge e, the probability p of an incorrect triangle existing with e as one of its edges is $1 - (1 - \frac{\Delta w}{n})^{\Delta w} \approx (\Delta w)^2/n$, where Δ (a constant) is the maximum number of correct edges from any vertex. Furthermore, the probability that there exists a correct edge with more than one wrong triangle is $\approx \frac{(\Delta w)^4}{n} = o(1)$.*

PROOF. Let edge $e = (i, j)$. The number of incorrect edges to j is Δw. An incorrect triangle with edge e exists if for any of the wrong edges (k, j), there exists an incorrect edge (i, k). The probability that this edge exists is $\Delta w/n$ by our random graph model. Therefore, the probability that no such triangle exists is $p_0 = (1 - \frac{\Delta w}{n})^{\Delta w}$. The probability p that it does is therefore $1 - (1 - \frac{\Delta w}{n})^{\Delta w} \approx (\Delta w)^2/n$ (for large n). The probability that exactly one such triangle exists is $p_1 = \Delta w(1 - \frac{\Delta w}{n})^{(\Delta w - 1)}(\frac{\Delta w}{n})$. The probability that at least two wrong triangles exist is therefore $1 - (p_0 + p_1) \approx 1 - (1 - \frac{(\Delta w)^2}{n} + \frac{(\Delta w)^2}{n}(1 - \frac{(\Delta w)^2}{n})) = \frac{(\Delta w)^4}{n^2}$. By the union bound, the probability of at least two wrong triangles for any correct edge is $\frac{(\Delta w)^4}{n}$. □

THEOREM 1. *Let M and V be the mean and variance of the number of correct triangles removed in an iteration. As assumed earlier, let M and V be bounded above by a constant (independent of n). If $w = o(n^{1/4})$ and $0 \leq M \leq 1/3$, then the algorithm finds all n correct triangles in expected polynomial time.*

PROOF. In a progressive state the expected number of correct triangles added is at least $1/2$ since, according to our assumption, with probability at least $1/2$ we add a correct triangle, and the probability of removing another correct triangle conditioned on the event of adding a correct triangle is zero (e.g., Pandurangan, 2005). The expected number of correct triangles lost in a progressive state is at most $(1/2)M$. Hence the expected gain in correct triangles in a progressive state is at least $(1/2)(1 - M)$.

Now consider a regressive state. We cannot add a correct triangle, but we can bound the number of correct triangles lost in a sequence of regressive states before returning to a progressive state. If $w = o(n^{1/4})$, the probability that we enter a regressive state from any state can be bounded by p (by Lemma 1) and hence the expected number of successive regressive states before we return to a progressive one is bounded by $1/(1 - p)$. The expected number of correct triangles removed in a sequence of regressive states can be bounded by $\frac{1}{1-p}M$.

For a net positive drift in progressive and regressive states we want

$$1/2(1 - M) - (1/(1 - p))M > 0$$

By the Lemma, $p \approx (\Delta w)^2/n$, so

$$(\Delta w)^2/n < 1 - \frac{2M}{1 - M},$$

which yields the condition that

$$w < \frac{1}{\Delta}\sqrt{n\left(1 - \frac{2M}{1 - M}\right)} = O(n^{1/2})$$

Having $w = o(n^{1/4})$ satisfies both the condition of Lemma 1 and the above progress condition. For expected drift to be positive, p needs to be positive and hence $0 \leq M \leq 1/3$. Polynomial expected time can be shown using standard probabilistic techniques (e.g., Feller, 1968, Chapter 14), since each iteration takes polynomial time, the variance V is bounded, and the expected drift is positive over an expected constant $(1 + 1/(1 - p))$ number of steps. □

In effect, the theorem also shows the conditions under which there is enough information in the data, i.e., conditions under which it is possible to perform resonance assignment using sparse data sets. In a qualitative sense, our theoretical analysis predicts that if the noise is not large, our randomized rotation algorithm will converge to the optimum in a polynomial number of steps. Our simulation studies, described next, agree well with this prediction, showing that our algorithm tolerates noise as high as 500%.

3.2 Simulation Studies

We started with a β-sheet graph with 100 triangles, configured as eleven 6-residue strands, for a total of 66 vertices and 215 edges. This large graph allows us to readily study the behavior of our algorithm under varying noise and sparsity. It also demonstrates that our algorithm is efficient enough to handle graphs much larger than would be possible with exhaustive search algorithms.

We generated noise edges for this graph according to the random graph model described in the methods section, varying the window size w to control the amount of chemical shift degeneracy. We tested with window sizes up to 5, yielding up to 1075 noise edges obscuring the 215 true edges. For each test, we formed random permutations of the residues individually for H^{α} and H^{N} (representing sorting by the corresponding chemical shift), and generated noise edges for each true edge by replacing the from-node with each other node within a distance of w in the appropriate permutation. This simulates ambiguity introduced by chemical shift degeneracy. We set an edge's score to $e^{-(x-\mu)^2}$ where x is the position in the window generating the edge, $\{-w/2, \ldots, -1, 0, 1, \ldots, w/2\}$ (0 for the true edge), and $\mu \sim \mathcal{N}(0, (w/4)^2)$. This approach follows typical scoring rules (e.g., Zimmerman et al., 1997; Güntert et al., 2000; Vitek et al., 2004, 2005).

Our theoretical analysis (in a simplified setting) predicts that in the low noise case, the algorithm makes progress, i.e., the number of correct triangles it uncovers increases with time until it recovers the entire structure. We measured the progress of the algorithm for varying amounts of noise with complete data (no missing edges).

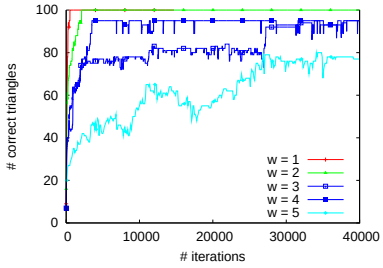

Fig. 4. Number of correct triangles selected by the algorithm as a function of the number of iterations. The amount of noise varies from 1 to 5 incorrect edges for every correct one.

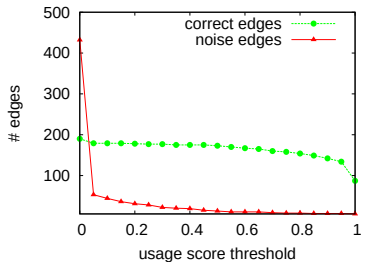

Fig. 5. Number of correct (red) and wrong edges (green) exceeding usage score values, for $w = 4$ and 20 missing edges.

Fig. 4 shows that for the low noise cases ($w = 1$ or 2, i.e., 100–200% noise), the algorithm makes progress and uncovers the entire sheet quickly (less than 1000 iterations). As the noise increases, the rate of progress of the algorithm generally decreases while the number of mistakes (due to which the rate of progress is not monotone) increases. (Due to randomization, progress is better for this particular $w = 4$ run than for the $w = 3$ one.) The algorithm recovers most of the structure, even for high values of w (as much as 500% noise).

The algorithm takes significantly longer for $w = 5$ than for the other cases. To gather intuition for the abrupt change, we measured how often the algorithm chose the correct triangle at each step (closely related to the probability of progress when $\sum kf(k)$ is low, as in our case). We found that the average frequency of making the correct choice was 0.48 when $w = 4$ but only around 0.37 when $w = 5$. Thus $w = 5$ was the first case when the probability of progress fell well below 0.5, resulting in a significant increase in the running time as predicted by our analysis.

While our focus here is on handling significant noise and ambiguity, in order to analyze experimental data we need to handle the missing edges of incomplete data sets. To simulate this, we randomly removed differing fractions (up to 10%) of the correct edges. Fig. 6(left) shows the number of correct triangles recovered in each case. The results show that while in the low noise case the algorithm is able to recover the rest of the correct triangles, as the noise increases the number of triangles found decreases rapidly. For example, with w = 2, when 15 edges are missing, the algorithm is able to recover around 70 out of the 85 correct triangles present. However, with w = 4, the algorithm recovers only 60 triangles and with w = 5, the number of triangles recovered falls further to 40.

With missing data and high amounts of noise, it might not be possible to recover the entire sheet. However, we still expect to find

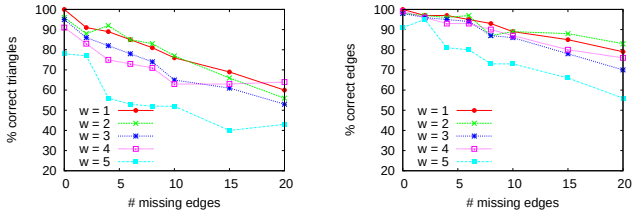

Fig. 6. Performance of the algorithm under varying amounts of noise (w from 1 to 5) and missing edges (0 to 20): (left) without and (right) with edge reconstruction.

sizable chunks of the sheet during the process of growing. We then need to estimate our confidence for every edge. To do this, we use the following observations. (1) A good triangle is consistent with the global structure of a sheet and should therefore be used frequently (in different polytope states) during growing. (2) The dependency between adjacent triangles is strong. Thus, we have more confidence in a triangle which is surrounded by (good) triangles. (3) Since it is less likely that a large number of bad triangles can conspire to form a consistent polytope, we have more confidence in a triangle which appears in a big polytope than one which appears in a small polytope. We encapsulate these observations in a "usage score": for each step of the algorithm in which a triangle is used in some polytope, we increment its usage score by the sum of its score and those of its neighbors, weighted by the size of the polytope. A triangle that appears frequently, in a big polytope, surrounded by good neighbors thus gets higher usage score than one that does not. We split a triangle's score equally among its edges, and normalize edges generated for the same spectral peak (i.e., those produced by chemical shift degeneracy) to obtain confidences. Any edge with a confidence greater than a cutoff value is classified as a correct edge.

In Fig. 5, we show the number of correct and wrong edges exceeding different usage score thresholds for one of the noisier cases (w = 4, 20 missing edges). Most wrong edges have low scores, and their number falls off rapidly as the threshold increases; on the other hand, most correct edges have high scores. Since the results are fairly insensitive to a broad range of thresholds, we use 2/3 as the threshold in the remainder. For a peak to be assigned, the highest scoring edge for the peak must therefore have a score at least twice as high as the sum of the scores of the remaining edges for the peak.

To counter the problem of missing edges, we "reconstruct" missing edges for which there is reasonable evidence that they exist. Since experimental data (and geometric intuition) suggest that in β-sheets, H^N edges are more prone to be missing, we reconstruct only that type of edge. If either of the two H^N edges needed to form a triangle is missing, we create it. Each such reconstructed edge gets a low score (one standard deviation out, a value empirically consistent with experimental data). To prevent the addition of too many incorrect edges in this process, we add edges only if the existing edges have a high score. We first grow with edges reconstructed according to the input score, and we then iterate the process, using the usage score from one round to restrict reconstruction for the next round. Note that we use the usage scores from the previous run only in reconstruction; our growing algorithm uses the original scores in all its decisions.

Fig. 6(right) shows the percentage of correct edges identified using this method. The performance of the algorithm improves

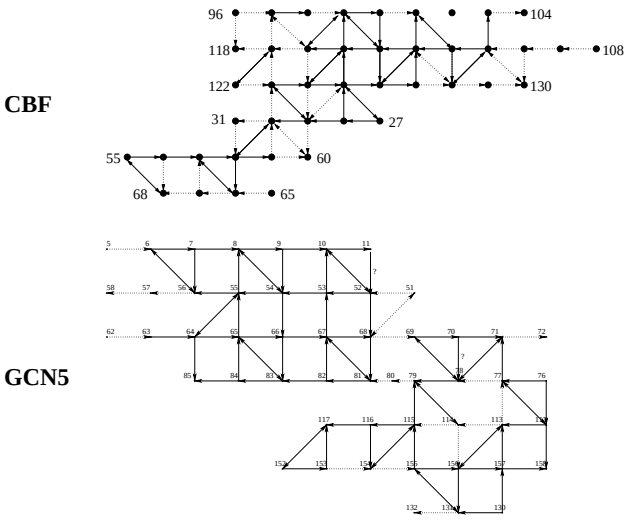

Fig. 7. Experimental results for β-sheets of (top) CBF and (bottom) GCN5 as computed by our algorithm. Solid lines indicate true positives and dotted lines indicate false negatives. Alternative assignments are indicated with a question mark. For clarity, false positives are not shown.

significantly when there are a reasonable number of missing edges. For example, when there are 6 (3%) missing edges, the algorithm identifies around 20 (10%) more edges correctly when edges are reconstructed than when they are not. However, as the number of missing edges increases, the improvement in the performance of the algorithm decreases. For example, when there are around 15 (7%) missing edges, the improvement in performance is only around 2–6 more edges (1–3%). Thus while reconstructing edges does improve performance, it is most effective when there is only a moderate amount of sparsity.

3.3 Experimental Data

We study the results of our algorithm on experimental data sets from four proteins: Human Glutaredoxin (HUGRX) and Core Binding Factor β (CBF), as previously reported (Bailey-Kellogg et al., 2000), as well as two larger data sets, the catalytic domain of GCN5 histone acetyltranferase (GCN5) and single chain T cell receptor (SCTCR). In each case, a graph was constructed from the experimental NOESY peaks, restricted to known β-sheet residues (simulating high-quality HNHA) and scored according to our Gaussian model. We ran our algorithm to compute usage scores, and used the determined scores to reconstruct edges. We then ran our algorithm again with the modified graph, and used the resulting usage scores to select edges for output.

Fig. 7 illustrates the β-sheets identified for CBF and GCN5 using the edges selected by our algorithm. For CBF, 58 of the 89 correct edges are recovered and all but 17 of the 111 incorrect edges are eliminated. Most of the false positives (not illustrated) are due to the fact that the last strand (connecting vertices 55–58 to vertices 65–68) is very poorly connected to the rest of the sheet. Therefore the two polytopes corresponding to this strand and the rest of the sheet do not form a consistent set (residue 58 is present in both). Instead, the algorithm selects edges from these residues to other parts of the sheet. The other errors occur around residues at the end of the first strand (103, 104, 108, 109) which again are loosely connected to

Protein	correct	noise	true-+	false-+	Spec.	Sens.
HUGRX	29	26	10	1	0.95	0.34
CBF	89	111	58	17	0.65	0.65
GCN5	124	142	79	11	0.80	0.64
SCTCR	174	373	117	68	0.46	0.68

Table 1. Summary of results on experimental datasets. Each protein is characterized by the number of correct and noise edges in the input graph. Results indicate the number of true and false positive edges identified by our algorithm, summarized in specificity (true-$-$ / (true-$-$ + false-+)) and sensitivity (true-+ / (true-+ + false-$-$)).

the sheet. For GCN5, we identify 79 of the 124 correct edges and eliminate all but 11 of the 142 incorrect ones. Our algorithm yields some reasonable alternate assignments for a few peaks — not the same as those determined by the experimentalist, but still consistent with the structure of the β-sheet. As with CBF, most of the errors are in incomplete regions. For example, residues 5, 58, 57 and 62 are connected to the rest of the sheet with only one correct edge each. However they are connected with noise edges to other parts of the sheet, causing errors in assignment. Another source of errors is violations of canonical β-sheet structure between 79 and 80. Again, due to noise edges, there are alternate assignments (selected by the algorithm) which meet the canonical structure. The reasons for some errors aren't immediately obvious. For example, residue 51 is connected by two edges to the sheet. It is however incorrectly assigned. This was due to the fact that the peak corresponding to the (correct) edge from residue to 51 to 52 also explains the (wrong) edge between 11 and 52. Furthermore, residue 52 has noise edges with good scores connecting it to 154, leading to a feasible alternate assignment.

Tab. 1 summarizes the results on all experimental datasets. The sensitivity on HUGRX and CBF is similar to that of Jigsaw (Bailey-Kellogg et al., 2000); the specificity is a bit lower, in a trade-off for being able to handle larger and noisier datasets such as GCN5 and SCTCR. For SCTCR, our largest, noisiest, and sparsest dataset, the algorithm yields a number of false positives. One of the reasons is that, unlike the other test cases, SCTCR has 3 different β-sheets. It is possible to consistently connect vertices at the ends of two sheets, and our algorithm could be fooled in this manner. A peculiarity of this dataset is the number of deviations from the canonical β-sheet structure. This, along with missing edges, explains many of the assignment errors. For example, due to incomplete data, residues 92 and 242 are poorly connected to their sheets. However, 92 has noise edges to 243 and 75, both residues being towards the end of their strands. Thus these noise edges are consistent with the expected structure of a β-sheet, and form a larger β-sheet using some of the correct edges. We discuss below possible extensions for non-canonical β-sheet structure.

4 DISCUSSION

This paper shows that a natural *reuse*-based randomized algorithm can be quite successful in overcoming significant noise and ambiguity in contact-based resonance assignment. We demonstrated in earlier work (Bailey-Kellogg et al., 2005) that a special case of the reuse approach was successful in finding long paths for sequential backbone assignment and for uncovering α-helices in NOESY data.

The present work significantly generalizes the paradigm of subgraph reuse and, in fact, shows that this is even more effective for uncovering higher-order patterns such as β-sheets.

We focus here on the information content available in hypothesized connectivity alone. This complements investigations into the proper modeling and use of amino acid type information in assignment (e.g., Pons and Delsuc, 1999; Güntert et al., 2000; Marin et al., 2004; Vitek et al., 2004, 2005). In practice, an integrated approach could help reduce the impact of incorrect edges, and could provide another constraint on hypothesized missing edges.

In our study of connectivity information, we accounted for chemical shift degeneracy, the key source of combinatorial explosion, and developed an algorithm able to handle a large amount of degeneracy (up to 500%). However, we did not study the impact of such noise factors as missing and extra vertices and incorrect secondary structure labels. In the experimental datasets we have used here and in other studies (Vitek et al., 2005), we have found only a small number of missing vertices, randomly distributed. Their effect on the algorithm would simply be to artificially break polytopes, and there appears to be little to be gained in explicitly accounting for them. Uncertain secondary structure labels present a more important practical concern, and we could readily extend our graph and algorithm to incorporate a probabilistic label. Results with Jigsaw (Bailey-Kellogg et al., 2000) show that enforcing consistent β-sheet patterns is the key to successful assignment, trumping noise in secondary structure labels.

Although our study concentrates on the effects of noisy edges, practical concerns require us to deal with missing edges as well. Since our analysis and simulation studies suggest that our algorithm is robust to noise, we hypothesize that errors for the experimental datasets are due primarily to missing edges. We devised a simple test, extending the experimental graphs with simulated data for the missing edges, along with noisy edges ($w = 2$) due to chemical shift degeneracy according to the chemical shift data in the BioMagResBank (Seavey et al., 1991). In the case of GCN5, our algorithm found a total of 130 (up from 79) correct edges with 20 false positives. For CBF, our algorithm found 78 correct edges (up from 57) with 21 false positives. This significant improvement suggests that missing edges do cause most of the errors.

The presented algorithm deals with the regular graph patterns displayed by β-sheets. Our general mechanism for subgraph reuse could also be applied in more complex cases in which the desired graph (or its subgraph components) is known, and the goal is to find it within a corrupted instance. A three-dimensional (tetrahedral) version would support assignment given a known structure or high-quality homology model. A non-homogeneous representation would be required to stich together subgraphs of mixed dimension. Similarly, to allow "non-canonical" β-sheets without artificially reconstructing missing edges, we could move from a simplex representation to a more general cell complex representation. Edges for interactions with side-chain protons could be incorporated, and might be particularly useful in the three-dimensional case. These extensions do come at some increase in implementation complexity, since regularity yields a simple data structure and efficient conflict test (current polytopes have a unique embedding up to rigid motions). The work presented here represents a major step in the development of an algorithmic basis for contact-based backbone assignment and shows the way forward for these future goals.

ACKNOWLEDGMENTS

We gratefully acknowledge the contribution of experimental data from Drs. John H. Bushweller, Bruce Randall Donald, Brian Hare, and Gerhard Wagner. CBK is supported in part by a CAREER award from the National Science Foundation (IIS-0444544). We thank Shobha Potluri and Fei Xiong for helpful comments on the manuscript.

REFERENCES

Angluin, D. and L. Valiant (1979). Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences 18*, 155–193.

Bailey-Kellogg, C., S. Chainraj, and G. Pandurangan (2005). A random graph approach to NMR sequential assignment. *J. Comp. Biol. 12*, 569–583. Conference version: Proc. RECOMB 2004, pp. 58–67.

Bailey-Kellogg, C., A. Widge, J. J. K. III, M. J. Berardi, J. H. Bushweller, and B. R. Donald (2000). The NOESY Jigsaw: automated protein secondary structure and main-chain assignment from sparce, unassigned NMR data. *J. Comp. Biol. 7*, 537–558. Conference version: Proc. RECOMB 2000, pp. 33–44.

Erdmann, M. and G. Rule (2002). Rapid protein structure detection and assignment using residual dipolar couplings. Technical Report CMU-CS-02-195, Computer Science Department, Carnegie Mellon University.

Feller, W. (1968). *An Introduction to Probability Theory and its Applications* (Third ed.). Wiley.

Grimmett, G. and D. Stirzaker (1992). *Probability and Random Processes* (Second ed.). Oxford University Press.

Grishaev, A. and M. Llinás (2002). CLOUDS, a protocol for deriving a molecular proton density via NMR. *PNAS 99*, 6707–6712.

Güntert, P., M. Saltzmann, D. Braun, and K. Wüthrich (2000). Sequence-specific NMR assignment of proteins by global fragment mapping with program Mapper. *J. Biomol. NMR 17*, 129–137.

Langmead, C. and B. Donald (2004). An expectation/maximization nuclear vector replacement algorithm for automated NMR resonance assignments. *J. Biomol. NMR 29*(2), 111–138. Conference version: Proc. RECOMB, 2003, pp. 176-187.

Lin, G., D. Xu, Z.-Z. Chen, T. Jiang, and Y. Xu (2002). A branch-and-bound algorithm for assignment of protein backbone NMR peaks. In *First IEEE Bioinformatics Conference*, pp. 165–174.

Marin, A., T. Malliavin, P. Nicolas, and M. Delsuc (2004). From NMR chemical shifts to amino acid types: investigation of the predictive power carried by nuclei. *J. Biomol. NMR 30*, 47–60.

Montelione, G. T., D. Zheng, Y. J. Huang, K. Gunsalus, and T. Szyperski (2000). Protein NMR spectroscopy in structural genomics. *Nature Structural Biology 7 Suppl*, 982–985.

Moseley, H. N. B. and G. T. Montelione (1999). Automated analysis of NMR assignments and structures for proteins. *Curr. Opin. Struct. Biol. 9*, 635–642.

Nelson, S., D. Schneider, and A. Wand (1991). Implementation of the main chain directed assignment strategy. *Biophysical Journal 59*, 1113–1122.

Pandurangan, G. (2005). On a simple randomized algorithm for finding a 2-factor in sparse graphs. *Information Processing Letters 95*(1), 321–327.

Pons, J. and M. Delsuc (1999). RESCUE: an artificial neural network tool for the NMR spectral assignment of proteins. *J. Biomol. NMR 15*, 15–26.

Seavey, B. R., E. A. Farr, W. M. Westler, and J. Markley (1991). A relational database for sequence-specific protein NMR data. *Journal Biomolecular NMR 1*, 217–236. http://www.bmrb.wisc.edu.

Stefano, D. D. and A. Wand (1987). Two-dimensional ^{1}H NMR study of human ubiquitin: a main-chain directed assignment and structure analysis. *Biochemistry 26*, 7272–7281.

Stevens, R. C., S. Yokoyama, and I. A. Wilson (2001). Global efforts in structural genomics. *Science 294*(5540), 89–92.

Vitek, O., C. Bailey-Kellogg, B. Craig, P. Kuliniewicz, and J. Vitek (2005). Reconsidering complete search algorithms for protein backbone NMR Assignment. *Bioinformatics 21*, ii230–ii236. Conference version: Proc. ECCB, 2005.

Vitek, O., J. Vitek, B. Craig, and C. Bailey-Kellogg (2004). Model-based assignment and inference of protein backbone nuclear magnetic resonances. *Statistical Applications in Genetics and Molecular Biology 3*(1), article 6, 1–33. `http://www.bepress.com/sagmb/vol3/iss1/art6/`.

Zimmerman, D., C. Kulikowski, Y. Huang, W. Feng, M. Tashiro, S. S. Shimotakahara, C. Chien, R. Powers, and G. T. Montelione (1997). Automated analysis of protein NMR assignments using methods from artificial intelligence. *J. Mol. Biol. 269*, 592–610.