Noname manuscript No.
(will be inserted by the editor)

NMR Assignment through Linear Programming

José F. S. Bravo-Ferreira · David Cowburn · Yuehaw Khoo · Amit Singer

Received: date / Accepted: date

Abstract Nuclear Magnetic Resonance (NMR) Spectroscopy is the second most used technique (after X-ray crystallography) for structural determination of proteins. A computational challenge in this technique involves solving a discrete optimization problem that assigns the resonance frequency to each atom in the protein. This paper introduces LIAN (LInear programming Assignment for NMR), a novel linear programming formulation of the problem which yields state-of-the-art results in simulated and experimental datasets.

Keywords NMR spectroscopy · Shortest path problem · Resonance assignment problem · Linear programming relaxation

1 Introduction

We investigate a type of constraint satisfaction problem on certain graphs arising in NMR Spectroscopy. Crucial to NMR spectroscopy is the time-consuming chemical shift assignment problem (also known as the spectral or resonance assignment problem) [14], which inhibits the wider application of this technique. To date, this procedure is done largely in a semi-manual way, even though approaches using exhaustive search [25], integer programming [2], genetic algorithms [29],

José F. S. Bravo-Ferreira
PACM, Princeton University, NJ 08540
E-mail: josesf@princeton.edu

David Cowburn
Departments of Biochemistry and of Physiology and Biophysics, Albert Einstein College of Medicine, NY 10461
E-mail: cowburn@cowburnlab.org

Yuehaw Khoo
Department of Statistics, University of Chicago, IL 60637
E-mail: ykhoo@uchicago.edu

Amit Singer
Department of Mathematics and PACM, Princeton University, NJ 08540
E-mail: amits@math.princeton.edu

variants of belief propagation [4], among others, have all shown promise in different experimental datasets. However, these approaches typically lack either a principled definition of the cost function, or a way to determine whether the global optimizer is every attained. In this paper, we attempt to address these issues in the search for a more rigorous algorithm.

1.1 The assignment problem

The spectral assignment problem is the problem of determining the resonance frequencies of individual atoms in the protein. These frequencies are typically defined by their chemical shifts, measured in parts per million (ppm) relative to a reference compound since they depend on the local environment of individual nuclei [12]. Therefore, the resonance frequencies are often referred to as *chemical shifts*.

To extract constraints from which one can deduce a global protein structure, NMR spectroscopists make use of interactions between atom nuclei, such as the nuclear Overhauser effect (NOE), among others. This effect arises from the dipolar relaxation of a two-spin system, and manifests itself as an off-diagonal peak in NOE spectroscopy experiments (NOESY) [12], as illustrated in Figure 1.

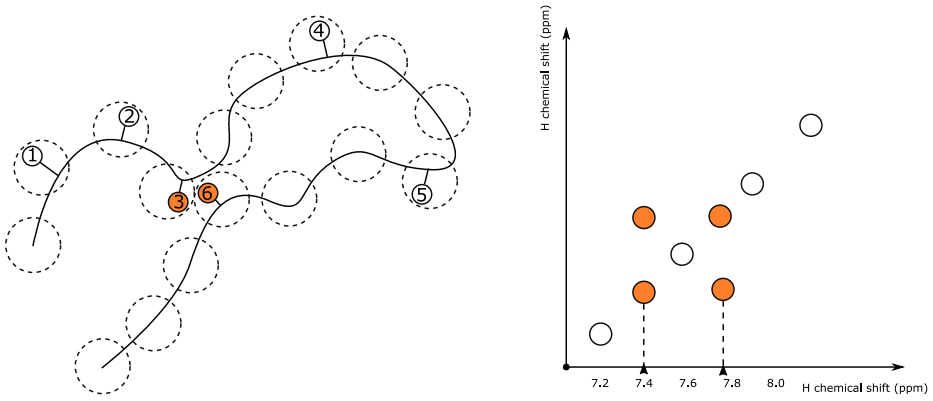

Fig. 1 Illustration of a protein with two hydrogen atoms in close spatial proximity (left), which induce off-diagonal peaks in H -H NOESY spectrum (right).

The NOE between two hydrogens (H) depends on the distance [12], such that cross-peaks in H-H NOESY spectra are indicative of the existence of two hydrogen atoms within close proximity. However, this information is not immediately useful geometrically without the knowledge of which hydrogen atoms induce the cross-peak. The assignment problem provides this information, by mapping the chemical shifts observed in this and other NMR spectroscopy experiments to the corresponding atoms in the protein. The assignment of all hydrogen and other *backbone* atoms, is a crucial first-step for high-resolution structure determination in NMR [28].

The process of NMR assignment (especially for larger proteins) relies on a set of experiments known as heteronuclear resonance experiments. Before describing

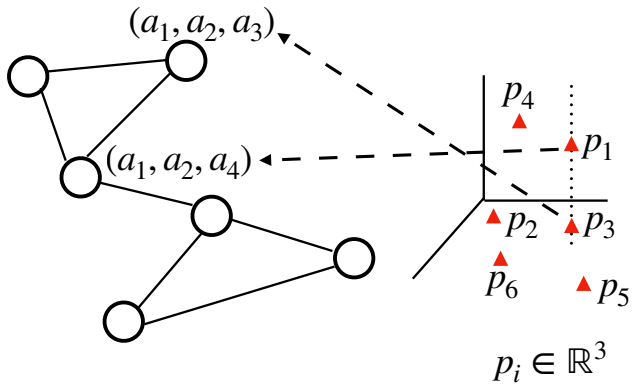

Fig. 2 Representing the assignment problem as a graph. Each node is associated with a triplet of atoms (e.g. (a_1, a_2, a_3) or (a_1, a_2, a_4)). When two triplets share some atoms, there is an edge joining them. Each triplet gives rise to a resonance peak in the 3-dimensional spectra. The goal is to assign the measured peaks to the nodes or triplets. Crucially, when two peaks result from two triplets that share some common atoms, they will share certain coordinates. For example, (a_1, a_2, a_3) and (a_1, a_2, a_4) share (a_1, a_2), so the coordinates in the horizontal plane of the resulting peaks p_1, p_3 are the same (indicated by the vertical dotted line).

these experiments in detail, we first provide some background information on protein structure. A protein is composed of a chain of *residues*. Every residue contains the same set of atoms H^N, N, C^α, C^β (with the exception of Proline). These repeated elements then form the *protein backbone*. Basic heteronuclear experiments couple (H^N, N), (H^N, N, C^α), or (H^N, N, C^β) from a single residue or two adjacent residues. Ideally, these pairs or triplets contribute to the resonance peaks on a 2- or n-dimensional spectra, where the coordinates of the peak are the resonance frequencies of the hydrogen, nitrogen and carbon (similar to the case in Figure 1). As different triplets (or pairs) may share common atoms, this results in a graph such as the one depicted in Figure 2, where a node resembles a triplet (pair), and an edge between two nodes means two triplets (pairs) share one or two atoms. *The goal of the assignment procedure is to take the measured peaks in* $\mathbb{R}^3$ *(peaks in* $\mathbb{R}^2$ *resulted from (N,* H^N*) can be embedded into* $\mathbb{R}^3$*), and assign them to the appropriate nodes in the graph.* An edge between two nodes induces a constraint that the two assigned peaks must share coordinates across certain dimensions.

In the next subsection, we describe different kinds of measurements one can perform to couple different pairs or triplets, giving rise to peaks in 2- or 3-dimensional spectra that facilitate the assignment procedure. Readers unfamiliar with the chemistry involved may skip the rest of the next subsection, and read Section 2 where we elucidate the general philosophy of how these spectra can be used in graph theoretic notions.

1.2 Typical spectra used for assignment

In this section we detail three basic experiments which are commonly used for backbone assignment of small proteins (<150 residues). As we shall see, these

three sets of experiments can provide an assignment of the peaks. They give rise to the three types of spectra detailed below.

HSQC The heteronuclear single quantum coherence experiment [10] involves a transfer of magnetization between the base amide proton, H^N, and the nitrogen N and back, as illustrated in Figure 3. With the exception of proline, all basic amino acids feature this amide pair, such that a distinct peak can be expected for most residues, leading to the use of HSQC as a fingerprinting experiment.

HNCACB This experiment involves magnetization transfer from H^α and H^β to C^α and C^β, respectively, and then from C^β to C^α and finally to N and to H^N of the same or subsequent residue, as illustrated in Figure 3, and described in [18]. The polarities of the C^α and C^β peaks are opposite, which allows these to be distinguished. Importantly, note that C^α and C^β peaks are observed with the same root N–H^N pair. This means there should be four peaks in HNCACB spectra, having the same frequency in the N and H^N dimension. This allows for *sequential walking* (Section 1.2.1), the process of matching residues with their neighbors through matching carbon frequencies.

HN(CO)CACB The last of the experimental toolset for backbone assignment of medium-size proteins also gives rise to C^α and C^β peaks [17], as illustrated in Figure 3. Magnetization transfer happens from H^α and H^β to C^α and C^β, onto CO' and finally the base amide pair. Chemical shifts are evolved only on C^α and C^β before detection, so no CO' peaks are observed.

1.2.1 Basic assignment procedure

Given this set of experiments, a greedy way of assignment (Figure 4) is summarized in this section. This procedure forms the backbone of many assignment algorithms. It is as follows:

(1) HSQC is used as a fingerprint experiment due to high sensitivity and resolution, allowing for accurate determination of base N–H^N pairs. As we can see in Figure 4, peaks in HNCACB and HN(CO)CACB can be grouped according to frequency in the N and H^N. Therefore, peaks in HSQC are matched with peaks in HNCACB and HN(CO)CACB spectra which satisfy tolerance bounds (typically 0.02 - 0.03 ppm for hydrogen and 0.20 - 0.30 ppm for nitrogen).
(2) After grouping the peaks in HNCACB and HN(CO)CACB, within the same N–H^N grouping, the peaks are further correlated and disambiguated using phase information, allowing for the assembly of spin systems. Firstly, HN(CO)CACB tells which of the four peaks in HNCACB comes from the carbons of previous residue, while the +/- sign in HNCACB distinguishes C^α from C^β.
(3) After steps (1) and (2), peaks from HSQC, HN(CO)CACB, HNCACB are combined, resulting in groups of peaks where each group has four peaks. Each group is called a *spin system*. We re-emphasize that the peaks within the same spin system have the same N and H^N frequency, but the frequency along the carbon axis differs. There should be as many spin systems as the number of residues (with a few exceptions), since each residue has exactly one pair of N–H^N.

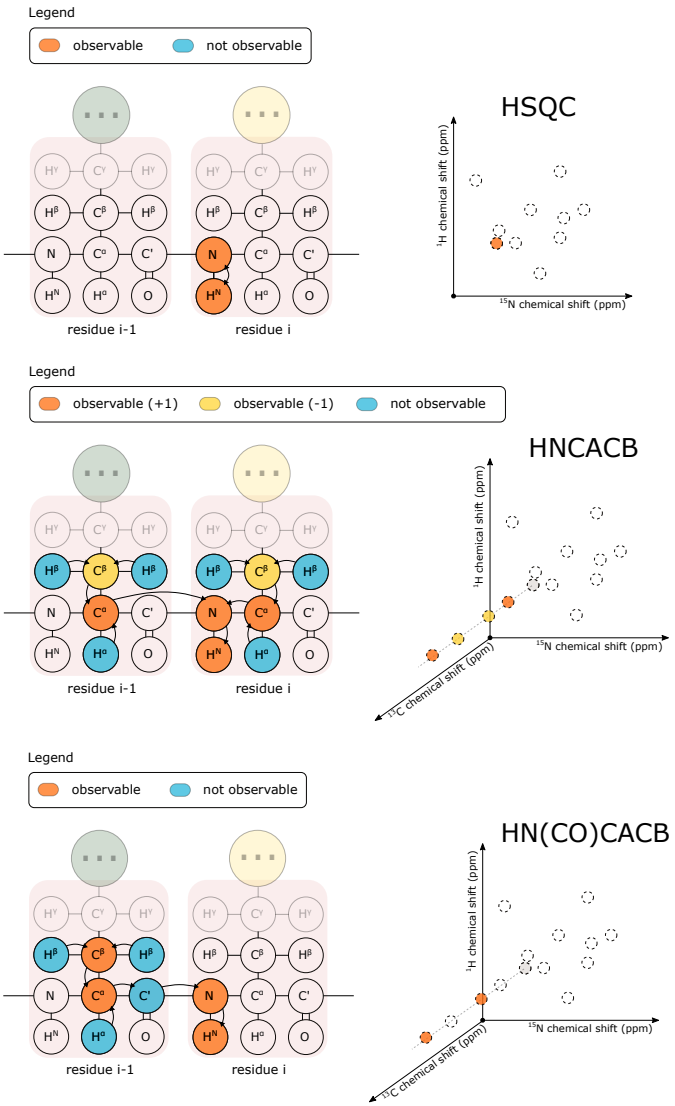

Fig. 3 Illustration of three heteronuclear spectra for backbone assignment. HSQC is used as a fingerprinting experiment. Peaks in HNCACB and HN(CO)CACB develop off the N – H^N plane, along the carbon dimension. Polarity differences of antiphase peaks in HNCACB disambiguate C^α from C^β, and HN(CO)CACB further disambiguates intra- from inter-residue atoms.

(4) Now we want to associate the spin systems to the residues in the protein. As depicted in Figure 4, if two spin systems come from two adjacent residues, they share two peaks with the same carbon chemical shifts. This gives a criteria to create fragments through sequential walking along the C^α and C^β chains. Now the fragments should come from a certain segment of residues in the protein chain. This is done via *statistical typing*, which compares the measured chemical

shifts of atoms in the identified fragments with the expected chemical shift of the residues collected in a public database such as BMRB [30]. The fragments are placed optimally according to that prior.

We remark that that the widespread availability of NMR data collected in databases such as BMRB is fundamental in assignment. The distributions of chemical shifts in different amino acid types is not the same, due to the unique environment induced by the different chemical structures. Certain amino acids, including alanine, glycine, isoleucine, leucine, proline, serine, threonine, and valine, present particularly distinct signatures. Among these, Glycine and Proline are unique in that they do not feature certain peaks. Specifically, Glycine is characterized by its unique C^{α} shift and the absence of a C^{β} signal (see, e.g. [19]). Proline, as the only secondary amine among proteinogenic amino acids, does not yield peaks in the experiments described above. As a result, such residues are only identified as neighbors through HN(CO)CACB spectra or via specific experiments (e.g. [26]).

We also note, however, that the local chemical environment can shift the resonance frequencies of certain atoms, even if they belong to the same residue type. In fact, local chemical shifts can be used as predictors for the chemical shift of a specific atom (see, e.g., [34]), which means that sophisticated probabilistic descriptions of the resonance frequencies may be necessary for certain proteins, or that an iterative process taking into account the primary and secondary structure of the protein should be employed.

1.2.2 Challenges in sequential assignment

As described above, accurate assignment relies on the correct identification of peaks in NMR experiments, and their assembly into consistent spin systems that can be sequentially assigned.

In practice, as the quality of NMR spectra deteriorates, some peaks will overlap, and others cannot be detected at all, due to peak broadening and lower SNR. Artifacts included in automatically selected peak lists further hamper sequential assignment. Even with a decent set of spin systems, sequential assignment itself is not as simple as solving a one-dimensional puzzle, as experimental noise, erroneous spin systems, overlapping chemical shifts, and missing spin systems introduce ambiguity to the process.

1.3 Our contributions

The contribution of the paper is two-fold:

1. We formulate the spectral assignment problem as a constraint satisfaction problem, with cost being defined on a graph $G = (V, E)$:

$$\min_{\{z_i\}_{i\in V}\subset\{0,1\}^m} \sum_{(i,j)\in E} f_{ij}(z_i, z_j), \quad \text{s.t. linear constraints on } z_1, \ldots, z_{|V|}. \quad (1)$$

   Here z_i's are indicator vectors associated with nodes V.

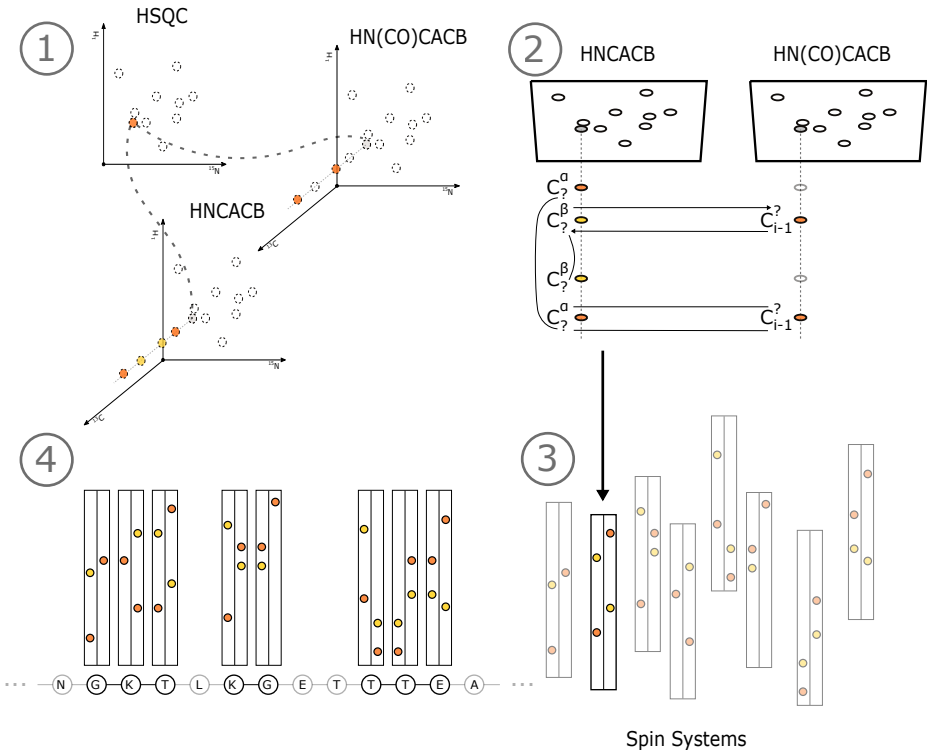

Fig. 4 Sketch of backbone assignment procedure through heteronuclear NMR. (1) HSQC peaks are used as fingerprints and linked to matching peaks in HN(CO)CACB and HNCACB spectra. (2) The carbon dimension in HNCACB and HN(CO)CACB is used along with phase information to identify specific atom frequencies. (3) Spin systems are created from these fragments, each containing carbon frequencies for two adjacent residues. (4) Spin systems are ordered into fragments (based on matching carbon frequencies) and placed in their correct position through statistical typing (using prior information from a public database of chemical shift statistics).

2. In general, this type of problem is hard to solve, unless G is tree-like. However, the adjacency matrix of graph G in the assignment problem forms a band matrix, which allows us to reformulate (1) as a problem on a path graph, by clustering the nodes in G. Such a reformulation allows (1) to be solved either via dynamic programming or linear programming (LP), depending on the structure of the constraints.

1.4 Organization

In Section 2, we formulate the assignment problem as a constraint satisfaction problem over discrete domain. In Section 3, we present a few versions of LPs to solve the constraint satisfaction problem. In Section 4, we demonstrate the performance of our algorithm in simulated and experimental datasets. However, we begin by surveying a few works that are most relevant to the proposed method.

1.5 Prior work

As early as 2004, a detailed review identified twelve important works on automated NMR assignment [7]. A more recent protocol overview [20] cited 44 works on automated chemical shift assignment, which is still not a complete list. Nearly all of the works cited leveraged a similar pipeline of: (1) registering peaks across different dimensions, (2) spin system construction, (3) fragment building through sequential walking, and finally (4) mapping of fragments through probabilistic typing, where a variety of different techniques have been explored, including exhaustive search [22], best-fit heuristics [35], simulated annealing/monte carlo [15], [24], [27], and genetic algorithms [8], [33], [31]. Among these, only a small subset has seen extensive use reported on the protein data bank (PDB, [9]) including AutoAssign [35], CYANA [21], GARANT [8], and PINE [4]. This highlights how automated assignment techniques have not yet managed to achieve widespread adoption.

The development of new automated assignment tools is challenging on a technical level, but there are additional barriers that must be mitigated. Namely, it is currently challenging to fairly compare assignment algorithms, due to discrepancies in input formats, simulation assumptions, and the lack of reproducibility standards or benchmarking datasets. Many state-of-the-art tools, such as CYANA [21] (or FLYA, [29], which is available as part of the CYANA package) lie behind a pay wall. Benchmark datasets are rarely open sourced, such that reliable comparisons can only be made through simulations. Since simulation code is rarely open sourced, comparisons require replicating the simulation frameworks adopted in other works, which is time consuming and error prone.

In [32], the authors attempted to rectify this by introducing a standardized simulated test suite of spin systems, produced according to empirically accepted experimental noise margins. The authors tested their algorithm against three other assignment tools: an iterative, connectivity-based approach called PACES [13], the random-graph theoretic approach RANDOM [5], and MARS [25], yet another iterative, connectivity-based method using random permutations to progressively nudge assignments into better ones. An integer programming approach called IPASS [2] later tested on this same experimental suite. We include comparisons on this same test suite for our algorithm.

Out of all automated assignment methodologies, our approach shares the most similarities with IPASS and FLYA. These are all algorithms that adopt a global optimization view of the assignment problem, rather than optimizing locally through fragment building. We describe them briefly below.

IPASS: This algorithm begins with a graph-based procedure to build spin systems from peak lists of HSQC, HNCACB, and HN(CO)CACB spectra [2]. Distances between peaks are calculated based on the chemical shifts of carbon, nitrogen, and hydrogen atoms, with any distances smaller than twice the nearest-neighbor distance converting into an edge. Spin systems are obtained through brute-force search of consistent C^{α} and C^{β} values. A connectivity graph is established by creating edges between any two spin systems where the C^{α} and C^{β} connections satisfy a loose threshold of $\delta = 0.5$ppm, and a heuristic connectivity score is computed for each edge, with edges scoring below a threshold score trimmed from the graph. Finally, all combinations of fragments where no spin system appears in more than one fragment and no fragments overlap are enumerated. An integer

linear program is then solved for each such combination to compute an assignment that best agrees with a probabilistic prior on the frequencies of each protein residue.

FLYA: Unlike IPASS, FLYA attempts to optimize a global score directly from peak lists, without the intermediate steps of spin system construction or fragment building [29], with state-of-the-art results. Given a set of measured peak lists, FLYA compares it directly to a hypothetical set of peak lists which one would expect to observe given the NMR experiments that were carried out. Expected peaks are matched to measured peaks with the goal of maximizing the global score, and chemical shift values are inferred from this matching. The score is based on a likelihood computed from a generative model assumed to produce the experimental data (we refer the reader to [29] for details). The optimization process itself is a combination of heuristic local optimizations (which remap local regions of the assignment) and a genetic algorithm, which probabilistically recombines and mixes existing assignments from generation to generation.

1.5.1 A note on generative assumptions

While probabilistic assumptions are commonly made across automated assignment tools, they do not always coincide. In particular, CISA [32] and, by extension, IPASS [2] generate simulated data by adding white noise chemical shifts at the spin system level. FLYA, on the other hand, assumes truncated Gaussian noise on measured peaks to ensure valid assignments are possible under their evaluation framework [29]. As a result, great care is required when comparing different algorithms.

2 Spectral assignment problem as constraint satisfaction problem

In this section, we formulate the spectral assignment problem as a constraint satisfaction problem on a graph, where the goal is to determine a set of expected resonance peaks $q_1, \ldots, q_{m_1} \in \mathbb{R}^3$, from an ensemble of experimentally measured peaks $p_1, \ldots, p_{m_2} \in \mathbb{R}^3$. Each q_i is the frequencies for the coupled triplets associated on each node in Figure 2. One can consider the list of peaks $P := \left[p_1, \ldots p_{m_2}\right] \in \mathbb{R}^{3\times m_2}$ returned from experiment as a shuffled list of $q_1, \ldots, q_{m_1}$. In the following, we assume $m_2 \geq m_1$, which implies the set of expected resonance peaks is a subset of the set of measured resonance peaks (this is common in practice due to the existence of artifact peaks). This assumption is made to simplify the exposition and is not crucial to the development of the algorithm. The discussion above indicates that the experimental peaks $p_1, \ldots, p_{m_2}$ are related to $q_1, \ldots, q_{m_1}$ via

$$q_i = Pz_i \quad i = 1, \ldots, m_1, \tag{2}$$

where $z_i \in \{0,1\}^{m_2}$ and $z_i^T \mathbf{1}_{m_2} = 1$. In other words, z_i is an indicator variable that selects the measured peak from P which in turn provides the values for q_i. Therefore determining the indicator variables $z_1, \ldots, z_{m_1}$ is called the spectral assignment problem. When a peak q_i gets assigned a value from one of the

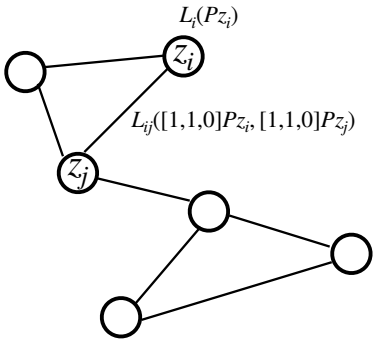

Fig. 5 Definition of cost for the assignment problem depicted in Figure 2. On each edge (i, j) in $G = (V, E)$, there is a loss function L_{ij} penalizing the difference in peaks' frequencies, while on each node a prior term L_i is used to encourage the selection of certain peaks from P.

$p_1, \ldots, p_{m_2}$, the three atoms that generate peak q_i get assigned with resonance frequencies.

In order to determine the indicator variables $z_i, i = 1, \ldots, m_1$, we solve a constraint satisfaction type problem on the graph defined in Figure 2. Each edge (i, j) is associated with a penalty

$$L_{ij}(B_{ij}^T q_i, B_{ij}^T q_j), \tag{3}$$

where each $B_{ij} \in \left\{ \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T, \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T, \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^T, \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T, \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^T \right\}$, and L_{ij} is some loss function. In words, we want to penalize the difference of certain coordinates between q_i and q_j. Furthermore, each node i is associated with a regularization $L_i : \mathbb{R}^3 \to \mathbb{R}$ that is used to impose some prior beliefs on q_i. Therefore determining the indicator variables $z_1, \ldots, z_{m_1}$ can be done via solving

$$\min_{\{z_i\}_{i=1}^{m_1}} \quad \sum_{(i,j)\in E} L_{ij}(B_{ij}^T P z_i, B_{ij}^T P z_j) + \sum_{i=1}^{m_1} L_i(P z_i) \tag{4}$$

$$\text{s.t.} \quad z_i \in \{0,1\}^{m_2},\ z_i^T \mathbf{1}_{m_2} = 1\ \forall i,$$

$$\sum_{i=1}^{m_1} z_i \leq \mathbf{1}_{m_2}, \tag{5}$$

an inference problem on a graphical model defined by $G = (V, E)$. The last constraint prevents selecting more than one measured peak from P for each q_i. We illustrate this construction in Figure 5.

We now turn to a reformulation of (4) that is closer in spirit to the most common assignment procedure outlined in Section 1.2.1. Suppose there are n residues in the protein. Based on the types of coupling detailed in Section 1.2, graph G in fact takes the form in Figure 6, which is a path graph after an appropriate clustering of the nodes into subgraphs $G_1, \ldots, G_n$, where we assume that each subgraph has c nodes.

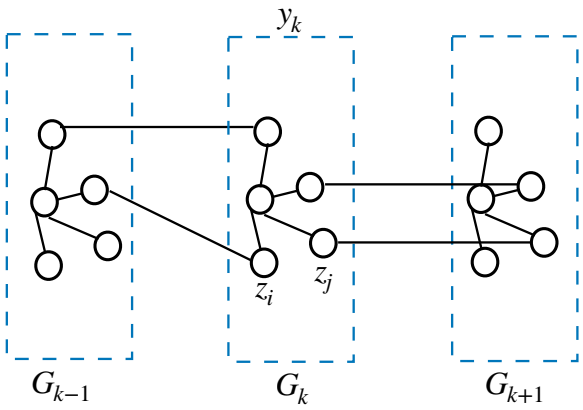

Fig. 6 Redefining the variables $z_1, \ldots, z_{m_1}$ in (4) as $y_1, \ldots, y_n$ in (6) according to subgraph $G_1, \ldots, G_n$. Grouping the nodes according to the dotted box induces a path graph.

Consider unit vectors $y_k, y_{k+1} \in \{0,1\}^{m_2^c}$, each representing the m_2^c ways of associating c measured peaks to each of the node clusters G_k, G_{k+1}, respectively (where the dimensionality follows from the fact that there are a total of m_2 measured peaks to assign to c nodes). Given some specific choice of y_k, denoted by i, and some choice of y_{k+1}, denoted by j, there is some total cost $W_{k,k+1}(i,j)$ associated with the evaluation of the loss functions on our choice of peak assignments. Therefore, we have

$$\min_{\{y_k\}_{i=1}^n} \quad \sum_{k=1}^{n-1} \mathrm{Tr}(W_{k,k+1} y_{k+1} y_k^T) \tag{6}$$

$$\text{s.t.} \quad y_k \in \{0,1\}^{m_2^c},\ y_k^T \mathbf{1}_{m_2^c} = 1, \quad k = 1, \ldots, n.$$
$$\mathcal{A}(y_1, \ldots, y_n) \leq \mathbf{1}_{m_2} \tag{7}$$

That is, once we group the variables, as depicted in Figure 6, there are only cost functions defined between the adjacent $y_k, y_{k+1}, k = 1, \ldots, n-1$. For each possible assignment of measured peaks to each set of adjacent subgraphs there is an associated cost, with the matrix $W_{k,k+1}$ representing the individual costs of all such assignment possibilities. The linear constraint $\mathcal{A}(y_1, \ldots, y_n) \leq \mathbf{1}_{m_2}$ captures (5).

2.1 Outline of proposed method for solving (6)

Without constraint (7), the optimization problem (6) has cost defined on a path graph (since only y_k and y_{k+1}, $k = 1, \ldots, n-1$ are coupled via some cost functions). This type of optimization problem can be solved using dynamic programming [6], and has a complexity of $O(nm_2^{2c})$. More precisely, in order to get a dynamic programming problem, we turn to the construction of a new weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Figure 7 with $nm_2^c + 2$ nodes. There are $n+2$ layers, where each layer consists of m_2^c nodes (except the first and last layer), and within each layer there are no edges. Edges are formed between two adjacent layers of nodes with weights

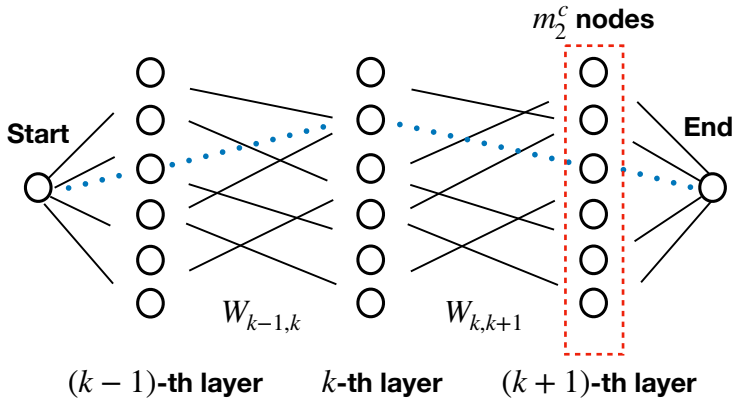

Fig. 7 Illustration of finding a solution to (6) by solving a shortest path problem. Here, the shortest path is indicated in blue by dotted line that traverses from the "start" to the "end" node. Each set of nodes, e.g. nodes in the box, depicts all possible choices for each y_i in (6).

defined by $\{W_{k,k+1}\}_{k=1}^{n-1}$, where each $W_{k,k+1} \in \mathbb{R}^{m_2^c \times m_2^c}$. There are two extra nodes in addition to the nm_2^c nodes, denoted as the "start" and "end" nodes. They are connected to the first and last groups of the nm_2^c nodes as depicted in Figure 7. The minimization problem in (6) *without* constraint (7) thus becomes a problem of tracing the shortest path from the start node to the end node, as depicted in Figure 7. While this problem can be solved efficiently using dynamic programming, such an approach is not possible due to constraint (7). Therefore, in the next section, we turn to a linear programming formulation of the shortest path problem, where (7) can be easily addressed.

3 Methodology

This section presents our general approach to NMR assignment, which we call LIAN (**LI**near Programming **A**ssignment for **N**MR). We first describe how we construct an *assignment graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (different from $G = (V, E)$ in (1) or (4)) on which we efficiently solve (6), a constrained shortest path problem whose solution yields a valid assignment which approximately maximizes the expected log-likelihood under a given probabilistic model. We divide this section into two parts: (1) defining the structure of the assignment graph (Section 3.1), and (2) solving the constrained shortest path problem (Section 3.2).

3.1 Building the assignment graph

The assignment graph $\mathcal{G}$ is a directed graph with $n+2$ layers, where n is the known number of residues in the protein. This is illustrated in Figure 7, where a layer is a group of nodes, for example those in the red dotted box. The construction of the graph proceeds in three steps:

1. **Initial peak groupings**: We first enumerate groups of measured peaks whose frequencies are internally consistent. More precisely, we partition the variables

$z_1, \ldots, z_{m_1}$ in (4) associated with nodes of G, by partitioning G into n subgraphs $G_1, \ldots, G_n$ (corresponding to n residues) as in Figure 6, and for each part we enumerate all the possible choices of assignments. For example, if each G_k has c number of z_i's associated with it, and each z_i has m_2 choices, then there are at most m_2^c choices for all the variables in G_k. This is too large in general, therefore we pick the possible values for all z_i, z_j associated with G_k such that $L_{ij}(B_{ij}^T P z_i, B_{ij}^T P z_j)$ is smaller than some threshold, for all $i, j \in G_k$. The possible values of c z_i's within each part essentially corresponds to a choice of c peaks from $p_1, \ldots, p_{m_2}$. Therefore this is called the peak grouping procedure. We assume there are g choices for the variables in each G_k, which give g nodes in the k-th layer of the assignment graph.

2. **Creating the graph nodes**: A possible combination of peaks in P, that can be assigned to the nodes in G_k, forms a node in a k-layer of $\mathcal{G}$. Again, corresponding to each layer (i.e. each residue), there are g nodes. To further cut down the number of nodes, for each residue we enumerate the peak groupings that are sufficiently consistent with each residue, as determined by the difference between the frequencies in the peak grouping and a prior, which is derived from chemical shift statistics for each residue type stored in BMRB [30]. This step associates a cost related to the log-likelihood of the assignment under the prior for each node. For the k-th residue, such a cost basically comes from $L_i(Pz_i), i \in G_k$ in (4). After this step, the k-th layer is left with g_k nodes.
3. **Creating the graph edges**: Edges *between* two layers are added to the graph between any two nodes which have sufficiently consistent frequency assignments for the same atoms. Each edge contributes a cost commensurate with the relevant level of consistency. Such a cost comes from $L_{ij}(B_i j^T P z_i, B_{ij}^T P z_j), i \in G_k, j \in G_{k+1}$ in (4).

We describe each of these steps in greater detail below.

3.1.1 Initial peak groupings

As explained in Section 2, our experimental data consists of a list of peaks, $P := [p_1, \ldots, p_m]$, where each $p_i \in \mathbb{R}^3$ corresponds to a set of atom frequencies. In order to form nodes from this list of peaks, we group them in groups that are internally consistent, i.e., groups of peaks which assign approximately the same frequency to the same atom.

As mentioned previously, a protein consists of n residues that have repeated sets of atoms. The k-th residue r_k contains atoms N_k, H_k^N, C_k^α, C_k^β. In subgraph G_k, the nodes come from residues r_k and r_{k+1} with triplets forming by N_k, H_k^N, C_k^α, C_k^β, C_{k+1}^α, C_{k+1}^β. When considering an NMR dataset with three spectra, HSQC, HNCACB, and HN(CO)CACB, we expect G_k to contain seven nodes, coming from the fact that there are seven triplet interactions all involving the same N_k and H_k^N. Therefore, each G_k contributes to seven peaks in the three spectra. Some of these peaks will also share C_k^α and C_k^β frequencies. This is illustrated in Figure 3, where all seven peaks have consistent frequency values in N–H^N plane (and there are two peaks agreeing along the C dimension). This allows us to guess valid peak groupings associated with G_k. To this end, we make use of an enumeration procedure (described in Appendix A) to enumerate all consistent peak groupings, which are defined as groupings of seven peaks (or more, depending

on the experiment set) where the frequencies associated with certain atoms do not differ by more than an experimentally-accepted threshold. We describe it more concisely here in the context of our example:

1. Select a reference spectrum (often called a *fingerprint* spectrum). This is typically a spectrum from an experiment such as HSQC, which contains the peaks generated from the pairs $(\mathrm{N}_k, \mathrm{H}_k^N)$ as these spectra have higher sensitivity than other experiments and are therefore less likely to be missing peaks. In principle, there should be n peaks in this spectra.
2. For each HSQC peak, enumerate all peaks in other spectra which are consistent with peaks in the fingerprint spectrum along the N–H^N dimensions, within appropriate experimental thresholds (δ_1, δ_2).
3. Among all consistent peaks, identify all subsets which are consistent (within experimental threshold δ_3) along the corresponding C dimension.

Spin systems: On some occasions, NMR practitioners perform this grouping procedure manually (or in a human-in-the-loop, computer-guided fashion). The groupings of measured peaks are then summarized in the form of *spin systems*, by averaging the frequencies assigned to each atom, thus producing a simple vector of "consensus" atom frequencies. As a result, data is sometimes summarized in the spin system format rather as peak lists, and we can skip the grouping step described in this section. However, we note that this leads to some information loss, as we lose information related to the level of agreement between frequencies assigned to the same atom by different peaks.

3.1.2 Creating the graph nodes

Nodes in the assignment graph are subdivided into $n + 2$ layers, one for each of the n residues in the protein, and additional *start* and *end* layers to simplify the formulation of the problem. There are three broad classes of nodes:

1. **Start and End nodes**: The first layer and the $n + 2$-th layer consist of a single node, used for convenience. These nodes help define the start and end position of the shortest path we seek to find.
2. **Dummy nodes**: There is one such node for each of the n inner layers, and their function is to ensure the shortest path problem is feasible. There is a path from every node in layer $k - 1$ to the dummy node of layer k, and from this dummy node to every node in layer $k + 1$. If included in the final path, no frequencies will be assigned to the atoms in residue k, such that this node is equivalent to a *null* assignment for residue k, and incurs a high associated cost.
3. **Regular nodes**: All other nodes in the graph represent a grouping of measured peaks, as defined in Section 3.1.1, which is consistent with the given residue.

In sum, there is exactly one start and one end node, and there are exactly n dummy nodes, one for each residue of the protein. However, given g valid peak groupings (as defined in Section 3.1.1) we create $g_k \leq g$ nodes in each layer. This is because any peak groupings which are not consistent with the prior on the atom frequencies for a given residue are not instantiated, in order to reduce the overall size of the graph. We formalize the process for eliminating nodes below, upon introducing the edge cost definitions.

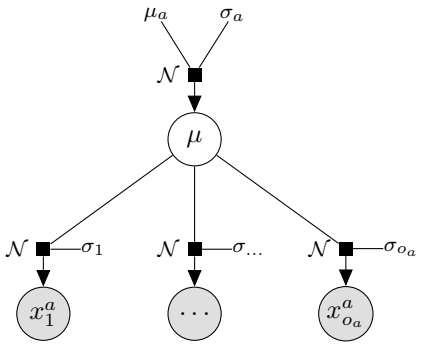

Fig. 8 Generative model for an atom observed by o_a peaks. A Gaussian prior for the frequency of each atom, a, is assumed, with parameters μ_a and σ_a derived from chemical shift statistics deposited in BMRB [30]. The observed frequencies $\{x_1^a, \dots, x_{o_a}^a\}$ of each atom are also assumed to be normally distributed, centered around the latent frequency of the atom and with (assumed known) experimental variance $\sigma_1, \dots, \sigma_{o_a}$.

3.1.3 Creating the graph edges

After creating all $n + 2$ node layers, we connect nodes between each layer and the subsequent layer. The edge creation step is most important because it is also where we define the costs associated with each edge. In sum, we want this cost to represent some notion of probabilistic agreement between our assumed generative model for the data and our set of observations.

Generative model: In Figure 2, atom a_1 is shared by two nodes in graph G. That means two peaks observed in the spectra are associated with a_1. We want to model the probability distribution of the observed peaks associated with an atom in common. Many probabilistic cost functions would be reasonable, but for the purposes of this paper we assume that the prior on each atom's frequency is Gaussian, and that the experimental noise is also Gaussian (with mean 0). This is depicted in Figure 8 for an atom, a, for which we have o_a distinct observations of its frequency, $\{x_1^a, \dots, x_{o_a}^a\}$. This implies that this atom is associated with o_a peaks (or in other words associated with o_a nodes in graph G in Figure 2). Such a generative model prescribes a graphical model on graph G. Solving (4) amounts to performing inference on z_i's under such a probabilistic model. This generative model is consistent with much of the automated assignment literature (see, e.g. [32]) with the notable exception of FLYA, which assumes the experimental noise is a truncated Gaussian, in order to guarantee feasible assignments under its definition of a valid assignment [29].

Under this model, we define the score associated with each atom as

Definition 1 (Atom cost) *The cost associated with atom a, with a normally distributed prior $\mathcal{N}(\mu_a, \sigma_a)$, and o_a observations $\{x_l^a\}_{l=1}^{o_a}$ defined by the peak grouping, also assumed to be normally distributed around the true frequency, μ, according to $\mathcal{N}(\mu, \sigma_l)$ is defined as*

$$cost\,(a, \{x_l^a\}_{l=1}^{o_a}) \triangleq -\log \mathbb{E}_{\mu \sim \mathcal{N}(\mu_a, \sigma_a)} \left[\prod_{l=1}^{o_a} f(x_l^a \mid \mu, \sigma_l) \right]. \tag{8}$$

where $f(\cdot \mid u, v)$ is the Gaussian density with mean u and standard deviation v. This expectation works out to a simple expression involving the observations, experimental noise parameters, and the parameters of the prior distribution, as explained in Appendix A.

Now, recall that if we select an edge between two nodes, node i in layer k, and node j in layer $k+1$ in Figure 7, to be included in our path, we are indeed assigning observed peaks to the nodes in G_k and G_{k+1}. This implies that all the atoms involved in establishing the nodes (recall that each node is associated with a triplet or a pair of atoms) are assigned a frequency valued obtained from the observed peaks. Then the generative model in Definition 1 determines how likely these frequency assignments are under the assumed generative model, which will, in turn, help determine the likelihood of an edge (i, j) in Figure 7.

This provides a cost for selecting edge between the k-th and $(k+1)$-th layers. Let r_{k+1} be the set of backbone atoms associated with residue $k+1$. The peak groupings in the two nodes connected by edge (i, j) in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ imply the assignment of a set of observations $\{x_l^a\}_{l=1}^{o_a}$ for each atom in the set of backbone atoms r_{k+1}. Since only nodes in layers k and $k+1$ include observations for the atoms in r_{k+1}, we define the cost of the edge as follows:

Definition 2 (Edge cost) *Each edge between node i in layer k and node j in layer $k+1$ in in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ assigned frequencies $\{x_l^a\}_{l=1}^{o_a}$ to each atom $a \in r_{k+1}$. The edge cost is then defined as*

$$\textit{edge cost}\,(r_{k+1}, i, j) \triangleq \sum_{a \in r_{k+1}} \textit{cost}\,(a, \{x_l^a\}_{l=1}^{o_a})\,. \tag{9}$$

As such, each edge between layers k and $k+1$ incorporates the cost associated with *all* observations on the atoms in residue $k+1$ induced by the peak groupings in the relevant nodes.

3.1.4 Statistical Typing

In order to further manage the size of the assignment graph, edge whose associated cost is too large (representing an extremely unlikely assignment) can be discarded at this stage. These are typically edges between nodes whose induced frequencies disagree strongly with the prior distributions for a residue's atoms. Details about how we set the threshold for inclusion can be found in Appendix C.

3.2 Finding a shortest path in a directed graph

Having constructed the assignment graph, we formulate the assignment problem as one of finding a shortest path in a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, subject to some utilization constraints. The cost of any path from the start to the end node is equal to the expected negative log-likelihood of the assignment induced by that path. As a result, finding the shortest path in this graph amounts of finding the path of greatest expected log-likelihood. We highlight that alternative formulations of the cost are also possible (such as, for example, directly maximizing the log-likelihood,

rather than the expected likelihood). However, the formulation presented here is computationally straightforward to implement, allowing us to quickly build large graphs.

While an unconstrained shortest path problem is straightforward to solve through dynamic programming, our problem is not unconstrained, due to the fact that each observed datum (i.e. a measured peak) can only be utilized once in a valid assignment. This practical limitation can be concisely written as a linear constraint, which fits into the constraint satisfaction framework we describe in Section 2.

In particular, recall that we had defined the assignment problem (6), where $y_k \in \{0,1\}^{m_2^c}$ is a unit vector selecting a group of peaks for G_k. After reducing the possible choices, we actually redefine our y_k variables into $y_k \in \{0,1\}^{g_k}$, assuming g_k valid combinations for layer k. We can further define

$$X_{k,k+1} \triangleq y_k y_{k+1}^T \in \{0,1\}^{g_k \times g_{k+1}}, \qquad \mathbf{1}_{g_k}^T X_{k,k+1} \mathbf{1}_{g_{k+1}} = 1 \tag{10}$$

as a selection matrix, where $X_{k,k+1}(i,j) = 1$ implies that node i is selected in layer k and node j is selected in layer $k+1$. Under this simple redefinition, $W_{k,k+1} \in \mathbb{R}^{g_k \times g_{k+1}}$ can also be easily understood as the matrix of edge costs between layers k and $k+1$. That is

$$W_{k,k+1}(i,j) \triangleq \text{edge cost}(i,j) \tag{11}$$

where edge cost(i,j) is the cost associated with the edge between node i in layer k and node j in layer $k+1$. Also note that we need not worry about layers 0 and $n+1$ to express the linear programming formulation of the problem, since there is a single node in these two layers.

Finally, we can formulate the NMR assignment problem in terms of these new variables:

Problem 1 (NMR assignment)

$$\min_{\{X_{k,k+1}\}_{k=0}^{n}} \quad \sum_{k=1}^{n} Tr(W_{k,k+1}^T X_{k,k+1}) \tag{12}$$

$$\begin{aligned} s.t. \quad & X_{k,k+1} \in \{0,1\}^{g_k \times g_{k+1}},\ \mathbf{1}_{g_k}^T X_{k,k+1} \mathbf{1}_{g_{k+1}} = 1, k = 1, \ldots, n \\ & X_{k-1,k}^T \mathbf{1}_{g_{k-1}} = X_{k,k+1} \mathbf{1}_{g_{k+1}}, k = 1, \ldots, n \\ & \mathcal{A}(X_{1,2}\mathbf{1}_{g_2}, \ldots, X_{n,n+1}\mathbf{1}_{g_{n+1}}) \leq \mathbf{1}_{m_2} \end{aligned} \tag{13}$$

Note the following details: A *path* constraint is included in the formulation, enforcing that the end node selected using $X_{k-1,k}$ must coincide with the start node selected using $X_{k,k+1}$. Compare to (6), the summation in the cost of (1) goes from $k = 0$ to $k = n$, which takes into account of the extra "start" and "end" node, and $g_0 = g_{n+1} = 1$.

The problem as formulated above is equivalent to a constrained shortest path problem, and is NP-hard [1]. For small enough problems, integer linear programming (ILP) solvers such as Gurobi [23] can successfully solve the problem with short runtimes. In our experience, this is often feasible whenever the input consists of a high quality set of nodes in each layer of $\mathcal{G}$ (e.g. when we have a set of reliable spin systems as input). However, for larger problems we can instead make use of linear programming relaxations of Problem 1. These relaxations can occasionally return integer solutions (in which case the solution coincides with

the solution for Problem 1) or, more commonly, partially-integer solutions (i.e. a solution in which many of the entries in each $X_{k,k+1}$ are integer, with most being zero), from which one can then derive a satisfactory solution as we describe shortly below. We make use of the following relaxation:

Problem 2 (NMR assignment, LIAN-1)

$$\begin{aligned} \min_{\{X_{k,k+1}\}_{k=0}^{n}} \quad & \sum_{k=0}^{n} Tr(W_{k,k+1}^T X_{k,k+1}) && (14) \\ s.t. \quad & X_{k,k+1} \geq 0, X_{k,k+1} \leq 1, k = 1, \ldots, n \\ & \mathbf{1}_{g_k}^T X_{k,k+1} \mathbf{1}_{g_{k+1}} = 1, k = 1, \ldots, n \\ & X_{k-1,k}^T \mathbf{1}_{g_{k-1}} = X_{k,k+1} \mathbf{1}_{g_{k+1}}, k = 1, \ldots, n \\ & \mathcal{A}(X_{1,2}\mathbf{1}_{g_2}, \ldots, X_{n,n+1}\mathbf{1}_{g_{n+1}}) \leq \mathbf{1}_{m_2} && (15) \end{aligned}$$

This follows from relaxing the original (matrix-integer) variables $X_{k,k+1}$ to their convex hull, where $X_{k,k+1} \in [0,1]^{g_k \times g_{k+1}}$ and $\mathbf{1}_{g_k}^T X_{k,k+1} \mathbf{1}_{g_{k+1}} = 1$. As alluded to above, solving this problem typically results in a sparse, partially-integer solution. That is: a solution in which some of the entries in each $X_{k,k+1}^*$ that solves Problem 2 are integer, with most being zero. Such a solution induces a much smaller subgraph of the original assignment graph by retaining only edges, (i, j), for which $X_{k,k+1}^*(i, j) \neq 0$. We then solve Problem 1 on that induced subgraph.

We note that the utilization constraint (15) can sometimes be too strict as there are often peaks which are overlapping, resulting less peaks than expected. As a result, strictly preventing data from being re-utilized can hurt, rather than help, the solution. To address this issue, we also consider an alternative relaxation, as follows:

Problem 3 (NMR assignment, LIAN-2)

$$\begin{aligned} \min_{\{X_{k,k+1}\}_{k=0}^{n}} \quad & \sum_{k=0}^{n} Tr(W_{k,k+1}^T X_{k,k+1}) + \lambda \mathbf{1}_{m_2}^T \epsilon && (16) \\ s.t. \quad & X_{k,k+1} \geq 0, X_{k,k+1} \leq 1, k = 1, \ldots, n \\ & \mathbf{1}_{g_k}^T X_{k,k+1} \mathbf{1}_{g_{k+1}} = 1, k = 1, \ldots, n \\ & X_{k-1,k}^T \mathbf{1}_{g_{k-1}} = X_{k,k+1} \mathbf{1}_{g_{k+1}}, k = 1, \ldots, n \\ & \mathcal{A}(X_{1,2}\mathbf{1}_{g_2}, \ldots, X_{n,n+1}\mathbf{1}_{g_{n+1}}) - \epsilon = \mathbf{1}_{m_2} \\ & \epsilon \geq 0, \epsilon \in \mathbb{R}^{m_2} && (17) \end{aligned}$$

Note that this relaxation penalizes (but allows for) the reutilization of measured peaks/spin systems at multiple points of the assignment through the use of slack variable ϵ. The reutilization of peaks is penalized in the cost function, with each reutilization costing λ in added cost. This λ thus becomes a user-set parameter.

An end-to-end description of the full assignment procedure is summarized in Figure 9.

3.3 Notes on the methodology

We emphasize that the graph-based approach presented in this paper extends to any objective function that can be expressed in the form of Problem 1. As a result,

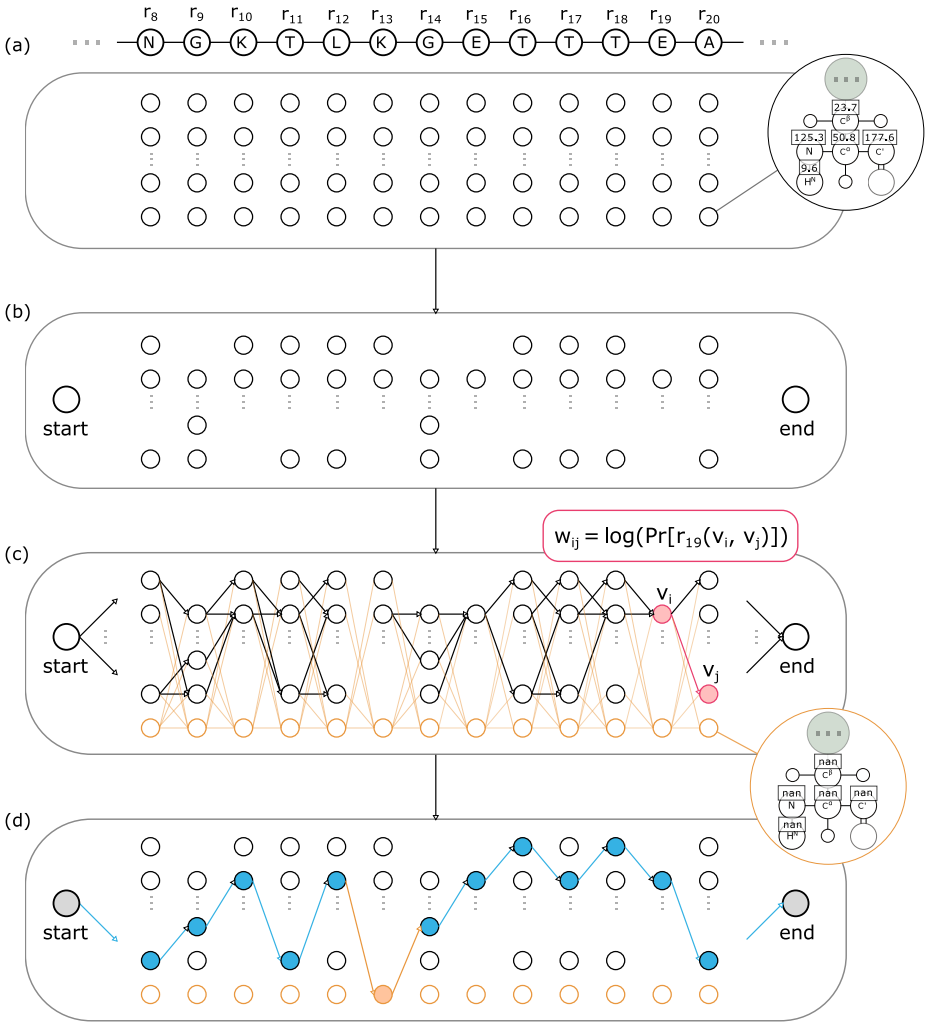

Fig. 9 Illustration of the full assignment procedure. (a) Possible chemical shift assignments are determined and enumerated for each residue, creating nodes in a graph. (b) Each node is statistically typed against its residue's distribution, and very low likelihood nodes are eliminated. (c) Edges are placed between nodes i, j in adjacent layers, k and $k+1$ with weight equal to the posterior log-likelihood of the respective assignment. Empty nodes are added to each residue and connected to every node in the preceding and succeeding layers with edge weights equal to the threshold. (d) A longest path is found between the start and end nodes, subject to any additional constraints (e.g. that spin systems cannot be used more than once).

many different generative models can easily be accommodated by this methodology, and one is also not limited to maximizing an expected log-likelihood. As an example, for each edge in the assignment graph connecting two nodes between layers k and $k+1$, which assign a set of observed chemical shifts to the atoms of residue k, one could solve a local maximum likelihood problem for that residue under the given observations. Doing so would enable a maximum likelihood estimate instead.

We also note that prior information or information from supporting experiments can also be accommodated, either by modifying the edge costs (through the matrices $W_{k,k+1}$) or by editing the assignment graph. As a concrete example, if one had accurate information about the chemical shifts in a particular residue, r_{k+1}, then one could specify tight priors, $\mathcal{N}(\mu_a, \sigma_a)$, for each atom a in that residue, where μ_a would be set to the chemical shift implied by the prior information, and σ_a would be set to a value that accurately represents the uncertainty of that prior information. By doing so, Equations 8, 9, and 11 would result in entries of $W_{k,k+1}$ which are high for any chemical shift assignment that violates the prior information about residue r_{k+1}, and low for any assignment consistent with that prior information, as desired. Additionally, one could also use this prior information to directly edit the assignment graph (i.e. by removing any nodes which represent an assignment inconsistent with prior knowledge). This not only helps enforce prior knowledge, but also helps the efficiency of the algorithm.

In sum, while we describe a concrete end-to-end methodology using a specific generative model, the building blocks of our approach should prove more broadly applicable to a wider range of problems in spectral assignment.

4 Results

4.1 Simulated data

4.1.1 CISA

As a first sanity check, we tested our approach on the entirety of the benchmark dataset developed by the authors of CISA in [32], as it provides a useful comparison to many other fully automated algorithms on problems of small, medium, and large scale. This synthetic dataset is created by generating a simulated list of spin systems from the ground-truth assignment values recorded in BMRB [30]. White Gaussian noise is then added to the carbon atoms, with standard deviations $\sigma_\alpha = 0.08$ ppm, $\sigma_\beta = 0.16$ ppm for the C^α and C^β atoms, respectively, in the *low-noise* simulation, and $\sigma_\alpha = 0.16$ ppm, $\sigma_\beta = 0.32$ ppm in the *high-noise* simulation. For full details on the simulation scenario, we refer the reader to [32].

We compare the performance of LIAN to that of 4 different algorithms. Results for MARS [25] and CISA [32] are both retrieved from the original CISA paper. We also compare with IPASS [2], although we note that the relevant paper does not mention adding noise to the spin systems (and, indeed, refers to a "perfect connectivity" scenario, which suggests that no noise is added). Finally, we also include a partial comparison with C-SDP [16], which is an earlier semidefinite programming relaxation approach to the NMR assignment problem which we developed, but which does not scale to larger proteins. The results for LIAN were obtained by solving Problem 2, which typically produces a partially integer solution. This partially integer solution induces a (much) smaller subgraph on which we can efficiently solve the original integer linear programming problem. Each row is averaged over 100 simulations.

The results are summarized in Tables 1 and 2, for the two distinct noise levels considered in [32]. We evaluate the results by calculating the precision and recall of the assignment algorithm. Let m_{assigned} be the number of assigned residues

(i.e. non-dummy nodes in the path), m_{correct} be the number of *correctly* assigned residues, and $m_{\text{assignable}}$ be the number of assignable residues (i.e. residues which have a ground-truth assignment). Then

$$\text{precision} \triangleq m_{\text{correct}}/m_{\text{assigned}}$$
$$\text{recall} \triangleq m_{\text{correct}}/m_{\text{assignable}}.$$

Table 1 Accuracy of assignment (precision/recall) of various algorithms and LP on synthetic spin systems with noise level $(\sigma_\alpha, \sigma_\beta) = (0.08, 0.16)$. Results for MARS [25] and CISA taken from Table 2 in [32]. Results for IPASS taken from Table 3 in [2]. Results for C-SDP taken from Table 1 in [16].

Protein ID	Length	N[1]	MARS	CISA	IPASS	C-SDP	LIAN-1
bmr4391	66	59	91/97	97/97	93/90	99/99	90/90
bmr4752	68	66	98/97	96/94	100/94	100/100	100/100
bmr4144	78	68	100/97	100/99	98/85	100/100	99/96
bmr4579	86	83	97/91	98/98	100/98	100/100	100/99
bmr4316	89	85	97/96	100/99	99/98	99/99	100/100
bmr4288	105	94	97/95	98/98	100/98		99/99
bmr4929	114	110	99/97	93/91	100/100		100/98
bmr4302	115	107	95/92	96/95	100/99		100/99
bmr4670	120	102	94/88	96/95	98/97		99/99
bmr4353	126	98	91/85	96/95	99/93		95/95
bmr4207	158	148	96/93	100/99	100/97		99/99
bmr4318	215	191	88/81	87/84	100/98		98/98

[1] Number of assignable spin systems in the BMRB data.

Table 2 Accuracy of assignment (precision/recall) of various algorithms and LP on synthetic spin systems with noise level $(\sigma_\alpha, \sigma_\beta) = (0.16, 0.32)$. Results for MARS [25] and CISA taken from Table 2 in [32]. Results for IPASS taken from Table 3 in [2]. Results for C-SDP taken from Table 2 in [16].

Protein ID	Length	N[1]	MARS	CISA	IPASS	C-SDP	LIAN-1
bmr4391	66	59	86/85	91/91	93/90	100/100	86/86
bmr4752	68	66	91/90	90/88	100/94	99/99	100/100
bmr4144	78	68	100/97	100/99	98/85	96/96	96/94
bmr4579	86	83	79/75	80/80	100/98	100/100	100/99
bmr4316	89	85	95/92	83/83	99/98	98/98	99/99
bmr4288	105	94	95/93	91/91	100/98		99/99
bmr4929	114	110	99/97	96/94	100/100		100/98
bmr4302	115	107	82/80	91/91	100/99		99/99
bmr4670	120	102	83/81	88/87	98/97		98/97
bmr4353	126	98	83/80	90/90	99/93		95/95
bmr4207	158	148	82/81	88/85	100/97		99/99
bmr4318	215	191	84/75	74/70	100/98		98/98

[1] Number of assignable spin systems in the BMRB data.

It can be seen that LIAN-1 achieves an assignment performance that generally exceeds that of both CISA and IPASS, particularly on recall (with the aforementioned caveat about the IPASS results, which may be overestimated by virtue

of not including random noise). In particular, our algorithm performs strongly on `bmr4353`, which is particularly challenging due to the large number of Proline residues (which, as mentioned in 1.2.1, do not yield N–H^N interactions in the chosen spectra).

One notable exception is the smallest protein, `bmr4391`. The reason for the lower performance on this particular instance appears to be that LIAN-1 finds an assignment of significantly higher likelihood than the ground-truth assignment (at least according to the generative model we selected). In fact, under the chosen generative model, our relaxation-based algorithm finds the optimal solution to the original (un-relaxed) Problem 1, which is small enough in this instance to solve directly. This is a useful reminder that the ground-truth assignment (often determined manually) may not maximize likelihood under our probabilistic model.

4.1.2 FLYA simulated framework

The simulated framework described for the protein SH2 in [29] was used to generate noisy peak lists, as validation of the peak list graph model described in 3.1. In particular, artificial peak lists were generated for HSQC, HN(CO)CACB, HNCACB, HNCO, HN(CO)CA, HN(CA)CO, and HNCA spectra at the positions specified by the reference chemical shifts as listed in the corresponding BMRB entry [30]. The measured frequencies for each peak were then randomly shifted by adding white Gaussian noise, with standard deviations of 0.4/4 ppm for C and N atoms, and 0.03/4 ppm for H^N atoms. Deviations that exceeded 0.4 ppm (for C and N atoms) or 0.04 ppm (for H^N atoms) were discarded, as per the simulation description in [29]. This is a best-effort approach at replicating the exact simulation framework in that paper.

The node enumerator described in Appendix A was used with only the 4 largest maximal cliques for each connected component considered as a node. The results are summarized in Table 3, where we can see that LIAN-1 appears to deliver comparable performance to FLYA. Three scores are presented for the LIAN-1 approach, corresponding to the lowest, average, and highest % correctness in the assignments over a set of 20 simulations.

Table 3 Percentage of correct atom assignments for LIAN-1 and FLYA on simulated SH2 peak list datasets. For LIAN-1 we show the lowest, **average**, and highest correctness scores achieved over 20 simulations.

Protein ID	Length	FLYA	LIAN-1
SH2	114	97.2%	94.5%, **95.4%**, 97.5%

4.2 Experimental data

To validate the performance of LIAN on experimental data, we make use of the experimental dataset used by the authors of IPASS in [2]. This is a challenging dataset, as there are several missing spin systems. We also note that some of the spin systems that were manually assigned are significantly distinct from the

prior (this could be a legitimate biological phenomenon, as shielding effects resulting from the specific electronic environment of the protein shift the resonance frequencies of atoms). LIAN-2 (Problem 3) was used throughout, with $\lambda = 5$. Results are summarized in Table 4, which shows the number of correctly assigned residues alongside the number of assigned residues for each methodology.

Table 4 Accuracy of assignment on four distinct spin system datasets provided by the authors of IPASS. Results for other algorithms were obtained from [2].

Protein	Length	Manual[1]	Spins[2]	MARS	IPASS	C-SDP	LIAN-2
TM1112	89	83	81/85	55/63	71/72	50/85	74/81
CASKIN	67	54	47/48	23/25	29/39	27/48	36/44
VRAR	72	60	47/47	6/17	30/37	19/47	29/51
HACS1	74	61	48/61	15/16	37/50	19/61	39/53

For each method, we show the results in the form #Correct/#Assigned, where #Correct is the number of correctly assigned residues and #Assigned is the total number of assigned residues.
[1] Number of manually assigned residues in the BMRB file.
[2] Correct/Total available spin systems, where spin systems are considered correct (i.e. not artifacts) if they were manually assigned by NMR practitioners. These numbers are taken from [2]. We note that there are meaningful differences between the spin system values and the chemical shifts available publicly on BMRB, so the first number should be interpreted as an upper bound on the number of potentially assignable residues.

We see that LIAN-2 delivers state-of-the-art performance on this dataset in terms of recall (albeit at the expense of lower precision relative to IPASS). However, we note that the precision-recall threshold can be easily tuned by adjusting the threshold score in the dummy nodes, and that once an assignment is produced, validation of assigned spin systems can be made more easily by referring to the residue-level likelihood scores for debugging (which is why higher recall was selected for in our thresholds).

An important observation provided by these experiments is that LIAN-2 is particularly useful when datasets are of poor quality. In fact, we observe that the final solution in all these experiments reused several of the spin systems in multiple positions, which would not have been possible under the standard formulation presented in Problem 1. This illustrates the importance of correctly characterizing the quality of the dataset through appropriate constraints on the problem.

5 Conclusion

This paper introduced a novel formulation of the spectral assignment problem in NMR as a constraint satisfaction problem. More specifically, we formulate it as a constrained shortest path problem, for which near-optimal solutions can be found via linear programming relaxations. This approach has significant advantages over existing approaches, as it treats spectral assignment as a global optimization problem, without the need for intermediate steps (such as spin system creation) which can lead to information loss. Furthermore, the approach is amenable to multiple probabilistic characterizations and could therefore accommodate complex characterizations of the generative model for the data (such as aminoacid-specific chemical shift correlations), which would simply result in different edge weights for the

assignment graph introduced in Section 3. This approach could also straightforwardly accommodate other interactions often useful for assignment, such as the existence of hydrogen-hydrogen interactions in NOE spectra, through additional linear costs in the objective function.

Testing of our approach with a simplistic generative model on both simulated and experimental data showed state-of-the-art performance. For synthetic, spin system data, our methodology's performance matched or surpassed the best performing algorithms (IPASS, [2] and CISA, [32]), with a notable exception where our algorithm found a higher likelihood assignment than the reference assignment, under our probabilistic model. For experimental, spin system data, our methodology improved upon state-of-the-art. For the higher dimensional problem of peak list data, our preliminary studies indicate performance on par with state-of-the-art algorithm, FLYA.

Our reformulation of the assignment problem permits a more realistic basis for assessment of complete automated structure determination, including ambiguous assignment and constraint methods [3].

Acknowledgements A.S. was partially supported by NSF BIGDATA award IIS-1837992, NIH/NIGMS award 1R01GM136780-01, award FA9550-17-1-0291 from AFOSR, the Simons Foundation Math+X Investigator Award, and the Moore Foundation Data-Driven Discovery Investigator Award. DC was supported by NIH GM-117212.

Conflict of interest

The authors declare that they have no conflict of interest.

Data and code availability

Data and preliminary (non-production) code used in simulations and tests is available in the author's repository at `https://github.com/fsbravo/lipras`.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1993)
2. Alipanahi, B., Gao, X., Karakoc, E., Li, S.C., Balbach, F., Feng, G., Donaldson, L., Li, M.: Error tolerant NMR backbone resonance assignment and automated structure generation. J Bioinform Comput Biol **9**(1), 15–41 (2011)
3. Allain, F., Mareuil, F., Ménager, H., Nilges, M., Bardiaux, B.: ARIAweb: a server for automated NMR structure calculation. Nucleic Acids Research **48**(W1), W41–W47 (2020). DOI 10.1093/nar/gkaa362. URL `https://doi.org/10.1093/nar/gkaa362`
4. Bahrami, A., Assadi, A.H., Markley, J.L., Eghbalnia, H.R.: Probabilistic interaction network of evidence algorithm and its application to complete labeling of peak lists from protein nmr spectroscopy. PLOS Computational Biology **5**(3), 1–15 (2009). DOI 10.1371/journal.pcbi.1000307. URL `https://doi.org/10.1371/journal.pcbi.1000307`
5. Bailey-Kellogg, C., Chainraj, S., Pandurangan, G.: A random graph approach to NMR sequential assignment. J. Comput. Biol. **12**(6), 569–583 (2005)
6. Bang-Jensen, J., Gutin, G.Z.: Digraphs: theory, algorithms and applications. Springer Science & Business Media (2008)

7. Baran, M.C., Huang, Y.J., Moseley, H.N.B., Montelione, G.T.: Automated analysis of protein nmr assignments and structures. Chemical Reviews **104**(8), 3541–3556 (2004). DOI 10.1021/cr030408p. URL https://doi.org/10.1021/cr030408p. PMID: 15303826
8. Bartels, C., Güntert, P., Billeter, M., Wüthrich, K.: Garant-a general algorithm for resonance assignment of multidimensional nuclear magnetic resonance spectra. Journal of Computational Chemistry **18**(1), 139–149. DOI 10.1002/(SICI)1096-987X(19970115)18:1⟨139::AID-JCC13⟩3.0.CO;2-H
9. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Res **28**(1), 235–242 (2000). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC102472/
10. Bodenhausen, G., J. Ruben, D.: Natural abundance nitrogen-15 nmr by enhanced heteronuclear spectroscopy **69**, 185–189 (1980)
11. Bromiley, P.: Products and convolutions of gaussian probability density functions. Tina-Vision Memo **3**(4), 1 (2003)
12. Cavanagh, J., Fairbrother, W.J., Palmer, A.G., Rance, M., Skelton, N.J.: Protein NMR Spectroscopy, first edn. Academic Press Limited, 24-28 Oval Road, London NW1 7DX (1996)
13. Coggins, B.E., Zhou, P.: PACES: Protein sequential assignment by computer-assisted exhaustive search. Journal of Biomolecular NMR **26**(2), 93–111 (2003)
14. Donald, B.R.: Algorithms in Structural Molecular Biology. The MIT Press (2011)
15. Donald, B.R., Martin, J.: Automated nmr assignment and protein structure determination using sparse dipolar coupling constraints. Progress in Nuclear Magnetic Resonance Spectroscopy **55**(2), 101 – 127 (2009). DOI https://doi.org/10.1016/j.pnmrs.2008.12.001. URL http://www.sciencedirect.com/science/article/pii/S0079656509000119
16. Ferreira, J.F.S.B., Khoo, Y., Singer, A.: Semidefinite programming approach for the quadratic assignment problem with a sparse graph. Comp. Opt. and Appl. **69**(3), 677–712 (2018). DOI 10.1007/s10589-017-9968-8. URL https://doi.org/10.1007/s10589-017-9968-8
17. Grzesiek, S., Bax, A.: Correlating backbone amide and side chain resonances in larger proteins by multiple relayed triple resonance nmr. Journal of the American Chemical Society **114**(16), 6291–6293 (1992). DOI 10.1021/ja00042a003. URL https://doi.org/10.1021/ja00042a003
18. Grzesiek, S., Bax, A.: An efficient experiment for sequential backbone assignment of medium-sized isotopically enriched proteins. Journal of Magnetic Resonance (1969) **99**(1), 201 – 207 (1992). DOI https://doi.org/10.1016/0022-2364(92)90169-8. URL http://www.sciencedirect.com/science/article/pii/0022236492901698
19. Grzesiek, S., Bax, A.: Amino acid type determination in the sequential assignment procedure of uniformly 13C/15N-enriched proteins. J Biomol NMR **3**(2), 185–204 (1993)
20. Guerry, P., Herrmann, T.: Comprehensive Automation for NMR Structure Determination of Proteins, pp. 429–451. Humana Press, Totowa, NJ (2012). DOI 10.1007/978-1-61779-480-3_22. URL https://doi.org/10.1007/978-1-61779-480-3_22
21. Güntert, P., Buchner, L.: Combined automated noe assignment and structure calculation with cyana. Journal of Biomolecular NMR **62**(4), 453–471 (2015). DOI 10.1007/s10858-015-9924-9. URL https://doi.org/10.1007/s10858-015-9924-9
22. Güntert, P., Salzmann, M., Braun, D., Wüthrich, K.: Sequence-specific nmr assignment of proteins by global fragment mapping with the program mapper. Journal of Biomolecular NMR **18**(2), 129–137 (2000). DOI 10.1023/A:1008318805889. URL https://doi.org/10.1023/A:1008318805889
23. Gurobi Optimization, L.: Gurobi optimizer reference manual (2020). URL http://www.gurobi.com
24. Hitchens, T.K., Lukin, J.A., Zhan, Y., McCallum, S.A., Rule, G.S.: MONTE: An automated Monte Carlo based approach to nuclear magnetic resonance assignment of proteins. Journal of Biomolecular NMR **25**(1), 1–9 (2003)
25. Jung, Y.S., Zweckstetter, M.: Mars - robust automatic backbone assignment of proteins. Journal of Biomolecular NMR **30**(1), 11–23 (2004). DOI 10.1023/B:JNMR.0000042954.99056.ad. URL http://dx.doi.org/10.1023/B%3AJNMR.0000042954.99056.ad
26. Karjalainen, M., Tossavainen, H., Hellman, M., Permi, P.: HACANCOi: a new HÎ±-detected experiment for backbone resonance assignment of intrinsically disordered proteins. J Biomol NMR (2020)
27. Leutner, M., Gschwind, R.M., Liermann, J., Schwarz, C., Gemmecker, G., Kessler, H.: Automated backbone assignment of labeled proteins using the threshold accepting algorithm. Journal of Biomolecular NMR **11**(1), 31–43 (1998)

28. Lian, L.Y., Barsukov, I.L.: Resonance Assignments, chap. 3, pp. 55–82. Wiley-Blackwell (2011). DOI 10.1002/9781119972006.ch3. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119972006.ch3
29. Schmidt, E., Güntert, P.: A new algorithm for reliable and general NMR resonance assignment. Journal of the American Chemical Society **134**(30), 12817–12829 (2012). DOI 10.1021/ja305091n. URL https://doi.org/10.1021/ja305091n. PMID: 22794163
30. Ulrich, E.L., Akutsu, H., Doreleijers, J.F., Harano, Y., Ioannidis, Y.E., Lin, J., Livny, M., Mading, S., Maziuk, D., Miller, Z., Nakatani, E., Schulte, C.F., Tolmie, D.E., Kent Wenger, R., Yao, H., Markley, J.L.: Biomagresbank. Nucleic Acids Research **36**(suppl 1), D402–D408 (2008). DOI 10.1093/nar/gkm957. URL http://nar.oxfordjournals.org/content/36/suppl_1/D402.abstract
31. Volk, J., Herrmann, T., Wuthrich, K.: Automated sequence-specific protein NMR assignment using the memetic algorithm MATCH. Journal of Biomolecular NMR **41**(3), 127–138 (2008)
32. Wan, X., Lin, G.: CISA: Combined NMR resonance connectivity information determination and sequential assignment. IEEE/ACM Transactions on Computational Biology and Bioinformatics **4**(3), 336–348 (2007). DOI 10.1109/tcbb.2007.1047
33. Yang, Y., Fritzsching, K.J., Hong, M.: Resonance assignment of the NMR spectra of disordered proteins using a multi-objective non-dominated sorting genetic algorithm. Journal of Biomolecular NMR **57**(3), 281–296 (2013)
34. Zeng, J., Zhou, P., Donald, B.R.: HASH: a program to accurately predict protein Hα shifts from neighboring backbone shifts. J. Biomol. NMR **55**(1), 105–118 (2013)
35. Zimmerman, D.E., Kulikowski, C.A., Huang, Y., Feng, W., Tashiro, M., Shimotakahara, S., ya Chien, C., Powers, R., Montelione, G.T.: Automated analysis of protein nmr assignments using methods from artificial intelligence. Journal of Molecular Biology **269**(4), 592 – 610 (1997). DOI https://doi.org/10.1006/jmbi.1997.1052. URL http://www.sciencedirect.com/science/article/pii/S0022283697910524

A - Grouping Peaks

As we mentioned in Section 3.1.1, grouping consistent peaks together is a crucial step in the graph creation process for $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. One would wish the enumeration of valid assignments to be as thorough as possible. We can effectively enumerate peak groupings to construct nodes in $\mathcal{G}$ by matching measured and expected peaks in a self-consistent way. In particular, we expect a specific set of peaks due to N–H^N from residue k (see Figure 10 for a standard example with three experiments) where the values of these peaks in $\mathbb{R}^3$ along certain dimensions are consistent. If there are n residues, we should have n sets of such expected peaks. Therefore, each layer in $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in principle should have n nodes, although in practice there are more nodes due ambiguities.

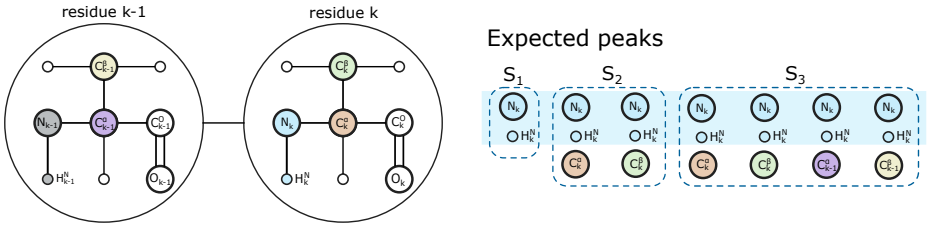

Fig. 10 With three NMR experiments (often HSQC, HNCACB, and HN(CO)CACB) we generally expect 7 distinct peaks for each base N–H^N pair in a residue, k. These peaks must be consistent - that is, the frequencies assigned to the same atom by two different peaks must be approximately the same up to some experimental tolerance. In principle, there should be n sets of such 7 peaks, one for each residue.

The notion of consistency can help significantly simplify the enumeration process (which would otherwise result in an exponential number of nodes). In order to efficiently enumerate consistent peak groupings, we do the following. Let $\mathcal{S}_1, \ldots, \mathcal{S}_L$ be collections of *measured* peak lists corresponding to different heteronuclear experiments, i.e. $\cup_{l=1}^{L} \mathcal{S}_l := [p_1, \ldots, p_{m_2}]$. In the case of Figure 10, $L = 3$, as we have peaks from three experiments. Now from these m_2 experimental peaks we form all combinations of seven peaks that each consists of one peak from $\mathcal{S}_1$, two peaks from $\mathcal{S}_2$, and four peaks from $\mathcal{S}_3$ using the following criteria.

– For any pair of p_u, p_v in a combination of seven peaks,

$$|p_u(1) - p_v(1)| \leq \delta_1$$
$$|p_u(2) - p_v(2)| \leq \delta_2.$$

This means that the frequencies of the seven peaks in the N–H^N dimension have to coincide up to tolerance δ_1, δ_2.

– Furthermore, for a combination of seven peaks, let p_u, p_v be the two peaks in $\mathcal{S}_2$. These peaks should coincide with two of the peaks in $\mathcal{S}_3$ (denoted p_i, p_j) up to tolerance δ_3, i.e.

$$|p_u(3) - p_i(3)| \leq \delta_3$$
$$|p_v(3) - p_j(3)| \leq \delta_3$$

along the C dimension.

B - Atom cost

Recall that we defined the cost of an atom, a, under a given set of assigned observations, $\{x_l\}_{l=1}^{o_a}$ as

Definition 3 (Atom cost) *The cost associated with atom a, with a normally distributed prior $\mathcal{N}(\mu_a, \sigma_a)$, and o_a observations $\{x_l^a\}_{l=1}^{o_a}$ defined by the peak grouping, also assumed to be normally distributed around the true frequency, μ, according to $\mathcal{N}(\mu, \sigma_l)$ is defined as*

$$cost\left(a, \{x_l^a\}_{l=1}^{o_a}\right) \triangleq -\log \mathbb{E}_{\mu\sim\mathcal{N}(\mu_a,\sigma_a)}\left[\prod_{l=1}^{o_a} f(x_l^a \mid \mu, \sigma_l)\right]. \tag{18}$$

where $f(\cdot \mid u, v)$ is the Gaussian density with mean u and standard deviation v.

This is Definition 1 in the main text. Note that the term inside the expectation is a product of o_a univariate Gaussian probability density functions. Furthermore, expanding the expectation, we note that

$$\mathbb{E}_\mu\left[\prod_{l=1}^{o_a} f(x_l^a \mid \mu, \sigma_l)\right] = \int_{-\infty}^{+\infty} f(\mu \mid \mu_a, \sigma_a) \prod_{l}^{o_a} f(x_l^a \mid \mu, \sigma_l) d\mu \tag{19}$$

$$= \int_{-\infty}^{+\infty} f(\mu \mid \mu_a, \sigma_a) \prod_{l}^{o_a} f(\mu \mid x_l^a, \sigma_l) d\mu \tag{20}$$

by symmetry. Using a standard result regarding the product of univariate Gaussian PDFs (see, e.g., [11]), we can write

$$\mathbb{E}_\mu\left[\prod_{l=1}^{o_a} f(x_l^a \mid \mu, \sigma_l)\right] = \int_{-\infty}^{+\infty} f(\mu \mid \mu_a, \sigma_a) \prod_{l}^{o_a} f(\mu \mid x_l^a, \sigma_l) d\mu \tag{21}$$

$$= \int_{-\infty}^{+\infty} Z_a f(\mu \mid M_a, \Sigma_a) d\mu \tag{22}$$

$$= Z_a \tag{23}$$

where

$$\Sigma_a = \left(\frac{1}{\sigma_a^2} + \sum_{l=1}^{o_a} \frac{1}{\sigma_l^2}\right)^{-1/2} \tag{24}$$

$$M_a = \left(\frac{\mu_a}{\sigma_a^2} + \sum_{l=1}^{o_a} \frac{x_l}{\sigma_l^2}\right) \Sigma_a^2 \tag{25}$$

$$Z_a = \frac{1}{(2\pi)^{o_a/2}} \sqrt{\frac{\Sigma_a^2}{\sigma_a^2 \prod_{l=1}^{o_a} \sigma_l^2}} \exp\left[-\frac{1}{2}\left(\frac{\mu_a^2}{\sigma_a^2} + \sum_{l=1}^{o_a} \frac{x_l^2}{\sigma_l^2} - \frac{M_a^2}{\Sigma_a^2}\right)\right]. \tag{26}$$

We see that this choice of cost function is therefore computationally advantageous, as the desired expectation is a simple function of the observations, $\{x_l\}_{l=1}^{o_a}$ and of the distributional parameters of the prior, (μ_a, σ_a) and experiments, $\{\sigma_l\}_{l=1}^{o_a}$. That said, it is certainly not the only cost function that one could use. As an example, we could instead solve a maximum likelihood problem for each peak grouping that would assign the highest likelihood frequency to each atom, given the prior and the observations. The exploration of alternative cost functions is left for future work.

C - Statistical Typing

Statistical typing is a process that happens both during the node and edge creation steps. In particular, we want to avoid the creation of nodes and edges which are too unlikely to constitute a valid assignment. The way we action on this notion is to define a threshold below which we would rather have a *null* assignment than the assignment induced by the relevant nodes. This threshold also determines the cost of the edges to (and from) the *dummy* nodes, which are therefore the highest cost edges in the graph.

For all simulations in this paper, we use the following definition:

Definition 4 (Atom cost threshold) *The maximum allowable cost associated with atom a, with an expected frequency, μ, distributed according to the normally distributed prior $\mathcal{N}(\mu_a, \sigma_a)$, and a total of o_a expected observations is given by:*

$$threshold(a) \triangleq cost(a, \{w_l^a\}_{l=1}^{o_a}) \tag{27}$$

where

$$w_l^a = \mu_a + \delta\sigma_a + (-1)^{l+1}\delta\sigma_l. \tag{28}$$

That is, we define the maximum allowable cost for atom a by setting $\{x_l^a\}_{l=1}^{o_a}$ in Definition 1 to $\{w_l^a\}_{l=1}^{o_a}$, which constitute an adversarial realization of the observations. In this realization, the mean of the observations is $\approx \delta$ standard deviations away from the prior mean, and the observations are split into two clusters, 2δ experimental standard deviations apart.