Utilizing Machine Learning to Accelerate Automated Assignment of Backbone NMR Data

Joel Venzke[a,b], *David Mascharka*[a], *Paxten Johnson*[a,b], *Rachel Davis*[a,*], *Katie Roth*[a], *Leah Robison*[a], *Adina Kilpatrick*[b] *and Timothy Urness*[a]

[a] *Department of Mathematics and Computer Science, Drake University, Des Moines, IA*
[b] *Department of Physics and Astromony, Drake University, Des Moines, IA*

*Students: joel.venze@drake.edu, david.mascharka@drake.edu, paxten.johnson@drake.edu, rachel.davis@drake.edu**, *katherine.roth@drake.edu, leah.robison@drake.edu*
Mentors: adina.kilpatrick@drake.edu, timothy.urness@drake.edu

ABSTRACT

Nuclear magnetic resonance (NMR) spectroscopy is a powerful method for determining three-dimensional structures of biomolecules, including proteins. The protein structure determination process requires measured NMR values to be assigned to specific amino acids in the primary protein sequence. Unfortunately, current manual techniques for the assignment of NMR data are time-consuming and susceptible to error. Many algorithms have been developed to automate the process, with various strengths and weaknesses. The algorithm described in this paper addresses the challenges of previous programs by utilizing machine learning to predict amino acid type, thereby increasing assignment speed. The program also generates place-holders to accommodate missing data and amino acids with unique chemical characteristics, namely proline. Through machine learning and residue-type tagging, the assignment process is greatly sped up, while maintaining high accuracy.

KEYWORDS

Chemical Shift; Machine Learning; NMR; Artificial Intelligence; Proteins; Bioinformatics

INTRODUCTION

Nuclear Magnetic Resonance (NMR) spectroscopy is a powerful method for obtaining atomic-resolution three-dimensional protein structures,[1] as well as assessing changes in protein conformations or motions due to mutations or interactions with ligands or other biomolecules. Determining a protein's structure is essential for understanding its function and alterations in function, which often lead to disease. The analysis of NMR data, in particular the sequence-specific assignment of backbone and side-chain protein resonances, is an error-prone and time-consuming step during protein structure determination by NMR spectroscopy.[2] This paper describes a computational algorithm that utilizes machine learning in the process of automating the assignment of backbone protein NMR resonances.

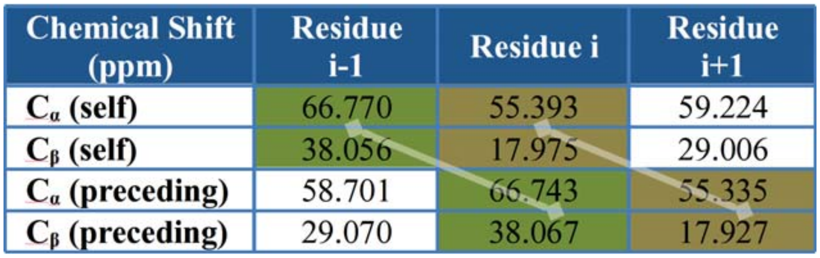
BACKGROUND

NMR experiments generate information on several variables that can be used in the determination of protein structures.[1] In particular, essential information is provided by the chemical shifts of NMR-active nuclei present in proteins, including hydrogen and isotopes of carbon and nitrogen.[3] The chemical shift is a quantifier for the deviation in the resonant frequency of a nucleus from its value in a structure-free environment, and therefore provides information on the local conformation. Measuring the chemical shifts of all or most of the nuclei in a protein is the first step in determining its structure by NMR spectroscopy. An important set of protein chemical shifts are those corresponding to the nuclei in the backbone of the protein polypeptide chain, including the amide nitrogen (N), attached hydrogen (H), and the alpha and beta carbons (C_α and C_β) of each residue. Chemical shift values are measured using various three-dimensional NMR experiments,[4] and then matched to the individual residues in the protein in a process called sequential assignment.

Triple resonance experiments on hydrogen, nitrogen, and carbon nuclei are the method of choice for proteins and other large biomolecules, because they can greatly decrease the amount of spectral overlap in data.[5] Typical experiments for the assignment of backbone resonances include HNCA, HN(CO)CA, HNCACB, and CBCA(CO)NH or HN(CO)CACB.[6] These experiments transfer magnetization over the protein polypeptide chain, and thus connect different spin systems through covalent bonds. A spin system contains all resonances belonging to a particular residue in the protein sequence. The order of elements in experiment names indicates the order in which magnetization is passed down the line of nuclei, resonances in parentheses being used only for transfer to the next nucleus. All these experiments produce NMR spectra with common H-N resonance correlations, and provide connectivities between neighboring residues.[7] For example, the HNCACB experiment identifies the chemical shifts corresponding to the C_α and C_β nuclei of each residue in the protein chain (residue i), as well as the immediately preceding residue (residue $i-1$).[8] In this experiment, resonances corresponding to residue i can usually be distinguished from the $i-1$ values due to their higher intensity. However, ambiguities can arise if the intensities are comparable or if chemical shift ranges overlap. These ambiguities can be resolved by using additional experiments. For example, the CBCA(CO)NH experiment yields the chemical shifts of the preceding residue only, unambiguously identifying i and $i-1$ values.[9] Using all inter-residue connectivities, a chain of correlations through the protein backbone chemical shifts can be established. The pattern of sequentially-linked chemical shift values reflects the linear arrangement of individual residues in the protein sequence. This pattern is then matched to the protein sequence through certain residues with characteristic C_α and C_β chemical shift values that uniquely identify them. Thus, each measured chemical shift is assigned to a protein residue and can then be used to infer structural information about the biomolecule.

SEQUENTIAL ASSIGNMENT STRATEGIES

The sequential assignment of backbone chemical shifts can be done manually or in an automated fashion. Both approaches follow similar strategies. The process starts by identifying the chemical shifts of i and $i-1$ residues from a set of three-dimensional NMR experiments. Many computer programs, such as NMRPipe[10] and SPARKY,[11] can be used to process, analyze, and visualize NMR data. After grouping chemical shifts into spin systems, they are linked into increasingly larger segments by matching i and $i-1$ values. This process can be seen in **Figure 1**. In parallel, the spin systems are classified into possible residue types, using a set of established chemical shift values for each of the 20 common amino acids found in proteins. For example, alanines have C_α and C_β chemical shifts ranging from approximately 50 to 56 ppm (parts-per-million), and 17 to 25 ppm, respectively; glycines have unique C_α values in the 45 ppm region, and threonines have distinctive C_β values in the 68-73 ppm range.[3,12] Using the residue type information, the linked segments from the resonance sequential walk are iteratively mapped onto the primary protein sequence.



Chemical Shift (ppm)	Residue i-1	Residue i	Residue i+1
C_α (self)	66.770	55.393	59.224
C_β (self)	38.056	17.975	29.006
C_α (preceding)	58.701	66.743	55.335
C_β (preceding)	29.070	38.067	17.927

Figure 1. Sequential residues are linked by matching i and $i-1$ chemical shifts

When performed manually, the assignment process is time-consuming and prone to error. If chemical shift values overlap, multiple matches between i and $i-1$ values are possible. The classification of spin systems into amino acid types also has a high potential for error, as the spin systems can look identical or very similar, even for very different amino acids. In recent years, advances in computer technology, accompanied by the increased use of NMR spectroscopy in drug design and structural genomics initiatives, have created a push for the automation of various steps in the NMR structure determination process.[13] These automated methods decrease the time needed to complete the assignment, and attempt to minimize the risk of human error and subjectivity. A wide variety of algorithms and software packages for the assignment of backbone chemical shifts exist,[14,15] including GARANT,[16] AutoAssign[13,17] and MARS.[18]

RELATED WORK

GARANT is an example which has three key elements in an algorithm.[16] GARANT starts by matching observed peaks to expected peaks in the backbone sequence, scoring these values, and then running through an optimization routine. This optimization is similar to simulated annealing, which makes repeated selections until a solution is reached. If the current selection decreases error, it becomes the new structure. If it is worse, a new selection is made based on probability.[13]

SPARKY uses a method called AUTOASSIGN, a heuristic, best-first mapping algorithm.[11] This program utilizes five basic steps. It first filters peaks and aligns resonances from different spectra.[17] Then, it groups these resonances based on their spin systems and identifies the amino acid types. Next, the already assigned segments are found and linked together into short chains. Finally, a solution is obtained.

MARS looks to optimize the quality of assignments overall.[18] This program breaks the assignment into segments, assigning up to five pieces into a chain and using them as units in assignment. Then,by combining secondary structure analysis with assignment, the most accurate model is selected and expanded upon. Lastly, MARS tests the chemical shifts for noise and reliability.

Programs such as GARANT, AUTOASSIGN, and MARS, have limitations. The best-first strategies employed by some of these programs may abandon a promising set too early and the optimum solution will not be selected. Many programs also experience difficulty with handling local minimums; not being able to identify these outliers in the assignment process can throw off the optimization process. Further issues also arise when programs try to assign larger protein chains, often needing days to finish assignment and still having very high error calculations. Even though these issues exist, these programs have paved the way for the automation of chemical shift assignments and made significant advances in the field of protein NMR spectroscopy.

Our algorithm utilizes elements from these existing programs, such as cost analysis and sorting amino acids into groups, but employs novel concepts from machine learning to accelerate the assignment process.

MACHINE LEARNING

Machine learning provides algorithms that learn from attributes in the input data to increase performance. Supervised learning, a field of machine learning, builds a model based on numerous data elements and their respective labels. The result is a mathematical model that predicts a label given a set of input attributes. Machine learning algorithms provide excellent solutions for building models that generalize well given large amounts of data with potentially many attributes by discovering patterns and trends in the data; a task that is often difficult or impossible by other means.

Machine learning offers a natural solution to the problem of determining amino acid type from NMR chemical shift values. The overlap in the normal range of C_α and C_β values among many amino acids makes it difficult to infer the type of residue based solely on chemical shift information. Machine learning algorithms offer a unique approach to this problem and achieve excellent accuracy.

There are several supervised machine learning algorithms that can be applied to improve automated assignment strategies. The J4.8 algorithm[19] builds a decision tree model. This means that the data is split based on a comparison to an attribute of the data, with a branch for each possible outcome of the test. At the end of the tree, the leaf, is the predicted label. In our case this is the amino acid type. To classify a new value, a datapoint begins at the root of the tree and moves through until a leaf is encountered. The encountered leaf is the predicted value for that datapoint. To construct the tree, the attribute test used at each branch is the one that partitions the set in the most useful manner.

The Logistic Model Tree, or LMT,[20] is another tree-based algorithm. In contrast to the J4.8 algorithm, LMT constructs a tree with logistic regression functions at each branch rather than an attribute test. Logistic regression attempts to model class probabilities with linear functions. The weights used for each function can be learned to split each class in a way somewhat analogous to the split in J4.8 described above.

The Decision Table algorithm[21] is comprised of a set of features and a set of labeled data. To classify a new datapoint, the set of labeled data is searched for a match with the new point, considering only the features in the feature set. If no matches are found, the majority class of the labeled data is used. Otherwise, the majority class of the matches is used.

PROGRAM DEVELOPMENT

Our goal is to create an automated program for the assignment of protein backbone chemical shifts that can deliver quality results in a small amount of time without the use of a supercomputer. Our program implements group sorting, machine learning, filtered amino acid selection, and careful cost calculations. The algorithm completes resonance assignment in six steps: (1) the NMR chemical shifts and the protein sequence are used to fill data structures; (2) the protein sequence is processed and empty data structures are initialized for missing data; (3) model created utilizing machine learning is employed to assign possible amino acid types to each spin system; (4) filtering is applied to locate chemical shift values that could potentially be assigned to residue 1 in the protein sequence; (5) the search process continues and identifies the closest-matching sequence until all chemical shift data is assigned; (6) the best solution is recorded and the process is terminated.

Before the algorithm begins the assignment process, machine learning is used to build a model for predicting amino acid type. After the model is trained, the pre-processing component of the algorithm (steps 1 to 3) begins. Pre-processing is where our research has made significant advances; by predicting amino acid types with machine learning algorithms, the assignment time is decreased significantly. Machine learning allows the data to be filtered, significantly reducing the search space and consequently the running time of the algorithm. The algorithm is then able to intelligently search the remaining possibilities for the best assignment (**steps 4** to **6**).

METHODS

Machine Learning Data Collection

The training dataset for the machine learning algorithm was obtained from the Biological Magnetic Resonance Bank (BMRB), a database of NMR chemical shifts hosted by the University of Wisconsin-Madison.[22] We initially identified 9,736 datasets containing chemical shifts for the C_α and C_β resonances of 689,977 residues. In order to improve both accuracy and generalization, and to prevent the algorithm from fitting extraneous data, it was necessary to remove outliers from the published datasets. By inspecting statistics available on the BMRB site, we excluded chemical shift values outside three standard deviations of the mean for each amino acid type. This gave us 681,363 pairs of C_α and C_β values to use for training.

Pre-processing

Step 1 of our algorithm consists of reading the chemical shift values and the protein amino acid sequence from an input file. We use the C_α and C_β values for each pair of i and $i-1$ residues to create an object we will refer to as a *tile*. A tile holds all the available chemical shift information corresponding to a single residue in the protein sequence to be assigned.

Step 2 of the algorithm converts the primary protein sequence into C_α and C_β chemical shift values, using statistics available in BMRB. These statistics provide average chemical shift values for each of the twenty common amino acids found in proteins. For example, we assign alanine a C_α chemical shift of 53.19 ppm and a C_β chemical shift of 18.96 ppm. Next, the algorithm searches the protein sequence for prolines. As prolines lack H-N spin systems, HNCACB and CBCA(CO)NH experiments do not provide C_α and C_β chemical shifts for this residue. Special tiles, designed to specifically identify the proline residue, are created to handle this case. Identifiers in the proline tiles ensure that these tiles are placed only when the corresponding residue in the protein sequence is a proline. The identifier limits the number of possibilities where the tile can be assigned. The length of the protein sequence is then compared to the total number of tiles created thus far. If fewer tiles exist than the overall number of residues, blank tiles are created to fill the difference. Blank tiles can fit in any location in the assignment. However, large amounts of missing data will deteriorate the algorithm's performance as every blank tile would need to be checked for the best fit in every position in the assignment.

Step 3 assigns possible amino acid types to each tile. The C_α and C_β values for residue i in each tile are processed by our machine learning model, producing a list of probabilities that a tile represents a certain amino acid. The probabilities correspond to confidence levels used for filtering during the assignment process.

Preprocessing the dataset takes a minimal amount of time (less than a second on a standard laptop) and drastically reduces the time required to assign the chemical shifts without affecting accuracy. The search for the optimal assignment then begins.

The Search

The algorithm initiates an intelligent search through filtered combinations of all possible chemical shifts. The search begins with **step 4** by placing the first tile, which is selected based on the filtering process. Only tiles that could correspond to the first amino acid based on a confidence level threshold (0.4% match or better) are placed at residue one. The threshold was chosen by determining the lowest probability for a correct amino acid classification.

At this point, a "cost" of assignment is generated. The cost of placing a tile consists of two parts: (1) the difference between the tile's residue $i-1$ values and the previous tile's residue i values, and (2) the difference between the residue i values and the values predicted from the protein sequence. In the case of blank and proline tiles, a fixed cost is added instead of the above calculation. This value is set to ensure a blank tile is neither the best nor worst option, allowing the blank and non-blank tiles to be considered simultaneously by the algorithm. When the first tile is placed, the cost is based solely on a comparison to the protein sequence, and the search moves on to **step 5**.

In **step 5**, the algorithm selects the assignment with the lowest cost to continue the search process. The solution is reached when all tiles have been placed in an assignment and the solution has the lowest cost. If the assignment is not a solution, the amino acid type of the next residue in the protein sequence is retrieved. Any unplaced tile that corresponds to that amino acid type with a confidence level above the threshold is placed at the next location in the sequence. In the special case that the next amino acid type is proline, only the special proline tiles are considered for placement. The cost is then adjusted to include the newly placed tile. **Step 5** is repeated until a solution has been reached.

In the final step the search records the solution. The solution assignment, along with the performance (the number of possible assignments searched) is output. Then the algorithm terminates.

RESULTS

The chemical shift dataset used in this study consisted of C_α and C_β values for the 62-residue long C-terminal domain of the Tfg1 subunit of the yeast transcription factor TFIIF. The chemical shifts were previously obtained by Kilpatrick et al. from HNCACB and CBCA(CO)NH experiments, and manually assigned to 99% completeness.[23] The chemical shift dataset was divided into sub-sections, randomized, and used for algorithmic analysis.

The results of assigning this dataset with different filtering methods are shown in.**Figure 2** Each of the tests described in **Figure 2** follows a smooth trend, as opposed to a highly irregular trend which would indicate an impediment in the processing and assignment of data. A comparison of the methods indicates that our filtering process results in a significant decrease in number of generated nodes compared to an unfiltered generic search algorithm similar to a predecessor to this algorithm.[24] Since the most time-consuming part of the search is node generation, there is a direct correlation between assignment time and the number of nodes generated. With a 2.3 GHz Intel Core i7 processor and 8 GB or RAM, generating 7036 nodes requires approximately 1 second. The graph in **Figure 2** indicates that the LMT machine learning algorithm has the best performance, outperforming the unfiltered method by almost two orders of magnitude, without loss of assignment accuracy. LMT was able to assign all 62 residues within forty-five minutes, whereas the other filters were unable to complete the full assignment within a 12 hour time limit. The LMT model not only accelerates assignment, but also allows for larger datasets to be assigned in the same amount of time.

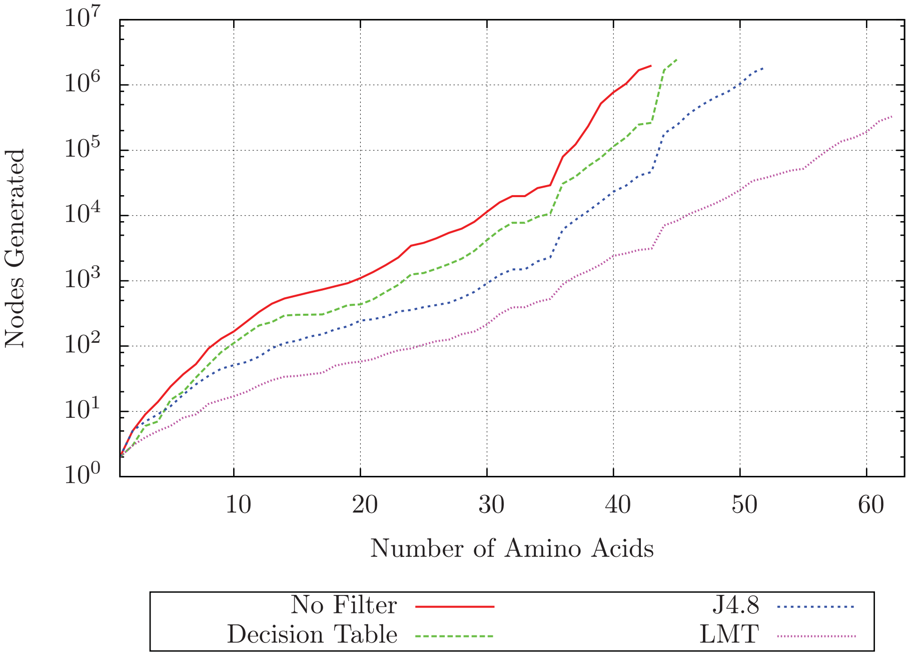


Figure 2. Impact of filtering methods on assignment time. Number of nodes (y-axis) is plotted as a function of number of residues (protein sequence length) used in the assignment (x-axis). The search without machine learning filtering is depicted in red. The other three groups (DecisionTable, J4.8 and LMT) are the machine learning algorithms that were used for filtering, e.g. DecisionTable used a Decision Table for the filtering process.

The impact of proline identification on our algorithm's performance is shown in **Figure 3**. In the sequence of the studied protein, residue 33 is a proline, for which chemical shifts are missing. The same unfiltered and LMT data from **Figure 2** is plotted for comparison. The large jump in assignment time between residues 32 and 33 shows the impact of missing data on performance without filtering and using the LMT model. If the algorithm identifies and handles prolines as a special case, only one more node is generated for the 33-residue long sequence. However, if the proline is not dealt with separately, the algorithm's performance is significantly impacted. Without proline checking, the proline tile is placed in every position in the assignment. The result is a major increase in the branching factor that leads to the jump observed in **Figure 3**. With proline checking, prolines are no longer problematic to the assignment process. This indicates that our algorithm can accurately obtain assignments with reasonably fast assignment times even when chemical shift data is missing.

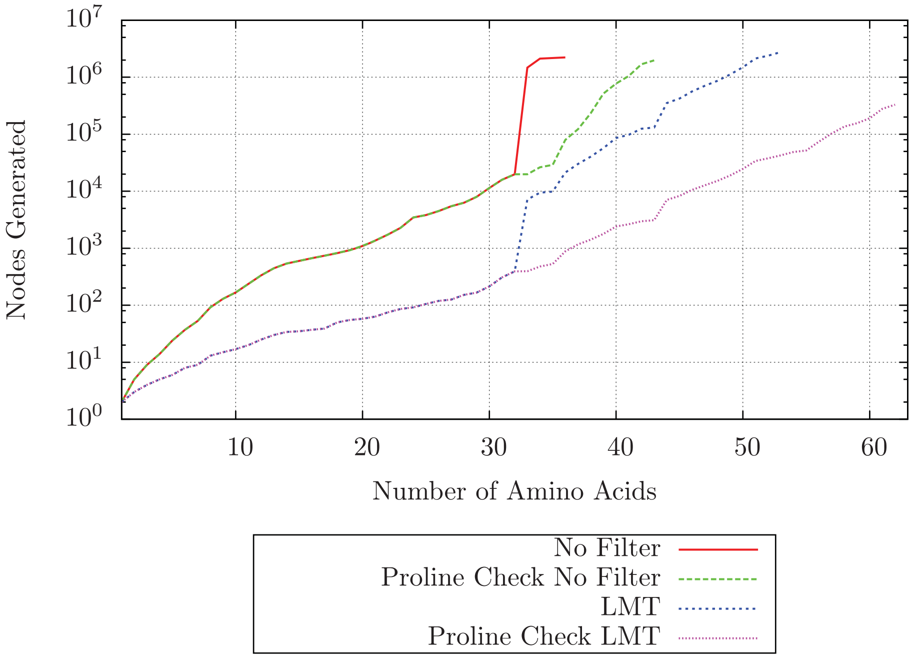


Figure 3. Impact of proline identification on assignment time. Number of nodes (y-axis) is plotted as a function of number of residues used in the assignment (x-axis). Red shows the search without proline checking and without machine learning filtering. Similarly, the search with proline checking without machine learning filtering is depicted in green. The search without proline checking using LMT machine learning filtering is depicted in blue. Utilizing proline checking and LMT machine learning filtering is depicted in pink.

In order to evalueate our algorithm's performance, and effectiveness, we compare the number of nodes generated in our search to the total possible number of nodes. To calculate the total possible number of nodes, we first consider n to be the number of amino acids. The search space will start with a single node, the root node, which has no tiles placed. Call this level 0. The next level of the search space, level 1, will have n nodes, one for each of the amino acids that could go in the first spot. We now have $n + 1$ total nodes in our search space. Off of each node in level 1, there will be $n - 1$ nodes. So level 2 will be $n(n - 1)$ nodes in size. We can rewrite this in the form

$$n(n-1) = \frac{n!}{(n-2)!} \qquad \textbf{Equation 1.}$$

We can extend this to the ith level by replacing 2 in the denominator with i since there will be $(n - i)$ tiles left to place at the ith level. By doing so, the number of nodes at a given level is

$$n(n-1)\cdots(n-i+1) = \frac{n!}{(n-i)!} \qquad \textbf{Equation 2.}$$

To get the total search space, we sum the levels from $i = 0$, the root level, until $i = n$, the level which all amino acids are placed. This yields

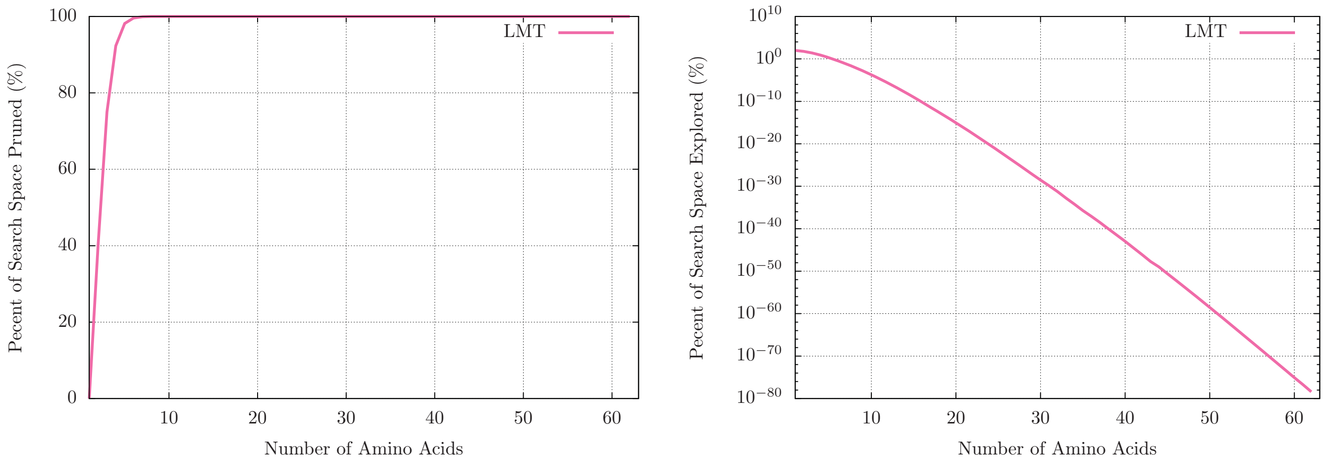


Figure 4. The plot on the right shows the percent of the search space not explored by the LMT model. On the left is a plot showing the percent of the Search Space that was explored by the LMT model. DecisionTable, J48 and No Filter all behave in a similar matter and are not shown for simplicity.

$$Search\ Space = \sum_{i=0}^{n} \frac{n!}{(n-i)!} \qquad \textbf{Equation 3.}$$

We can now evaluate the effectiveness of our algorithm by comparing the number of nodes generated for each of our machine learning filters to the total possible number of nodes. In **Table 1**, we have shown the results for 43 amino acids (the largest sequence that the “No Filter” for proline detection completed). The search space given by **Equation 3** for $n = 43$ is $1.64 \cdot 10^{53}$ nodes. The table shows that no filter explored a little over 10^{-45}% of the search space. DecisionTable, J48, and LMT each preformed approximately an order magnitude better than the one before it, with LMT exploring $1.89 \cdot 10^{-48}$% of the search space.

Filtering Model	Nodes Generated	Percent of Search Space
No Filter	1,977,233	$1.20 \cdot 10^{-45}$%
DecisionTable	262,961	$1.60 \cdot 10^{-46}$%
J48	46,372	$2.82 \cdot 10^{-47}$%
LMT	3,105	$1.89 \cdot 10^{-48}$%

Table 1. A comparison of the filtering algorithms with respect to the search space. The results shown are for 43 amino acids, the largest set completed by the no filter algorithm.

Figure 4 shows the relationship of the number of amino acids assigned and the amount of the search space explored by the LMT model. DecisionTable, J48, and No Filter all show the same general trend. The trend shows that the more amino acids assigned, the more of the search space is removed due to filtering. Due to the factorial growth of the search space, even 10^{-79}% of the search space is over $300,000$ nodes in size when taken for $n = 62$. As we can see, even an extremely small percentage of the search space can have a major impact on the assignment process. The addition of filtering to our algorithm removes large portions of the search space and drastically reduces assignment times.

Our algorithm with LMT filtering and proline checking can complete the 62-residue assignment in approximately 40 minutes on a 2.3 GHz Intel Core i7 processor with 8Gb of RAM. The LMT model with proline checking can complete the assignment of a 43-residue long sub-sequence in less than 1 second, compared to 27, 29 and 30 seconds without filtering, using a Decision Table for filtering and using the J4.8 model for filtering, respectively. In all cases, the solution from the automated algorithm is identical to the manual assignment, indicating that our algorithm is both fast and accurate.

CONCLUSIONS AND FUTURE DIRECTIONS

Our algorithm has made significant advances in the field of automated assignment of protein backbone chemical shifts. We implemented machine learning to filter NMR data in order to reduce the branching factor in a search-based algorithm. This increased our assignment rate by approximately three orders of magnitude, as seen in **Figure 3**, while still maintaining the accuracy of the solution.

The use of proline checking and the utilization of machine learning to filter data has shown to be extremely effective in accelerating the assignment. Our algorithm has successfully completed assignments of up to 62 amino acids in 40 minutes without the use of a supercomputer. The same sequence took several days to manually assign. If allotted more time, more residues from a larger dataset could be assigned.

One of our main focuses for the future is handling missing data in a more efficient manner. By examining characteristics of the amino acids in the sequence, we hope to predict where missing data will end up in the final assignment. This would reduce the number of assignments attempted by greatly reducing the number of nodes in the search. Furthermore, we will work to improve the overall performance of our algorithm by optimizing cost calculations and investigate assignment via parallel processing.

We are currently measuring the backbone chemical shifts of a protein previously uncharacterized by NMR spectroscopy. This new dataset will include additional chemical shifts that can be used in the assignment process, such as the backbone carbonyl and H-alpha values. This data will be used to further validate and improve our algorithm. Since the cost calculation is crucial to the effectiveness of our algorithm, additional chemical shifts may prove invaluable to the success of the algorithm for longer protein sequences or incomplete datasets. We are also investigating methods of predicting the final cost of an assignment in order to remove unrealistic assignments early on. Having this information available will help optimize our cost calculation, resulting in a further decrease in assignment times.

ACKNOWLEDGEMENTS

The authors of this paper thank Drake University for being a conducive space to research, and John Emmons for initiating this research.

REFERENCES

1. Wüthrich, K. (1990), Protein structure determination in solution by NMR spectroscopy., *The Journal of Biological Chemistry 265*, 22059–22062.

2. Linge, J. P., Habeck, M., Rieping, W., Nilges, M. (2003), ARIA: Automated NOE assignment and NMR structure calculation, *Bioinformatics.*

3. Wishart, D., Nip, A. (1998), Protein chemical shift analysis: a practical guide., *Biochemistry and Cell Biology 76*, 153–163.

4. Nagayama, K. (1988), Three-dimensional NMR spectroscopy, US Patent 4,789,832.

5. Kay, L. E., Ikura, M., Tschudin, R., Bax, A. (1990), Three-dimensional triple-resonance NMR spectroscopy of isotopically enriched proteins, *Journal of Magnetic Resonance (1969) 89*, 496–514.

6. Cavanagh, J., Fairbrother, W. J., III, A. G. P., Rance, M., Skelton, N. J. (2007), *Protein NMR Spectroscopy*, second edition ed., Academic Press: Burlington.

7. Bax, A. (2011), Triple resonance three-dimensional protein NMR: Before it became a black box, *Journal of Magnetic Resonance 213*, 442–445.

8. Wittekind, M., Mueller, L. (1993), HNCACB, a high-sensitivity 3D NMR experiment to correlate amide-proton and nitrogen resonances with the alpha-and beta-carbon resonances in proteins, *Journal of Magnetic Resonance, Series B 101*, 201–205.

9. Rios, C. B., Feng, W., Tashiro, M., Shang, Z., Montelione, G. T. (1996), Phase labeling of C- H and C- C spin-system topologies: Application in constant-time PFG-CBCA (CO) NH experiments for discriminating amino acid spin-system types, *Journal of Biomolecular NMR 8*, 345–350.

10. Delaglio, F., Grzesiek, S., Vuister, G., Zhu, G., Pfeifer, J., Bax, A. (1995), NMRPipe: A multidimensional spectral processing system based on UNIX pipes, *Journal of Biomolecular NMR 6*, 277–293.

11. Goddard, T., Kneller, D. SPARKY 3, University of California, San Francisco, *https://www.cgl.ucsf.edu/home/sparky/* (Accessed February 2015).

12. Wang, Y., Jardetzky, O. (2002), Probability-based protein secondary structure identification using combined NMR chemical-shift data., *Protein science : a publication of the Protein Society 11*, 852–861.

13. Moseley, H. N. B., Monleon, D., Montelione, G. T. (2001), Automatic determination of protein backbone resonance assignments from triple resonance nuclear magnetic resonance data, Elsevier, 339, pp 91–108.

14. Güntert, P. (2009), Automated structure determination from NMR spectra, *European Biophysics Journal 38*, 129–143.

15. Emmons, J., Johnson, S., Urness, T. (2013), Automated Assignment Of Backbone NMR Data using Artificial Intelligence, *http://micsymposium.org/mics_2013_Proceedings/submissions/mics20130_submission_26.pdf* (Accessed February 2015).

16. Bartels, C., GÃijntert, P., Billeter, M., WÃijthrich, K. (1997), GARANT - A General Algorithm for Resonance Assignment of Multidimensional Nuclear Magnetic Resonance Spectra, *Journal of Computational Chemistry 18*, 139–149.

17. Zimmerman, D. E., Kulikowski, C. A., Huang, Y., Feng, W., Tashiro, M., Shimotakahara, S., ya Chien, C., Powers, R., Montelione, G. T. (1997), Automated Analysis of Protein NMR Assignments Using Methods from Artificial Intelligence, 269, *"J Mol Bio"*, pp 592–610.

18. Jung, Y., Zweckstetter, M. (2004), Mars-robust automatic backbone assignment of proteins, *Journal of Biomolecular NMR 30*, 11–23.

19. Quinlan, R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers: San Mateo, CA.

20. Landwehr, N., Hall, M., Frank, E. (2005), Logistic Model Trees, *Machine Learning 95*, 161–205.

21. Kohavi, R. (1995), The Power of Decision Tables, 8th European Conference on Machine Learning, pp 174–189.

22. Ulrich, E. L. et al. (2008), BioMagResBank., *Nucleic Acids Research 36*, 402–408.

23. Kilpatrick, A. M., Koharudin, L. M., Calero, G. A., Gronenborn, A. M. (2012), Structural and binding studies of the C-terminal domains of yeast TFIIF subunits Tfg1 and Tfg2, *Proteins: Structure, Function, and Bioinformatics 80*, 519–529.

24. Emmons, J., Venzke, J., Johnson, P., Davis, R., Roth, K., Mascharka, D., Robison, L., Urness, T. (2014), Accelerating Biomolecular Nuclear Magnetic Resonance Assignment with A*, *http://www.micsymposium.org/mics2014/ProceedingsMICS_2014/mics2014_submission_35.pdf* (Accessed February 2015).

ABOUT THE STUDENT AUTHORS

Joel Venzke is a senior pursuing a B.S. in Physics, Computer Science, and Mathematics. Outside of conducting research, Joel is president of the Drake University chapter of the Society of Physics Students and vice president of the math club. He also works as an award winning photojournalist for Drake University Communications and The Times-Delphic. Joel plans to pursue a Ph.D. in computational science.

David Mascharka is a junior seeking a B.S. in Computer Science and Mathematics and B.A. in Philosophy. David also pursues his love for mathematics as the president of the math club at Drake University, where is he able to hone his leadership skills. His research interests include machine learning and mobile computation.

Paxten Johnson is on track to graduate in the Spring of 2016 with a B.S. in Physics, Computer Science, and Mathematics. Aside from conducting this research for the past two years, she has been involved with Air Force ROTC, Delta Gamma Fraternity, and the Drake Honors Program. She hopes to use the knowledge and experience gained from this research to help her advance into the military intelligence branch of the Air Force.

Rachel Davis is currently on the path to obtaining a B.S. in both Computer Science and Mathematics, with a minor in Data Analytics. This is the second year conducting this sequencing research. Her research includes not only this primary structuring of proteins, but also a collaborative tertiary structuring algorithm. Rachel is also the president of the Women in Mathematics and Computer Science and an active member of the Math Club.

Katie Roth is pursuing a B.A. in Mathematics and Computer Science. She has been working on this research for two years, and enjoys it immensely. Katie is the vice president of the Women in Mathematics and Computer Science and an active member of Math Club.

Leah Robison is on her way to graduate with a B.S. in Environmental Science and a minor in Computer Science. Apart from studying abroad in Denmark for a semester, she has been involved with this research group for the past two years. She hopes to apply the knowledge gained from this research to future studies in the field of Environmental Science.

PRESS SUMMARY

Manually assigning nuclear magnetic resonance data to a protein sequence is time consuming and error prone. Although current algorithms have made advances in this area, a student research group at Drake University is improving the process by utilizing machine learning to identify amino acids before assignment. Using both old and new methods has resulted in an algorithm that is fast and accurate.