J Biomol NMR (2010) 46:281–298
DOI 10.1007/s10858-010-9403-2

ARTICLE

SAGA: rapid automatic mainchain NMR assignment for large proteins

Gordon M. Crippen · Aikaterini Rousaki · Matthew Revington · Yongbo Zhang · Erik R. P. Zuiderweg

Received: 7 December 2009 / Accepted: 23 February 2010 / Published online: 16 March 2010

Abstract Here we describe a new algorithm for automatically determining the mainchain sequential assignment of NMR spectra for proteins. Using only the customary triple resonance experiments, assignments can be quickly found for not only small proteins having rather complete data, but also for large proteins, even when only half the residues can be assigned. The result of the calculation is not the single best assignment according to some criterion, but rather a large number of satisfactory assignments that are summarized in such a way as to help the user identify portions of the sequence that are assigned with confidence, vs. other portions where the assignment has some correlated alternatives. Thus very imperfect initial data can be used to suggest future experiments.

Keywords Automatic assignment · Generic spin system · Triple resonance · Large proteins

Electronic supplementary material The online version of this article (doi:10.1007/s10858-010-9403-2) contains supplementary material, which is available to authorized users.

G. M. Crippen (✉)
College of Pharmacy, University of Michigan,
Ann Arbor, MI 48109, USA
e-mail: gcrippen@umich.edu

A. Rousaki · M. Revington · Y. Zhang
LSA Biophysics, University of Michigan,
Ann Arbor, MI 48109, USA

Present Address:
M. Revington
Department of Chemistry and Biochemistry,
University of Windsor, Windsor, ON N9B 3P4, Canada

Present Address:
Y. Zhang
Department of Biochemistry, Molecular Biology
and Cell Biology, Northwestern University, Evanston,
IL 60208, USA

E. R. P. Zuiderweg (✉)
Department of Biological Chemistry, University of Michigan
Medical School, Ann Arbor, MI 48109, USA
e-mail: zuiderwe@umich.edu

Introduction

Presently, the backbone resonances of virtually all proteins in solution are assigned using a combination of 3D triple resonance NMR spectra pairs HNCA/HN(CO)CA, HNCA CB/HN(CO)CACB, HN(CA)CO/HNCO (for perdeuterated proteins) or HN(CA)HA/HN(COCA)HA for small proteated proteins (e.g., Cavanagh et al. 2007). Similar combinations of experiments have been devised for the assignments of the backbone resonances of proteins in the solid state (e.g., Baldus 2007). Smaller or unfolded proteins in solution are increasingly assigned using higher dimensional data with similar or extended coherence pathways, typically obtained by using projection methods on three frequency coordinates (e.g., Atreya and Szyperski 2004).

Here, we present an automated assignment algorithm for a combination of the "classical" 3D triple resonance experiments from proteins in solution. We focus on these 3D experiments because their relative simplicity allows the highest sensitivity per unit time, which is important for the assignment of the larger proteins we study in our laboratory (>300 residues), which are typically soluble at 300 μM concentration or less.

In theory, the 3D experiments provide more than sufficient resolution to assign the spectra of such large proteins: all $^{13}C\alpha$ resonance frequencies of a 50 residue protein are likely to be unique within an achievable 0.1 ppm precision

(see Fig. S1); similar values hold for $^{13}C\beta$ frequencies (within an achievable 0.2 ppm precision) and ^{13}CO frequencies (within an achievable 0.1 ppm precision). Hence, if the sequence specific shifts of the Cα, Cβ and CO resonance positions may be assumed to be uncorrelated, a combination of an HNCA/HN(CO)CA pair with HN(CA)CO/HNCO pair should suffice to assign a $50 \times 50 = 2500$ residue protein. Combined with Cβ connectivity information from an HNCACB/HN(CO)CACB pair, the size limit increases further, and when the latter is combined with chemical shift statistics which allows the distinction of 6 different amino acid groups (A; G; P; S+T; D+F+I+L+N+Y; E+C+H+K+M+Q+R+V+W, see Fig. S2), the size limit of assignments for proteins with just triple resonance methods seems much larger than any protein that will likely be ever studied by solution NMR methods.

While the information content of such spectra is sufficient, manual assignment of the spectra to the known amino acid sequence is extremely tedious for large proteins, and it has been hard to automate, especially for spectra of large proteins with incomplete data. Many computer algorithms have been devised (Friedrichs et al. 1994; Hare and Prestegard 1994; Hyberts and Wagner 2003; Leutner et al. 1998; Lukin et al. 1997; Meadows et al. 1994; Moseley et al. 2001; Olson and Markley 1994; Oschkinat and Croft 1994; Stratmann et al. 2010; Zimmerman et al. 1993, 1994, 1997; Zimmerman and Montelione 1995). See also recent reviews (Billeter et al. 2008; Williamson and Craven 2009; Güntert 2009).

Here we adopt the nomenclature of the AutoAssign program (Zimmerman et al. 1997; Moseley et al. 2001) to explain in broad terms common and distinguishing features. The general paradigm has been to first assemble peaks from the triple resonance experiments, and group these into generic spin systems (GS) that specify the chemical shifts of various nuclei of a particular residue and the sequentially previous residue. Next, the GSs are arranged into sequentially ordered groups (segments), where there is a unique match in chemical shifts between successive GSs in a segment. Finally, the segments are placed on the sequence in such a way as to optimize some scoring function based on amino-acid residue chemical shift statistics, such as shown in Fig. S2. A common theme is that the objective is the single best assignment according to some criteria, such as number of residues assigned and accuracy of agreement between chemical shifts in a GS and the sequence position to which it is assigned. Thus many different optimization procedures are employed by the various assignment programs to find a relatively good solution. The other common theme is to group GSs into segments before attempting to place them on the sequence. The motivation is that any individual GS may be quite compatible with dozens of sequence positions, whereas a segment of several GSs may have only a few options, thus drastically reducing the combinatorial search.

Without making a comprehensive review of automatic sequential assignment methods, it is worth noting recent activity in this lively field. The CRAACK program (Benod et al. 2006) takes the usual input data and forms GSs, but then its objective is to assign them to residue types and secondary structure states, rather than assigning them to the sequence. Other methods aim to assign GSs to the sequence but employ additional inputs to the problem. For example, ABACUS (Lemak et al. 2008) uses the usual GSs plus NOESY cross peaks to help determine sequence separations in the assignment. NOEnet (Stratmann et al. 2010) uses largely NOEs plus residual dipolar couplings and chemical shifts. Xiong et al. (2008) start with a model of the three-dimensional structure of the protein, perhaps derived from homology modeling, plus NMR data from HSQC, TOCSY, and NOESY experiments. From this they deduce rough residue types and choose from the many possible NOESY cross peaks those that are consistent with the residue types and interresidue contacts in the model.

Another issue is the assessment of reliability of the one or more assignments produced by a method. In ABACUS (Lemak et al. 2008) the Monte Carlo search for assignments gives the probability of assigning a particular segment of GSs to a sequence position. The IDA method (Lin et al. 2006) gives exactly one assignment for a given set of GSs plus a score reflecting how good the assignment is. Gaussian perturbations of the rungs (chemical shifts) of the GSs produce different assignments and different scores, so an alternative assignment can be associated with a Z-score relative to the assignment from unperturbed GSs.

Another question is how best to structure the assignment search. Our earlier program, CASA (Wang et al. 2005), follows the standard organization of peaks into GSs into segments, but the subsequent branch-and-bound search for assignments was carefully structured for speed and robustness. In PISTACHIO (Eghbalnia et al. 2005) GSs are associated with tripeptides so that the assignment amounts to assigning GSs to mutually consistent overlapping tripeptides in the given sequence. In GASA (Wan and Lin 2007) the usual sequence of steps (peaks to GSs, GSs to segments, segments to sequence positions) are overlapped in order to improve performance. For example, the best way to link GSs together into a segment is judged by a score for how well such a segment might fit onto the sequence. The usual logic for joining GS_i to GS_j to form a segment GS_iGS_j is that (1) the two sets of rungs agree well enough in terms of numbers of rungs involved and similarity of corresponding chemical shifts, and (2) there is no other GS_k that would equally well form segments GS_iGS_k or GS_kGS_j. Vitek et al. (2005) argue that even though such a unique link seems too good to be coincidental, one can

never exclude the possibility that the GS that really follows GS_i in sequence is simply missing from the input data. Hence they structure the assignment search by breaking the whole sequence up into smaller non-overlapping windows and then considering what alternative assignments of GSs suit each window. Deeper in the branch-and-bound search, two sequentially adjacent windows having relatively few alternative assignments can be merged into a larger window, and so on.

Methods

Here we present a new automatic backbone assignment procedure, SAGA (sequential assignment of GSs algorithm). The inputs are the amino acid sequence of the single polypeptide chain and at least some of the customary triple resonance peaks. Additional information is not used, such as NOEs, secondary structure, or homology models. As explained in detail below, the first step is to assemble the spectral data into GSs, as do most assignment programs. However, unlike most methods and our earlier work (Wang et al. 2005), the GSs are not subsequently assembled into apparently unambiguous segments, as suggested by Vitek et al. (2005). The second step is to assign the GSs to the sequence. Most methods define a quantitative measure of quality for an assignment and then use stochastic global optimization methods such as Monte Carlo or simulated annealing to search for the single best quality assignment. The second unusual feature of SAGA is that the user instead defines criteria that an acceptable assignment satisfies, and the output consists of a relatively broad sampling of perhaps very many acceptable assignments. In this way, we hope to learn what parts of the assignment are known with high confidence, what parts may be the result of slowly interconverting conformations, and what features suggest the need for scrutiny of the input data or additional experiments. In particular, one of the acceptability criteria is a lower bound on the fraction of the GSs that are assigned to residues. This allows one to focus only on rather complete assignments in the case of high-quality data, or to see what parts of the assignment have been well established at an early stage of a study where, e.g., only half the residues can be assigned. The third unusual feature of SAGA is that more than one algorithm is provided for searching for acceptable assignments. While most methods center around a particular search or optimization algorithm that works well for at least some test cases, we find that quite different algorithms are needed for thorough searches given a small protein and high-quality peak data, as opposed to broad samplings of assignments given a large protein with many peaks lost in the noise. The three search algorithms presented below are based on well-established general combinatorial methods, such as clique finding, branch and bound tree searching, and greedy searches. However, tailoring these general approaches to the backbone sequential assignment problem constitutes the fourth unusual feature of SAGA, and this accounts for SAGA's pleasing performance, even on challenging data.

Peaks to GSs

The first step is to assemble the peaks gleaned from several triple-resonance experiments into data structures called generic spin systems (GS) (Zimmerman et al. 1997). A GS consists of the chemical shifts of several atoms in some unspecified amino acid residue i (intra-residue) and the sequentially previous residue $i-1$ (sequential). These always include the intra-residue HN root, H_i^N and N_i, and optionally various rungs on the i and $i-1$ sides for Cα, CO, and Cβ atoms.

One way to input peak information is via NMRPipe-format (Delaglio et al. 1995) peak-pick lists derived from corresponding pairs of intra-residue and sequential experiments: HNCA and HN(CO)CA for Cα, HNCACB and HN(CO)CACB for Cβ, and HN(CA)CO and HNCO for CO, respectively. We require the first pair of experiments so that the GSs all have at least an intra-residue or sequential Cα rung. For each available pair of experiments, allowance is made for a possible uniform additive shift between the two experiments of no more than 0.1 ppm in H^N and 0.75 ppm in N. Pairs of peaks are considered to match if their (shifted) H^N and N chemical shifts agree within a given tolerance while the third chemical shifts (e.g. Cα) differ by at least 0.25 ppm. Tolerances in H^N and N are varied from 0.005 to 0.03 and from 0.05 to 0.3 ppm, respectively, until the maximum number of uniquely matched pairs of peaks is achieved. In this way, each pair of available peak lists produces a set of so-called Ts, which are GSs having an HN root and one or both rungs for the intra-residue and sequential chemical shifts of either Cα, Cβ, or CO.

Exploring the same overall shift range and tolerance ranges used for constructing Ts, the Cα Ts are combined with the available Cβ and then CO Ts, choosing the shifts and tolerances that give the maximal number of unambiguous matches. If a resulting GS has no rungs on either the intra-residue or sequential side, it is still accepted.

An alternative format for peak information is Sparky (Goddard and Kneller 2003), where one Sparky file substitutes for one of the three pairs of NMRPipe files because the intra-residue and sequential pairings are already specified by the user through the use of matching tag pairs in the file. However, the resulting Ts need not always have a matching i and $i-1$ rung, due to experimental noise considerations. Missing rungs are denoted by entries larger than 500 ppm. This input style is especially suited for noisy

and incomplete data from large proteins, where an investigator needs to curate peak-pick files by hand. The HN root chemical shifts are taken to be those of the intra-residue peak when both peaks are available. Associated with each T is the common tag specified by the user. Using Sparky files instead of peak pick lists affords the spectroscopist greater control over building GSs. Ts from more than one Sparky file are joined together into a GS by the program by matching up their tags, where there must be a Cα T, and the resulting GS has its HN root. That is, in the Sparky input mode, the ^{15}N–^{1}H frequencies are not used to construct GSs, which relieves some of the criteria on the quality of the input NMR data. In this input mode, the investigator can also easily assess the precision of the peak matching. The Sparky program allows hand-adjustment of the center of the peak-pick positions. After these optimizations are carried out, we find that in properly zero-filled data, the center of ^{13}Cα peaks can be defined with a 0.1 ppm, ^{13}Cβ with a 0.2 ppm and ^{13}CO with a 0.1 ppm precision, even for data sets of large proteins. The ^{15}N–^{1}H frequencies data are taken from the Cα T and are kept by the program for amino acid identification purposes (significant for the ^{15}N frequencies of glycyls only), and are returned at the output stage.

If the GSs are constructed from NMRPipe files, they don't have the unique identifier tags seen in GSs built from Sparky files. As such tags are helpful later in communicating the results to the user, the program automatically constructs tags for any GSs that lack them.

GSs to matching graph

Only after the GSs are prepared is the sequence considered at all. This is read from a FASTA format file, where in addition to the standard 20 single character symbols for residue types, B is recognized as either D or N, and Z is recognized as either E or Q. Each GS may occupy residue i in the sequence if its intra-residue rungs are in agreement with the chemical shifts expected for the type of that residue, and if its sequential rungs agree with the type of residue $i-1$ in the sequence. A GS that fits nowhere in the sequence is deleted, but most are compatible with multiple sequence positions. In any case, compatibility of a GS with a residue is treated as a qualitative yes/no decision, rather than assigning a quantitative compatibility score.

As in our previous algorithm (Wang et al. 2005), GS occupancy is based on chemical shift statistics from the BioMagResBank (Seavey et al. 1991; http://www.bmrb.wisc.edu). For each atom type in each residue type they provide the chemical shift a = lower bound, b = upper bound, μ = mean, and σ = standard deviation. Then a chemical shift from a GS rung is in agreement with the corresponding data bank statistics if it falls in the interval = $[\max(a, \mu-s\sigma), \min(b, \mu+s\sigma)]$, where by default $s = 4.0$ except $s = 5.5$ for H because of the larger dispersion. The user of SAGA can optionally choose other values for s (see Fig. S2). The data bank values of σ generally are much smaller than the standard deviation for a uniform distribution over the interval $[a,b]$, so accepting chemical shifts that are four standard deviations from the mean is not so permissive as it might seem. In our tests, however, we used $s = 6$ for ^{1}HN, $s = 6$ for ^{15}N, $s = 4$ for ^{13}Cα, $s = 2.5$ for ^{13}Cβ and $s = 3$ for ^{13}CO shift ranges, based on our own experience with assigning spectra of large proteins.

Each GS is tested for possible occupancy at each assignable residue position, excluding the N-terminus and prolines, of course. The chemical shifts of the N and all available intra-residue and sequential Cα, Cβ, and CO rungs must fall within the corresponding allowed intervals. The amide proton chemical shift is not considered in determining occupancy. As a special case, if a GS has an intra-residue Cα chemical shift less than 50 ppm and no intra-residue Cβ rung, then it must occupy a glycyl residue, as long as the other sequential rungs match satisfactorily. Another special case is a GS having an intra-residue Cβ rung less than 20 ppm, in which case it must occupy an alanyl residue where the sequential rungs match.

SAGA allows the user to optionally include further constraints on occupancy derived from other experiments. Each constraint specifies a GS by giving its tag and HN root chemical shifts. The constraint requires it to occupy certain residue types and/or certain sequence positions. Ordinarily this is used to narrow down the list of possible occupancies already determined, but in exceptional circumstances, the constraints can completely override them.

Once the possible occupancies of all GSs have been established, the links between GSs must be determined. That is, for each GS_j, can GS_k immediately follow it in the sequence? One necessary condition is that there is at least one sequence position i that GS_j can occupy, and GS_k can also occupy position $i + 1$. The other necessary condition is that all available intra-residue Cα, Cβ, and CO rungs of GS_j must match the corresponding available sequential rungs of GS_k within default tolerances of 0.1, 0.2, and 0.1 ppm, respectively. The user may optionally choose different tolerances. In the vacuous case of no corresponding rungs between the two GSs, the link is still considered possible.

Matching graph

One way to visualize the sequence, GSs, occupancies of GSs, and links between GSs is as a bipartite graph of GS vertices vs. residue vertices. There may be more or fewer GSs than assignable residues, depending on factors such as

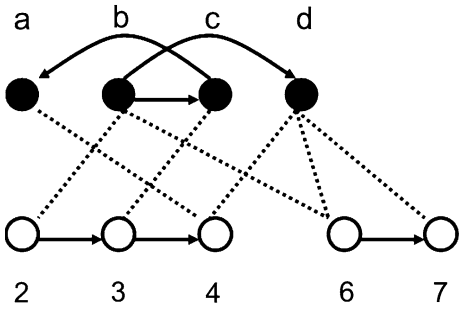

Fig. 1 A simple bipartite assignment graph. *Open circles* are assignable residues with corresponding sequence positions, and *arrows* between them indicate sequentially adjacent residues. *Filled circles* are GSs with links between them shown as *arrows*. Edges between GSs and assignable residues are *dashed lines*

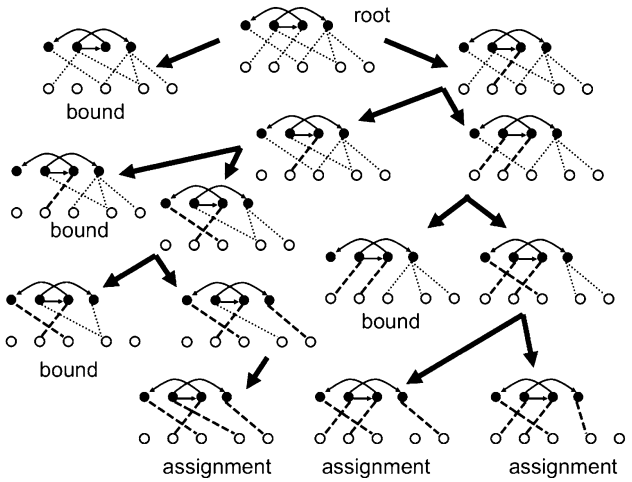

Fig. 2 An example branch-and-bound binary search tree starting with the bipartite graph in Fig. 1. Each branch involves either adding some GS to residue assignment, *highlighted* as a *bold dashed line*, or eliminating that edge from the bipartite graph. A bounding condition requiring that all GSs eventually be used eliminated from further consideration those tree vertices marked "bound"

the noise floor for detecting peaks, peaks rendered unobservable due to conformational exchange broadening, and possible multiple conformations in slow exchange. Each assignable residue may be occupied by zero or more different GSs, each GS may occupy at least one residue, and each GS may be sequentially linked to zero or more other GSs. Before describing the assignment algorithms in general, consider a very simple example shown in Fig. 1. Suppose the polypeptide chain consists of only seven residues, sequence being ADREPLE, so that five residues are assignable, $i = 2$, 3, 4, 6, and 7. Suppose there are four GSs having tags *a*, *b*, *c*, and *d*, where *a* and *c* have only a single possible occupancy, *b* may occupy residues 2 or 6, and *d* may occupy residues 4, 6, or 7. As for linking, *b* may be followed by *c* or *d*, but none follows *a*. Converting the initial bipartite graph to an assignment amounts to removing some of the GS to residue edges until each GS vertex and each residue vertex has zero or one edge. If two GSs are thus assigned to sequentially adjacent residues, there must be a corresponding link from the first to the second GS. As a shorthand notation for assignments, write five characters for the five assignable residues, where "#" means "unassigned" and comes first in the alphabet. Then there are 35 possible assignments in this simple example, ranging from ##### to bcad#, of which five are maximal in the sense that no additional residue can be assigned. The maximal assignments are ##db#, b#d##, bcad#, bca#d, and #cabd, the last three being shown in Fig. 2.

Clique algorithm

There is a very long history of algorithms for bipartite matching, where the initial graph looks like Fig. 1 with only the GS vertices and residue vertices joined by dashed edges between a GS and a residue, but without the directed edge arrows between residue vertices for sequential adjacency or the directed edges between GS vertices for link compatibility. The standard bipartite matching algorithms try to find the largest subset of the dashed edges such that no more than one edge connects to each vertex, or given weights associated with each such edge, they try to maximize the sum of the weights in the final matching graph (Kuhn 1955; Hopcroft and Karp 1973; Kao et al. 2001). Our sequential assignment problem is qualitatively different because of the link compatibility feature between GSs, which excludes certain *pairs* of dashed edges in the matching graph. This problem was treated in the equivalent context of calculating ways to dock a small molecule (represented as one set of vertices) to a binding site (represented as another set of vertices), where the edges correspond to energetically favorable ligand-receptor interactions, but geometric constraints exclude certain pairs of edges from the solutions (Kuhl et al. 1984). In that work, it was shown that for arbitrary pair exclusions, the problem in general mapped to an NP-complete one, so that any algorithm would require exponentially increasing computer time as the problem's size increased. However, for the assignment problem the situation is by no means hopeless. First, if even a non-polynomial algorithm runs fast enough for practical application to the largest proteins NMR can handle (around 800 residues), then that is adequate, and we need not worry about the algorithm's behavior in the limit of infinite proteins. Second, in the assignment problem the pair exclusions are by no means arbitrary, so for that restricted class of problems there may be a polynomial time algorithm.

As in Kuhl et al. (1984), the matching graph, such as in Fig. 1, can be transformed into an assignment graph, where every vertex in the assignment graph represents an edge of the matching graph, and there are edges between vertices of the assignment graph whenever the two vertices are mutually compatible. In this way, we can encode the constraint that each GS and each residue is used at most once, and GSs assigned to sequentially adjacent residues must be link compatible. Then an assignment of the GSs to the sequence corresponds to a clique of the assignment graph, where a clique is defined to be a maximal, completely connected subgraph.

More specifically, the transformation of the matching graph into the assignment graph is quite straightforward, but the assignment graph tends to be rather large. If GS_j can occupy m_j residues in the sequence, the assignment graph has $\sum_j m_j = k$ vertices and has nearly $k(k–1)/2$ edges because assigning one GS to a particular residue is compatible with most other assignments of another GS to its possible residues. There are a number of different general clique finding algorithms available, and we used that of Bron and Kerbosch (1973), which performs well for large graphs. Without going into the intricate details of the algorithm, we did not attempt to find all cliques, as that would include many small ones, corresponding to few assigned GSs. Instead, we used the option to guide the algorithm's branch-and-bound search for cliques toward those that would be at least as large as the largest so far found, and in the end we retain only the ten largest cliques. Converting a clique into an assignment involves nothing more than assigning the GS of each vertex to the residue for that vertex, and then automatically each GS and each residue is used at most once, and sequentially adjacent assigned residues are paired with link-compatible GSs.

Greedy assignment algorithm

One standard paradigm for solving combinatorial problems is the so-called greedy approach, where the best available choice is made at each stage according to some criterion regardless of the ultimate consequences, and the choice at an earlier stage is never revised. Obviously there is no guarantee that the best solution will be found, but it is a fast and reliable way of finding a fairly good one, even in the face of a vast number of possibilities. This general approach has long been applied to many different combinatorial or discrete optimization problems, so it is discussed in many standard textbooks (Parker and Rardin 1988). To apply the greedy approach to a particular class of problems, one must specify what constitutes a choice and how to measure the quality of a choice. How well it performs on that class of problems depends critically on these two definitions.

For the sequential assignment problem, the starting point amounts to a bipartite graph such as Fig. 1, where there are many edges (shown as dashed lines) between GS vertices and the residue vertices they *may* occupy. Even if a GS has only one such edge to a residue that has no alternative edges, the final assignment might not involve that edge. The definition of a choice in our tailored greedy algorithm amounts to selecting one such edge, marking it as "required" (shown as a bold dashed line in Fig. 2), and then simplifying the graph by deleting any alternative edges from the GS and its required assigned residue. In Fig. 2, different choices are illustrated as the right-hand branches in the tree.

The crucial second definition is a measure of how good a choice is. A good choice amounts to assigning some GS to a residue where the GS to residue compatibility is relatively certain, and where there are adjacent residues which have been or could be assigned to other GSs that link together relatively certainly to form a contiguous assigned segment of the chain. Note that choices involving a residue near a proline or near a chain terminus tend to be worse than for a residue in the middle of a five-residue assignable segment. As a measure of the certainty of assigning GS i to residue r, we use w_i = the total number of intra-residue and sequential rungs for GS_i, all of which must match the types of residues r and r–1 for any allowed occupancy of that GS. As a measure of the certainty a link from GS_i to GS_j, we use the number of matching rungs available, $v_{i,j}$. Then the total score $s(i,r)$ for choosing GS i and residue r looks at the best combination of link-compatible GSs that may occupy the previous and subsequent two residues, if available. In order to favor a choice that extends a segment having some already assigned residues, let $x_{j,r+1} = 2$ if the GS_j to residue $r + 1$ edge is required, or 0 otherwise.

$$s(i,r) = w_i + \max_{\substack{j,r+1\\ k,r+2\\ m,r-1\\ n,r-2}} (w_j + v_{i,j} + x_{j,r+1} + w_k + v_{j,k} + x_{k,r+2} + w_m + v_{m,i} + x_{m,r-1} + w_n + v_{n,m} + x_{n,r-2})$$

The greedy assignment algorithm is then very simple. (1) Start with the initial bipartite graph, where no edges are required. (2) Calculate the score for all edges that are not required. (3) Generally, more than one edge has the maximum score. Choose at random one of these. (4) Mark the chosen edge as required and remove any other edges from that GS and that residue. (5) If all edges are required, an assignment has been found. Otherwise, return to step 2.

As a simple example, consider the bipartite graph in Fig. 1. Suppose $w_i = 1$ for all GSs, and all links are equally strong, $v_{i,j} = 1$. Then $s(b,2) = s(c,3) = s(a,4) =$

5 because for each of these edges, the other two are link compatible to sequentially adjacent residues. In contrast, $s(d,6) = 1$ because no other edge could be required that would have a link compatible GS to a sequentially adjacent residue. Thus the greedy algorithm first requires c to 3, then b to 2, and then a to 4. At this point $s(d,6) = s(d,7) = 1$, and either can be chosen. In our shorthand notation, the assignment proceeds from ##### to #c### to bc### to bca## to either bca#d or bcad#. This sequence of events is illustrated in the rightmost branch of Fig. 2.

As shown in the results section, for sequential assignment problems involving large proteins and GSs having many missing rungs, the greedy approach is a recommendable way to explore the different possible assignments. Because of the random selection in step 3, the different assignments found tend to be a broad sampling of the possibilities.

Branch-and-bound assignment algorithm

Another combinatorial optimization approach discussed in standard textbooks (Parker and Rardin 1988) is branch-and-bound. As illustrated in Fig. 2, the idea is to search all possibilities arranged like a family tree, starting at the initial problem (the root) drawn at the top, and working down the branches toward solutions (leaves of the tree). To be efficient for a particular class of problems, one must choose a definition for branching and a test for bounding. We also choose a depth-first search, as opposed to a breadth-first search, so that relatively promising branches are explored downward in search of solutions in case the full examination of the tree is infeasible.

For the sequential assignment problem, a branching procedure we have found to be suitable is just the assignment choice used in the greedy algorithm above. Thus a search tree vertex splits into two children vertices: one where the graph edge between the chosen residue r and GS i has been removed, and the other where the i to r assignment has been required and all other edges to vertices i and r have been removed. The edge that is required and deleted in the two children vertices of the search tree is simply the first edge with the maximum score calculated as in the greedy procedure. This tends to require relatively promising edges early in the search tree that will lead to segments of assigned residues. However, the branch-and-bound exhaustive search tree examines all the possible choices if time permits, so the ultimate results do not depend critically on the ordering of edge choices.

The bounding test amounts to an upper bound on the number of GSs that could possibly be used. Considering only the GS to residue edges in the bipartite graph, the whole graph may consist of one or more connected, mutually disjoint subgraphs. For each subgraph, the maximal number of GSs that can be assigned is the lesser of the number of GS vertices and residue vertices. The upper bound on the whole graph is the sum of these estimates for all the subgraphs. The user specifies the minimal fraction of GSs that must be assigned to residues in order to be an acceptable assignment. If the upper bound is less than that fraction, then further exploration from this vertex in the search tree is pointless.

Having chosen a branching procedure and a bound test, the search procedure is very straightforward. (1) Start with the initial bipartite graph at the root. (2) If the current search tree node is an assignment (all edges are required), then test it (step 3) and backtrack (step 4); otherwise apply the bound test (step 5). (3) If the required fraction of GSs have been assigned, and this assignment has not been seen before, add it to the list of solutions. (4) Move up the tree to the first higher node having an unexplored child, and move to it for a repeat of step 2. If no such node can be found, the backtracking has returned to the root, and the search is complete. (5) If the current node fails the bound test, backtrack (step 4); otherwise branch (step 6). (6) Branch by creating two child search nodes by deleting and requiring the first edge with highest score. Then move to the child with the required edge for a repeat of step 2.

The search is illustrated on an extremely simple example in Fig. 2, where the bounding test requires that 100% of the GSs must be used in each assignment, and the branching choices are prioritized by the score described in the greedy algorithm. As shown in the figure, the right-hand branches correspond to requiring an edge, while the left-hand branches eliminate that edge. Note how the graphs become simpler as more edges are required. The greedy algorithm would travel down the right side of the tree always making assignments, whereas the branch-and-bound algorithm searches the tree more thoroughly and discovers a third assignment.

The total size of the search tree grows rapidly with increasing protein size and numbers of GSs, and with decreasing numbers of rungs on the GSs. For high quality data on a small protein, it is possible in a reasonable time to search the whole tree for assignments using nearly all the GSs. Otherwise, branching continues until a preset time limit is reached, but this produces a deceptive set of assignments that all agree on the earlier parts of the tree simply because the branch-and-bound search did not have a chance to backtrack very far from the leaves corresponding to acceptable assignments. In order to get the broader sampling of the greedy algorithm while also enjoying the branch-and-bound algorithm's better local searching for adequately good assignments, we have combined them. The allotted time is divided into, say, tenths for ten different attempts. Each attempt consists of a greedy descent of the search tree with different random choices of required edges,

followed by a branch-and-bound search from that vertex in the tree, enumerating alternative acceptable assignments until the time for that attempt expires. For challenging assignment problems, this gives a fairer assessment of the variety of acceptable assignments where any consistencies are more likely to be genuine features of the data and less likely to be artifacts of the search procedure. So either by the purely greedy algorithm or this combination procedure, the general result is either some set of acceptable assignments or none when the time limit was too low, or the demands on fraction of GSs to be used were too high.

Examining multiple assignments

If multiple assignments are found, some subsequent analysis is helpful to determine which features are consistent throughout all assignments, and can therefore be given high confidence. Also when some alternative features are consistently associated with certain other features, such a set may characterize one of the alternative conformations of the protein. Consider the three assignments shown at the bottom of Fig. 2. GS *c* is always assigned to residue 3, and GS *a* is always assigned to residue 4. GS *b* can be assigned to either residue 2 or 6. In the former case, GS *d* can be assigned to either residue 6 or 7, while in the latter case, GS *d* is consistently assigned to residue 7.

For large proteins and incomplete data, SAGA can sometimes find thousands of different assignments, but they can be summarized in a readable form in the following way. Subsets of residues consistently assigned the same way are given an identifier, starting with 0 for the set of residues that are assigned consistently in all assignments, then 1 for the set of residues consistently assigned in the next largest subset of assignments, and so on. Then for each residue there are zero or more different GSs assigned to it, each alternative being marked with an identifier to show what other residue assignments are consistent with it. Table 1 shows how the three assignments in Fig. 2 can be presented, for example. Note that residue R3 is consistently assigned to GS c, and P5 is consistently unassigned of course, so they share the assignment identifier 0. However in 33% of the assignments, residue L6 is assigned to GS b, but whenever this happens, residue D2 is unassigned, so to show this correlation, they share the identifier 3.

Software implementation

Saga is written in Python v. 2.6 as a standalone program, and it should be fairly operating system independent. It has been tested on several Macintosh computers with Intel CPUs. For academic use, the program is available from the authors free of charge, with citation obligation. Industrial users should contact the University of Michigan Technology Transfer Office. Most of the tests in the following section were run on an Apple MacBookPro5 with a 2.4 GHz Intel Core 2 Duo processor and 2 GB of memory.

Table 1 Summary of multiple assignments from Fig. 2

Residue	GS	Identifier	%
A1		0	100
D2	b	1	67
D2		3	33
R3	c	0	100
E4	a	0	100
P5		0	100
L6	b	3	33
L6	d	4	33
L6		5	33
E7	d	2	67
E7		4	33

Results and discussion

Viability

For a simple but realistic test of SAGA, consider the favorite test protein, ubiquitin. There are 78 residues in the chain, of which 74 are assignable. Constructing GSs from a full set of Sparky files (HNCA, HNCB, and HNCO) produces 69 GSs, all of which can occupy one or more residues. The mean number of edges associated with each GS or residue vertex is 11, so there is nontrivial ambiguity in even such a high quality dataset for a small protein. The more traditional approach taken by our earlier CASA algorithm first forms the 69 GSs into five segments of sequentially joined GSs, and these are quickly and unambiguously placed on the sequence, thereby assigning 69 residues. The single assignment agrees exactly with the manual assignment. SAGA, however, does not insist on initially forming segments of GSs. Yet the greedy algorithm finds the same assignment in just 1 min (results not shown). The branch-and-bound search using 10 random greedy starts finds 13 different assignments in a total of 2 h. In each of these 13, all 69 GSs were required to be used, and they differ from the manual assignment mostly by assigning one or two GSs to residues near the C-terminus, which were unassigned in the manual assignment.

Even in this example of a small protein with high quality data, the clique algorithm shows how enormous the task is to examine all assignments. The 69 GSs and the 74 assignable residues correspond to 792 vertices in the assignment graph, which is much less than $69 \times 74 = 5{,}106$ because the GSs are on average compatible

with only 11 sequence positions. There are only 43356 pairwise exclusions between assignment vertices, so there are 583908 edges. Even confining the Bron and Kerbosch combinatorial search to the largest cliques, some 6 million cliques can be found in 2 h, corresponding to assignments of 53–64 GSs. Much more time would be required to search the whole tree of possibilities, including the 69 vertex cliques.

To test the performance of SAGA on a larger protein, we chose *E. Coli* Chaperone LolA, with 186 residues, as deposited in the BioMagResBank (Nakada et al. 2007). We constructed complete Cα, Cβ and CO connectivities for all 163 assignments tabulated. Figure 3 shows that SAGA's greedy algorithm achieves an assignment in 10 min, completely in agreement with literature data.

Commonly, proteins contain areas that can exist in multiple conformations in solution. Depending on the timescale of the conformational exchange dynamics, all cross peaks belonging to the affected NH root will broaden beyond detection when the process is on the microsecond timescale. Typically, several residues in sequence or area will be affected as a group. We simulated this behavior for LolA by deleting the first 50 assignments from the input. As Fig. 4 shows, SAGA's greedy algorithm still performs well. It is especially pleasing to see that the "vacuum" of the first 50 unassigned residues did not "attract" many GSs from the proper areas. That was achieved, as explained above, by emphasizing assignments of several GSs is a row.

If the dynamical process in a certain area of a protein is very slow, the NMR spectra will show multiple cross peaks for that area. Again, experience indicates that several residues in sequence or area will be affected as a group. We simulated this behavior for LolA by giving duplicate GSs for the sequence 57–72 (shifted as a group in all dimensions by 0.2 ppm). As Fig. 5 shows, SAGA's greedy algorithm still performs well overall, and correctly gives two possible assignments for most of the sequence 57–72.

Our group has worked with proteins that after assignment were found to have areas in two-site slow exchange, with cross peaks missing for other areas due to

M	1		
A	2		
D	3		
A	4		
A	5	A5	1
S	6	S6	1
D	7	D7	1
L	8	L8	1
K	9	K9	1
S	10	S10	1
R	11	R11	1
L	12	L12	1
D	13	D13	1
K	14	K14	1
V	15	V15	1
S	16		
S	17	S17	1
F	18	F18	1
H	19	H19	1
A	20	A20	1
S	21	S21	1
F	22	F22	1
T	23	T23	1
Q	24	Q24	1
K	25	K25	1
V	26	V26	1
T	27	T27	1
D	28	D28	1
G	29	G29	1
S	30	S30	1
G	31	G31	1
A	32	A32	1
A	33	A33	1
V	34	V34	1
Q	35	Q35	1
E	36	E36	1
G	37	G37	1
Q	38	Q38	1
G	39	G39	1
D	40		

L	41		
W	42	W42	1
V	43	V43	1
K	44	K44	1
R	45	R45	1
P	46		
N	47	N47	1
L	48	L48	1
F	49	F49	1
N	50	N50	1
W	51	W51	1
H	52	H52	1
M	53	M53	1
T	54	T54	1
Q	55	Q55	1
P	56		
D	57	D57	1
E	58	E58	1
S	59	S59	1
I	60	I60	1
L	61	L61	1
V	62	V62	1
S	63	S63	1
D	64	D64	1
G	65	G65	1
K	66	K66	1
T	67	T67	1
L	68	L68	1
W	69	W69	1
F	70	F70	1
Y	71	Y71	1
N	72	N72	1
P	73		
F	74	F74	1
V	75	V75	1
E	76	E76	1
Q	77	Q77	1
A	78	A78	1
T	79		
A	80	A80	1

T	81	T81	1
W	82	W82	1
L	83	L83	1
K	84	K84	1
D	85	D85	1
A	86	A86	1
T	87	T87	1
G	88	G88	1
N	89		
T	90		
P	91		
F	92	F92	1
M	93	M93	1
L	94	L94	1
I	95	I95	1
A	96	A96	1
R	97	R97	1
N	98	N98	1
Q	99	Q99	1
S	100		
S	101		
D	102	D102	1
W	103	W103	1
Q	104	Q104	1
Q	105	Q105	1
Y	106	Y106	1
N	107	N107	1
I	108	I108	1
K	109	K109	1
Q	110	Q110	1
N	111	N111	1
G	112		
D	113	D113	1
D	114	D114	1
F	115	F115	1
V	116	V116	1
L	117	L117	1
T	118	T118	1
P	119		
K	120	K120	1

A	121	A121	1
S	122		
N	123	N123	1
N	124		
N	125		
L	126	L126	1
K	127	K127	1
Q	128	Q128	1
F	129	F129	1
T	130	T130	1
I	131	I131	1
N	132	N132	1
V	133	V133	1
G	134	G134	1
R	135	R135	1
D	136	D136	1
G	137	G137	1
T	138	T138	1
I	139	I139	1
H	140	H140	1
Q	141	Q141	1
F	142	F142	1
S	143	S143	1
A	144	A144	1
V	145	V145	1
E	146	E146	1
Q	147	Q147	1
D	148	D148	1
D	149	D149	1
Q	150	Q150	1
R	151	R151	1
S	152	S152	1
S	153	S153	1
Y	154	Y154	1
Q	155	Q155	1
L	156	L156	1
K	157	K157	1
S	158	S158	1
Q	159	Q159	1
Q	160	Q160	1

N	161		
G	162		
A	163		
V	164	V164	1
D	165	D165	1
A	166	A166	1
A	167	A167	1
K	168	K168	1
F	169	F169	1
T	170	T170	1
F	171		
T	172		
P	173		
P	174		
Q	175	Q175	1
G	176	G176	1
V	177	V177	1
T	178	T178	1
V	179	V179	1
D	180	D180	1
D	181	D181	1
Q	182	Q182	1
R	183	R183	1
K	184	K184	1
L	185	L185	1
E	186	E186	1

Fig. 3 10 min greedy assignment of 186-residue *E. Coli* Chaperone LolA (BMR10078). All Cα, Cβ and CO connectivities were given for the *white fields* in the first columns. No information was given for the *black fields*. *White fields* in the second columns show SAGA's assignments corresponding to the literature assignment. The numbers in these fields show the fraction of those assignments

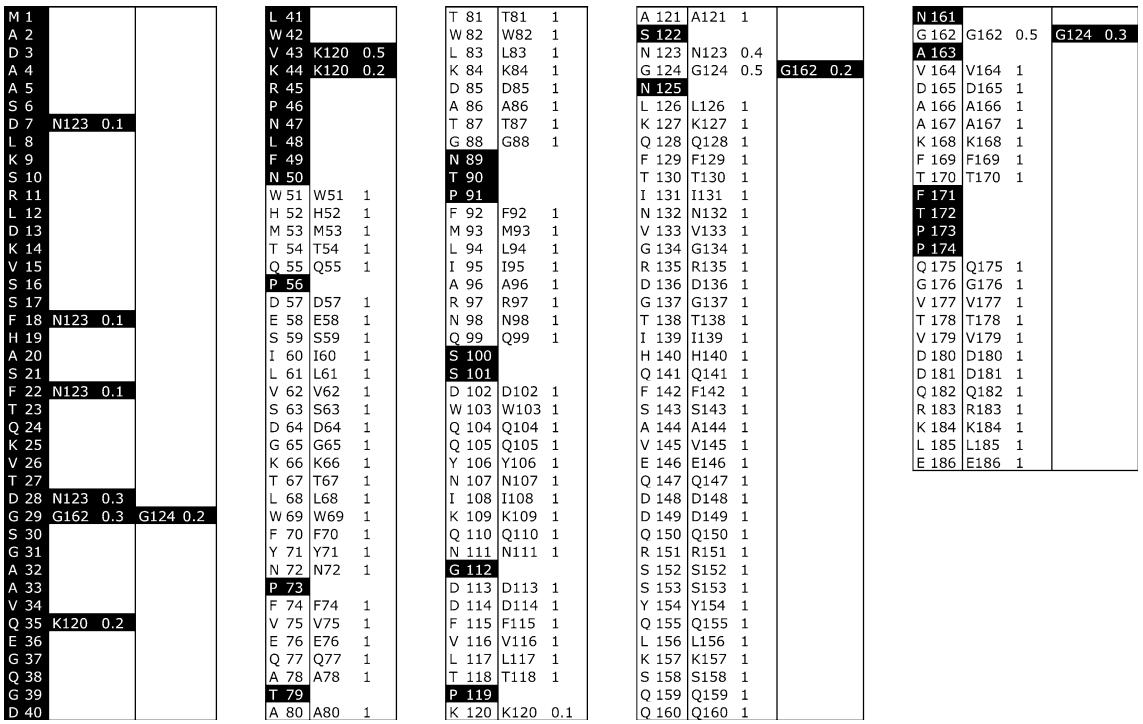

Fig. 4 10 min greedy assignment of 186-residue *E. Coli* Chaperone LolA (BMR 10078). All Cα, Cβ and CO connectivities were given for the *white fields* in the first columns. No information was given for the *black fields*. *White fields* in the second columns show SAGA's assignments corresponding to the literature assignment. *Black fields* in the second or third columns show alternative assignments. The numbers in these fields show the fraction of those assignments

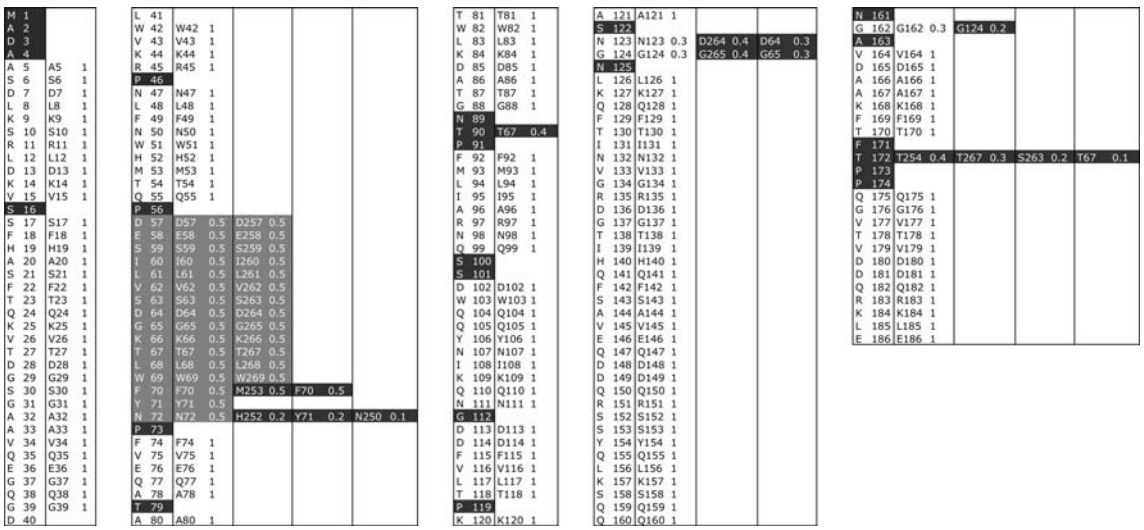

Fig. 5 10 min greedy assignment of 186-residue *E. Coli* Chaperone LolA (BMR 10078). All Cα, Cβ and CO connectivities were given as described in the legend to Fig. 3. Duplicate GSs were given for the *grey area*, residues 57–72 (shifted as a group in all dimensions by 0.2 ppm and labeled with numbers 257–272)

intermediate exchange. We simulated this "devilish" situation and show SAGA's performance in Fig. 6. While not as clean as in Fig. 4, the assignment is still completely credible. To show the degeneracy in the assignment, we have represented the sequence with the six NMR-distinguishable residue types (See Fig. S2).

Size limits

Having established that the algorithm performs very well in realistic situations with missing and/or duplicated data, we tested the size limits of the program. As a large example, consider maleate synthase G of *E. Coli*, which contains 723

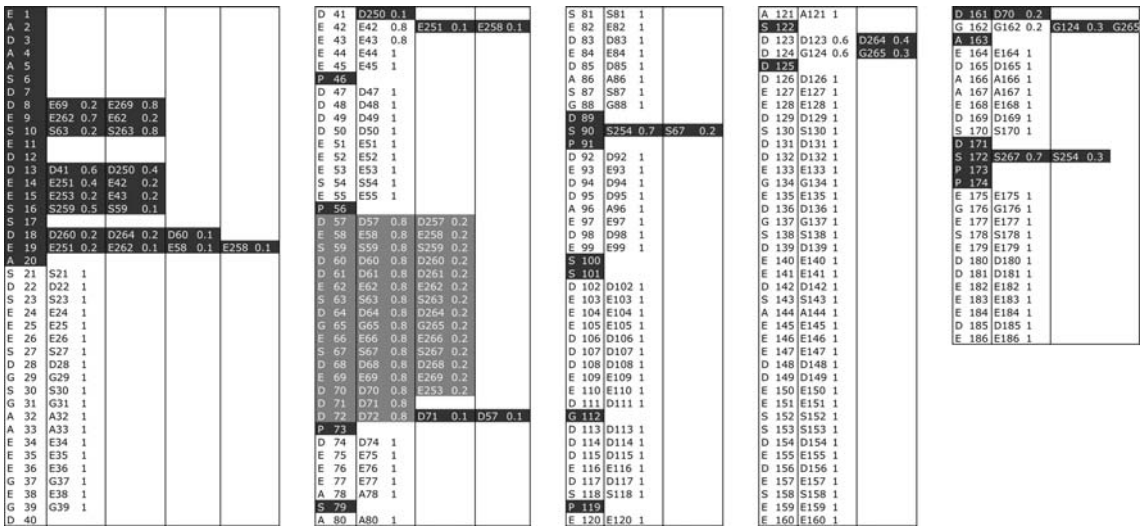

Fig. 6 30 min greedy assignment of 186-residue *E. Coli* Chaperone LolA (BMR 10078). All Cα, Cβ and CO connectivities were given as described in the legend to Fig. 3. Duplicate GSs were given for the *grey area*, residues 57–72 (shifted as a group in all dimensions by 0.2 ppm and labeled with numbers 257–272). The amino acid sequence is represented as an "NMR" sequence, in which indistinguishable residue types have been collected (A = A; G = G; P = P; S+T = S; D+F+I+L+N+Y = D; E+C+H+K+M+Q+R+V+W = E. See Fig. S2)

residues. The original spectra were assigned using a combination of 4D and 3D NMR methods (Tugarinov et al. 2002), and the assignment list was deposited as BMR 5471. Of the 723 residues, 692 are possibly assignable, and an artificially complete set of three Sparky files based on the BMRB entry produced 654 GSs, 615 of which have 3-rung connectivities, 53 have two-rung connectivities and 5 have 1-rung connectivities.

Using the greedy algorithm, we find in 60 min virtually complete assignments that for the most part agree with the literature assignments (see Fig. 7). Interestingly, the program generated also several stretches of alternative assignments that comprise more than several residues in a row and are hence worth considering. This illustrates the advantage of the program: it will point out areas that are assigned with confidence, but also areas for which additional experiments need to be carried out (e.g. NOESY connectivities or residue-specific labeling), or which should not be used in subsequent work.

Using SAGA to test and extend partial assignments in large proteins

We used SAGA to assess the viability of partial hand assignments of large proteins, using original Sparky peak pick files. As explained in methods, these files were hand-curated to remove noise and sidechain peaks, to add an occasional missed cross peak, and were synchronized between the six triple resonance spectra. That is, HCN cross peaks at the same (H,N) coordinates were given the same root name on authority of the expert spectroscopist.

In the case of a 501-residue construct of the Hsp70 chaperone DnaK from *Thermos Thermophilus,* (called TTH-501 hereafter), there are 474 assignable residues and 405 GSs. The hand assignments were based on 351 GSs with Cα-matches, 88 GSs with Cβ-matches and 275 GSs with CO-matches. These combined into 82 GSs with 3-rung connectivities, 194 GSs with 2-rung connectivities, and 80 GSs with single-rung connectivities. A 10 h greedy run produced 30 assignments with probabilities between 1 and 0.2. The complete results are shown in Fig. 8, which shows that most of the hand assignments are indeed "found" by the program. However, there are many more valid assignments, and the question arises how one would go about picking the "correct" ones if one does not know the answer. We proceeded as follows. (1) We keep all assignments of probability 0.6 or higher. (2) We discard all assignments with probability less than 0.2. (3) Of the remaining set, we only retain those GSs that are assigned in stretches of 3 or more residues. (4) The results are checked for mutual compatibility. (5) If the same stretch of linked GSs occurs more than once, we retain the one that occurs in a longer overall stretch. The results of this editing are shown in Fig. 9. The retained assignments are very compatible with the hand assignment, but several other credible assignments are found as well.

SAGA clearly discloses the risk of obtaining partial assignments without confirming them with further scrutiny, such as ^{13}C lineshape comparisons in the different spectra, independent assignment experiments at different temperature or pH, extra information from selective labeling, 4D methods (Tugarinov et al. 2002), NOE connectivities and

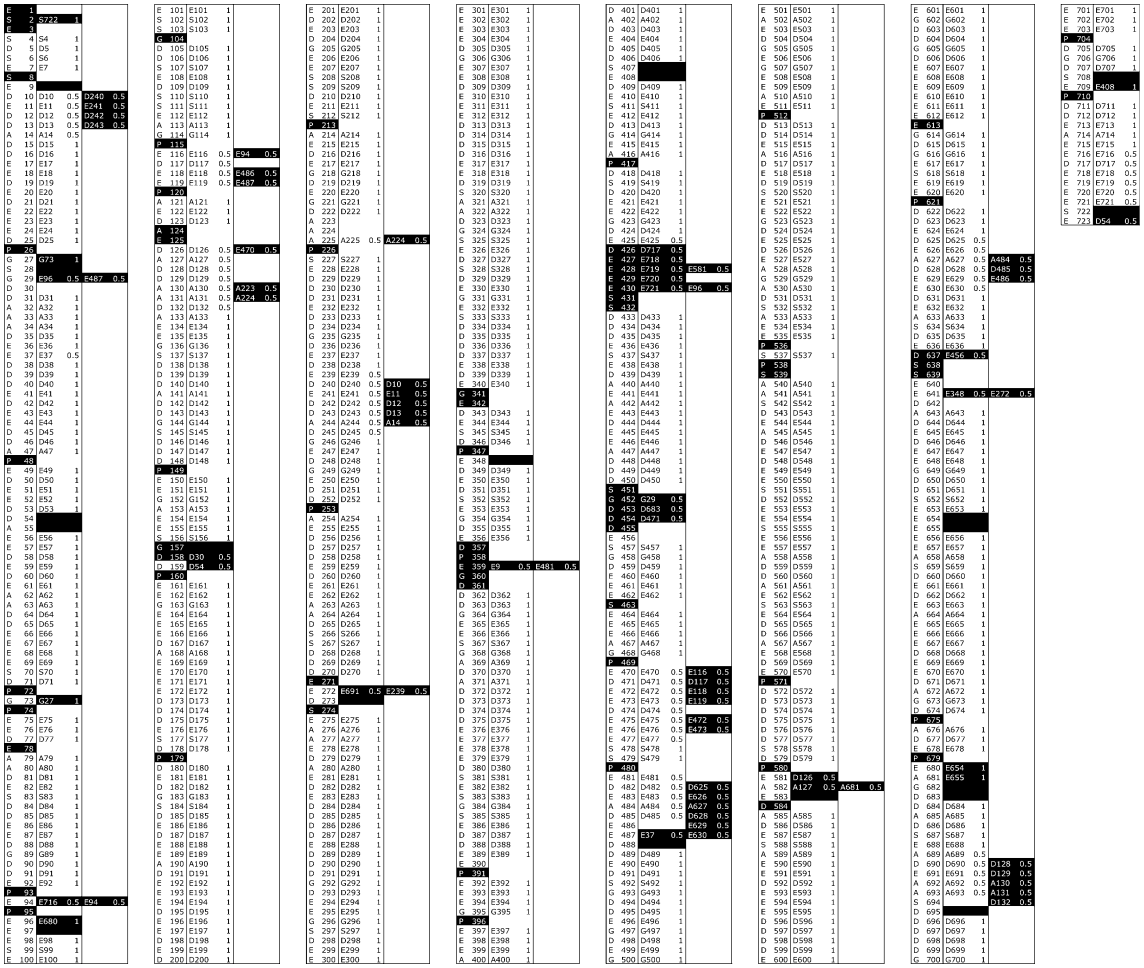

Fig. 7 60 min greedy assignment of Maleate Synthase G of *E. Coli*, which contains 723 residues. The original spectra were assigned using a combination of 4D and 3D NMR methods (Tugarinov et al. 2002) and the assignment was deposited as BMR 5471. Of the 723 residues, 692 are possibly assignable, and an artificially complete set of three Sparky files based on the BMRB entry produced 654 GSs, 615 of which have 3 rung connectivities, 53 have two-rung connectivities and 5 have 1-rung connectivities. GSs with connectivities were given for the *white fields* in the first columns. No information was given for the *black fields*. *White fields* in the second columns show SAGA's assignments corresponding to the literature assignment. *Black fields* in the second or third columns show alternative assignments. The numbers in these fields show the fraction of those assignments

HNCAHN experiments (Frueh et al. 2009). For TTH-501, the hand assignments were verified from triple resonance experiments at different pH, from assignments from a subdomain (Revington and Zuiderweg 2004), and with NOE experiments before being used in structural studies (Revington et al. 2005).

In the case of the 386-residue nucleotide binding domain of the chaperone Hsc70, we have a hand assignment of just 206 GSs for 374 assignable residues. The hand assignments were based on 147 GSs with Cα-matches, 130 GSs with Cβ-matches and 134 GSs with CO-matches. These combined into 110 GSs with 3-rung connectivities, 30 GSs with 2-rung connectivities, and 21 GSs with single-rung connectivities. SAGA produced 162 assignments in 2 h, 88 of which had probabilities of 0.5 or larger. Sorting these by the largest number of mutually compatible assignments following the rules described above for TTH-501, we obtained the result as shown in Fig. 10. It shows that the great majority of the hand assignments are reproduced. Several are not, but that should be no surprise, since the sequence is quite degenerate, which is obvious when written as a sequence of NMR-distinguishable residues as is shown in Fig. S2. In addition, when rungs are incomplete, the degeneracy is even higher, accounting for some of the apparent type-violations in Fig. 10. Nevertheless, we trust the hand assignments, since these were based on several complete datasets from different samples and contain ^{13}C lineshape matching information. The NMR spectra contained 101 additional GSs that were not be assigned by hand. We let SAGA place these in a 2-h greedy run after we constrained the original hand assignments to their sequence positions. We used larger matching

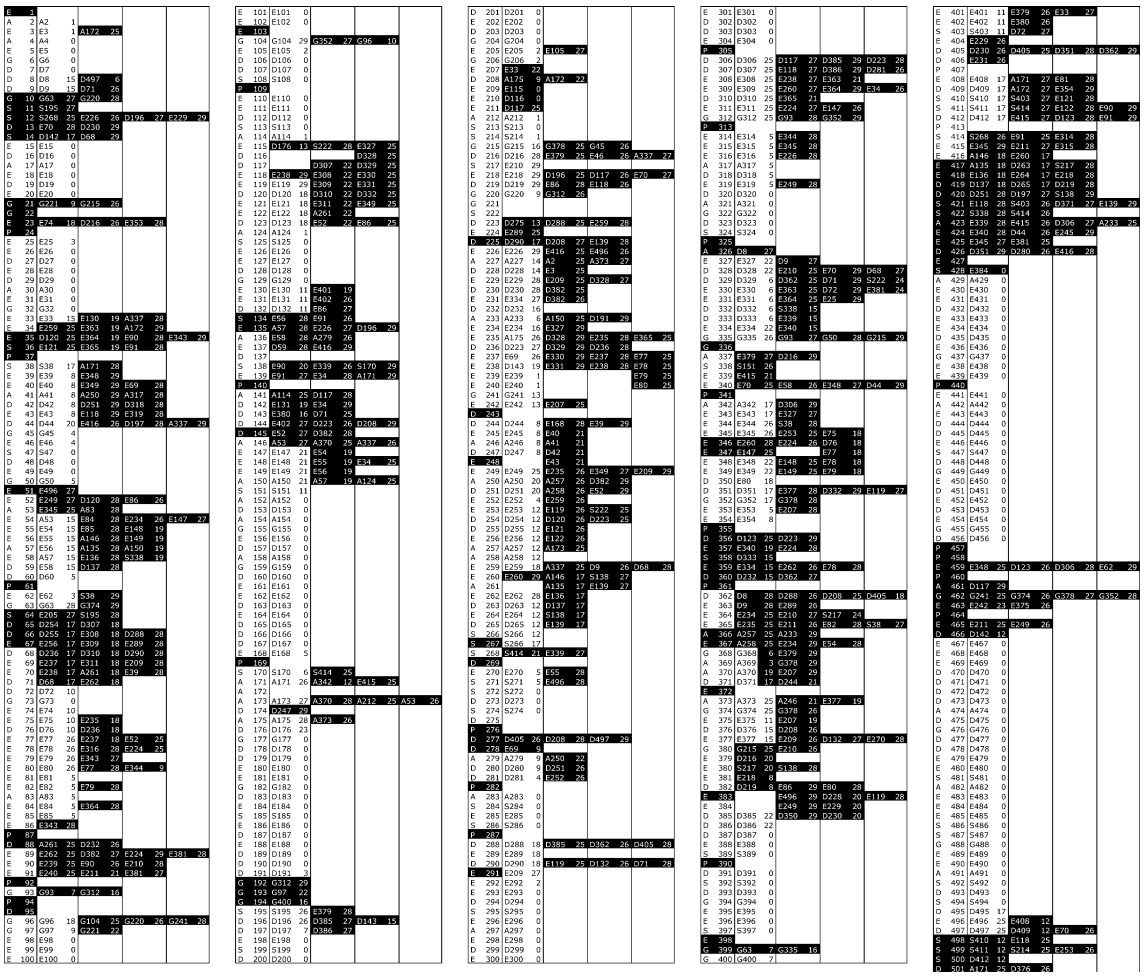

Fig. 8 10 h greedy assignment of a 501-residue construct of the Hsp70 chaperone DnaK from *Thermos Thermophilus,* with 474 assignable residues and 405 GSs. The hand assignments were based on 82 GSs with 3-rung connectivities, 194 GSs with 2-rung connectivities, and 80 GSs with single-rung connectivities. The connected GSs were given for the *white fields* in the first columns. No information was given for the *black fields*. *White fields* in the second columns show SAGA's assignments corresponding to the literature assignment. *Black fields* in the second and following columns show alternative assignments. The numbers in these fields show the rank order of those assignments, with 0 being the best. The amino acid sequence is represented as an "NMR" sequence, in which indistinguishable residue types have been collected

tolerances and larger type tolerances in this additional run. The resulting assignment is shown in Fig. 11. Of the 101 additional GSs, 46 could be placed with a confidence of 0.5 or better. Clearly, some of these could not ever have been found by "hand": Consider GS X954, which is placed with 100% confidence on V18. This GS has no Cα(i) or Cβ(i) rungs, while its CO(i) rung is unmatched, because G19 does not show any CO(i-1) rungs. SAGA, however, placed GS X954 on the basis of its (unmatched) i−1 rungs that put it next to a residue of that type.

Using the program to assess experimental assignment strategies

Having established the reliability of the program, we assess here the inverse question: how complete do NMR data sets need to be in order to be uniquely assignable? As is shown in Figures S3 and S4, there is no real need for CO connectivities, even for assigning the resonances of a 381-residue protein. This substantiates the assertion in the introduction that, theoretically, two sets of complete rung connectivities are sufficient to uniquely sequentially link the GSs of proteins, maybe even larger than 1,000 residues. The loss of the CO-rungs does not affect the placement of the GSs on the sequence, as the CO chemical shifts have almost no predictive value for residue type (see Fig. S2).

However, as Fig. S5 shows, one cannot rely on just complete Cα and CO for an assignment of a 20 kDa protein or larger. The reason, of course, is that the Cα resonance statistics are not sufficiently restrictive to allow placement of the GSs on the sequence (see Fig. S2). Figure S5 also shows how well SAGA performs: it finds many feasible

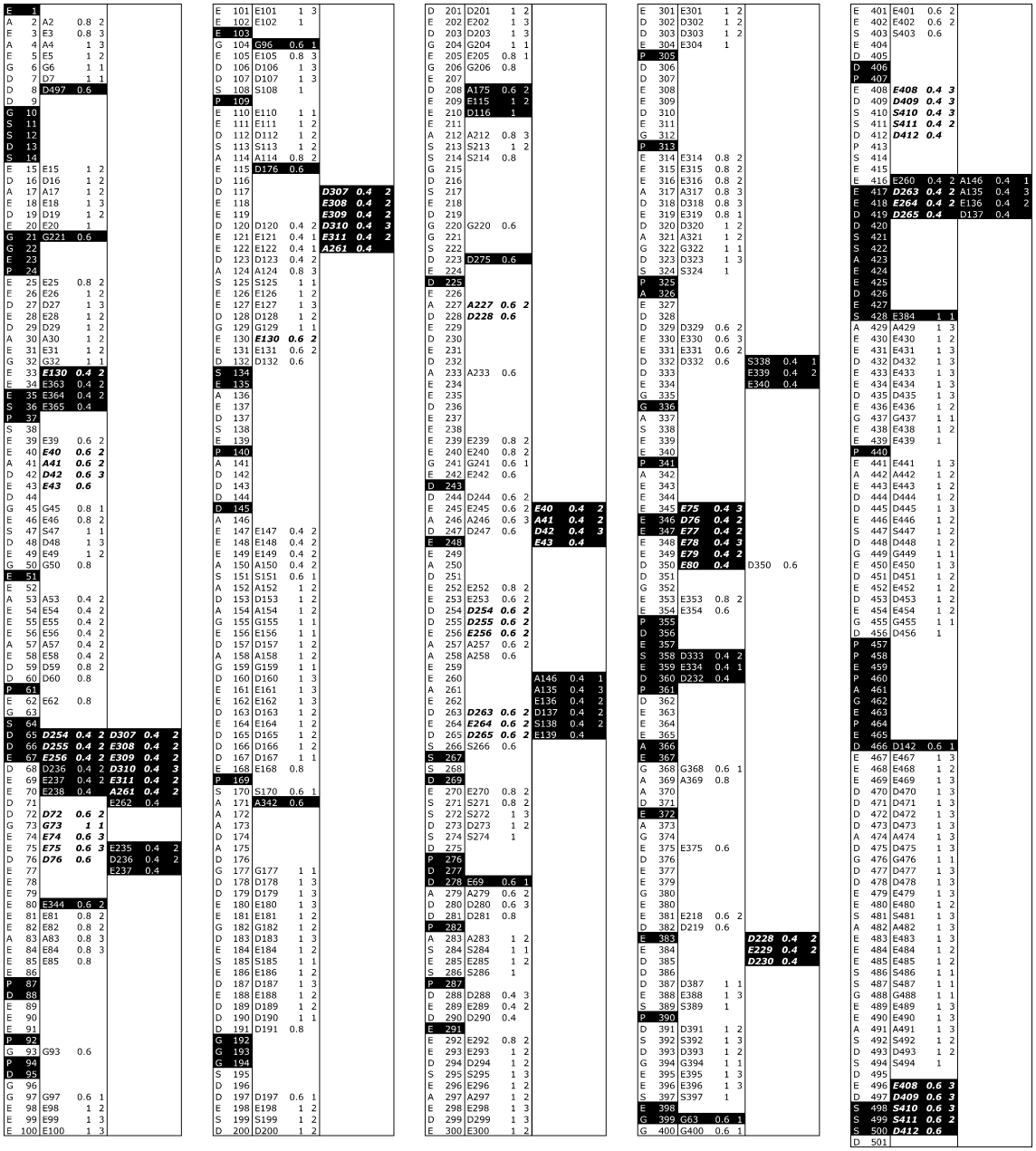

Fig. 9 10 h greedy assignment of a 501-residue construct of the Hsp70 chaperone DnaK from *Thermos Thermophilus,* as in Fig. 8, but edited as described in the text. The connected GSs were given for the *white fields* in the first columns. No information was given for the *black fields*. *White fields* in the second columns show SAGA's assignments corresponding to the literature assignment. *Black fields* in the second or third columns show alternative assignments. The middle numbers in the these fields show probabilities of the assignments. The last number gives the number of rung connectivities between the GSs. The amino acid sequence is represented as an "NMR" sequence. Stretches of GSs that occur more than once in this assignment are indicated in bold italic

assignments, as it should, and in such a case could be used to design experiments to resolve the ambiguities.

Figures S6 and S7 show that it is not the Cβ connectivities, but the Cβ resonance frequencies that are needed to place the GSs on the sequence. Even for a large protein of 381 residues, the assignments are still virtually complete without Cβ(i-1) information. This is encouraging: the Cβ(i-1) connectivities are the most difficult to obtain for any protein for sensitivity reasons, and are the first ones to be incomplete for large proteins (see above for TTH-501).

Possible future extensions of the program

The NMR spectroscopist has also available cross peak intensity and ^{13}C resonance line width information. Both Sparky and NMRPipe provide this information in the peak

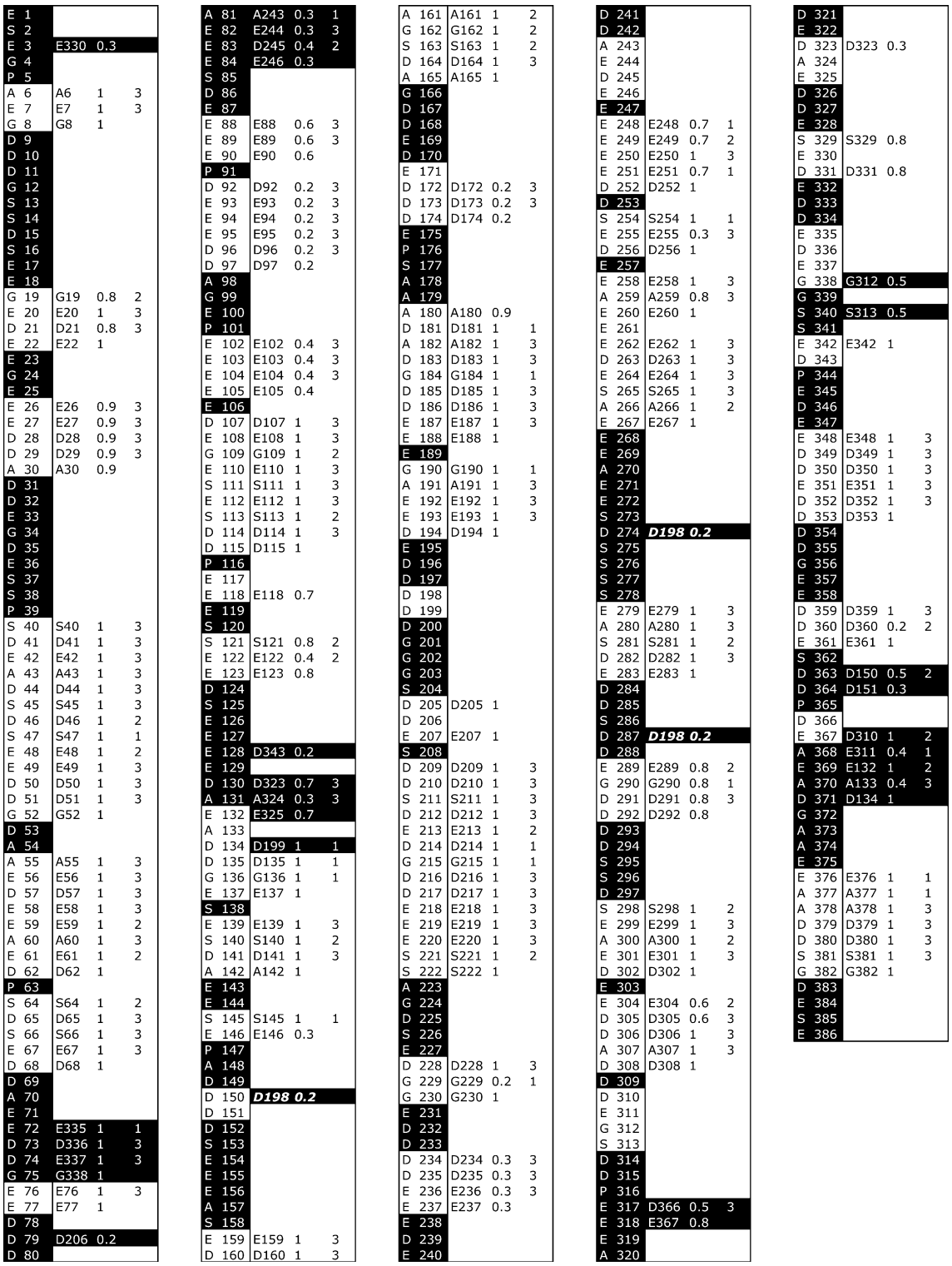

Fig. 10 120 min greedy check of an unpublished, partial hand assignment of the 386-residue nucleotide binding domain of human Hsc70 (206 GSs/374 assignable residues). The hand assignments are based on 110 GSs with 3-rung connectivities, 30 GSs with 2-rung connectivities, and 21 GSs with single-rung connectivities (they are given in the *white fields* in the left columns). Tolerances used: Cα, Cβ and CO range, 2.5, 3 and 3 sigma, respectively; Cα, Cβ and CO match 0.1, 0.2 and 0.1 ppm respectively. *White fields* in the second columns show SAGA's assignments corresponding to the literature assignment. *Black fields* in the second or third columns show alternative assignments. The middle numbers in these fields show the fraction of those assignments. The last number gives the number of rung connectivities between the GSs. The amino acid sequence is represented as an "NMR" sequence. Duplicate assignments have been removed on the basis of frequency and/or number of connecting rungs, except for D198 (*italic*)

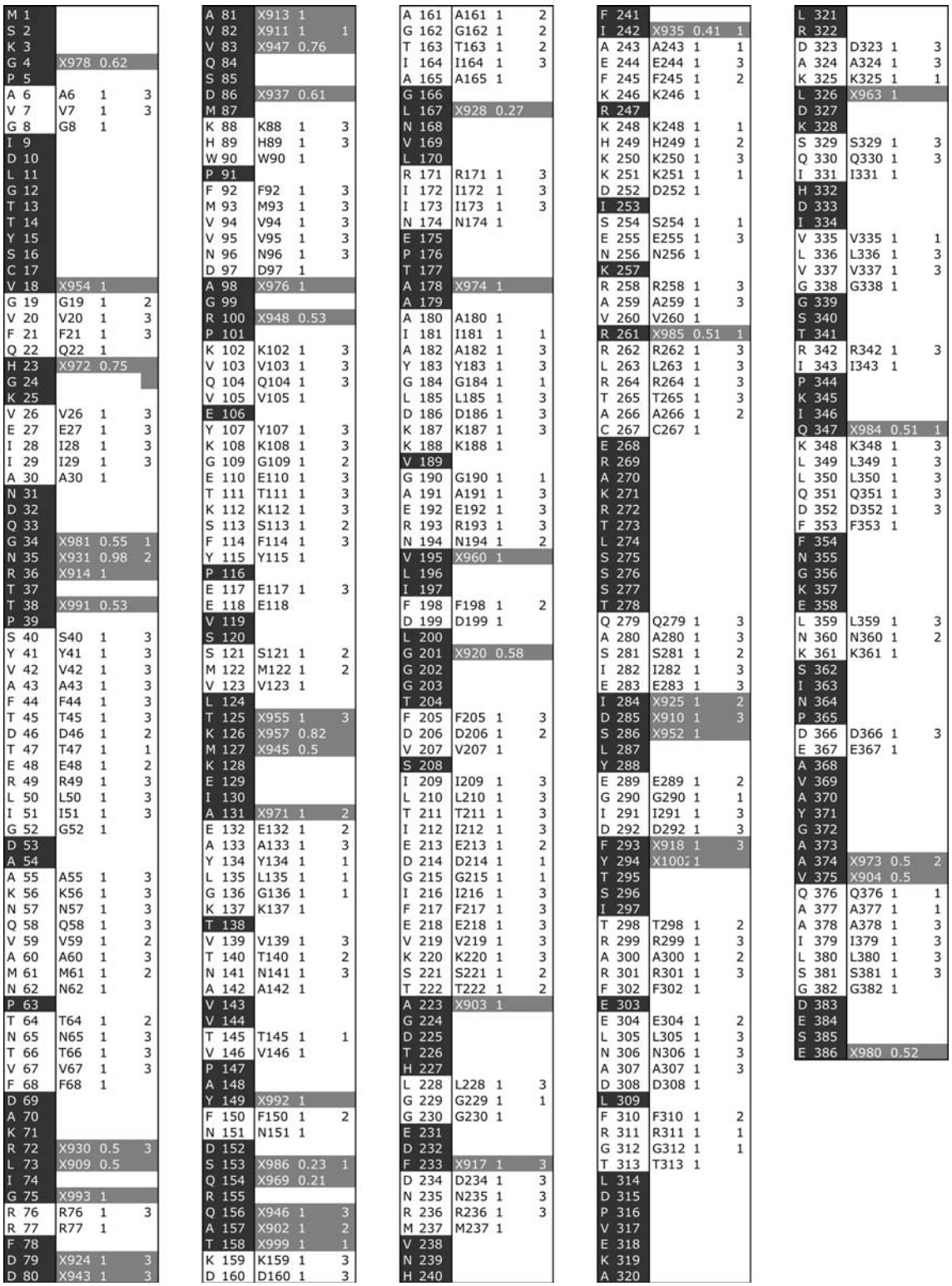

Fig. 11 Improvement of the assignment of Hsc70 NBD (386 residues). The 206 hand assigned GSs were constrained to their positions, and an additional 101 GSs present in the spectra (mostly with incomplete rungs) were allowed to fill in the open stretches using default tolerances (see Methods). 46 of these were placed with high confidence. They are shown in the *grey fields*. The middle numbers in the fields show the fraction of those assignments. The last number gives the number of rung connectivities between the GSs. Duplicate assignments have been removed on the basis of frequency and/or number of connecting rungs. The final result is the assignment of 249 out of the 374 assignable residues, based on 120 GSs with 3-rung connectivities, 37 GSs with 2-rung connectivities, and 28 GSs with single-rung connectivities

pick tables. Currently, the Sparky input mode of SAGA already accepts intensity data. In future enhancements of the program, we envision using this information to weigh against assignments that place GSs of very different intensities next to each other. This is to encapsulate the common knowledge that intensity differences in (a single class of) triple resonance spectra are due to dynamical processes, which, as argued before, often occur in stretches rather than at isolated residues. ^{13}C-line width differences are also mostly determined by dynamics, and can potentially distinguish between correct and wrong GS linkages even when the chemical shifts are identical. Elegant computer-assisted hand-assignment programs such as XEASY have long incorporated an easy visualization of this parameter (Bartels et al. 1995). Coding it into SAGA will be achieved in future versions, and will likely improve the reliability of GS linkages where only few rungs are available.

Conclusions

SAGA is a versatile program for automatic sequential assignment that can handle not only small proteins with high quality data, but even the largest feasible proteins with realistic, flawed data. No single search algorithm is optimal for all datasets, so the branch-and-bound search gives a very thorough search on rather unambiguous data, while the greedy search produces very useful results on large proteins with lower quality data. In real applications, many different assignments satisfy reasonable acceptance criteria, so SAGA summarizes them all, highlighting consistently assigned residues, seldom assigned residues, and different alternative assignments for parts of the polypeptide chain.

Acknowledgments E.R.P.Z. acknowledges support from NIH grants GM063027 and GM063027-08S1 (E.R.P.Z., PI) and NS059690 (J.E. Gestwicki, PI). We thank Drs. A.V. Kurochkin and D.S. Weaver for the preparation of the Hsc70 NMR samples.

References

Atreya HS, Szyperski T (2004) G-matrix fourier transform NMR spectroscopy for complete protein resonance assignment. Proc Natl Acad Sci USA 101:9642–9647

Baldus M (2007) ICMRBS founder's medal 2006: biological solid-state NMR, methods and applications. J Biomol NMR 39:73–86

Bartels C, Xia T, Billeter M, Güntert P, Wüthrich K (1995) The program XEASY for computer-supported NMR spectral-analysis of biological macromolecules. J Biomol NMR 6:1–10

Benod C, Delsuc M-A, Pons J-L (2006) CRAACK: consensus program for NMR amino acid type assignment. J Chem Inf Model 46:1517–1522

Billeter M, Wagner G, Wüthrich K (2008) Solution NMR structure determination of proteins revisited. J Biomol NMR 42(3):155–158

Bron C, Kerbosch J (1973) Finding all cliques of an undirected graph. Comm ACM 16:575–577

Cavanagh J, Fairbrother WJ, Palmer AG, Rance M, Skelton NJ (2007) Protein NMR spectroscopy, 2nd edn. Academic Press, Amsterdam

Delaglio F, Grzesiek S, Vuister GW, Zhu G, Pfeifer J, Bax A (1995) NMRPipe: a multidimensional spectral processing system based on UNIX pipes. J Biomol NMR 6:277–293

Eghbalnia HR, Bahrami A, Wang L, Assadid A, Markley JL (2005) Probabilistic identification of spin systems and their assignments including coil–helix inference as output (PISTACHIO). J Biomol NMR 32:219–233

Friedrichs MS, Mueller L, Wittekind M (1994) An automated procedure for the assignment of protein ^{1}HN, ^{15}N, $^{13}C^{\alpha}$, $^{1}H^{\alpha}$, $^{13}C^{\beta}$ and $^{1}H^{\beta}$ resonances. J Biomol NMR 4:703–726

Frueh DP, Arthanari H, Koglin A, Walsh CT, Wagner G (2009) A double TROSY hNCAnH experiment for efficient assignment of large and challenging proteins. J Am Chem Soc 131:12880–12881

Goddard TD, Kneller DG (2003) Sparky–NMR assignment and integration software. University of California, San Francisco

Güntert P (2009) Automated structure determination from NMR spectra. Eur Biophys J 38(2):129–143

Hare BJ, Prestegard JH (1994) Application of neural networks to automated assignment of NMR spectra of proteins. J Biomol NMR 4:35–46

Hopcroft JE, Karp RM (1973) An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. SIAM J Comput 2:225–231

Hyberts SG, Wagner G (2003) IBIS–A tool for automated sequential assignment of protein spectra from triple resonance experiments. J Biomol NMR 26:335–344

Kao M-Y, Lam T-W, Sung W–K, Ting H–F (2001) A decomposition theorem for maximum weight bipartite matchings. SIAM J Comput 31:18–26

Kuhl FS, Crippen GM, Friesen DK (1984) A combinatorial algorithm for calculating ligand binding. J Comput Chem 5:24–34

Kuhn HW (1955) Variants of the hungarian method for assignment problems. Naval Research Logistics Quarterly 2:83–97

Lemak A, Steren CA, Arrowsmith CH, Llinás M (2008) Sequence specific resonance assignment via multicanonical Monte Carlo search using an ABACUS approach. J Biomol NMR 41:29–41

Leutner M, Gschwind RM, Liermann J, Schwarz C, Gemmecker G, Kessler H (1998) Automated backbone assignment of labeled proteins using the threshold accepting algorithm. J Biomol NMR 11:31–43

Lin G, Wan X, Tegos T, Li Y (2006) Statistical evaluation of NMR backbone resonance assignment. Int J Bioinf Res App 2:147–160

Lukin JA, Gove AP, Talukdar SN, Ho C (1997) Automated probabilistic method for assigning backbone resonances of (^{13}C, ^{15}N)-labeled proteins. J Biomol NMR 9:151–166

Meadows RP, Olejniczak ET, Fesik SW (1994) A computer-based protocol for semiautomated assignments and 3D structure determination of proteins. J Biomol NMR 4:79–96

Moseley HNB, Monleon D, Montelione GT (2001) Automatic determination of protein backbone resonance assignments from triple resonance nuclear magnetic resonance data. Meth in Enzymol 339:91–108

Nakada S, Sakakura M, Takahashi H, Tokuda H, Shimada I (2007) Backbone resonance assignment for the outer membrane lipoprotein receptor LolB from Escherichia coli. Biomol NMR Assign 1:121–123

Olson JB Jr, Markley JL (1994) Evaluation of an algorithm for the automated sequential assignment of protein backbone resonances:

A demonstration of the connectivity tracing assignment tools (CONTRAST) software package. J Biomol NMR 4:385–410

Oschkinat H, Croft D (1994) Automated assignment of multidimensional nuclear magnetic resonance spectra. Meth Enzymol 239:308–318

Parker RG, Rardin RL (1988) Discrete Optimization. Academic Press, New York

Revington M, Zuiderweg ERP (2004) TROSY-driven NMR backbone assignments of the 381-residue nucleotide-binding domain of the Thermus Thermophilus DnaK molecular chaperone. J Biomol NMR 30:113–114

Revington M, Zhang Y, Yip GN, Kurochkin AV, Zuiderweg ERP (2005) NMR investigations of allosteric processes in a two-domain Thermus thermophilus Hsp70 molecular chaperone. J Mol Biol 349:163–183

Seavey BR, Farr EA, Westler WM, Markley J (1991) A relational database for sequence-specific protein NMR data. J Biomol NMR 1:217–236

Stratmann D, Guittet E, van Heijenoort C (2010) Robust structure-based resonance assignment for functional protein studies by NMR. J Biomol NMR 46:157–173

Tugarinov V, Muhandiram R, Ayed A, Kay LE (2002) Four-dimensional NMR spectroscopy of a 723-residue protein: chemical shift assignments and secondary structure of malate synthase G. J Am Chem Soc 124:10025–10035

Vitek O, Bailey-Kellogg C, Craig B, Kuliniewicz P, Vitek J (2005) Reconsidering complete search algorithms for protein backbone NMR assignment. Bioinformatics 21:230–236

Wan X, Lin G (2007) GASA: A graph-based automated NMR backbone resonance sequential assignment program. J Bioinf Comput Biol 5:313–333

Wang J, Wang T, Zuiderweg ERP, Crippen GM (2005) CASA: An efficient automated assignment of protein mainchain NMR data using an ordered tree search algorithm. J Biomol NMR 33:261–279

Williamson M, Craven C (2009) Automated protein structure calculation from NMR data. J Biomol NMR 43:131–143

Xiong F, Pandurangan G, Bailey-Kellogg C (2008) Contact replacement for NMR resonance assignment. Bioinformatics 24:205–213

Zimmerman DE, Montelione GT (1995) Automated analysis of nuclear magnetic resonance assignments for proteins. Curr Opin Struct Biol 5:664–673

Zimmerman DE, Kulikowski CA, Montelione GT (1993) A constraint reasoning system for automating sequence-specific resonance assignments from multidimensional protein NMR spectra. Proc Int Conf Intell Syst Mol Biol 1:447–455

Zimmerman D, Kulikowski C, Wang L, Lyons B, Montelione GT (1994) Automated sequencing of amino acid spin systems in proteins using multidimensional HCC(CO)NH-TOCSY spectroscopy and constraint propagation methods from artificial intelligence. J Biomol NMR 4:241–256

Zimmerman DE, Kulikowski CA, Huang Y, Feng W, Tashiro M, Shimotakahara S, Chien CY, Powers R, Montelione GT (1997) Automated analysis of protein NMR assignments using methods from artificial intelligence. J Mol Biol 269:592–610