Methods in Enzymology

Automatic Determination of Protein Backbone Resonance Assignments From Triple Resonance NMR Data

Hunter N.B. Moseley, Daniel Monleon, and Gaetano T. Montelione*

Center for Advanced Biotechnology and Medicine,

and

Department of Molecular Biology and Biochemistry

Rutgers University

Running Title: Automatic Determination of Backbone Resonance Assignments

* Address correspondence to:
Prof. G. T. Montelione
CABM-Rutgers University
679 Hoes Lane
Piscataway, NJ 08854
Phone: (732) 235-5321
Fax: (732) 235-4850
e-mail: guy@cabm.rutgers.edu

Introduction.

Advances in molecular biophysics and genomics are providing expanded roles for NMR in structural biology. With the advent of multidimensional and triple-resonance strategies for determining resonance assignments and 3D structures[1-7], it became increasingly clear that the quality and information content of protein NMR spectra could allow largely automated, and thus faster and more routine, analyses of assignments and structures for small proteins. Over the last few years, several labs realized this potential to some degree, and many production structure analyses are now carried out using automated or semi-automated methods (for a recent review see ref 7). These advances have tremendous implications for NMR's use as a powerful and accessible tool in biophysical chemistry, drug design, and structural genomics. In this paper, we will discuss algorithms and practical considerations for using the assignment program AutoAssign [8, 9] to automate analysis of protein backbone resonance assignments from triple resonance NMR data.

Methodology of AutoAssign

AutoAssign[9] is a constraint-based expert system designed to determine backbone H^N, H^α, $^{13}C'$, $^{13}C^\alpha$, ^{15}N, and $^{13}C^\beta$ resonance assignments from peak lists derived from a set of triple resonance spectra with common H^N-^{15}N resonance correlations. The program can handle data obtained on uniformly ^{15}N-^{13}C doubly-labeled, uniformly ^{2}H-^{15}N-^{13}C triply-labeled, or partially-deuterated ^{15}N-^{13}C doubly-labeled protein samples. AutoAssign's general analysis scheme involves the following five basic steps: 1) filter peaks (filtering) and align resonances from different spectra (aligning); 2) group resonances into spin systems (grouping); 3) identify amino acid type of spin systems (typing); 4) find and link sequential spin systems into segments (linking); and 5) map spin system segments onto the primary sequence (mapping). Most automation methods use this same general

analysis scheme which originates from the classical strategy developed by Wüthrich and co-workers[10-12]. Across different implementations; the weakest of these key steps dictates overall robustness. Thus reliable performance of each step is necessary for good overall performance.

AutoAssign implements the general analysis scheme using a combination of artificial intelligence techniques and statistical methods. Each step has a tailor-made solution implemented with techniques amenable for solving that type of problem. Also, information is shared between steps for better overall performance. In Step one, AutoAssign performs most of its filtering (i.e. distinguishing real peaks from noise and other artifacts) using the amide ^{15}N-H^N projection common to all of the peak lists. For aligning the ^{1}H,^{15}N, and ^{13}C dimensions, AutoAssign first organizes the peak lists into a hierarchy representing nearest-neighbor set inclusion (superset/subset) relationships between peak lists. For example, intra and sequential $^{13}C^{\alpha}$'s and $^{13}C^{\beta}$'s of the CBCANH experiment are superset to intra and sequential $^{13}C^{\alpha}$'s of the CANH experiment, which are superset to sequential $^{13}C^{\alpha}$'s of the CA(CO)NH experiment. It then finds well-resolved peaks in the shared ^{15}N-H^N projections common to all of these triple resonance experiments. It uses these isolated peaks to calculate pair-wise alignments between peak lists with the hierarchical relationships described above. AutoAssign then uses (shares) this alignment information along with peak intensity information in later steps.

In Step two, AutoAssign creates a generic spin system object (GS) for each unique HSQC or HNCO peak. It then groups peaks from other peak lists having the same amide resonance frequencies (^{15}N-H^N root) into these GS's. Next, the program separates the resonance frequency data of each GS into a "CA-ladder" and a "CO-ladder" representing resonance frequencies of nuclei that are part of the same or previous amino acid residue, respectively, as the ^{15}N-H^N root (Figure 1). AutoAssign performs this separation based upon the set inclusion hierarchy established in the first step and the relative intensities of the peaks involved. It then identifies $^{13}C^{\alpha}$, $^{13}C^{\beta}$, $^{13}C'$ and H^{α}

resonance rungs in each ladder based upon their inclusion and exclusion in peak lists derived from specific types of triple resonance experiments and their resonance frequencies. Next, the program ranks and sorts GS's based upon their completeness and the intensity of the peaks comprising these resonances. Finally, AutoAssign divides the GS's into separate lists based upon the presence of “overlapped” GS's defined as those GS’s with similar or degenerate ^{15}N-H^N roots and “weak” extra GS's due to conformational and/or chemical heterogeneity. In later steps, AutoAssign uses different reasoning methods for each group of GS's (ranked as "distinct", "overlapped", and "weak", in that order), allowing for special treatment of overlapped and weak GS's[9].

In Step three, AutoAssign creates “possible amino acid type” lists for each CA- and CO-ladder in each GS using Bayesian posterior probabilities based upon $^{13}C^{\alpha}$ and $^{13}C^{\beta}$ chemical shifts[9]. Analysis of the BioMagResBank[13], which provides statistics on $^{13}C^{\alpha}$ and $^{13}C^{\beta}$ chemical shifts for each type of amino acid, makes such a Bayesian statistical method feasible. Furthermore, AutoAssign restricts the “possible amino acid type” lists based upon “phase” information (if available) characteristic of local spin system topologies. In this context, "phase" of the data refers to the sign of the peak intensities resulting from nuclear spin interactions during properly tuned coherence transfer periods[6, 14-17]. With “possible amino acid type” lists for each of its CA- and CO-ladders, each GS is restricted to a list of dipeptide sequence-specific sites (SS) in the protein sequence which are compatible with this typing information (Figure 2A). AutoAssign uses this information to define possible mappings of GS’s to SS’s and SS’s to GS’s. These “possible assignment” lists form the basis for implementation of a constraint propagation network (CPN)[8] which can infer all logically entailed assignments. The “possible amino acid type” lists represent the first set of constraints that are propagated through this CPN.

In Step four, AutoAssign compares all the CA- and CO-ladders from different GS's to create a list of possible nearest neighbor links. This comparison is a match

scoring function that returns a value based on how many and how well the resonance rungs between the two ladders match. The return value represents a Euclidean distance between the two sets of rungs, with unmatched rungs representing a categorical penalty. AutoAssign then sorts the list of possible nearest neighbor links first with respect to the number of matching rungs (rung category) and then to the match value. In this sorted order, the program starts "establishing links" between GS's that have a unique 1 to 1 match at the currrent and higher rung categories, creating segments of linked GS's. This represents a best-first search algorithm for linking that also satisfies a 1 to 1 uniqueness criterion. With each instantiated link, a new nearest neighbor consistency constraint propagates through the CPN, further limiting the "possible assignment" lists (Figure 2).

In Step five, AutoAssign uses another 1 to 1 uniqueness criterion for assigning GS's to SS's, and visa versa. When both a GS and an SS have the same single possible mapping to one another, AutoAssign assigns this GS <-> SS mapping. This conservative approach to mapping prevents errors by delaying the mapping decision until deductive reasoning from both the GS and the SS perspective can be applied. Also, as a natural consequence of the applied set logic, when AutoAssign establishes a unique GS <-> SS mapping for one GS in a segment, it consequently identifies mappings for the entire segment. Furthermore, AutoAssign interleaves Steps four and five such that, as the program establishes links, it immediately establishes mappings when the GS <-> SS uniqueness criterion is met. Due to the presence of "overlapped" GS's with similar ^{15}N-H^N roots and "extra" GS's arising from multiple conformations, AutoAssign performs three separate interleaved iterations of Steps four and five, first using only distinct GS's, then using distinct plus overlapped GS's, and finally using distinct, overlapped, plus extra GS's.

Description of Input Data for AutoAssign

AutoAssign uses up to nine different types of peak lists representing data obtained

from eight different types of triple resonance experiments and ^{15}N-H^N HSQC spectrum. These peak lists represent information from the following types of experiments: HSQC, HNCO, HNCACB HN(CO)CACB, HNCA, HN(CO)CA, HN(CA)CO, HNHA, and HN(CO)HA (for detailed description of these pulse sequences and practical aspects of their implementation, see ref 6). Figure 3 schematically defines these data sets in terms of the bonds and nuclei involved. Although the program handles data from as many as nine experiments, AutoAssign requires only the HSQC, HNCO, HNCA, HN(CO)CA, HNCACB, and HN(CO)CACB peak lists. However, using all nine types of data obtains the best performance. These peak lists come in the form of ASCII text files. Figure 4A shows the general format of these peak list files. Along with the peak lists, AutoAssign requires a general description file called a "table" file. This table file contains the protein sequence, a list of tolerances for aligning peak lists, a description of each peak list file, and possibly a flag indicating a fully-deuterated or partially-deuterated sample. The description of each peak list file also includes a description of the "phasing" (i.e. the spin topology information indicated by the signs of peak intensities) if it is present. Figure 4B shows the format of this table file.

Quality Control Issues of Input Data for AutoAssign

There are many quality control issues concerning the creation, editing, and validation of the peak lists. Many of these issues relate directly to the process of data collection[6]. Good performance from AutoAssign requires quality assurance at every step of data collection, processing, peak picking, and interactive editing of the peak lists.

Data Collection and Processing. The most critical consideration in data collection involves ensuring adequate spectral resolution and minimal tolerances when comparing peak resonance frequencies between two spectra. This is particularly important for constructing the CA- and CO-ladders of a GS and then comparing CA- and CO-ladder

resonance frequencies from different GS's for finding possible links. From a practical point of view, collecting the spectra in a back-to-back fashion on the same NMR sample is the best strategy for minimizing tolerances between spectra. For samples which provide sufficient signal-to-noise, this precaution minimizes most problems with samples except rapid degradation. In our laboratory, we generally prepare two identical samples, and execute the full set of NMR experiments back-to-back on one of these. If the sample decomposes or is accidentally spoiled in the course of data collection, the remaining spectra can be collected on the second sample. As a preventative measure, we validate sample stability by incubating a small quantity of sample under NMR data collection conditions for seven to twelve days and then evaluate for proteolysis and/or aggregation by SDS-PAGE. Also, we make efforts to collect all the spectra with identical sample temperatures. This is not always simple to do, as the different decoupling duty cycles used in the various triple-resonance experiments can result in varying degrees of sample heating. Nevertheless, this can be measured and the temperature of the probe offset appropriately to account for differential sample heating. Finally, we collect an HSQC at the beginning and at intermittent times during data collection to determine the amount of change, if any, that occurred during these measurements.

Other important considerations for quality control involve digital resolution and processing, the overall measuring time required to record the eight or nine spectra, and chemical shift referencing[6]. In our laboratory, the recommended minimal digital resolutions for data collection are 0.024, 1.00, 1.66, and 0.35 ppm / pt in the direct H^N, indirect ^{15}N, indirect aliphatic or carbonyl ^{13}C, and indirect aliphatic H^{α} dimensions, respectively. On a 500 MHz spectrometer, these correspond to sweepwidths of 6250 Hz, 2000 Hz, 8300 Hz, 8300 Hz, and 7000 Hz in the H^N, ^{15}N, ^{13}C-aliphatic, $^{13}C'$, and H^{α} dimensions, respectively, collecting 512 complex points in the direct dimension and 40 complex points in each of the indirect dimensions. We zero-fill the direct H^N dimension to 1024 complex points and 2-fold linear predict and zero-fill each indirect dimension to

128 complex points. This produces processed digital resolutions of 0.012, 0.32, 0.52, and 0.110 ppm / pt in the direct H^N, indirect ^{15}N, indirect aliphatic or carbonyl ^{13}C, and indirect aliphatic H^{α} dimensions, respectively. The use of linear prediction instead of zero filling results in less severe Fourier truncation artifacts and reduces line broadening effects caused by window functions[18]. In our hands, linear prediction produces cleaner spectra and better shaped peaks, thus improving the performance of the peak picking algorithms, providing higher quality peak lists, and ultimately improving the performance of AutoAssign. Successful analyses have been done with lower resolution data sets, but the above resolutions work well in our hands. Collecting identical digital resolution for matching dimensions is necessary to minimize tolerances between spectra. Processing with identical digital resolutions and Fourier transforming using identical window functions helps to minimize the required tolerances as well[6].

Of course, the signal-to-noise ratio available on a specific instrument for a specific sample determines the overall measuring time. For example, using a 500 MHz spectrometer and sample concentrations of 1 - 3 mM, typical total data collection times for seven to nine spectra in our laboratory range from 10 to 14 days. It should be possible to reduce these collection times with developments in triply [^2H, ^{13}C, ^{15}N]-enriched protein samples, ^2H-decoupled triple-resonance experiments[19], cryoprobes, and TROSY-based experiments[20]. Efforts to minimize data collection times will speed up the whole structure determination process and are especially important for samples that degrade within the time frame of current collection strategies.

Chemical Shift Referencing. Proper chemical shift referencing for the $^{13}C^{\alpha}$ and $^{13}C^{\beta}$ resonances is essential for accurate amino acid typing[9, 14]. Secondary structure analysis based on chemical shift data also requires accurate chemical shift referencing for $^{13}C^{\alpha}$, $^{13}C^{\beta}$, $^{13}C'$ and H^{α} resonances[21]. Both of these chemical shift analyses rely on the use of

the recommended IUPAC chemical shift referencing method with dimethylsilapentane-5-sulfonic acid (DSS) as the reference compound[22].

Peak Picking. Peak picking is the final issue in quality control. Peak lists do not have to be perfect. AutoAssign can handle the presence of artifactual peaks and incompleteness; however, inaccurate or imprecise peak picking can considerably limit the performance of the program. This is especially true with regard to the key HSQC and HNCO spectra since AutoAssign restricts GS creation to the list of ^{15}N-H^N "root" frequencies defined from these spectra. In addition, the lack of $^{13}C^{\beta}$ resonances severely limits the ability of AutoAssign to define a restricted set of "possible amino acid types" for the CA- and CO-ladders. For most spectra, peak-picking software can generate the initial list of cross-peak resonance frequencies and intensities for each 2D and 3D spectrum using intensity and linewidth filters together with resonance lists from another peak-picked spectrum. This initial list is then edited manually to identify and eliminate extraneous peaks, using interactive graphics and various general features such as the approximate expected number of peaks, the visual quality of alignment across spectra, and peak shape criteria. However, as general specifications for peak picking are not yet established, the user-defined criteria for manual editing of peak-picked spectra vary considerably. For an experienced spectroscopist with good spectra, the interactive manual editing requires roughly 1 hour per 3D NMR spectrum. This is roughly 10 hours for a complete set of spectra. However, a user can carry out such editing while data collection is in progress. Thus, interactive editing may not increase the time required for the complete automated backbone resonance assignment process. Of course, poor quality, highly degenerate, or problematic spectra require considerably more time to accurately peak-pick. In our laboratory, we primarily use the interactive analysis program Sparky for automatic peak picking and manual editing of the peak lists[23]. Sparky has automatic restricted peak picking facilities that uses the resonance list from one peak-picked spectrum to aid in

peak picking a second spectrum. In addition, Sparky can save the peak lists in the proper AutoAssign format. We also provide filtering perl scripts with the AutoAssign distribution to significantly reduce the time for manual peak picking. This approach requires a careful peak picking of the ^{15}N-H^{N} HSQC spectrum, as well as H-^{13}C and H-H projections of some of the other triple resonance spectra. The user then performs a low level automatic peak picking of all the 3D triple-resonance spectra restricted on the ^{15}N-H^{N} resonances of the HSQC spectrum (Sparky can do this). Next, the user filters these peak lists using the extract_by_filter.pl perl script with respect to the ^{1}H and ^{13}C resonance lists derived from the carefully picked 2D ^{13}C-^{1}H HSQC or 2D projections of 3D spectra. In this way, the peak lists are filtered by restricted peak picking in all dimensions. Ridges and other baseline distortions in the spectra can significantly affect the performance of this approach and should be suppressed by appropriate baseline correction prior to analysis.

Validation of Peak Lists. In our experience, it is often useful and necessary to validate certain features of the input files before using AutoAssign. The most common causes of poor performance are inaccurate and/or inconsistent alignments between peak lists. Although the program includes an internal correction to optimize the alignment between spectra, it is critical to ensure that peak lists are properly aligned both with respect to one another and properly referenced with respect to standard IUPAC indirect ^{13}C and ^{15}N referencing relative to DSS[22].

Specific validation procedures that should be done on each set of peak list files include:

- The 2D ^{15}N-H^{N} HSQC spectrum and all 3D spectra should exhibit superimposable 2D ^{15}N-H^{N} projections, within tolerances of < 0.35 ppm and < 0.025 ppm in the ^{15}N and H^{N} dimension, respectively.

- $^{13}C^{\alpha}$ frequencies in 3D HNCA and (HA)CA(CO)NH spectra should fall in the range of 40 - 70 ppm, with an average value of ~ 57 ppm.
- $^{13}C^{\beta}$ frequencies in 3D CBCANH and CBCA(CO)NH spectra should fall in the range of 10 - 75 ppm, with an average value of ~39 ppm.
- The $^{13}C^{\alpha}$ regions of 2D ^{13}C-H^{N} and ^{13}C-^{15}N projections of the 3D HNCA and CBCANH experiments should exhibit many superimposable peaks, within tolerances of < 0.6 ppm, < 0.35 ppm, and < 0.025 ppm in the $^{13}C^{\alpha}$, ^{15}N, and H^{N} dimensions, respectively.
- The $^{13}C^{\alpha}$ regions of 2D ^{13}C-H^{N} and ^{13}C-^{15}N projections of the 3D (HA)CA(CO)NH and CBCA(CO)NH spectra should exhibit many superimposable peaks, within tolerances of < 0.6 ppm, < 0.35 ppm, and < 0.025 ppm in the $^{13}C^{\alpha}$, ^{15}N, and H^{N} dimensions, respectively.
- The distribution of $^{13}C'$ resonance frequencies (plot as histograms) in the 3D HNCO and HN(CA)CO spectra should be similar.
- The distribution of H^{α} resonance frequencies (plot as histograms) in the 3D HA(CA)NH and HA(CA)(CO)NH spectra should be similar.

To facilitate this validation, the AutoAssign distribution includes utility perl scripts called calculate_referencing.pl and validate_script.pl . The calculate_referencing.pl script calculates the shift between two peak lists in the specified peak dimensions that minimizes the peak frequency differences. The validate_script.pl script calculates various statistics about peak list file in comparison to the protein sequence. Both scripts give valuable information for validating the peak list files before using AutoAssign.

Using AutoAssign

AutoAssign has two major components, the processing server written in C++ and

the graphical user interface (GUI) written in Java 2.1. These choices of implementation languages provide the necessary processing speed to make AutoAssign interactive while leaving the GUI relatively platform independent. We compile the C++ server for Mips-IRIX (SGI), Spark-Solaris, and Pentium-Linux platforms and package the full AutoAssign distribution for these platforms. The complete AutoAssign distribution is freely available to academic users at http://www-nmr.cabm.rutgers.edu .

Expanded Strategies. AutoAssign was originally designed to use a set of peak lists from eight triple-resonance experiments plus a 2D ^{15}N-H^N HSQC spectrum. Desiring to carry out more rapid data collection strategies using fewer spectra, we are developing the flexibility to handle alternative experimental strategies. Table I shows a basic list of viable strategies available for the current AutoAssign package. We divide them into three categories defined as 4-rung, 3-rung, and 2-rung strategies. However, there are robustness issues with the 2-rung strategy and thus we do not recommend it except with pristine and complete peak lists. Strategies 1, 2, and 6 in Table I are 4-rung, strategy 10 is 2-rung, and all other strategies are 3-rung.

The expanded strategies also allow the use of data generated from other triple-resonance experiments. The simplest examples involve generating a required ^{15}N-H^N HSQC peak list from an HNCO peak list in strategies 2, 4, 6, 7, 8, 9, and 10. Any of the other experiments contain the required ^{15}N-H^N HSQC data; however, the HNCO is generally the most complete and, therefore, the logical choice to use. Another example is generating an HN(CO)CA-like peak list from a phase labeled version of an HN(CO)CACB-like peak list[14] where the phase (sign) of cross peaks distinguishes between the $^{13}C^{\alpha\prime}$s and $^{13}C^{\beta\prime}$s (e.g., in strategies 6, 8, 9, and 10). To aid in using these different strategies, we implemented a suite of perl scripts for generating the peak list of one experiment type from the peak lists of other experiments. The perl suite can generate all the peak lists for the strategies shown in Table I. Furthermore, it can generate the

required input for AutoAssign from even more compact data collection strategies; for example, generating an HN(CA)CO-like peak list from a combination of an (HB)CBCACO(CA)HA[24], HNHA-like[16], and HNCACB-like[14] peak lists. This specific example is useful because (in the absence of partial deuteration) the (HB)CBCACO(CA)HA experiment is generally more sensitive than other experiments commonly used to establish intraresidue HN_i -> CO_i connections. We also included various peak list validation tools in the perl suite along with tools for reformatting peak lists generated by various peak-picking packages. These tools allow quick creation and validation of peak lists.

It is important to point out that strategies 5, 7, 9, and 10 of Table I are amenable to experiments involving ^{2}H,^{13}C,^{15}N-enriched proteins since even partial deuteration results in essentially complete ^{2}H enrichment of H^{α} atoms during biosynthesis, thus precluding assignment of the H^{α}'s. The current version of AutoAssign has facilities for handling peak lists with ^{2}H isotope shift effects on ^{13}C resonance frequencies for fully- and partially-deuterated protein samples.

Executing AutoAssign. Once a user determines which strategy to use, they must create a table file and associated peak list files. AutoAssign has a set of perl scripts to aid in peak list file creation. The perl scripts are versatile enough to quickly convert row-based peak list files from any interactive spectral visualization/analysis program into the AutoAssign format with nothing more than nominal manual editing. For the table file, we recommend that the user copies a table file from one of the example data sets included with the AutoAssign distribution and modify it to fit their use. After validating the peak list files, the user can run AutoAssign via the autoclient script. This script starts the Java-implemented GUI (see Figure 5). From the main window (Figure 5A), the user will connect to the C++-implemented server.

Next the user opens the table file. Once the table file is opened and the data set properly loaded, the user can run the "Refined Execution" for full analysis of the dataset. AutoAssign will display the results of execution on a "Connectivity Map" (Figure 5B) while performing the "Refined Execution." There are several options for viewing and saving the resulting resonance assignment list including the NMRStar format for the BioMagResBank and several row-based formats. Web-based help documentation describes the various features of the graphical user interface. In addition, AutoAssign has reports to aid further manual assignments after its automated methods are exhausted.

The AutoAssign program is also designed to use the Sparky interactive analysis program[23] to access strip plots and other views of the NMR data. Sparky can peak pick spectra and prepare the required input files for AutoAssign. It also has facilities to run AutoAssign, read AutoAssign's assignment results, and directly incorporate them into its "spin graph" representations of assignments and spectral data. This provides the user access to Sparky's advanced interactive graphical tools to view the spectral data as strip plots, manually edit the peak lists, and/or manually extend assignments starting where AutoAssign left off. This Sparky/AutoAssign interface makes iterative manual editing of peak lists and rerunning of AutoAssign very easy.

Testing AutoAssign

We tested AutoAssign on data sets for eleven different proteins using several different experimental strategies, including strategies 1, 3, and 6 of Table I. Table II summarizes the results of these tests. The sizes of the proteins analyzed range from 6 to 19 kDa. Details related to some of these analyses are presented in published descriptions of their structure determinations[25-31]. We performed these tests on an SGI (Mips) 194 MHz R10000 processor running SGI IRIX; however, the same tests on a Pentium II-400 MHz processor running Linux gave similar results. "Default Execution" involves only 4-rung, 3-rung, and 2-rung matching. The CPU times required for "Default Execution"

were less than 30 seconds in all cases, and in most cases, it was less than 10 seconds. From a practical standpoint, such fast execution allows a human to perform real-time optimization and tweaking of the peak lists which is extremely useful for troubleshooting hard or problematic data sets. This feature is particularly valuable when used together with the Sparky interface. The "Refined Execution" involves extending these established assignments using additional 1-rung matching. Table 2 shows the number of $^{13}C^{\alpha}$ assignments obtained. We use $^{13}C^{\alpha}$ assignments as the statistical measure for the number of amino acid residues with assigned backbone resonances (generally H^{N}, ^{15}N, $^{13}C^{\alpha}$, H^{α}, and sometimes $^{13}C'$ and /or $^{13}C^{\beta}$) instead of counting H^{N} or ^{15}N assignments because prolines and N-terminal residues lack root (^{15}N-H^{N}) resonance pairs but gain $^{13}C^{\alpha}$ assignments through the CO-ladder of the next amino acid in the sequence. We determine error rates by comparing the results of automated analysis by AutoAssign with the final assignment list produced with additional (or exclusively) manual analysis. The results of these tests, for "Default Execution" only, are 92.8% of $^{13}C^{\alpha}$'s (i.e. sets of backbone resonance assignments) assigned on average with a 0.09% error rate in root resonances. After applying further refinement with looser matching criteria (i.e. allowing 1-rung matching), this average increases to 96.0% assignment with a 0.44% error rate. Minor differences between manual and automated assignments demonstrate that the output of AutoAssign should be viewed as *a hypothesis for assignments, which must then be validated by manual inspection of the NMR spectra using an interactive spectral analysis tool like Sparky*.

Conclusions and Future Prospects

The AutoAssign program is an extensively tested and reasonably robust approach for automatic backbone resonance assignments. Quality control issues remain with regard to data collection, referencing, processing, and peak picking. In our experience, the primary challenge to automated analysis with AutoAssign involves peak picking and

interactive editing of the peak lists. However, AutoAssign has an excellent track record with good peak lists. Also, the program is flexible enough to use a variety of experimental data, an important practical concern in NMR research where the experimental methods are under rapid development and improvement.

Future directions and improvements for AutoAssign include the following: full automatic resonance assignments; the use of HCC(CO)NH TOCSY[32] and HCCH-COSY[33] data for spin system typing and side chain resonance assignments; the use of 3D reduced dimensionality data[34]; the use of NOESY data to validate and extend sequential connectivities; development of a reduced set of experiments to determine resonance assignments for mutant and slightly perturbed proteins (i.e. ligand bound, etc.) based on available assignments for wild-type (or apo) forms; and built-in quality control and validation of peak lists, providing warnings of actual and potential problems in the quality of the data. It is reasonable to expect future developments to combine automatic determination of resonance assignments with automatic determination of NOESY cross peak assignments and protein structure determination, providing validation of resonance assignments by their consistency with the complete structure analysis.

A**cknowledgments**

We would like to thank Drs. Robert Powers and Robert Schisknis (Ayeth-Hyerst Pharmaceutical Company) for providing the data sets for FGF, collagenase, and the DNA-binding domain of the human progesterone receptor. We would also like to thank Dr. Luciano Mueller (Bristol-Myers Squibb Pharmaceutical Research Institute) for providing the data set for “Pharmaceutical Protein X”. This work was supported by grants from the National Institutes of Health (GM-47014), the National Science Foundation (MCB-9407569, DBI-9974200 Postdoctoral fellowship to HNBM), a New Jersey Commission on Science and Technology Research Excellence Award, and the Spanish Science and Education Ministry (Postdoctoral fellowship EX-2917991 to DM).

References

1. G. T. Montelione and G. Wagner, *J. Magn. Reson.* 87, 183 (1990).

2. M. Ikura, L. E. Kay and A. Bax, *Biochemistry* 29, 4659 (1990).

3. G. M. Clore and A. M. Gronenborn, *Science* 252, 1390 (1991).

4. K. Wüthrich, *Acta. Cryst.* D51, 249 (1995).

5. L. E. Kay, *Prog. Biophys. Mol. Biol.* 63, 277 (1995).

6. G. T. Montelione, C. B. Rios, G. V. T. Swapna and D. E. Zimmerman, in *Biological Magnetic Resonance*, Vol. 17: *Biological Magnetic Resonance: Structure Computation and Dynamics in Protein NMR* (N. R. Krishna and L. Berliner Eds.) p. 81 Plenum, New York, 1999.

7. H. N. B. Moseley and G. T. Montelione, *Curr. Opin. Struct. Biol.* 9, 635 (1999).

8. D. E. Zimmerman, C. A. Kulikowski, L. L. Wang, B. A. Lyons and G. T. Montelione, *J. Biomol. NMR* 3, 241 (1994).

9. D. E. Zimmerman, C. A. Kulikowski, W. Feng, M. Tashiro, R. Powers and G. T. Montelione, *J. Mol. Biol.* 269, 592 (1997).

10. M. Billeter, W. Braun and K. Wuthrich, *J. Mol. Biol.* 155, 321 (1982).

11. G. Wagner and K. Wüthrich, *J. Mol. Biol.* 155, 347 (1982).

12. K. Wüthrich, *NMR of Proteins and Nucleic Acids*, John Wiley & Sons, New York, 1986.

13. B. R. Seavey, E. A. Farr, W. M. Westler and J. Markley, *J. Biomol. NMR* 1, 217 (1991).

14. S. Grzesiek and A. Bax, *J. Biomol. NMR* 3, 185 (1993).

15. M. Tashiro, C. B. Rios and G. T. Montelione, *J. Biomol. NMR* 6, 211 (1995).

16. W. Feng, C. B. Rios and G. T. Montelione, *J. Biomol. NMR* 8, 98 (1996).

17. C. B. Rios, W. Feng, M. Tashiro, Z. Shang and G. T. Montelione, *J. Biomol. NMR* 8, 345 (1996).

18. P. Koehl, *Prog. NMR Spec.* 34, 257 (1999).

19. K. H. Gardner and L. E. Kay, *Annu. Rev. Biophys. Biomol. Struct.* 27, 357 (1998).

20. M. Salzmann, G. Wider, K. Pervushin, H. Senn and K. Wüthrich, *J. Am. Chem. Soc.* 121, 844 (1999).

21. D. S. Wishart and B. D. Sykes, *J. Biomol. NMR* 4, 171 (1994).

22. D. S. Wishart, C. G. Bigam, J. Yao, F. Abildgaard, H. J. Dyson, E. Oldfield, J. L. Markley and B. D. Sykes, *J. Biomol. NMR* 6, 135 (1995).

23. T. D. Goddard and D. G. Kneller, SPARKY 3, University of California, San Francisco (1999).

24. L. E. Kay, *J. Am. Chem. Soc.* 115, 2055 (1993).

25. B. A. Lyons, M. Tashiro, L. Cedergren, B. Nilsson and G. T. Montelione, *Biochemistry* 32, 7839 (1993).

26. K. Newkirk, W. Feng, W. Jiang, R. Tejero, S. D. Emerson, M. Inouye and G. T. Montelione, *Proc. Natl. Acad. Sci. USA* 91, 5114 (1994).

27. M. Tashiro, R. Tejero, D. E. Zimmerman, B. Celda, B. Nilsson and G. T. Montelione, *J. Mol. Biol.* 272, 573 (1997).

28. C.-Y. Chien, R. Tejero, Y. Hunagn, D. E. Zimmerman, R. M. Krug and G. T. Montelione, *Nature Struct. Biol.* 4, 891 (1997).

29. F. J. Moy, M. R. Pisano, P. K. Chanda, C. Urbano, L. M. Killar, M. L. Sung and R. Powers, *J. Biomol. NMR* 10, 9 (1997).

30. S. Shimotakahara, C. B. Rios, J. H. Laity, D. E. Zimmerman, H. A. Scheraga and G. T. Montelione, *Biochemistry* 36, 6915 (1997).

31. W. Feng, R. Tejero, D. E. Zimmerman, M. Inouye and G. T. Montelione, *Biochemistry* 37, 7834 (1998).

32. G. T. Montelione, B. A. Lyons, S. D. Emerson and M. Tashiro, *J. Am. Chem. Soc.* 114, 10974 (1992).

33. A. Bax, G. M. Clore, P. C. Driscoll, A. M. Gronenborn, M. Ikura and L. E. Kay, *J. Magn. Reson.* 87, 620 (1990).

34. T. Szyperski, B. Banecki, D. Braun and R. W. Glaser, *J. Biomol. NMR* 11, 387 (1998).

Table I: Experimental strategies for AutoAssign.

	Strategies									
Type of Experiment	**1[b]**	**2[b]**	**3**	**4**	**5[c]**	**6[b]**	**7[c]**	**8**	**9[c]**	**10[c]**
HSQC[a]	*	perl[d]	*	perl[d]	*	perl[d]	perl[d]	perl[d]	perl[d]	perl[d]
HNCO[a]	*	*	*	*	*	*	*	*	*	*
HN(CA)CO	*	*			*	*	*		*	
HNCA[a]	*	*	*	*	*	perl[e]	*	perl[e]	perl[e]	perl[e]
HN(CO)CA[a]	*	*	*	*	*	perl[f]	*	perl[f]	perl[f]	perl[f]
HNCACB[a]	*	*	*	*	*	phase required	*	phase required	phase required	phase required
HN(CO)CACB[a]	*	*	*	*	*	phase optional	*	phase optional	phase optional	phase optional
HNHA	*	*	*	*		*		*		
HN(CO)HA	*	*	*	*		*		*		

[*]**Strategy requires any implementation of an experiment of that type.**

[perl]**Generated from other experiments using perl scripts.**

[phase]**Phase sensitive experiment that distinguishes $^{13}C^{\alpha}$ from $^{13}C^{\beta}$.**

[a]**Data required for running AutoAssign. These data may be generated from the corresponding experiment or extracted from other triple resonance experiments using perl scripts as indicated in the table.**

[b]**Optimal strategies.**

[c]**Strategies amenable to experiments on samples with 100% ^{2}H,^{13}C,^{15}N-enrichment.**

[d]**Generated from HNCO.**

[e]**Generated from HNCACB.**

[f]**Generated from HN(CO)CACB.**

Figure Legends

Fig. 1. Generic Spin System Object (GS).

The GS is a set of resonance frequencies derived from resonances associated by a network of one-bond scalar couplings. Its provides a logical unit of organization to begin the assignment process. AutoAssign uses a dipeptide-based GS constructed with peaks from several triple-resonance spectra associated together via a common ^{15}N-H^N root. Resonances of nuclei within the same amino acid residue as the ^{15}N-H^N root make up the CA-ladder. Resonances of nuclei of the previous amino acid residue in the protein sequence make up the CO-ladder. AutoAssign also keeps track of the various sequence-specific sites (SS's) in the protein sequence that are compatible with the GS.

Fig. 2. Example of constraint-propagation resulting from establishing a link.

(A) Sequence-specific sites (SS's) consistent with data for two GS's after "amino acid typing" of their CA- and CO-ladders. The SS's are dipeptide sites centered on the backbone amide NH of each amino acid residue in the protein sequence.

(B) After identifying a match between the CA-ladder of the GS on the left to the CO-ladder of the GS on the right, AutoAssign establishes a link between the two GS's, which propogates a consistency constraint, which then creates a unique tripeptide mapping (double SS).

Fig. 3. Schematic representation of experimental input to AutoAssign.

This set of non-directed graphs depicts the types of triple-resonance data usable by AutoAssign (for experimental details see ref 6). The edges in the graphs reflect the transfer of magnetization through the participating nuclei. Atoms within parenthesis along a given coherence transfer pathway are *not* detected. Experiments (A) through (D)

correlate the H^{α}, $^{13}C^{\alpha}$, $^{13}C^{\beta}$, and $^{13}C'$ resonances of residue *i - 1* with the ^{15}N-H^{N} resonances of residue *i* and provide data that define the CO-ladder. Experiments (E) through (H) correlate the H^{α}, $^{13}C^{\alpha}$, $^{13}C^{\beta}$, and $^{13}C'$ resonances of residue (*i*) with its own ^{15}N-H^{N} resonances and provide data that define the CA-ladder.

Fig. 4. Example input files for AutoAssign.

(A) Example HNCO peak list file in the format that AutoAssign reads. The first column is a peak index. The next three columns are the ^{15}N, H^{N}, and $^{13}C'$ frequencies respectively. The fifth column is the intensity of the peak. The final column is a token describing the experimental origin of these data.

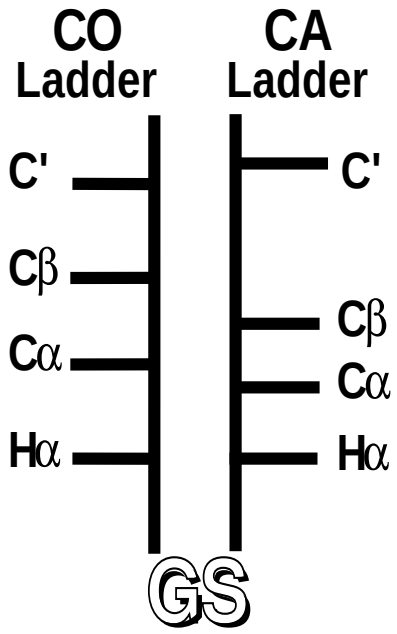
(B) Example AutoAssign Table File. The "Protein:" field provides a name for the protein. The "Sequence:" field is the amino acid sequence of the protein with the initial number indicating the residue position with which the sequence starts and an asterisk to mark the end of the sequence. The "Tolerance:" section is the initial global tolerances, in ppm, used for clustering and aligning peaks along their common dimensions. The "Spectra:" section has the description of the peak lists. The first field is the name of the spectrum. The second is the reference spectrum to which the ^{15}N and H^{N} frequencies should be aligned, generally an ^{15}N-H^{N} HSQC peak list, (defined as ROOT for the HSQC spectra). The third field provides the name of the peak file. The fourth, fifth, and sixth fields indicate the presence of intra, sequential, and through space interactions, respectively, in these experimental data. The seventh field is a description of how phasing (if any is present in the experiment) should be interpreted in terms of spin system topology. In this example, HNcoCA peak list has phasing (i.e. the sign of peak intensities) distinguishing glycine $^{13}C^{\alpha}_{i-1}$->H^{N}_{i} correlations from others. Each line thereafter has the description of the information content in each dimension the spectrum. For each dimension, the table file includes a list of resonance atom types, the spectral

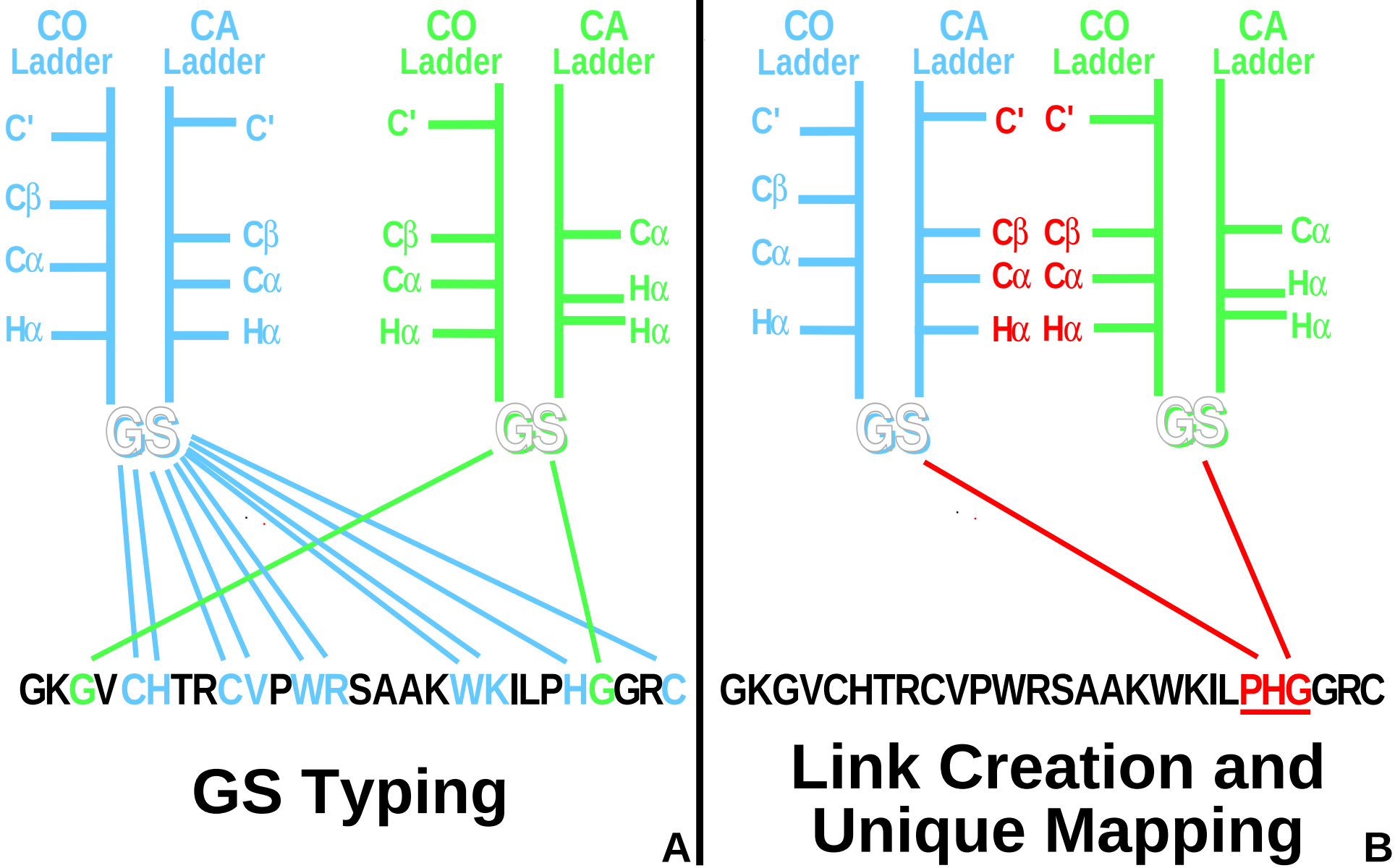
range in ppm, an offset in ppm, a spectrum-specific match tolerance in ppm, and whether the dimension is folded. An optional "Properties:" section contains a description of the protein sample's deuteration.

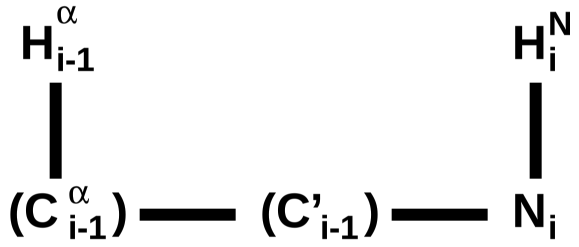
Fig. 5. Graphical user interface of AutoAssign.

(A) Main window for the AutoAssign graphical user interface. The "Open Table File" menu option is selected in the menu bar. This menu option will open the table file, read the spectral descriptions, read in the peak lists, and initialize the internal representation of the data set.

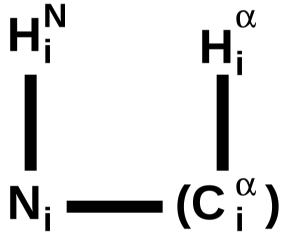
(B) "Connectivity Map" which graphically shows automated assignment results for human basic fibroblast growth factor. AutoAssign continually updates the "Connectivity Map" while executing. The grey bars in this window indicate the presence of intra (dark) and sequential (light) connectivity data used by AutoAssign to establish resonance assignments at each sequence site in the protein sequence.


CO
Ladder
CA
Ladder
C'
Cβ
Cα
Hα
C'
Cβ
Cα
Hα
GS

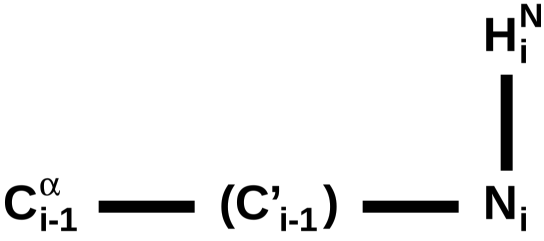

CO Ladder
CA Ladder
CO Ladder
CA Ladder
C'
Cβ
Cα
Hα
C'
Cβ
Cα
Hα
C'
Cβ
Cα
Hα
Cα
Hα
Hα
GS
GS
GKGVCHTRCVPWRSAAKWKILPHGGRC
GS Typing
A
CO Ladder
CA Ladder
CO Ladder
CA Ladder
C'
Cβ
Cα
Hα
C'
C'
Cβ
Cβ
Cα
Cα
Hα
Hα
Cα
Hα
Hα
GS
GS
GKGVCHTRCVPWRSAAKWKILPHGGRC
Link Creation and Unique Mapping
B



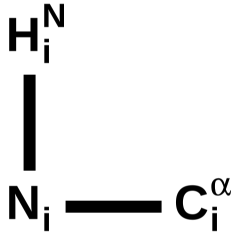
HA(CA)(CO)NH
A



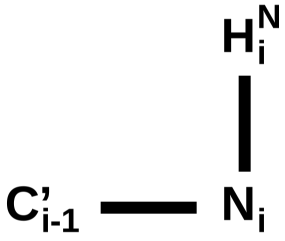
HA(CA)NH
E



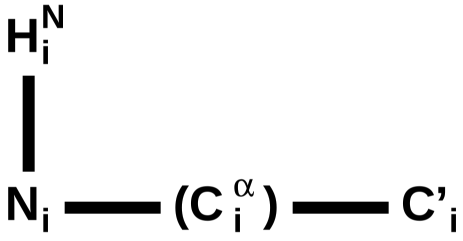
CA(CO)NH
B



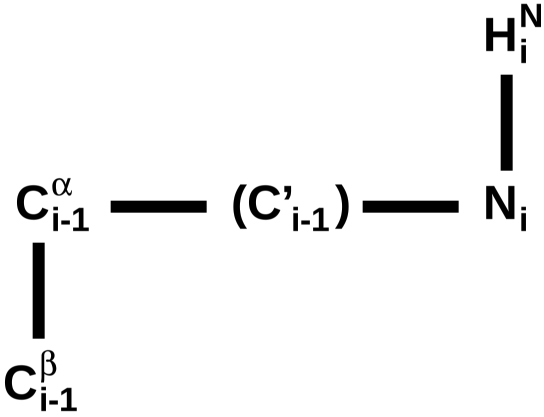
CANH
F



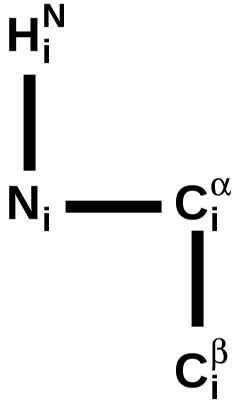
HNCO
C



HN(CA)CO
G



CBCA(CO)NH
D



CBCANH
H

#Index	1Dim	2Dim	3Dim	Intensity	Workbook
1	173.09	10.63	117.72	+1.88e+06	HNCO
2	173.38	10.47	120.30	+3.39e+06	HNCO
3	174.80	10.34	122.94	+2.82e+06	HNCO
4	174.13	10.25	126.37	+4.86e+06	HNCO
5	174.63	10.23	119.96	+4.68e+06	HNCO
6	175.48	9.77	118.64	+4.84e+06	HNCO
7	176.44	9.74	119.24	+5.53e+06	HNCO

A

```
#
#  FGF Table File for AutoAssign
#
Protein: FGF

Sequence: 2 AEGEITTLPALPEDGGSGAFPPGHFKDPKRLYBKNGGFFLRIHPDGRVDG
VREKSDPHIKLQLQAEERGVVSIKGVSANRYLAMKEDGRLLASKSVTDEBFFFERLE
SNNYNTYRSRKYTSWYVALKRTGQYKLGSKTGPGQKAILFLPMSAKS*

Tolerances: HN .02 N15 .25 CA .4 CB .6 HA .05 CO 0.2

Spectra:
HSQC  ROOT    hsqc.pks   1   0   0 phase: {} {
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNCO  HSQC    hnco.pks  0   1   0 phase: {} {
{CO     170 180   0  .25 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCA  HSQC   hncoca.pks 0   1   0 phase: { CA {G} } {
{CA      40  70   0  .40 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCACB HSQC  hncocacb.pks  0   1   0 phase: {} {
{{CA CB } 10 80   0  .60 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoHA  HSQC   hncoha.pks 0   1   0 phase: {} {
{HA      1   8   0  .05 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
```

B

#Index	1Dim	2Dim	3Dim	Intensity	Workbook
1	173.09	10.63	117.72	+1.88e+06	HNCO
2	173.38	10.47	120.30	+3.39e+06	HNCO
3	174.80	10.34	122.94	+2.82e+06	HNCO
4	174.13	10.25	126.37	+4.86e+06	HNCO
5	174.63	10.23	119.96	+4.68e+06	HNCO
6	175.48	9.77	118.64	+4.84e+06	HNCO
7	176.44	9.74	119.24	+5.53e+06	HNCO

A

```
#
#  FGF Table File for AutoAssign
#
Protein: FGF

Sequence: 2 AEGEITTLPALPEDGGSGAFPPGHFKDPKRLYBKNGGFFLRIHPDGRVDG
VREKSDPHIKLQLQAEERGVVSIKGVSANRYLAMKEDGRLLASKSVTDEBFFFERLE
SNNYNTYRSRKYTSWYVALKRTGQYKLGSKTGPGQKAILFLPMSAKS*

Tolerances: HN .02 N15 .25 CA .4 CB .6 HA .05 CO 0.2

Spectra:
HSQC  ROOT    hsqc.pks   1   0   0 phase: {} {
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNCO  HSQC    hnco.pks  0   1   0 phase: {} {
{CO     170 180   0  .25 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCA  HSQC   hncoca.pks 0   1   0 phase: { CA {G} } {
{CA      40  70   0  .40 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCACB HSQC  hncocacb.pks  0   1   0 phase: {} {
{{CA CB } 10 80   0  .60 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoHA  HSQC   hncoha.pks 0   1   0 phase: {} {
{HA      1   8   0  .05 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
```

B

#Index	1Dim	2Dim	3Dim	Intensity	Workbook
1	173.09	10.63	117.72	+1.88e+06	HNCO
2	173.38	10.47	120.30	+3.39e+06	HNCO
3	174.80	10.34	122.94	+2.82e+06	HNCO
4	174.13	10.25	126.37	+4.86e+06	HNCO
5	174.63	10.23	119.96	+4.68e+06	HNCO
6	175.48	9.77	118.64	+4.84e+06	HNCO
7	176.44	9.74	119.24	+5.53e+06	HNCO

A

```
#
#  FGF Table File for AutoAssign
#
Protein: FGF

Sequence: 2 AEGEITTLPALPEDGGSGAFPPGHFKDPKRLYBKNGGFFLRIHPDGRVDG
VREKSDPHIKLQLQAEERGVVSIKGVSANRYLAMKEDGRLLASKSVTDEBFFFERLE
SNNYNTYRSRKYTSWYVALKRTGQYKLGSKTGPGQKAILFLPMSAKS*

Tolerances: HN .02 N15 .25 CA .4 CB .6 HA .05 CO 0.2

Spectra:
HSQC  ROOT    hsqc.pks   1   0   0 phase: {} {
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNCO  HSQC    hnco.pks  0   1   0 phase: {} {
{CO     170 180   0  .25 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCA  HSQC   hncoca.pks 0   1   0 phase: { CA {G} } {
{CA      40  70   0  .40 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCACB HSQC  hncocacb.pks  0   1   0 phase: {} {
{{CA CB } 10 80   0  .60 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoHA  HSQC   hncoha.pks 0   1   0 phase: {} {
{HA      1   8   0  .05 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
```

B

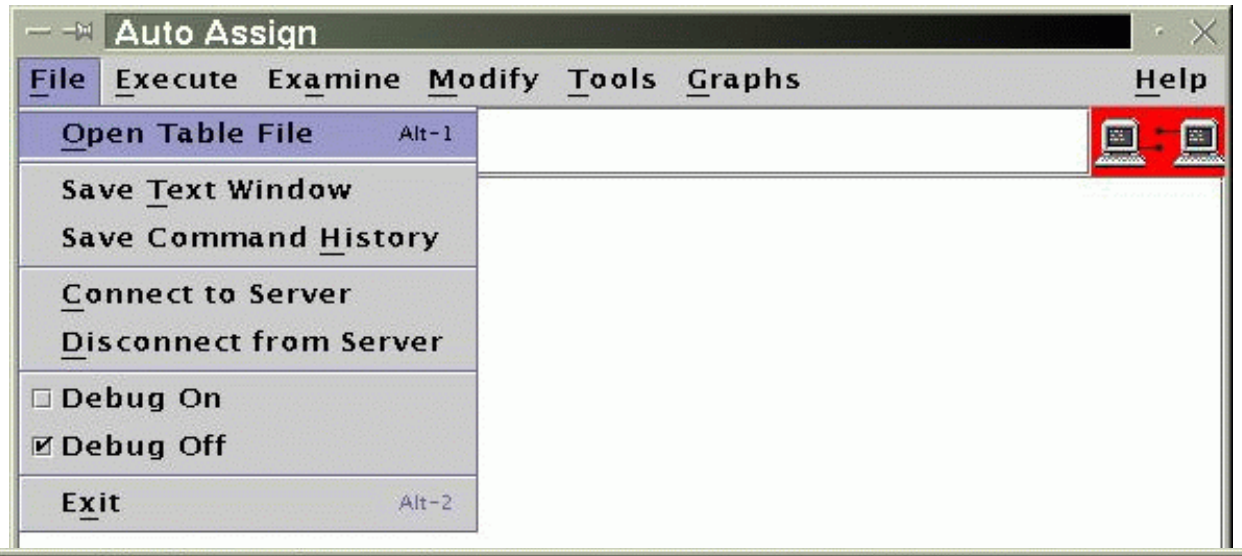
#Index	1Dim	2Dim	3Dim	Intensity	Workbook
1	173.09	10.63	117.72	+1.88e+06	HNCO
2	173.38	10.47	120.30	+3.39e+06	HNCO
3	174.80	10.34	122.94	+2.82e+06	HNCO
4	174.13	10.25	126.37	+4.86e+06	HNCO
5	174.63	10.23	119.96	+4.68e+06	HNCO
6	175.48	9.77	118.64	+4.84e+06	HNCO
7	176.44	9.74	119.24	+5.53e+06	HNCO

A

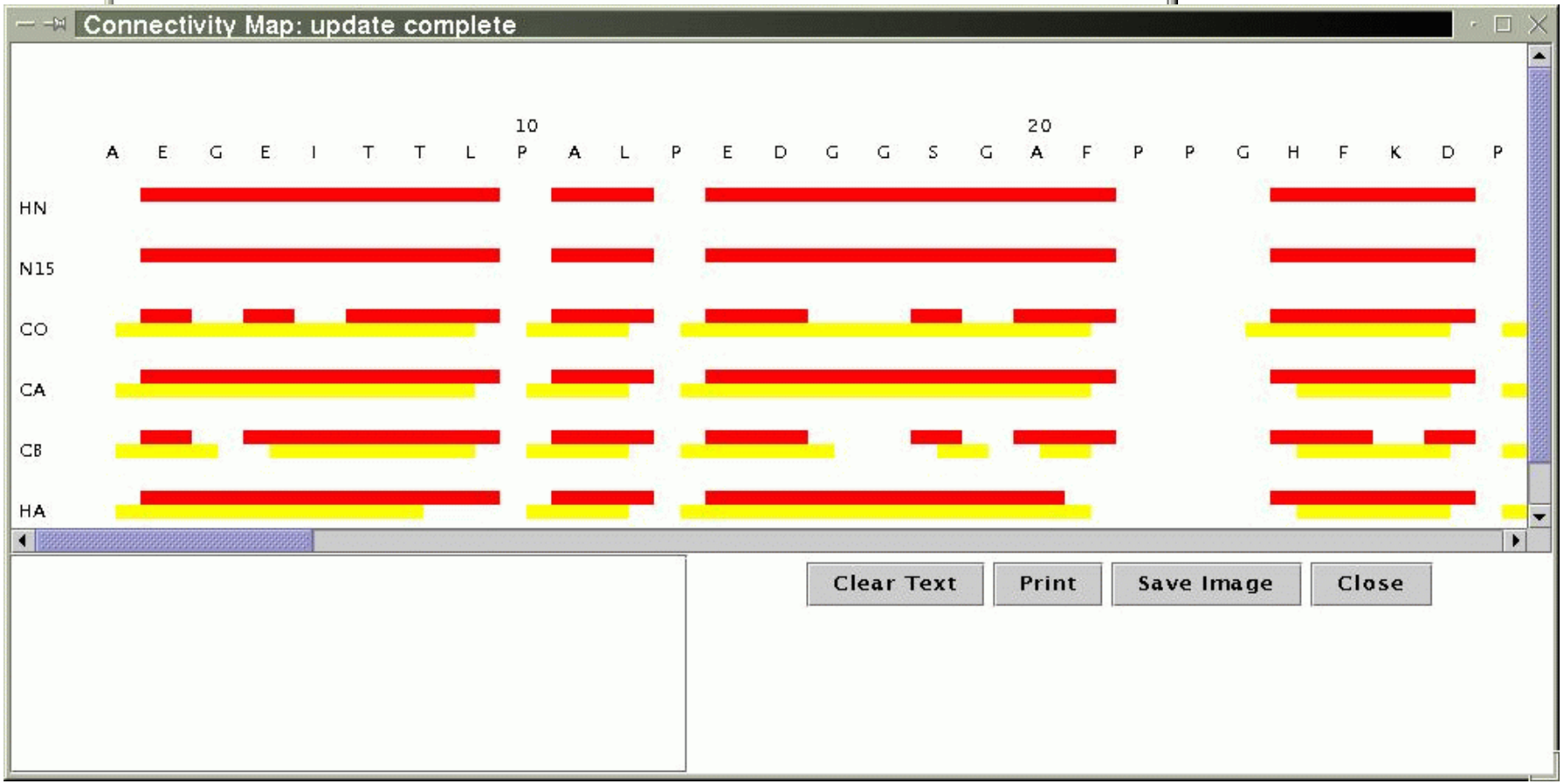
```
#
#  FGF Table File for AutoAssign
#
Protein: FGF

Sequence: 2 AEGEITTLPALPEDGGSGAFPPGHFKDPKRLYBKNGGFFLRIHPDGRVDG
VREKSDPHIKLQLQAEERGVVSIKGVSANRYLAMKEDGRLLASKSVTDEBFFFERLE
SNNYNTYRSRKYTSWYVALKRTGQYKLGSKTGPGQKAILFLPMSAKS*

Tolerances: HN .02 N15 .25 CA .4 CB .6 HA .05 CO 0.2

Spectra:
HSQC  ROOT    hsqc.pks   1   0   0 phase: {} {
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNCO  HSQC    hnco.pks  0   1   0 phase: {} {
{CO     170 180   0  .25 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCA  HSQC   hncoca.pks 0   1   0 phase: { CA {G} } {
{CA      40  70   0  .40 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoCACB HSQC  hncocacb.pks  0   1   0 phase: {} {
{{CA CB } 10 80   0  .60 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
HNcoHA  HSQC   hncoha.pks 0   1   0 phase: {} {
{HA      1   8   0  .05 unfolded }
{HN      4  12   0  .02 unfolded }
{N15    100 130   0  .25 unfolded }
}
```

B

A


Auto Assign
File
Execute
Examine
Modify
Tools
Graphs
Help
Open Table File
Alt-1
Save Text Window
Save Command History
Connect to Server
Disconnect from Server
Debug On
Debug Off
Exit
Alt-2

B


Connectivity Map: update complete
10
20
A E G E I T T L P A L P E D G G S G A F P P G H F K D P
HN
N15
CO
CA
CB
HA
Clear Text
Print
Save Image
Close