

ASSIGNMENT OF PROTEIN BACKBONE RESONANCES USING CONNECTIVITY,
TORSION ANGLES AND $^{13}\text{C}\alpha$ CHEMICAL SHIFTS

by

LAURA C. MORRIS

(Under the Direction of James H. Prestegard)

ABSTRACT

A computer program is presented that will return the most probable sequence location for a short connected set of residues in a protein given just $^{13}\text{C}\alpha$ chemical shifts ($\delta\text{C}\alpha$) and data restricting the phi and psi backbone angles. Data taken from both the BioMagResBank and the Protein Data Bank were used to create a probability density function (PDF) using a multivariate normal distribution in $\delta\text{C}\alpha$, ϕ , and ψ space for each amino acid. Extracting and combining probabilities for particular amino acids in a short proposed sequence yields a score indicative of the correctness of the proposed assignment. The program is illustrated using several proteins for which structure and $^{13}\text{C}\alpha$ chemical shift data are available.

INDEX WORDS: assignment, chemical shift, probability density function, protein backbone, structural genomics, torsion angle, nuclear magnetic resonance spectroscopy

ASSIGNMENT OF PROTEIN BACKBONE RESONANCES USING CONNECTIVITY,
TORSION ANGLES AND $^{13}\text{C}\alpha$ CHEMICAL SHIFTS

by

LAURA C. MORRIS

B.S., Georgia State University, 1996

A Thesis Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of
the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2003

© 2003

Laura C. Morris

All Rights Reserved

ASSIGNMENT OF PROTEIN BACKBONE RESONANCES USING CONNECTIVITY,
TORSION ANGLES AND $^{13}\text{C}\alpha$ CHEMICAL SHIFTS

by

LAURA C. MORRIS

Major Professor: James H. Prestegard

Committee: Jeffrey L. Urbauer
Lionel A. Carreira

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2003

DEDICATION

To my husband who has made all this possible.

ACKNOWLEDGEMENTS

First of all I would like to thank Dr. Homayou Valafar for all his assistance during the course of my graduate work. His knowledge and insight has been invaluable. In addition he has been a great friend and confidant. Hoday, you would be a great professor in any university and a true asset to your students. I wish you and Teresa the best of everything in the future.

For his constant good humor and positive attitude I must thank Dr. Jorge Gonzalez-Outeirino. He also made many helpful suggestions during the course of my writing. Any acknowledgements would be incomplete without expressing the pleasure I've had working with many of the past and present members of the Prestegard lab. Their camaraderie has been much appreciated over the years.

Lastly, and most importantly, I must express my sincere gratitude to my professor, Dr. James H. Prestegard. Having the right professor is everything. It's hard to over emphasize his importance. While I can certainly fail on my own there is no way I could have succeeded without his guidance. His breadth and depth of knowledge are beyond compare. The respect and kindness given to everyone is truly rare today, especially someone of his high stature. He is a great scientist and a good person. I can't even say half of that to more than a handful of people I've known throughout my life. But even without these qualities, though they cannot really be separated, it is the opportunity and freedom he provides that is so valuable. My sincere hope is that anyone from the Prestegard lab who reads this realizes what they have and works hard to take full advantage of what they've been given.

TABLE OF CONTENTS

	Page
DEDICATION	iv
ACKNOWLEDGEMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
ABBREVIATIONS	xi
 CHAPTER	
1 INTRODUCTION	1
1.1 Assignment of Protein NMR Spectra	1
1.2 Current Methodologies to Assist Assignment of Protein NMR Spectra	3
1.3 The Use of Dipolar Couplings to Determine Structure	5
1.4 The Use of $\delta C\alpha$, ϕ and ψ in Assignment of Protein NMR Spectra	6
1.5 Density Estimation used to create a Probability Density Function	7
1.6 Modifications and Additions to SEASCAPE	8
1.7 Combination of SEASCAPE with RDC Data	9
2 DEVELOPMENT OF SEASCAPE	14
2.1 Data Correlation	14
2.2 Probability Density Function Calculations	17
2.3 Calculation of Probabilities Using ($\delta C\alpha$, ϕ , ψ) or ($\delta C\alpha$, $^3J_{\text{HNHA}}$)	20
2.4 Calculation of Probabilities with Known Structure	21

2.5 Use of Residual Dipolar Couplings	21
2.6 Calculation of Probabilities with Residual Dipolar Couplings	23
3 APPLICATIONS	30
3.1 Distribution of Data Points	30
3.2 Assignment of Rubredoxin Fragments of Various Lengths	32
3.3 Assignment of Fragments for a Selection of Proteins	33
3.4 Correlation of Success with Amino Acid Composition and Secondary Structure	35
3.5 Raw Scores and Confidence	37
3.6 Correct versus Incorrect Connectivities in Fragments	37
3.7 Combining RDCs with SEASCAPE	38
4 CONCLUSIONS	76
4.1 The Current State of SEASCAPE	76
4.2 Data and Statistical Analysis	77
4.3 Incorporation of SEASCAPE into New and Existing NMR Programs	78
4.4 Future Modifications to SEASCAPE	78
4.5 Conclusion	79
REFERENCES	80
APPENDICES	82
A BMRB/PDB Sequence Alignment Program	82
B BMRB/PDB Data Extraction Program	88
C The SEASCAPE Program	92

LIST OF TABLES

	Page
Table 1.1: Traditional NMR experiments for protein structure determination.....	11
Table 1.2: Residual dipolar coupling experiments for protein structure determination	12
Table 2.1: Number of data points used for each amino acid	25
Table 3.1: Comparison of assignment results for 1M2Y	41
Table 3.2: Assignment results for a set of 10 proteins.....	42

LIST OF FIGURES

	Page
Figure 1.1: Protein backbone structure	13
Figure 2.1: NMR and heavy atom definitions of the protein backbone torsion angle ϕ	26
Figure 2.2: Histograms with different bin sizes.....	27
Figure 2.3: Histograms with different origins.....	28
Figure 2.4: Example assignment of a fragment to a position in a protein sequence.....	29
Figure 3.1: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for alanine	43
Figure 3.2: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for arginine	44
Figure 3.3: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for asparagine	45
Figure 3.4: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for aspartate	46
Figure 3.5: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for cystine	47
Figure 3.6: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for glutamate	48
Figure 3.7: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for glutamine	49
Figure 3.8: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for glycine	50
Figure 3.9: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for histidine	51
Figure 3.10: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for isoleucine	52
Figure 3.11: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for leucine	53
Figure 3.12: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for lysine	54
Figure 3.13: Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for methionine	55

Figure 3.14: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for phenylalanine	56
Figure 3.15: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for proline	57
Figure 3.16: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for serine	58
Figure 3.17: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for threonine	59
Figure 3.18: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for tryptophan	60
Figure 3.19: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for tyrosine	61
Figure 3.20: Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for valine	62
Figure 3.21: Assignment results for the protein 1M2Y	63
Figure 3.22: Assignment results for the protein 1C0V	64
Figure 3.23: Assignment results for the protein 1QJT	65
Figure 3.24: Assignment results for the protein 1SYM	66
Figure 3.25: Assignment results for the protein 1IRF	67
Figure 3.26: Assignment results for the protein 1DMO	68
Figure 3.27: Assignment results for the protein 1CDC	69
Figure 3.28: Assignment results for the protein 1EZA	70
Figure 3.29: Assignment results for the protein 1AZM	71
Figure 3.30: Assignment results for the protein 1L6N	72
Figure 3.31: Value of ($\delta C\alpha$, ϕ , ψ) in identifying an isolated single amino acid	73
Figure 3.32: Correct assignment probability as a function of raw score	74
Figure 3.33: RDC assisted assignment in SEASCAPE	75

ABBREVIATIONS

$^3J_{\text{HNHA}}$	$\text{H}^{\text{N}}\text{-H}^{\alpha}$ three bond homonuclear coupling constant
$\delta\text{C}\alpha$	$^{13}\text{C}\alpha$ chemical shift
$\delta\text{C}\beta$	$^{13}\text{C}\beta$ chemical shift
φ	$\text{N}^{\text{H}}\text{-C}^{\alpha}$ torsion angle (see Figure 1.1)
ψ	$\text{C}^{\alpha}\text{-C}'$ torsion angle (see Figure 1.1)
Hz	hertz
KDE	kernel density estimation
NMR	nuclear magnetic resonance
NOE	nuclear Overhauser effect
NOESY	nuclear Overhauser enhancement spectroscopy
PDF	probability density function
ppm	parts per million
RDC	residual dipolar coupling
REDCAT	REsidual Dipolar Coupling Analysis Tool
rmsd	root mean square deviation
SEASCAPE	SEquential Assignment by Structure and Chemical shift Assisted Probability Estimation

CHAPTER 1

INTRODUCTION

Nuclear magnetic resonance (NMR) spectroscopy is one of the most powerful tools in molecular structure determination. One of the first steps in analysis of protein structure by NMR is the assignment of resonances in the NMR spectra of proteins to specific amino acid residues. This is the most time consuming and frustrating step in determining the structure of a protein. It may take as long as several weeks to gather all the appropriate spectra for a sample and it can take weeks more to make the assignments. Therefore, it is easy to understand the pressure to find ways to simplify and accelerate the assignment process. This thesis will describe a new methodology to assign resonances of a short connected set of amino acid residues (of arbitrary length) given only their $^{13}\text{C}\alpha$ chemical shifts ($\delta\text{C}\alpha$) and data restricting backbone torsion angles (ϕ, ψ) of the amino acids (Figure 1.1). It meshes well with new residual dipolar coupling (RDC) approaches to structure determination (Tian et al., 2001) and facilitates use of NMR data in the drug design process (Stockman and Dalvit, 2002).

1.1 Assignment of Protein NMR Spectra

In the 1980's protein NMR spectroscopy was just being developed and most experiments were based on the excitation of hydrogen. At this time the most powerful experiment an NMR spectroscopist had was the two dimensional (2D) nuclear

Overhauser experiment (NOESY; Wagner et al., 1981; Wüthrich, 1986). This experiment provides a means of determining the distance between any two hydrogens typically no more than ~ 5 Å apart. Since the intensity of a cross-peak for short mixing times is inversely proportional to the sixth power of the distance between the hydrogens one may determine an approximate distance between the two hydrogens. The presumption is that one knows the chemical shifts of each pair of protons. Also, because of the steep distance dependence and consequent short range of interaction, the most important data for constraining the fold of the protein comes most often from side chain to side chain contact. Therefore one is required to fully assign the protein spectra to make use of an NOE approach. Despite this high barrier, the NOE experiment has been the basis for structure determination for many years.

In the 1990's isotopic labeling became commonplace thanks to the widespread availability of isotopically enriched starting materials and advancements in instrumentation. Usually ^{15}N enriched ammonium salts and/or ^{13}C enriched glucose are used during protein expression to incorporate these NMR active nuclei into the protein. This labeling allows experiments that utilize these nuclei to be performed; thereby expanding the capabilities of the spectroscopist. Typically proteins are both ^{15}N and ^{13}C labeled. This doubly labeled protein may then be used in an entire suite of two and three dimensional (3D) experiments which transfer magnetization along a portion of the backbone of a protein, allowing connectivities to be made between atoms and hence between amino acids in the protein. Such innovations have substantially facilitated the assignment of backbone atoms, in comparison with previous methodologies. Making side chain assignments is often a subsequent time consuming step.

All the advancements so far have come with a cost. Labeling the protein allows the spectroscopist to run many more experiments than before and obtain a more reliable and complete structures but it increases the expenses of protein production and requires more time to collect and process the data. This leads to a deluge of data with lists of peaks from spectra that can easily lead to thousands of numbers to process.

1.2 Current Methodologies to Assist Assignment of Protein NMR Spectra

There are several programs that have been written to assist in the assignment of backbone resonances in protein NMR spectra. AutoAssign (Zimmerman et al., 1997), by Montelione's group, is the most developed and well known among these. Under ideal circumstances the program requires five 3D experiments: HNCO, HNCACB, HN(CO)CACB, HNCA and HN(CO)CA. In more complex cases other experiments may be necessary to confirm assignments. For reference, Table 1.1 lists these experiments along with the correlations observed, magnetization transferred, couplings and references. In order to make assignments the program relies heavily on connectivities between amino acids. Amino acid type identification is mostly dependant upon $^{13}\text{C}\beta$ chemical shifts ($\delta\text{C}\beta$). Although the authors claim an assignment rate as high as 96% on the 11 proteins they have tested, they have all been fairly small proteins, ranging in size from 6 to 18.7 kDa (Mosley and Montelione, 1999). Since many other programs (e.g., TATAPRO II (Atreya et al., 2002), PACES (Coggins and Zhou, 2003)) utilize the same basic approach and have the similar data requirements they will not be discussed further. For further information the interested reader is directed to a review article by Moseley (Moseley and

Montelione, 1999). The most notable fact is that the assignment procedure requires ^{13}C and ^{15}N labels and a large number of experiments to be run.

Other programs emphasize the comparison of calculated to experimental shifts. Although a more manual than automated approach, these do have benefits. The greatest variety and simplicity can be seen in the collection of programs developed by the Wishart lab (Wishart et al., 1997; Neal et al., 2003). Chemical shifts from ^1H , ^{13}C , and ^{15}N may be calculated quickly and easily from the Wishart web site using his web based interfaces (<http://redpoll.pharmacy.ualberta.ca>). And although he has made chemical shift prediction easy to perform, assignment is not always straightforward, and in many cases relies on having $\delta\text{C}\beta$ data.

There are many applications which could benefit by enhancing the speed or accuracy of making assignments. Reducing the quantity of data required would also be welcome. Traditional manual assignment methods involving tedious assignment of hundreds or thousands of peaks could be streamlined with better graphical interfaces. Also, making an automated or semi-automated approach more practical by lowering data requirements would be advantageous. The latter would be especially helpful in cases where the protein has a short life span and multiple samples have to be made in order to obtain all the data traditionally required. The ability to provide a confidence level in examining a potential assignment would also give researchers another means by which to judge the progress and reliability of their results. In cases where there are only sections of data available a tool to help determine the position of a fragment in the protein could provide a valuable aid to interpretation of related data such as ligand binding.

1.3 The Use of Dipolar Couplings to Determine Structure

While assignment is a prerequisite for NOE based structure determination, a new structure determination methodology has emerged in which the assignment process may not need to precede the structure determination process (Tian et al., 2001). This approach is based on residual dipolar couplings (RDCs) and amino acid connectivities. Briefly, RDCs are additions (either positive or negative) to existing scalar couplings between pairs of magnetic nuclei which occur due to partial alignment of a protein in a liquid crystalline solution. Since RDCs provide information on the orientation of internuclear vectors with respect to a magnetic field they are suited for structure determination, but do not impose the requirement of close contact that NOEs do. Hence focus can remain on backbones and reduced sets of data can be used. Combining adequate numbers of RDC data with amino acid connectivities for short peptide fragments allow the calculation of both backbone torsion angles (Figure 1.1). In cases where a nearly complete set of fragment structures and a preferred orientation of the fragment is obtained it is possible to assemble fragments into a complete protein. The data acquired are based on a small number of experiments (phase-modulated HSQC, soft HNCA-E.COSY, 2D IP-HSQC, Table 1.2) and only partial ^{13}C labeling is necessary. This usually means that no $\text{C}\beta$ or other side chain connectivities are established. Still, it is advantageous to assign the fragments to sequential positions when assembling the structure. However if we are to do this with no additional experiments then it is imperative to extract as much assignment information from the existing data as possible, for example $\text{C}\alpha$ chemical shifts.

1.4 The Use of $\delta C\alpha$, ϕ and ψ in Assignment of Protein NMR Spectra

Currently $\delta C\alpha$ data are used mainly to predict secondary structure once residue type assignments are made (Wishart and Case, 2001) and are not considered to be as definitive as $\delta C\beta$ in making residue specific assignments. In principle, type assignments might be facilitated if secondary structure dependence could be removed first. Although Wishart has grouped the $\delta C\alpha$ s for a particular amino acid into helix, β strand and coil, the utility of these shifts in making type assignments could be increased by developing a function which fully describes their dependence upon backbone torsion angles (ϕ , ψ). This torsion angle information is available from the RDC based fragment structure approach and could be used in a new assignment protocol. This thesis will describe the development of a proper description of the angular dependence of $C\alpha$ chemical shift data and its use in a new assignment protocol.

At the present, there are two databases which, when combined, would provide the raw data necessary to create a function which describes the angular dependence of $\delta C\alpha$ in terms of the two backbone torsion angles, ϕ and ψ . From the Protein Data Bank (PDB URL: <http://www.rcsb.org/pdb>; Berman et al., 2000), a repository for 3D biomolecular structures, it will be possible to obtain backbone torsion angles for proteins whose structure has been determined by either x-ray or NMR. A younger database, the BioMagResBank (BMRB URL: <http://www.bmrb.wisc.edu>; Doreleijers, 2003) contains $\delta C\alpha$ data for many proteins, most of which have a corresponding structure in the PDB.

After obtaining this data it will be necessary to transform it into a form which will describe the probability (P) that an entity is a particular amino acid given the $\delta C\alpha$ and a particular set of torsion angles. To avoid problems associated with discrete points it is

necessary to develop a description of the four dimensional space ($\delta C\alpha$, ϕ , ψ , P) such that each possible data point in the range of interest has a definite non-zero value. Non-zero values would allow the computation of probabilities for a string of several data points. An easy solution would be to create a histogram. However, if a histogram were used, the appropriate bin size and placement would have to be determined. In addition, if some regions were completely devoid of data the value would need to be artificially adjusted to some arbitrary non-zero value. Also, the presence of definite bin beginning and end points could possibly lead to inaccuracies due to normal variations (experimental errors) in the data. Ideally it would be desirable to have an analytical function which describes the dependence of $\delta C\alpha$ on ϕ and ψ and can take normal variations into account.

1.5 Density Estimation used to create a Probability Density Function

A probability density function (PDF), or probability distribution, is a function ($f(x)$) which describes the distribution of the independent variable (x). In cases where this function is unknown it is estimated using a process called density estimation. In cases where a general knowledge of the distribution is known (e.g. a normal distribution), only parameters constituting that distribution (μ and σ^2 in this example) need be determined. This is the parametric approach. However in our case no assumptions have been made regarding the distribution of the data sets, therefore we will use a more generalized nonparametric approach, in which the data is not assumed to belong to any particular parametric family of functions.

Once the PDF for each amino acid has been determined it can be used to assist in the assignment of chemical shifts in spectra in several related ways. A program named

SEASCAPE (SEquential Assignment by Structure and Chemical shift Assisted Probability Estimation) was written to use the PDFs created for the amino acids to assist in the assignment of spectra based on known connectivities between any number of resonances using their $\delta C\alpha$ and backbone torsion angles. The program has been modified to fit a set of different analyses problems. These modifications will be introduced in the following paragraphs and discussed further during the course of the thesis.

The program calculates the probability that a fragment is correctly placed at a particular location in the sequence. The amino acid type for each residue in the fragment is determined by a trial placement of the fragment along the sequence, and a probability for this amino acid type with a given $\delta C\alpha$, ϕ , ψ is extracted. The overall probability for a particular fragment placement is the product of the probabilities for each residue (Equation 1.1) in the fragment. The most probable position

$$\prod_{i=1}^n P(\delta C\alpha_i, \phi_i, \psi_i) \quad (1.1)$$

is determined by exhaustively matching the fragment at each possible position in the protein. The development of SEASCAPE will be described in chapter 2 of this thesis.

1.6 Modifications and Additions to SEASCAPE

While the initial impetus for the program was as an addition to the work done in this lab (Tian et al., 2001) on *de novo* protein structure determination, it is also possible to obtain backbone torsion angles using other experiments or deposited data. For instance, ϕ angles are often determined based on the $1H^N$ - $1H^\alpha$ homonuclear three bond coupling constants ($^3J_{HNHA}$) obtained from the HNHA experiment (Vüister and Bax, 1993), and cross-correlation experiments (Schwalbe et al., 2001) can be used to

determine ψ angles. But since the HNHA type experiments are more common and easier to perform, a modification to the program in which $^3J_{\text{HNHA}}$ values are used instead of ϕ and ψ values has been made. The PDFs for this data were created in a manner analogous to the ϕ , ψ data with the only difference being that the $^3J_{\text{HNHA}}$ values were derived from ϕ angles in the original data set.

Another case arises when a structure has previously been determined and the spectroscopist wishes to assign the NMR spectra of the protein for purposes such as binding site studies. In this case the researcher only needs to obtain lists of connected residues (fragments) from an experiment such as an HNCA experiment which provides connectivities between two or more $\delta\text{C}\alpha$. The torsion angles can be easily obtained from the existing crystal structure and combined with the protein's sequence to enable calculation of the most probable location of the various fragments. In this case the only change in the program was in the pairing of the data. The backbone torsion angles are already paired with the correct amino acid and position in the protein. So the only variable is the $\delta\text{C}\alpha$ as the connected $\delta\text{C}\alpha$ s are positioned at each possible position along the sequence. All of these related applications of SEASCAPE will be described in chapter 2 of the thesis.

1.7 Combination of SEASCAPE with RDC Data

$\text{C}\alpha$ chemical shifts are not the only possible source of assignment information when structures are known from other sources. RDC data can also be used directly in the assignment process. Since the orientation of a molecule in a liquid crystalline environment can be calculated based on a few (theoretically 5 but more realistically 8)

RDCs it is possible to back-calculate the remaining RDCs based on the calculated orientation of the protein using the program REDCAT (Residual Dipolar Coupling Analysis Software Tool; Valafar and Prestegard, 2003). This assumes that a structure already exists and that there are enough RDCs from assigned portions of the molecule to obtain the order parameters. These back-calculated RDCs are used as a means of comparison to experimentally obtained RDCs. Analysis of N-H RDCs measured for a fragment may be combined with any of the three variations of SEASCAPE described above. In the analysis, the experimental N-H RDC is compared to the back-calculated RDC for each RDC in the fragment. The position with the lowest product of root mean square deviations (rmsds) for the fragment is considered to be the most probable. This data is combined with the calculated probabilities from chemical shift and angular data to determine an overall most probable position. This application will be described in chapter 2 of the thesis.

Table 1.1. Traditional NMR experiments for protein structure determination. (Adapted from Cavanagh et al., 1996)

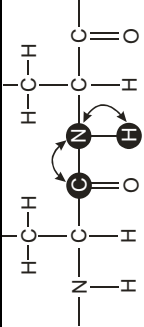
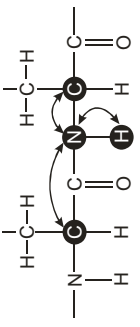
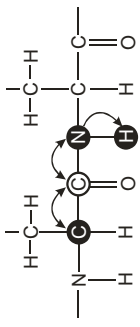
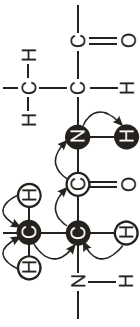
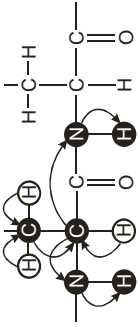
Experiment	Correlations observed	Magnetization transfer	J Couplings	Reference
HNCO	$^1H_i^N - ^{15}N_i - ^{13}C_{i-1}'$		$^1J_{NH}, ^1J_{NC'}$	Kay et al., 1990
HNCA	$^1H_i^N - ^{15}N_i - ^{13}C_i^\alpha$ $^1H_i^N - ^{15}N_i - ^{13}C_{i-1}^\alpha$		$^1J_{NH}, ^1J_{NC^\alpha}, ^2J_{NC^\alpha}$	Kay et al., 1990
HN(CO)CA	$^1H_i^N - ^{15}N_i - ^{13}C_{i-1}^\alpha$		$^1J_{NH}, ^1J_{NC'}, ^1J_{C^\alpha C'}$	Bax and Ikura, 1991
CBCA(CO)NH	$^{13}C_i^\beta - ^{13}C_i^\alpha - ^{15}N_{i+1} - ^1H_{i+1}^N$		$^1J_{CH}, ^1J_{C^\alpha C^\beta}, ^1J_{C^\alpha C'}, ^1J_{NC'}, ^1J_{NH}$	Grzesiek and Bax, 1992a
CBCANH	$^{13}C_i^\beta / ^{13}C_i^\alpha - ^{15}N_i - ^1H_i^N$ $^{13}C_i^\beta / ^{13}C_i^\alpha - ^{15}N_{i+1} - ^1H_{i+1}^N$		$^1J_{CH}, ^1J_{C^\alpha C^\beta}, ^1J_{NC^\alpha}, ^2J_{NC^\alpha}, ^1J_{NH}$	Grzesiek and Bax, 1992b

Table 1.2. Residual dipolar coupling experiments for protein structure determination.

(Taken from Tian et al., 2001)

Experiment	Correlations and Couplings* Measured for Structure Determination	Reference
phase-modulated HSQC	$^1D_{N_iH_i^N}$	Tolman et al., 1996
soft HNCA- E.COSY	$\delta C\alpha, {}^{13}C_i^\alpha - {}^{13}C_{i-1}^\alpha,$ $^1D_{C^\alpha H^\alpha}, {}^3D_{H_i^N H_i^\alpha}, {}^4D_{H_{i-1}^\alpha H_i^N}$	Weisemann et al., 1994
2D IP-HSQC	$^2D_{C_{i-1}H_i^N}, {}^2D_{N_iC_i}$	Wang et al., 1998

* D – dipolar couplings. Subscripts and superscripts are used to designate atoms in the protein backbone. See Figure 1.1 for details.

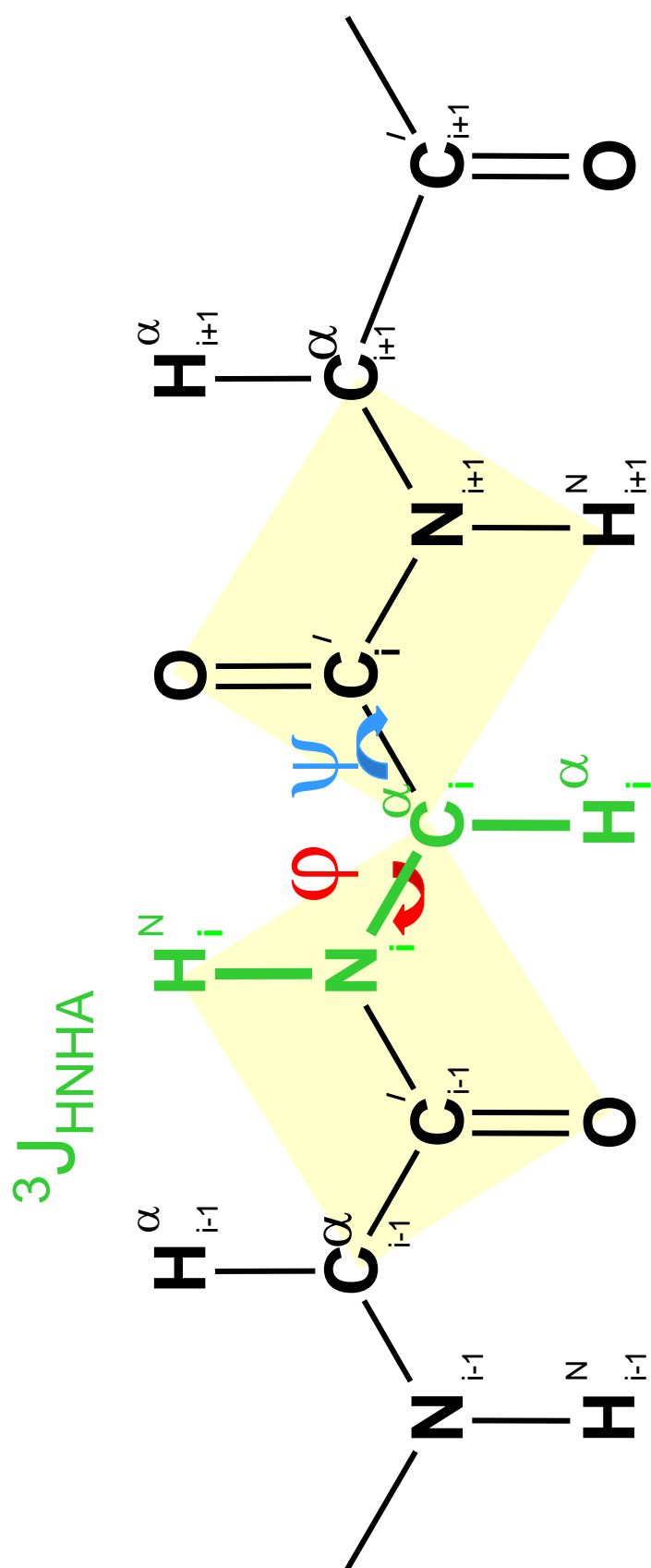


Figure 1.1 Structure of the Protein Backbone
 The atoms have been labeled with the NMR designations.
 The peptide planes are shown in yellow for reference.

CHAPTER 2

DEVELOPMENT OF SEASCAPE

It is well known that $^{13}\text{C}\alpha$ chemical shifts ($\delta\text{C}\alpha$) are sensitive to both amino acid type and local backbone (secondary) structure. Normally $^{13}\text{C}\alpha$ resonances are assigned to amino acid types and the deviation of chemical shifts from random coil values for the amino acid is used to deduce secondary structure (Spera, 1991; Wishart and Case, 2001). The objective of this thesis is to do the opposite, assign secondary structure, or ϕ , ψ angles, and use deviations in chemical shift at a given ϕ , ψ combination to identify amino acid type. Since the BioMagResBank's (BMRB's) database of chemical shifts and other NMR data contains thousands of entries, it seemed possible to combine shift data from the BMRB with structural data available from the Protein Data Bank (PDB) to provide a statistic that can predict amino acid type from $^{13}\text{C}\alpha$ shift and local structure characteristics.

2.1 Data Correlation

The first step in correlating the data from the BMRB and PDB is to determine which proteins are common to both databases. At the time this work was started, there were 1966 files for proteins in the BMRB database. Of these, 1433 contained five or more $^{13}\text{C}\alpha$ chemical shifts per file. Each of these files was searched for the text string "PDB" and the resulting list of BMRB and corresponding PDB file names was output to

a file. At the time this was done there was no percent homology information given in the BMRB file so it was impossible to know a priori if the “matched” BMRB and PDB names were indeed complete matches (to an NMR derived structure) or if the PDB listed was just a reference to a crystal structure someone had done of the same, although perhaps mutated, protein. There are cases in which the protein in the BMRB entry and the referenced PDB entry were the same (eg., *Clostridium pasteurianum* Rubredoxin C42S Mutant). In these cases another problem arose quite often, one in which the two files did not contain the same starting point, numbering scheme, or end point. There were often missing residues in one or both of the files as well. Most missing data was from BMRB files which rarely contain chemical shift data for all C α 's in the protein. In rarer instances, a PDB file from an x-ray structure would not contain data for an amino acid residue in the protein. This is most likely due to insufficient density to accurately determine the placement of the atoms.

Before attempting to correlate the two databases, all of the torsion angles from the protein structures in the list of PDB files created earlier were extracted using the program **dang** (URL: <http://kinemage.biochem.duke.edu>; Word, 2000). Only the ϕ and ψ torsion angles for each residue were needed so those values were extracted and placed in a separate file. But because the dihedral angle extraction program, **dang**, calculates torsion using the heavy atom convention ($\phi = C'_{i-1} - N_i^H - C_i^\alpha - C'_i$, $\psi = N_i^H - C_i^\alpha - C'_i - N_{i+1}^H$; Figure 2-1) it was adopted rather than using the $^1\text{H} - ^a\text{X} - ^b\text{Y} - ^1\text{H}$ torsion frequently seen in NMR applications. Then, to address the correlation issues listed above, a Perl program (Appendix A) was used to align the sequence in the respective BMRB and PDB files. Using a recursive algorithm, the first five residues in the BMRB file were compared to

the first five amino acids in the PDB sequence. If a complete match was not found the BMRB data was moved down the PDB sequence by one residue and the match was attempted again. This was repeated until a match was found or the BMRB data fragment could not be moved any further down the PDB sequence. When a match was found the BMRB and PDB file names as well as the correct alignment for the match was output to an intermediate file given the name of the BMRB file and “.align” appended to the file name. The alignment procedure was able to skip over missing sections of data in each file. If no match was found that information was printer to a file. As a result, 728 BMRB files were ultimately used to obtain the data employed in later analysis.

After the starting point and alignment list was compiled, another Perl program (Appendix B) was used to extract the data points. Each data point was checked to insure that the amino acid listed in the BMRB file was a match to the one listed in the PDB file. If a mismatch was found, the program was terminated for that particular pair of files. Each data triplet ($\delta C\alpha$, ϕ , ψ) was written to a file for that amino acid (eg., alaCAphipsi.txt). The number of data points for each amino acid is listed in Table 2.1 along with the total number of possible points from the BMRB files containing five or more $^{13}C\alpha$ chemical shifts. Approximately half of the possible data points were correlated for each amino acid. For each ($\delta C\alpha$, ϕ , ψ) point collected the ϕ value was used to calculate the corresponding $^3J_{HNHA}$ coupling constant using a standard Karplus equation (Equation 2.1; Pardi et al., 1984).

$$^3J_{HNHA} = 6.4 \cos^2(\phi - 60) - 1.4 \cos(\phi - 60) + 1.9 \quad (2.1)$$

A second set of parameters, $A = 6.51$ $B = -1.76$ $C = 1.6$ (Vuister and Bax, 1993), was also used to calculate a slightly different set of coupling constants in order to determine if they

would result in a different assignment. Since none was found, the original parameter set was used in all subsequent analyses.

2.2 Probability Density Function Calculations

To accurately predict the probability of obtaining a particular measurement, one must know the probability density function (PDF) for that variable. This PDF may have any number of variables and can be described by a well known analytical function, such as a normal (Gaussian) distribution, or it can be completely unknown. For known, or parametric, PDFs, such as the normal distribution, the parameters can be estimated with an increase in accuracy concomitant with an increase in the number of measurements. In cases where the function is unknown, or when it is preferable to avoid unjustifiable assumptions, one may use the method of nonparametric density estimation as a means of determining the PDF.

Histograms may be thought of as a crude, nevertheless often effective, method of nonparametric density estimation. The main problems inherent in the use of histograms are bin size and origin placement. Bin size is used as a means to smooth the data but in each case the size of the bin must be chosen. This could lead to artificial features, or blurring of features, in the distribution. An example of this is seen in Figure 2.2. Graphs (a) through (d) illustrate how the apparent distribution of a set of data can change based on the size of the bin. Data in the example are the number of $\delta C\alpha$'s obtained from BMRB files. For purposes of illustration only those within the range 5 to 50 $\delta C\alpha$'s per file are shown. The effects of origin placement (where the bins start) can be seen in Figure 2.3. The same data are used as in Figure 2.2, only the starting point for the bins is changed. These examples are very simple in part because they represent univariate data.

Issues of bin size and placement become much more important in multivariate cases, especially in fourth and higher dimensional data.

Kernel density estimation (KDE; Silverman, 1986) is a more generalized approach to nonparametric density estimation. The kernel (K) is a function which satisfies the condition

$$\int_{-\infty}^{\infty} K(x)dx = 1 \quad (2.2)$$

Often the kernel is a symmetric PDF such as the normal distribution shown in Equation 2.3. Here μ is the mean of x and σ is the standard deviation. This appears to be a logical choice for the kernel since one goal is to allow for normal variations in the data. In the univariate case, when a normal distribution is used as the kernel, the error inherent in the

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-(x - \mu)^2 / (2\sigma^2)\} \quad (2.3)$$

data may be taken into account by using that value in place of the standard deviation (σ_{er} replacing σ in Equation 2.3). Then, an estimate of the probability density at any position (x) is merely a summation (Equation 2.4) of the contribution of each data point (x_i). The entire PDF may also be estimated by summing the individual kernels over the range of interest (e.g., $40 \leq \delta C\alpha \leq 70$ ppm).

$$PDF(x) = \sum_{x_i=1}^n \frac{1}{\sqrt{2\pi}\sigma_{er}} \exp\{-(x - x_i)^2 / (2\sigma_{er}^2)\} \quad (2.4)$$

In this research, KDE was used to calculate the probability density function for each amino acid over $\delta C\alpha$, ϕ and ψ space. A multidimensional normal distribution function (Equation 2.5) was used as the kernel. It provided a means of compensating for

error in measurements and for regions containing sparse data. In Equation 2.5, \vec{x} is the point at which the PDF is being calculated ($[\delta C\alpha, \phi, \psi]$ or $[\delta C\alpha, {}^3J_{\text{HNHA}}]$), p is the dimensionality of the problem (2 or 3 in this study), Σ is the covariance matrix which contains errors and their correlations, and \vec{x}_i represents a point in the experimental data set ($[\delta C\alpha(i), \phi(i), \psi(i)]$ or $[\delta C\alpha(i), {}^3J_{\text{HNHA}}(i)]$). This kernel was then used to generate a PDF from the experimental data for each amino acid. The resulting PDF for each amino acid covers $^{13}\text{C}\alpha$ chemical shifts ranging from 40

$$PDF(\vec{x}) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\{-1/2 (\vec{x} - \vec{x}_i)' \Sigma^{-1} (\vec{x} - \vec{x}_i)\} \quad (2.5)$$

to 70 ppm and all of ϕ, ψ space (using the convention -180 to 180) for the first data set and the same chemical shift range and ${}^3J_{\text{HNHA}}$ values from 1.5 to 10 Hz for the second data set. Each PDF was normalized over this range for each amino acid. An experimental data point ($[\delta C\alpha(i), \phi(i), \psi(i)]$ or $[\delta C\alpha(i), {}^3J_{\text{HNHA}}(i)]$) can only belong to one amino acid at a time (i.e., there are no joint probabilities) and it must belong to one of the 20 amino acids. Therefore the probability densities for each point were normalized such that the probability that a point was one of the 20 amino acids was one (Equation 2.6).

$$\sum_{i=1}^{20} P(\delta C\alpha(i), \phi(i), \psi(i)) = 1 \quad (2.6)$$

A problem encountered in the calculation of the probability densities is that of applying a continuous function such as a normal distribution to a cyclical variable such as a torsion angle. While it is obvious to a human that 179° is only 2° away from -179° , it is not obvious to the function and it will treat the two values as if they are 358° apart with a corresponding reduction in probability density. Any probability densities calculated near

this point would be incorrectly lower than their true value. In order to minimize this error, each point calculated was transformed into the exact middle of the range of possible torsion angles. For example, if the $\delta C\alpha$, ϕ , ψ point to be calculated was (45.7, -179, 130) then all experimental data points in the ϕ and ψ dimensions would be recentered around -179° and 130° respectively for the calculation. This would result in ϕ values with the range (1°, -359°) while the range for ψ values would be (-50°, 310°).

2.3 Calculation of Probabilities Using ($\delta C\alpha$, ϕ , ψ) or ($\delta C\alpha$, $^3J_{\text{HNHA}}$)

Because, for a given \vec{x} , the probability for assignment to the best amino acid may be only marginally higher than the probability for assignment to the next best amino acid, it is necessary to improve discrimination by using the fact that data can be connected for several residues. This allows combined probabilities to be calculated for sequentially connected amino acid types appearing in the protein primary sequence. The program SEASCAPE (SEquential Assignment by Structure and Chemical shift Assisted Probability Estimation) was therefore written to make assignments based on known connectivities between any number of resonances, for which $\delta C\alpha$, ϕ and ψ or $\delta C\alpha$ and $^3J_{\text{HNHA}}$ are available. Given data from the fragment to be assigned, the program takes the protein sequence and PDFs for each amino acid and calculates the combined probability for the fragment to be placed at each possible position in the sequence by multiplying the likelihood of assignment for each individual amino acid as in Equation 2.7.

$$\prod_{i=1}^n P(\delta C\alpha_i, \phi_i, \psi_i) \quad (2.7)$$

An illustration of this procedure can be seen in Figure 2.4. A section of contiguous residues for which the connectivities and each residue's $\delta C\alpha$, ϕ , and ψ , or $\delta C\alpha$, $^3J_{\text{HNHA}}$

are known (hereafter called a fragment) is aligned along the beginning of the sequence, and the probability that the fragment is correctly positioned is calculated. The fragment is then repositioned by sliding it over one residue and the probability at that position is calculated. This procedure is repeated until all of the positional probabilities have been calculated. The most probable alignment is the position with the highest calculated probability. Although this program was originally written in Perl, later versions will be written in a combination of Tcl/Tk and C++ to add a graphical interface, combine functionalities, and ease additions and incorporation into other NMR analysis packages. The program along with example input files may be seen in Appendix C.

2.4 Calculation of Probabilities with Known Structure

A modification of SEASCAPE was made to utilize data from a previously determined structure. In this instance the fragment experimental data consists only of $^{13}\text{C}\alpha$ chemical shifts and connectivities. Torsion angles are taken directly from the structure and kept with the amino acid sequence. As before, the program takes the protein sequence and PDFs for each amino acid and calculates the combined probability for the fragment to be placed at each possible position in the sequence. The only difference now is that the torsion angles are no longer paired with a particular $\delta\text{C}\alpha$ but with the correct amino acid in the sequence.

2.5 Use of Residual Dipolar Couplings

In most high resolution NMR experiments, contributions to spin-spin couplings other than scalar couplings are ignored. This simplification is justified since molecules

tumble and sample orientations in space isotropically. Therefore the effects of through-space magnetic dipoles generated by each nucleus on other nuclei are averaged to zero. In more recent applications, departures from isotropic orientation distributions have been induced using anisotropic liquid crystal media. In these systems, the effects of the magnetic dipoles upon other nuclei do not average to zero. This introduces an additional coupling term. This additional coupling, residual dipolar coupling (RDC), adds to the preexisting scalar coupling. The magnitude (D_{ij}) of this additional coupling between two atoms, i and j , shown in Equation 2.8, varies with types of coupled nuclei

$$D_{ij} = \frac{-\mu_0 \gamma_i \gamma_j h}{(2\pi r_{ij})^3} \left\langle \frac{3 \cos^2 \theta(t) - 1}{2} \right\rangle \quad (2.8)$$

(i.e., ^{15}N - ^1H versus ^{13}C - ^1H) due to the difference in gyromagnetic ratios (γ) for various nuclei, but for a given set of vectors connected directly such as an ^{15}N - ^1H pair, the differences in magnitude are inversely proportional to the distance (r_{ij}) cubed between the two coupled nuclei and the function in brackets dependent on the angle between their internuclear vector and the magnetic field ($\theta(t)$). In the above equation, μ_0 is the permittivity of free space and h is Planck's constant. Since the internuclear distance is known for directly bonded nuclei, the difference in RDCs among a set of data are due to differing orientations of the vectors. As a result, relative orientations for portions of a protein may be determined, thereby aiding structure determination.

In order to measure RDCs one only needs to collect coupling data in a traditional way without partial alignment (isotropic media) and then collect a second data set using the exact same experiment but with the sample in an alignment medium (anisotropic media) such as bacteriophage or bicelles (Prestegard and Kishore, 2001). The RDC is

simply the difference between the experimental coupling constants in isotropic and anisotropic media and, depending upon the orientation of the vector, can be either positive or negative.

2.6 Calculation of Probabilities with Residual Dipolar Couplings

A second modification of SEASCAPE makes use of available residual dipolar couplings. If a protein structure is already known, and any type of RDCs are collected, this additional data may be used to assist in the assignment of fragments. If a small portion, theoretically five but realistically at least eight, of the RDCs are already assigned it might be possible to obtain the preferred orientation of the molecule using a program such as REDCAT. REDCAT will then calculate theoretical RDCs for the remainder of the protein using the previously calculated alignment tensor. From the “back calculated” RDCs, it will be possible to make use of this additional data to assist sequential placement of any connected fragment.

As in the previous version of SEASCAPE, the fragment is placed at each possible position in the sequence. This time instead of determining the probability by calculating the product of the probability densities, the root mean square deviations (rmsds) between the experimental and theoretical RDCs for the fragment is calculated at each position. A lower rmsd corresponds to a higher probability of that position being the correct placement of the fragment. Given that two or more sections of the protein may have a very similar local structural orientation, as in the case of proteins with two or more parallel beta strands or alpha helices, it is desirable to combine these results with another analysis, such as that provided chemical shift analysis versions of SEASCAPE.

In order to combine the results of chemical shift version of SEASCAPE with RDC comparisons, the rmsds calculated need to be modified. By taking the negative exponent of the rmsds, the resulting figures would be between zero and one, with higher values indicating a better match between experimental and calculated RDCs in the fragment. The RDCs which more closely matched experimental values would then have a higher score. The product of the scores from the RDC analysis and the SEASCAPE analysis would then produce a single probability type number which could be interpreted as an overall best fit for the data. As before, a higher score would indicate an increased likelihood of correct assignment.

Table 2.1. Number of data points used to determine the probability distribution over ($^{13}\text{C}\alpha$, ϕ , ψ) for each amino acid.

amino acid	# data points	% ^a occurrence	amino acid	# data points	% ^a occurrence
ala	1521	7.5	leu	1779	8.8
arg	919	4.5	lys	1590	7.8
asn	793	3.9	met	378	1.9
asp	1312	6.5	phe	757	3.7
cys	313	1.5	pro	870	4.3
gln	817	4.0	ser	1206	5.9
glu	1591	7.8	thr	1138	5.6
gly	1446	7.1	trp	217	1.1
his	467	2.3	tyr	601	3.0
ile	1137	5.6	val	1426	7.0

^a The percent occurrence of each amino acid.

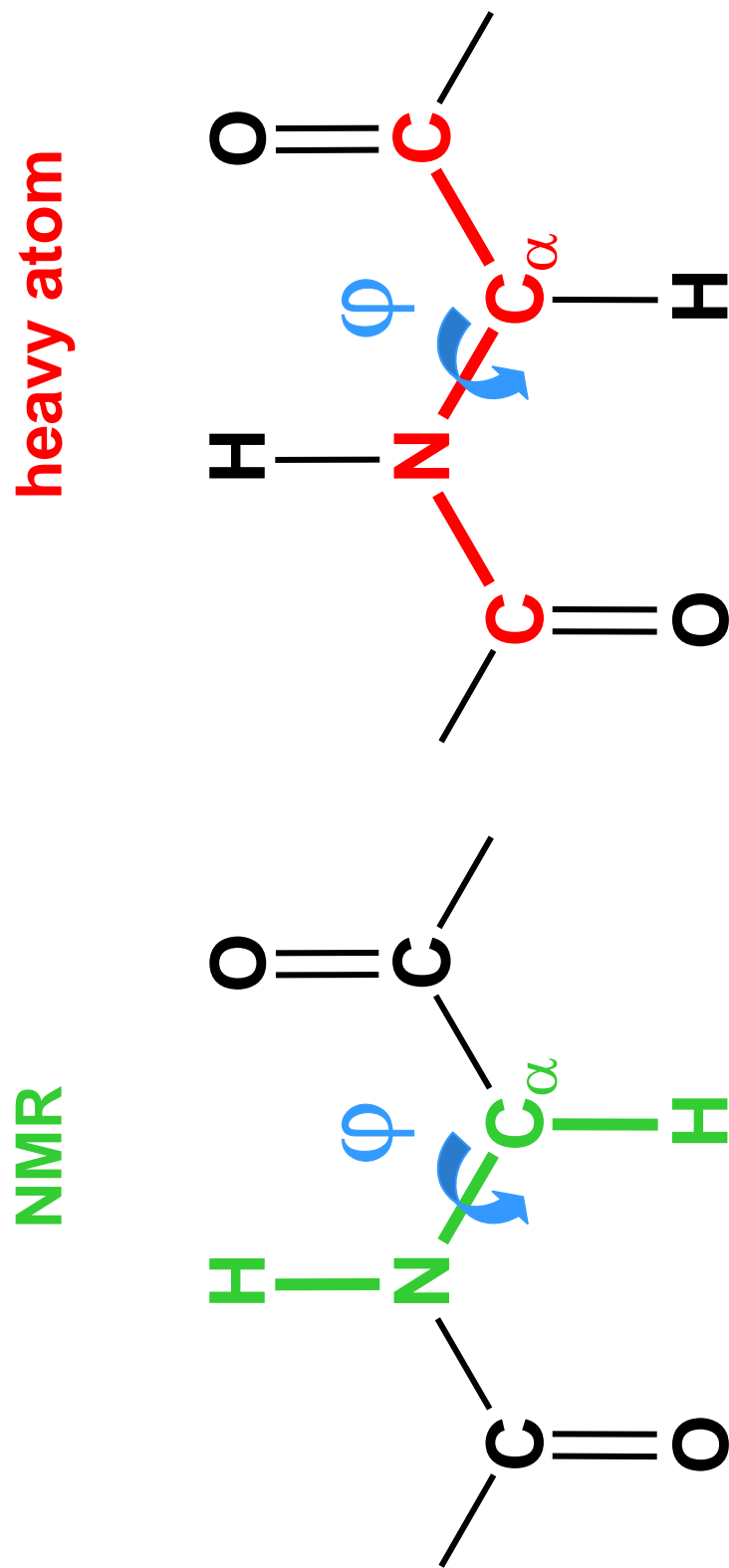


Figure 2.1 NMR and heavy atom definitions of the protein backbone torsion angle ϕ .

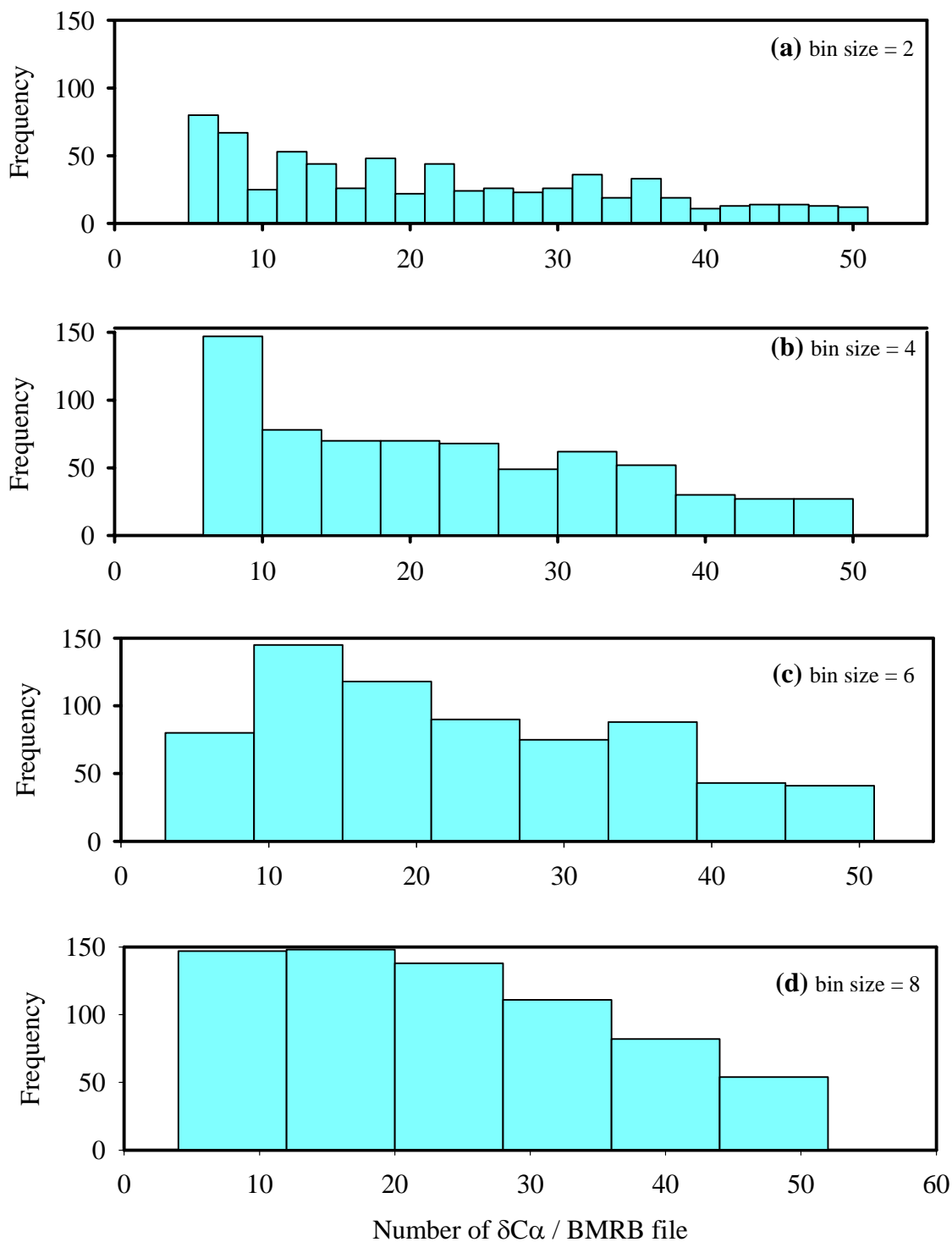


Figure 2.2 Histograms with different bin sizes. In all cases the x-axis is the same. (a) Distribution of the number of $^{13}\text{C}\alpha$ chemical shifts per BMRB file using a bin size of 2. (b) Same data as in (a) but with a bin size of 4. (c) Same data as in (a) but with a bin size of 6. (d) Same data as in (a) but with a bin size of 8.

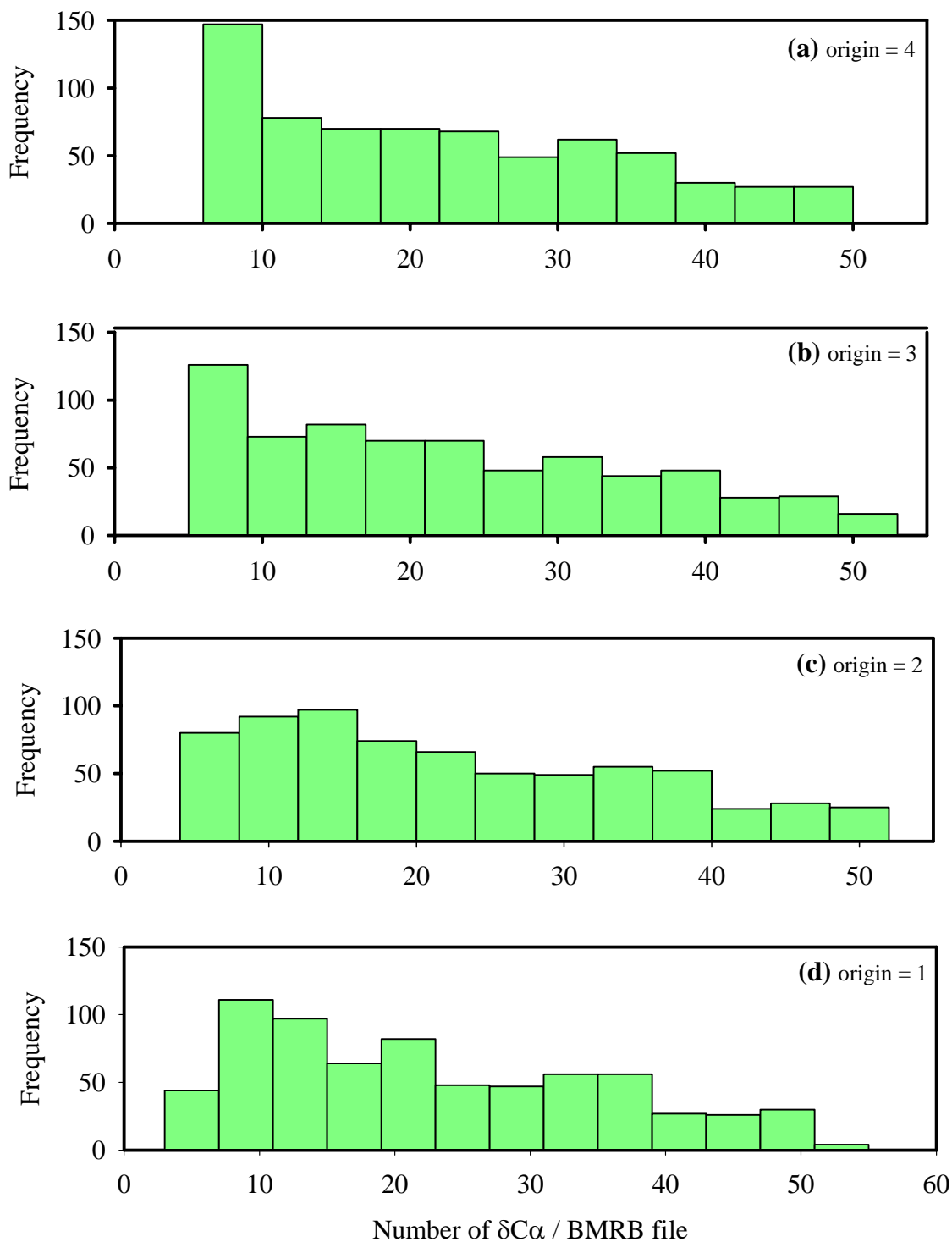


Figure 2.3 Histograms with different origins. In all cases the x-axis is the same.

(a) Distribution of the number of $^{13}\text{C}\alpha$ chemical shifts per BMRB file using an origin (bin starting point) of 4. (b) Same data as in (a) but with an origin of 3. (c) Same data as in (a) but with an origin of 2. (d) Same data as in (a) but with an origin of 1.

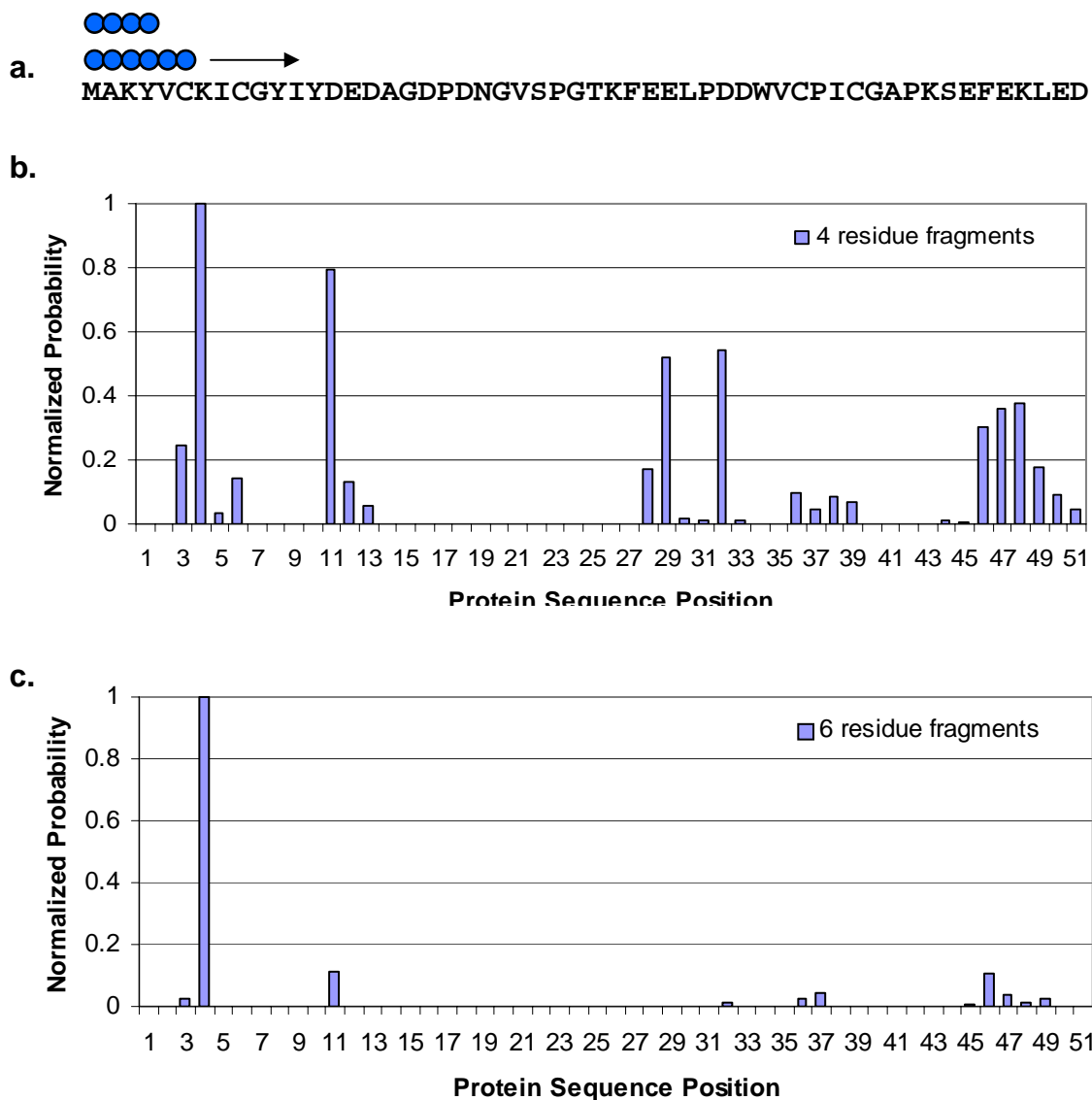


Figure 2.4. Example of assignment of a fragment to a position in a protein sequence. (a.) Sequence of a rubredoxin mutant (1M2Y) from *Pyrococcus furiosus*. Four and six residue fragments (circles) are moved along the sequence as probabilities are calculated for each possible position. Graphs of the normalized probabilities calculated for the 4 residue (b.) and 6 residue (c.) fragments, ICGY and ICGYIY respectively. The ($^{13}\text{C}\alpha$, ϕ , ψ) data used for the analysis are 4Y = (56.63, -125.1, 135.1), 5V = (58.28, -110.2, 140.1), 6C = (59.79, -75.1, 64.8), 7K = (59.08, -14.7, -9.9), 8I = (62.43, -115.1, 15.2), and 9C = (59.1, -140, -30.2).

CHAPTER 3

APPLICATIONS

A primary objective of this chapter is to provide an assessment of the accuracy with which SEASCAPE can assign peptide fragments to their proper sequential positions in proteins. A mutant of the protein rubredoxin from *Pyrococcus furiosus* (PDB ID 1M2Y) was used as the primary test case for all analyses performed. Nine other proteins were selected from the BMRB database for subsequent analysis, based on the amount of data present for the protein. In order to investigate the performance of SEASCAPE (Appendix 3), rubredoxin fragments varying in length from one to six amino acids were tested. Data used in these initial analyses were either ($\delta C\alpha$, ϕ , ψ) or ($\delta C\alpha$, $^3J_{HNHA}$). Later analyses, on the other nine proteins, used four or six residue fragments and ($\delta C\alpha$, ϕ , ψ) data exclusively. Investigations using residual dipolar coupling (RDC) data were assessed using the rubredoxin model.

3.1 Distribution of Data Points

The distribution of the data obtained from correlating the BMRB $^{13}C\alpha$ chemical shifts ($\delta C\alpha$) with ϕ and ψ data obtained from structures in the PDB can be seen in Figures 3.1 – 3.20. Graph (a) in each figure is the kernel density estimate of the probability density function (PDF) created to examine the distribution of $\delta C\alpha$'s obtained

for that amino acid. A normal distribution was used for the kernel. A more traditional Ramachandran plot of ϕ versus ψ values obtained from the databases can be seen in the (b) graphs. Scatter plots of $\delta C\alpha$ versus ϕ (c) and $\delta C\alpha$ versus ψ (d) are in the remaining graphs.

From plots of the chemical shift distribution it is easy to get a quick overview of the dispersion one is likely to see in the ^{13}C dimension of protein NMR spectra. The $\delta C\alpha$ versus ϕ and ψ graphs allow an informal look at possible correlations between $^{13}\text{C}\alpha$ chemical shift and one or both of the torsion angles. The expected preference for negative ϕ values is seen for all amino acids with the exception of glycine (Figure 3.8c). Groupings in the $\delta C\alpha$ versus ψ graphs illustrate the correlation between a protein's secondary structure and the ψ torsion angle. Two amino acids, cystine (Figure 3.5) and tryptophan (Figure 3.18), are described by only a few hundred data points each and the $^{13}\text{C}\alpha$ chemical shift values are dispersed. Methionine (Figure 3.13), on the other hand, has approximately the same number of points yet has a tighter clustering of data. This suggests that the conformational populations of cystine and tryptophan may be under represented by the data and those of methionine may not. For all amino acids a low density section may be seen running all the way through the ψ dimension of the (b) plots at approximately zero degrees in the ϕ dimension. The best illustration of this is seen in the data for glycine (Figure 3.8b). This region is sterically unfavorable due to the clash between the carbonyl oxygens in adjacent residues. A sparsely populated region may also be seen in the vicinity of $\psi = -120^\circ$. In this case the side chains prevent the amino acids from adopting ψ values in this region. The obvious exception is glycine, with only a single hydrogen in the side chain.

3.2 Assignment of Rubredoxin Fragments of Various Lengths

Known fragments, one to six residues in length, were created from the completely assigned protein rubredoxin (PDB ID 1M2Y, Figure 3.21). Two sets of fragments from this protein were created. The first contained $^{13}\text{C}\alpha$ chemical shifts and backbone torsion angles (ϕ , ψ) along with connectivity information. The second contained $^{13}\text{C}\alpha$ chemical shifts, $^3\text{J}_{\text{HNHA}}$ coupling constants and connectivity information. SEASCAPE was used to assign the fragments to positions in the protein based on these data. The robustness of the method was found to be steeply dependent upon the length of the fragment. Table 3.1 shows a comparison of how often the highest probability resulted in the correct assignment for fragments of different lengths. Column 1 contains the results for ($\delta\text{C}\alpha$, ϕ , ψ) fragments while column 2 contains results for ($\delta\text{C}\alpha$, $^3\text{J}_{\text{HNHA}}$) fragments. In all cases the results have been averaged over all possible assignments for 1M2Y. For this protein, ϕ and ψ had been directly determined from NMR data prior to this analysis and were associated with positions in the connected amino acids of the fragment when the NMR structure of the protein was determined. The utility of attempting assignment using just $^{13}\text{C}\alpha$ shifts and $^3\text{J}_{\text{HNHA}}$ data seems marginal unless very long stretches of connectivities can be established. However with ($\delta\text{C}\alpha$, ϕ , ψ) data, sequential stretches of five or more prove to give reliable assignments.

Also included in Table 3.1 are results from cases in which ϕ and ψ angles are associated with the sequence and not the fragments (column 3). In other words, assignment problems in which a previous NMR or x-ray structure is available. The backbone torsions remain fixed to their proper position in the protein while the remaining fragment data, $\delta\text{C}\alpha$ and connectivities, are moved along the sequence. The percentage of

correct assignments is slightly less in this case, but largely parallels the initial study in which ϕ and ψ angles are associated with a particular $\delta C\alpha$ in a particular fragment (column 1).

3.3 Assignment of Fragments for a Selection of Proteins

The program was further tested using 9 additional proteins chosen from a set having a significant percentage of their $^{13}C\alpha$ chemical shifts deposited in the BMRB and a structure available from the PDB. Here, angular constraints from either x-ray or NMR derived structures and $^{13}C\alpha$ data from the BMRB were used (Table 3.2). A detailed accounting of which fragments have been correctly positioned for all 10 proteins is given in Figures 3.21 – 3.30. Again, these cases represent those in which an x-ray structure is available and assignments may be sought for the purpose of ligand screening using HSQC data.

The best performance of the program was on the proteins 1M2Y, 1C0V and 1CDC (Figures 3.21, 3.22 and 3.27 respectively). These three proteins each gave nearly complete (or complete) correct assignments for all fragments 6 residues in length and approximately 80% for fragments 4 residues in length. While one might have expected these to be predominantly structures derived from x-ray data since it is the predominant source of data in the PDB, the structures of two of these proteins, 1M2Y and 1C0V, were derived from NMR data. This fact could reflect slight deviations of structures in crystals from those in solution, making the NMR structures a better fit, but more likely this reflects a more rigorous scrutiny of experimental assignments when they are subsequently used to produce a structure.

One protein, 1M2Y, has almost no secondary structure listed in the HEADER portion of its structure file, while 1CDC contains a significant amount of β -sheet structure and 1C0V is almost all α -helix. While this is a small sampling of the possible structural elements it is nevertheless encouraging to see successful application over a range of classical secondary structures.

The next best results were obtained on 1SYM (Figure 3.24), with 84% correct assignment for 6 residue fragments and 62% correct assignments for 4 residue fragments. As with other proteins, increasing the length of the fragment not only increases the likelihood of obtaining a correct assignment but the correct assignments are clustered together as well. One of the largest proteins attempted, 1AZM (Figure 3.29), gave fairly good results with assignment percentages of 80% (6 residue fragments) and 47% (4 residue fragments). Four of the proteins, 1QJT (Figure 3.23), 1DMO (Figure 3.26), 1EZA (Figure 3.28) and 1L6N (Figure 3.30) all had 6 residue fragments assigned correctly approximately 70% of the time and 4 residue fragments assigned correctly approximately 40% of the time.

Table 3.2 clearly illustrates the aforementioned variability in the level of successful assignments, but in all proteins (with one notable exception), a six residue connected fragment can be placed in the sequence with greater than 70% certainty using just $^{13}\text{C}\alpha$ shifts, ϕ and ψ data. The one exception is a DNA binding protein, 1IRF (Figure 3.25), whose binding domain has been categorized as a novel subgroup of the winged helix-turn-helix family (Furui, J., et al., 1998).

In order to understand why assignment of the 1IRF protein proved difficult we repeated the analysis with structural data from a corresponding crystal structure (2IRF).

The crystal structure is from a DNA bound form of the protein; the structural information was nevertheless combined with $^{13}\text{C}\alpha$ data from solution in the absence of DNA. Four and six residue fragments were again examined and the results are reported in Table 3.2. When using the crystal structure we obtained double the number of correct identifications of fragment position. It is possible that the dynamic nature of the protein in solution when not bound to DNA contributed to a set of averaged torsion angles that do not correlate well with chemical shift. Indeed, the torsion angles for half of the residues in the protein differ substantially ($>30^\circ$) between the unbound solution structure and the bound crystal structure. It is also possible that the NMR structure is a poor quality structure. Structure validation programs such as PROCHECK in fact show a significant number of unlikely peptide geometries for the deposited NMR structure. This possible ability to identify poor structure suggests some structure validation applications of SEASCAPE.

3.4 Correlation of Success with Amino Acid Composition and Secondary Structure

It is of some interest to examine possible variations in the success of assignment with variables such as amino acid content or secondary structure content. The PDFs of the individual amino acids give an indication of how distinctive the distributions are for each amino acid and how well the program might be expected to perform on a fragment of given composition. In order to simplify representation of data in the PDFs and produce a more user friendly form, the probability densities were summed across the entire ($\delta\text{C}\alpha$, ϕ , ψ) data set for each amino acid. Since the densities had previously been normalized over all amino acids at each point, the resulting sums provide an indication of

how well separated the distributions are. In Figure 3.31 the sums have been divided by leucine, which has the smallest sum, to produce a single number related to the value of $\delta C\alpha$, ϕ and ψ information in identifying each amino acid. Not surprisingly, glycine has the highest identification value by far (almost $16 \times$ leucine). Although this could easily be predicted by glycine's unique $^{13}C\alpha$ chemical shift, the reason that threonine has the second highest value is less obvious. Threonine valine, proline, and isoleucine, all have, on average, similar $^{13}C\alpha$ shifts. This suggests that the basis for distinction is more complex.

Surprisingly, regions with regular secondary structure (α helices and β sheets) do not result in more accurate assignments than regions that lack regular secondary structure (everything not defined as an α helix or β sheet). Of the proteins tested, about half of the data are for regions lacking regular secondary structure and the accuracy is no worse or better than structured regions. Among the fragments lacking regular secondary structure, 46% of the four residue fragments are assigned correctly while 76% of the six residue fragments are assigned correctly. In the fragments having regular secondary structure, 47% of the α helices and 48% of the β sheets are assigned correctly in the four residue fragments whereas 77% of the α helices and 68% of the β sheets are assigned correctly in six residue fragments. We still do not expect the program to perform well on highly flexible regions of proteins where chemical shifts may not correlate well with average structural parameters.

3.5 Raw Scores and Confidence

An indication of the confidence one should have in a given sequential assignment is also an important issue. The raw probability score obtained for a fragment can be used to give this indication. Figure 3.32 shows the probability of correct assignment for the top score given a fragment of four or six residues. Because individual probability densities in the data sets are always between zero and one, it is not unusual to obtain overall raw scores on the order of 10^{-4} for a four residue fragment and 10^{-6} for a six residue fragment. The highest scores obtained so far are 3×10^{-2} and 2×10^{-3} , for four and six residue fragments respectively. Respective scores for these fragments can be as low as 10^{-5} and 10^{-7} . In cases where the raw score is equal to or greater than 2×10^{-3} for a four residue fragment the probability of a correct assignment is greater than 80%. For a six residue fragment a score equal to or greater than 3×10^{-6} results in a more than 90% probability of correct assignment. In addition, scores at or above 10^{-4} for a six residue fragment lead to assignment with near certainty. Thus, we can evaluate the probability of successful assignment in the absence of results from more conventional strategies.

3.6 Correct versus Incorrect Connectivities in Fragments

A problem that requires a program to go a step beyond the proper sequential placement of correctly connected fragments is one in which there is some uncertainty in the connection, possibly due to degeneracy in $^{13}\text{C}\alpha$ shifts used to establish connections between the residues. Fragments can be generated with all possible connections and connectivities correctly representing the fragment may be determined by comparison of the probabilities of each of the proposed fragments. Each of the possible fragments is

threaded through the sequence taking the torsion angles, ϕ and ψ , from a proposed structure; the probabilities are calculated for each possible assembly and the highest score, or probability, is used to identify the correct assembly in addition to the proper sequential position. The rightmost column in Table 3.1 compared the results of this adaptation, using correctly assembled fragments, with the original program in which the $^{13}\text{C}\alpha$ chemical shifts were paired with experimentally determined ϕ and ψ . To test the method's ability to identify fragments that are not correctly connected several pairs of sequences 6 or 7 residues in length were threaded through the sequence with one member of the pair being correctly connected and the other incorrectly connected. In all cases the correctly connected sequence gave the highest score and was properly placed in the sequence. The same criteria given above for confidence in assignment, 3×10^{-6} for a six residue fragment and 2×10^{-3} for a four residue fragment, resulted in a confirmation of correct assignment for the six residue fragments but scores for the four residue fragment were below the 80% standard in all cases.

3.7 Combining RDC Analysis with SEASCAPE

Combining other data with $\text{C}\alpha$ chemical shifts to aid in assignment makes sense, particularly when those data are also acquired in the course of a structure determination. Residual dipolar coupling data for rubredoxin (1M2Y) has been used lately (Tian et. al., 2001) to fully determine the solution structure of the protein. Here the use of RDC data in resonance assignment is illustrated. From the RDC data set of this 54 amino acid protein eight one bond ^{15}N - ^1H dipolar couplings were selected at random. These RDCs were used to determine the orientation of the protein using the program REDCAT. Then REDCAT

was able to “back calculate” the entire set of ^{15}N - ^1H dipolar couplings for the protein. These values were used in comparison to experimental ^{15}N - ^1H RDCs to improve assignments.

Fragments of four connected residues which had not successfully been assigned previously by SEASCAPE were used in the analysis. Only three of the seven fragments in this category did not include a proline. The fragments were those beginning at amino acid positions 5, 26 and 27. The experimentally derived RDCs for these three fragments were compared to the back calculated RDCs at each possible position in the protein. The root mean square deviation (rmsd; Equation 3.1) was used to compute an overall score for the fragment at a

$$rmsd = \sqrt{\frac{\sum_{i=1}^n (RDC_{ex}^i - RDC_{bc}^i)^2}{n}} \quad (3.1)$$

particular position. Once these values were tabulated, the negative exponent (Equation 3.2) was taken of each rmsd to convert the values into a more suitable probability type

$$P_i = \exp(-rmsd_i) \quad (3.2)$$

number for combination with the chemical shift based results from SEASCAPE. In this initial application, probability from SEASCAPE chemical shift analysis and pseudo probability from RDC analysis were simply multiplied.

Application of this method to these fragments resulted in correct assignments in all cases. Representative results can be seen in Figure 3.33. The top plot contains the results from using the chemical shift version of SEASCAPE. The bottom plot illustrates the results from combining RDC analysis with the above data. Both plots have been normalized for easier comparison.

Although preliminary, these results suggest that by combining RDC analysis with SEASCAPE would result in a higher percentage of correct assignments and a higher confidence level. ^{15}N - ^1H RDCs are normally a part of the data collected in structure determination based on RDCs and these would normally be available in the course of these studies. ^{15}N - ^1H RDCs are also among the easiest RDC data acquired and acquisition for the purpose of assignment of resonances in proteins with x-ray structure is not beyond expectation.

Table 3.1. Comparison of assignment results for 1M2Y.

# residues in fragment	% correctly assigned		
	$^{13}\text{C}\alpha, \varphi, \psi$ ^a	$^{13}\text{C}\alpha, {}^3\text{J}_{\text{HNHA}}$ ^b	$^{13}\text{C}\alpha$ (+ structure) ^c
1	8	2	12
2	36	14	28
3	61	28	43
4	84	45	62
5	95	53	77
6	100	62	86

^a using φ, ψ data calculated from dipolar couplings

^b using experimental ${}^3\text{J}_{\text{HNHA}}$ values

^c using a modification in which the structure is already known and torsion angles are paired with the residues.

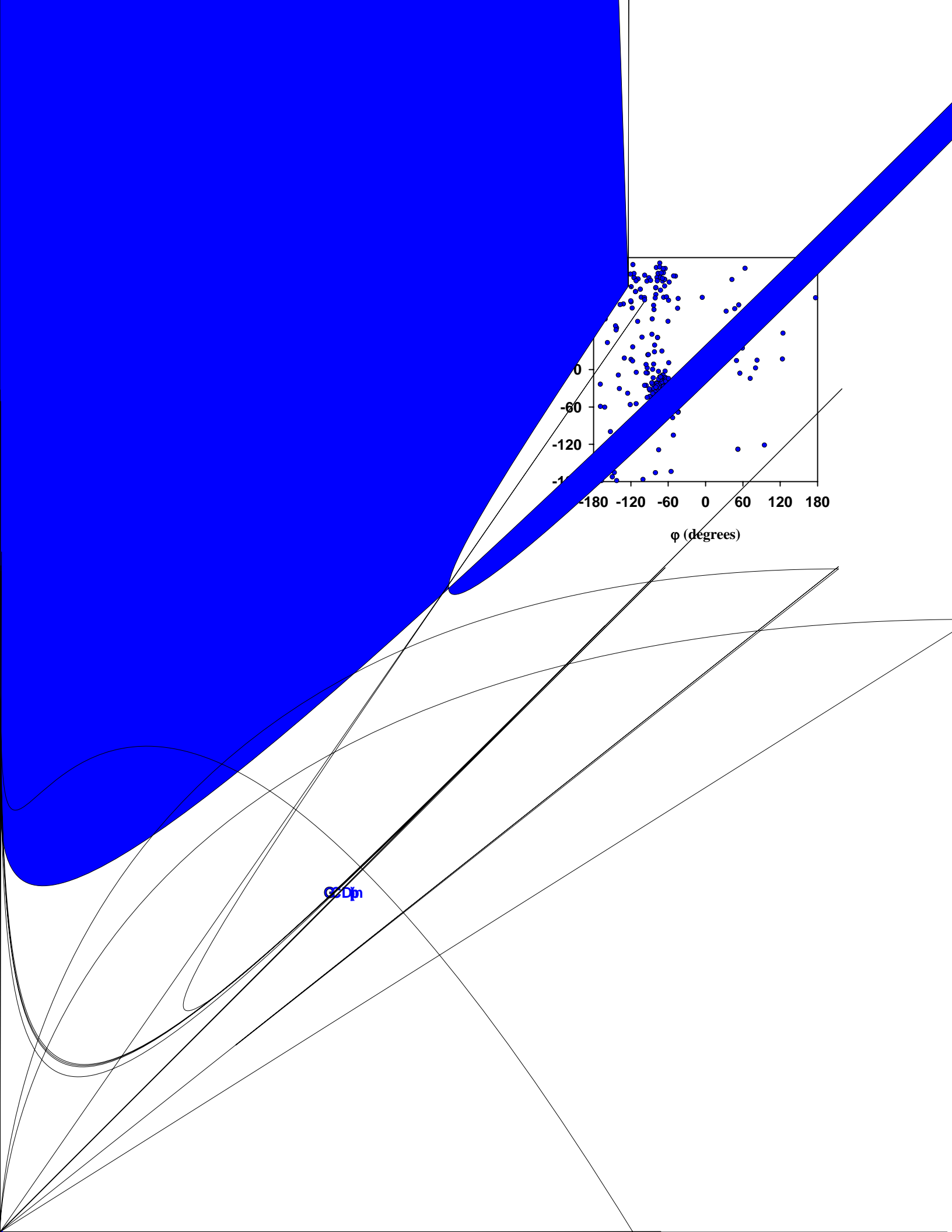
Table 3.2. Assignment results for a set of 10 proteins

PDB ID (#residues)	BMRB accession number	$(^{13}\text{C}\alpha, \varphi, \psi)$ % correct assignments		experimental structure	% secondary structure ^c	
		4 residues	6 residues		α helix	β sheet
1M2Y (54)	5601	84	100	NMR ^a	0	4
1C0V (79)	4146	85	96	NMR	82	0
1QJT (99)	4140	44	70	NMR	42	0
1IRF (112)	4161	23	43	NMR	29	13
2IRF ^b (113)	4161 ^b	52	81	X-ray	33	14
1DMO (148)	4056	39	71	NMR	55	5
1SYM (184)	4001	62	84	NMR	55	4
1CDC (198)	4109	78	98	X-ray	0	35
1EZA (259)	4264	39	76	NMR	50	10
1AZM (260)	4022	47	80	X-ray	8	28
1L6N (289)	5316	37	74	NMR	56	0

^a φ, ψ values determined using dipolar couplings, see Tian, F., et al., 2001.

^b Chemical shift data from the unbound solution structure (1IRF) was used in this analysis.

^c Secondary structure content as listed in the PDB in the Sequence Details section.



Arginine

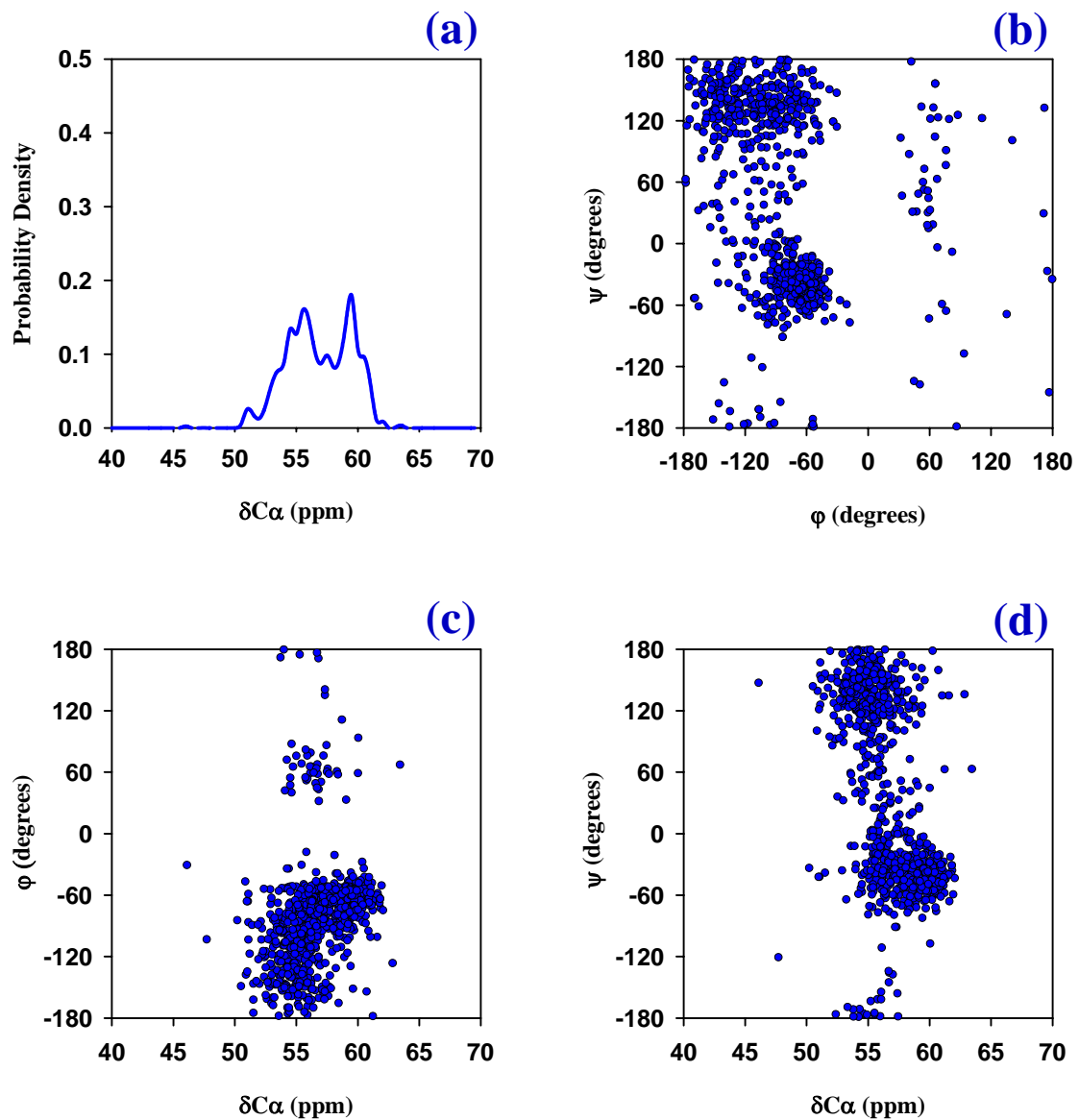


Figure 3.2 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for arginine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Asparagine

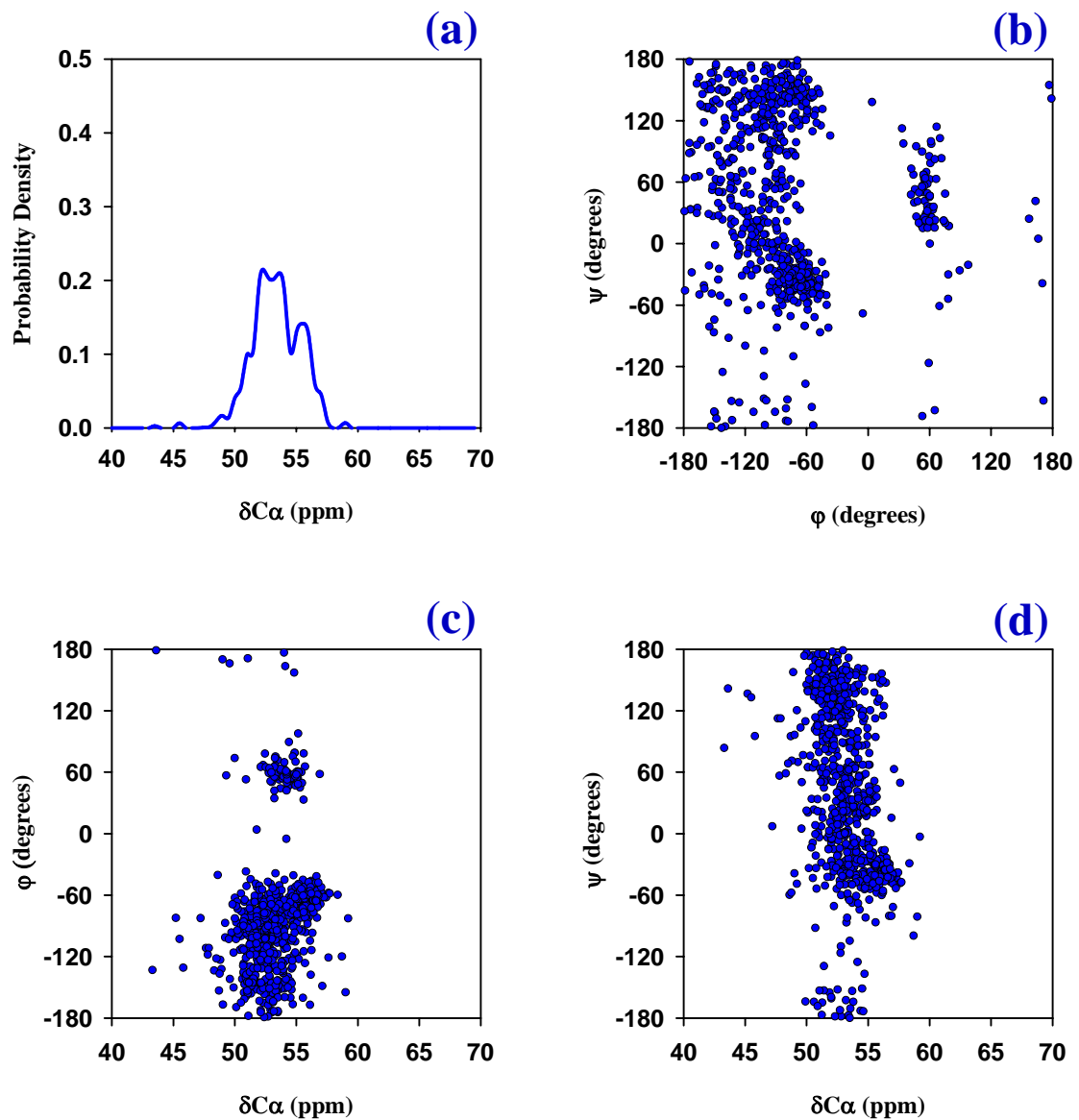


Figure 3.3 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for asparagine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Aspartate

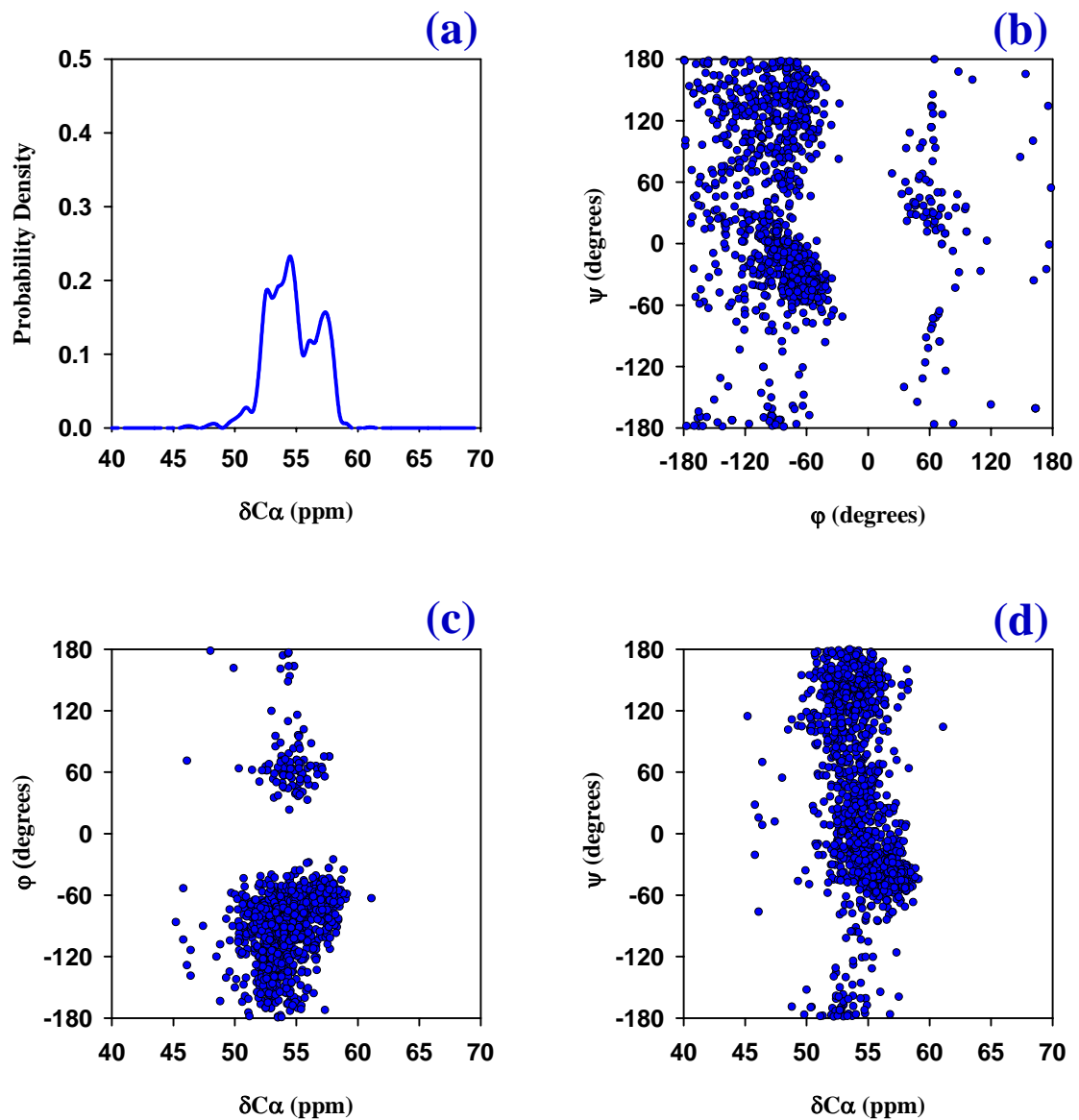


Figure 3.4 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for aspartate.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Cysteine

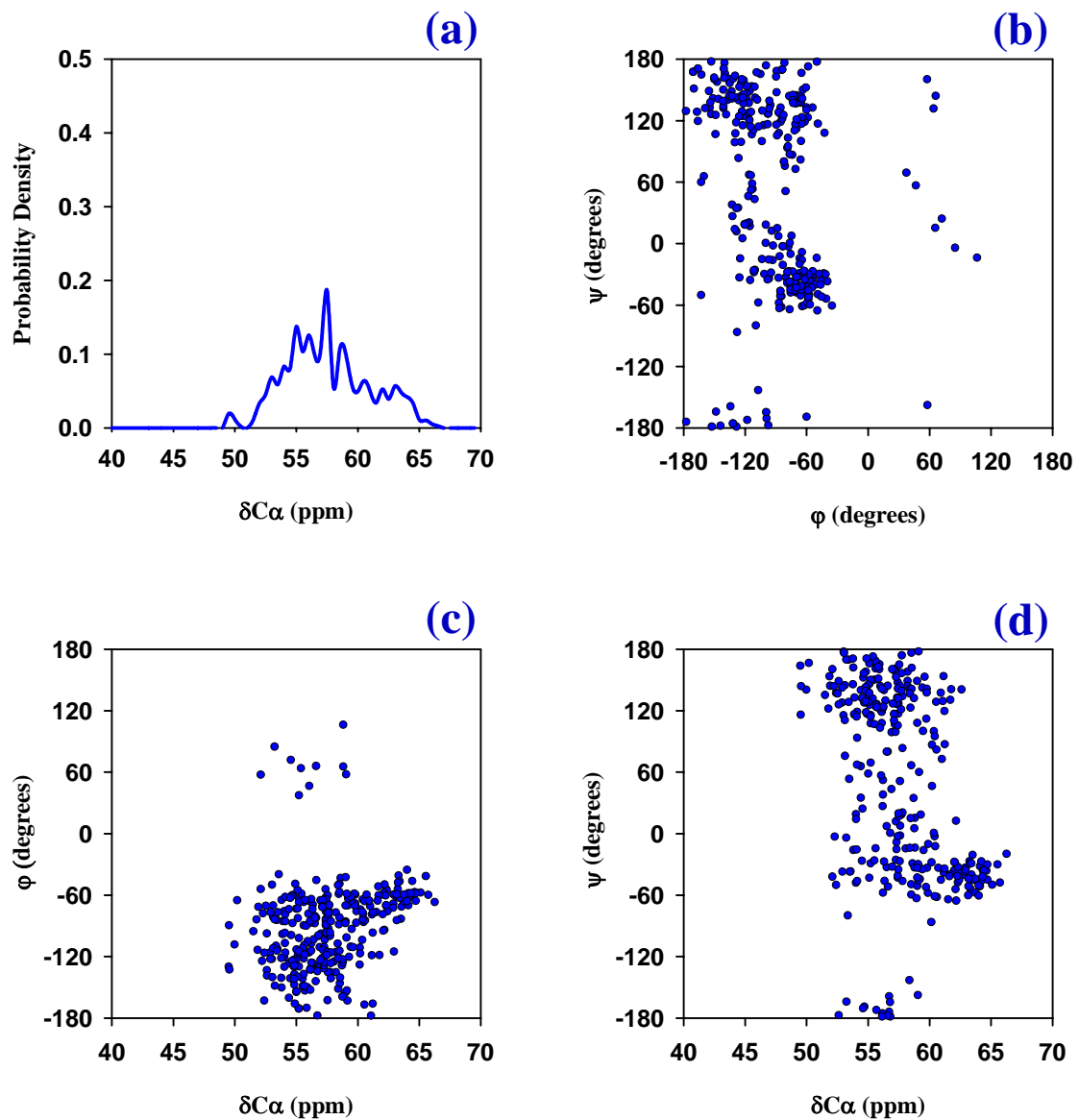


Figure 3.5 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for cysteine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Glutamate

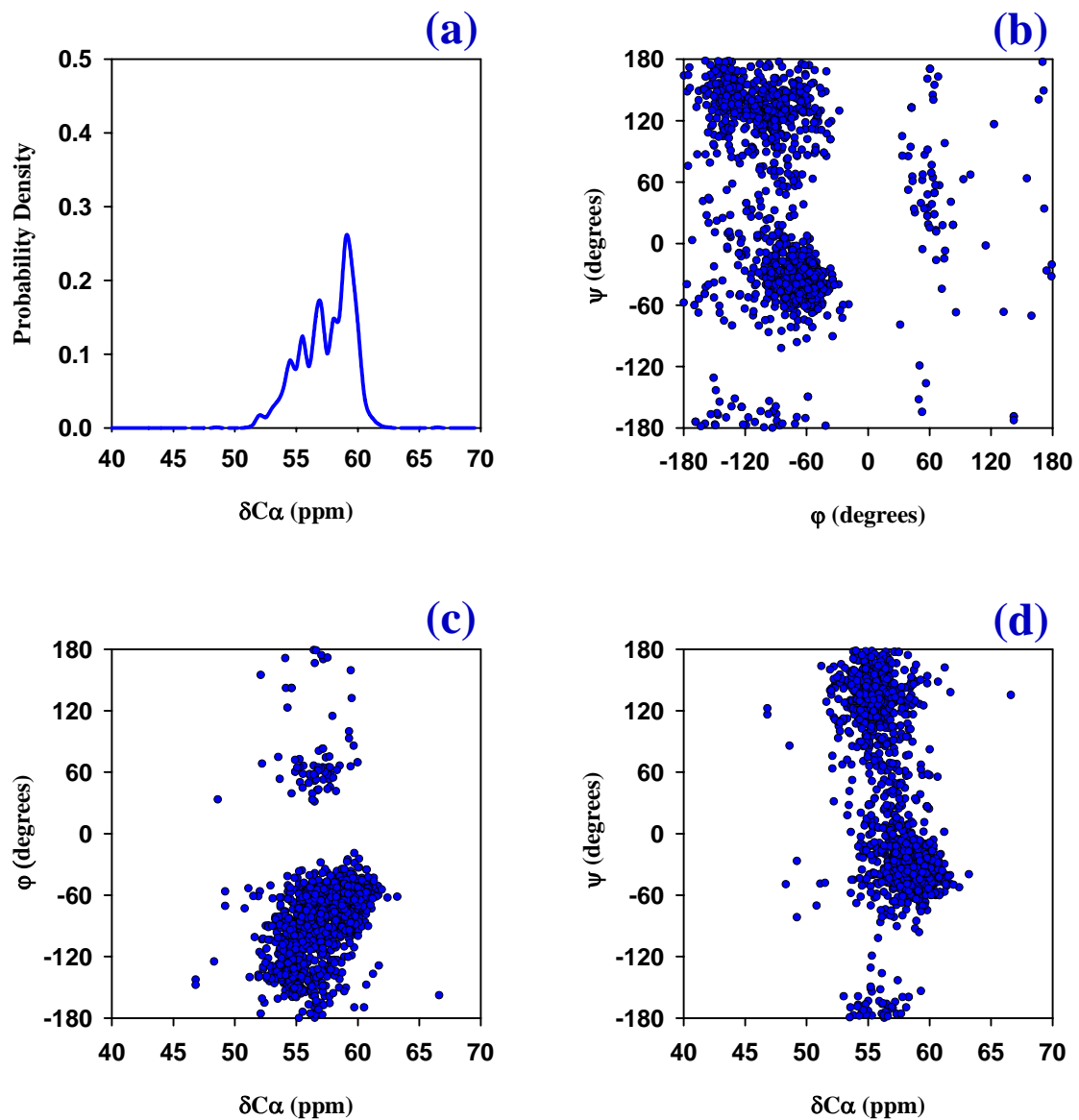


Figure 3.6 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for glutamate.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Glutamine

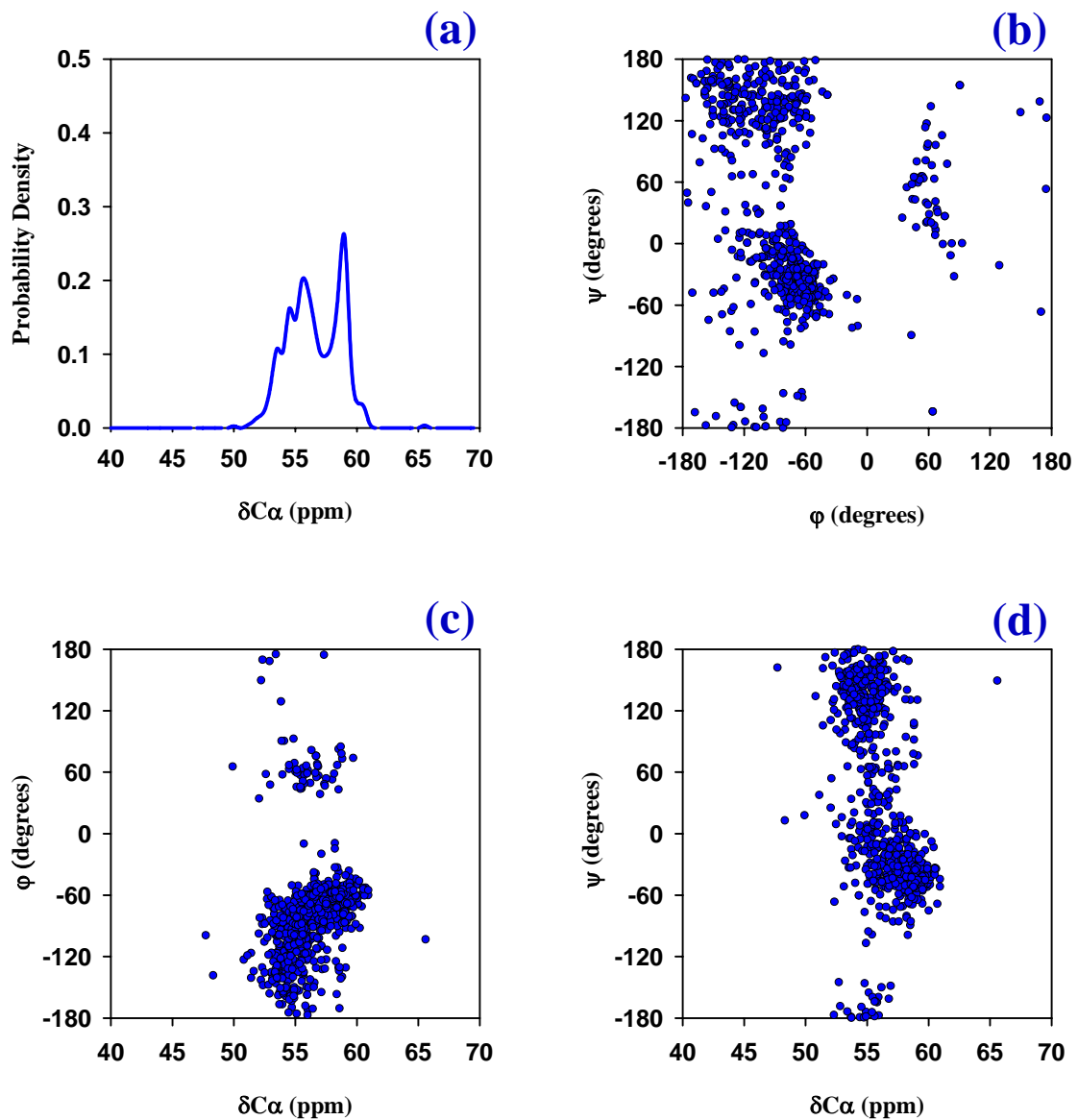


Figure 3.7 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for glutamine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Glycine

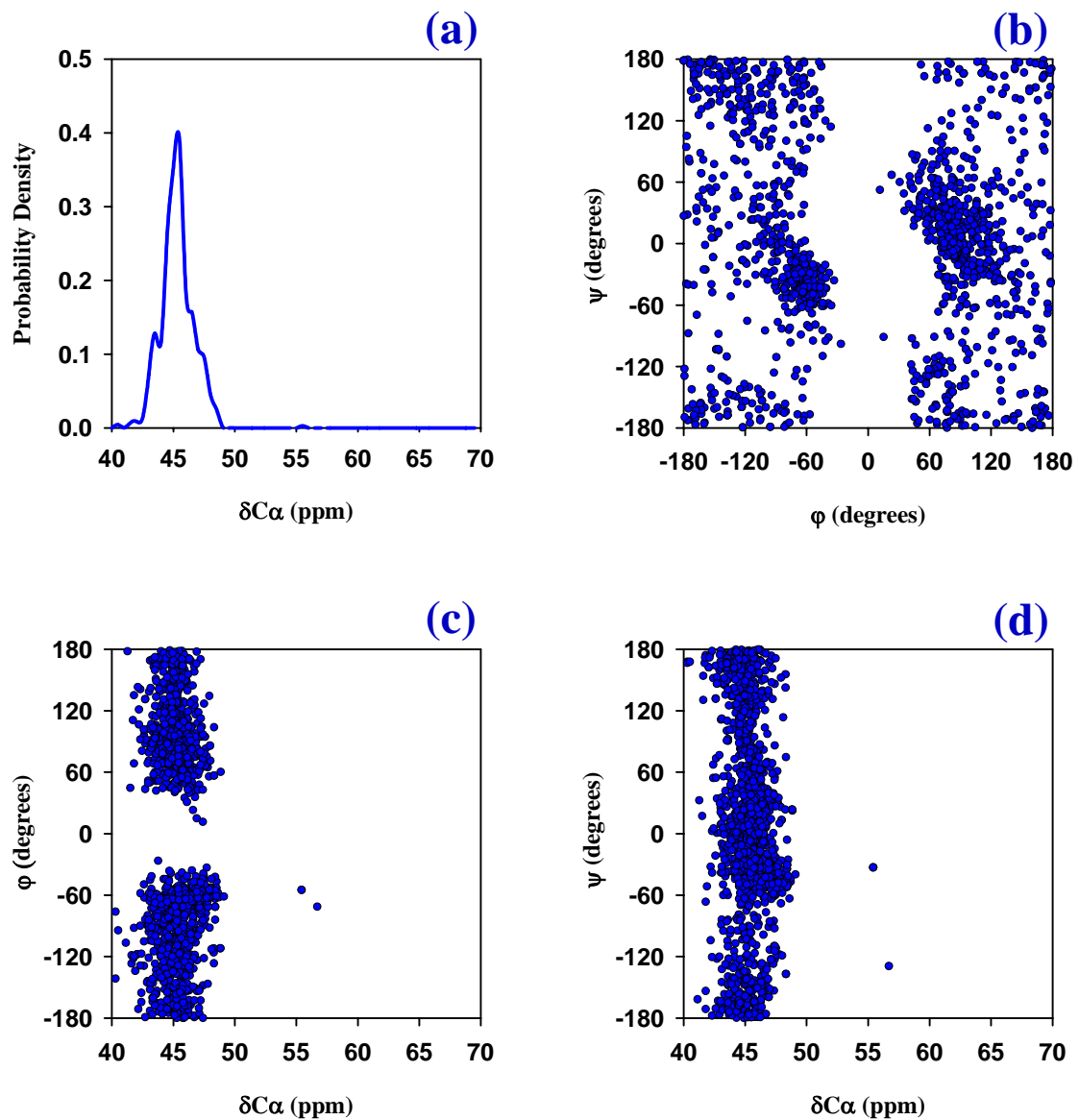


Figure 3.8 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for glycine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Histidine

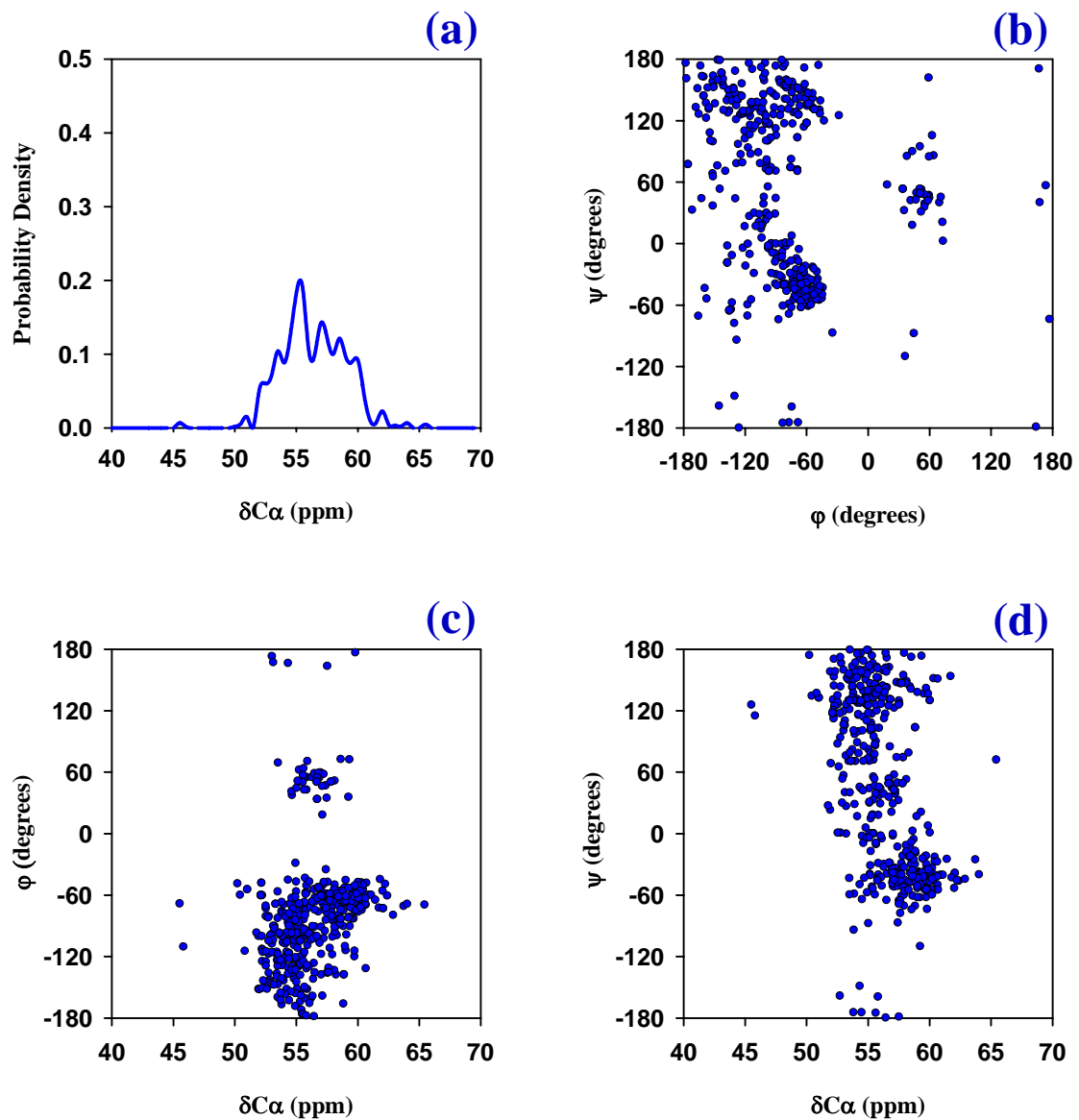


Figure 3.9 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for histidine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Isoleucine

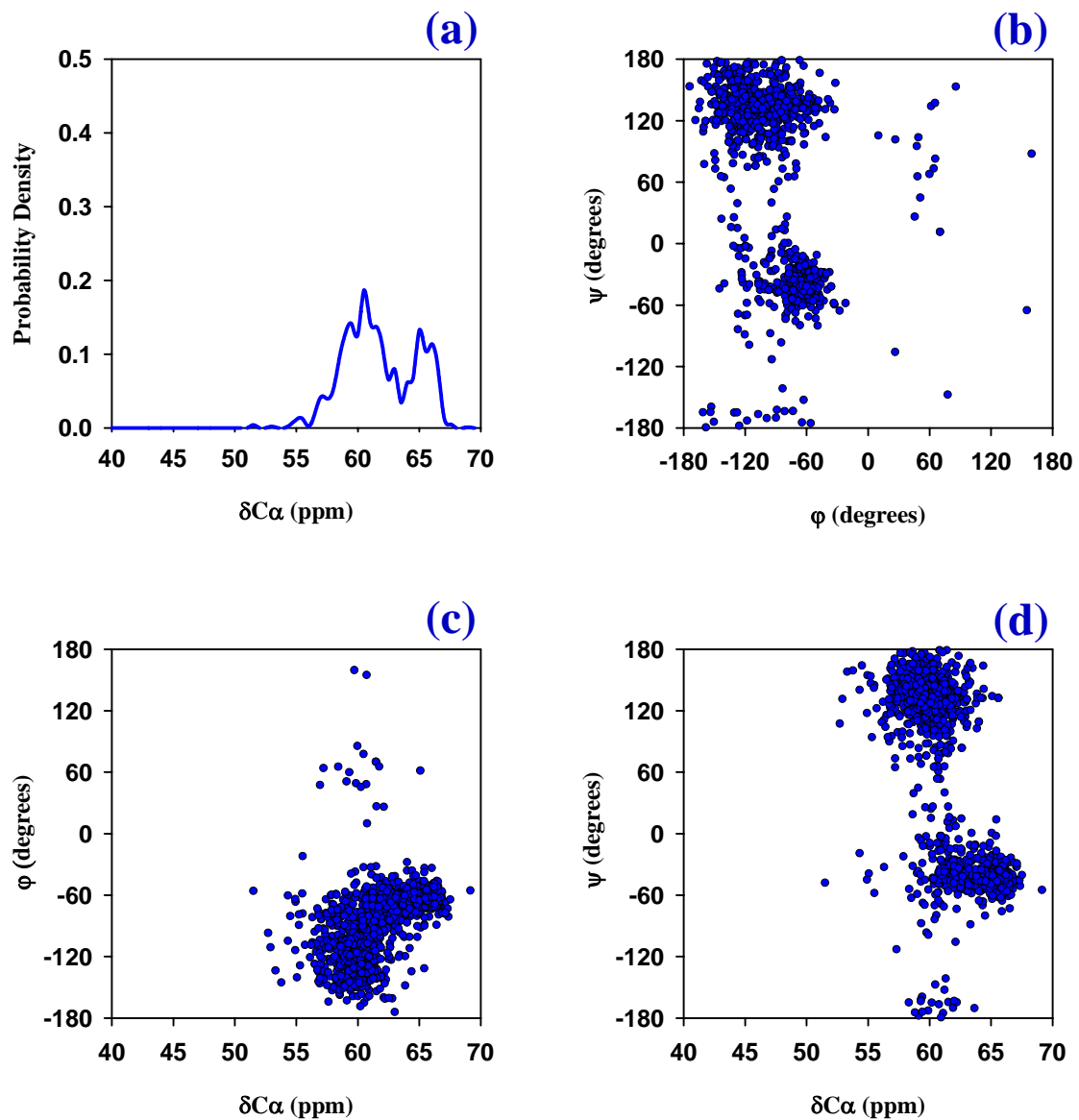


Figure 3.10 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for isoleucine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Leucine

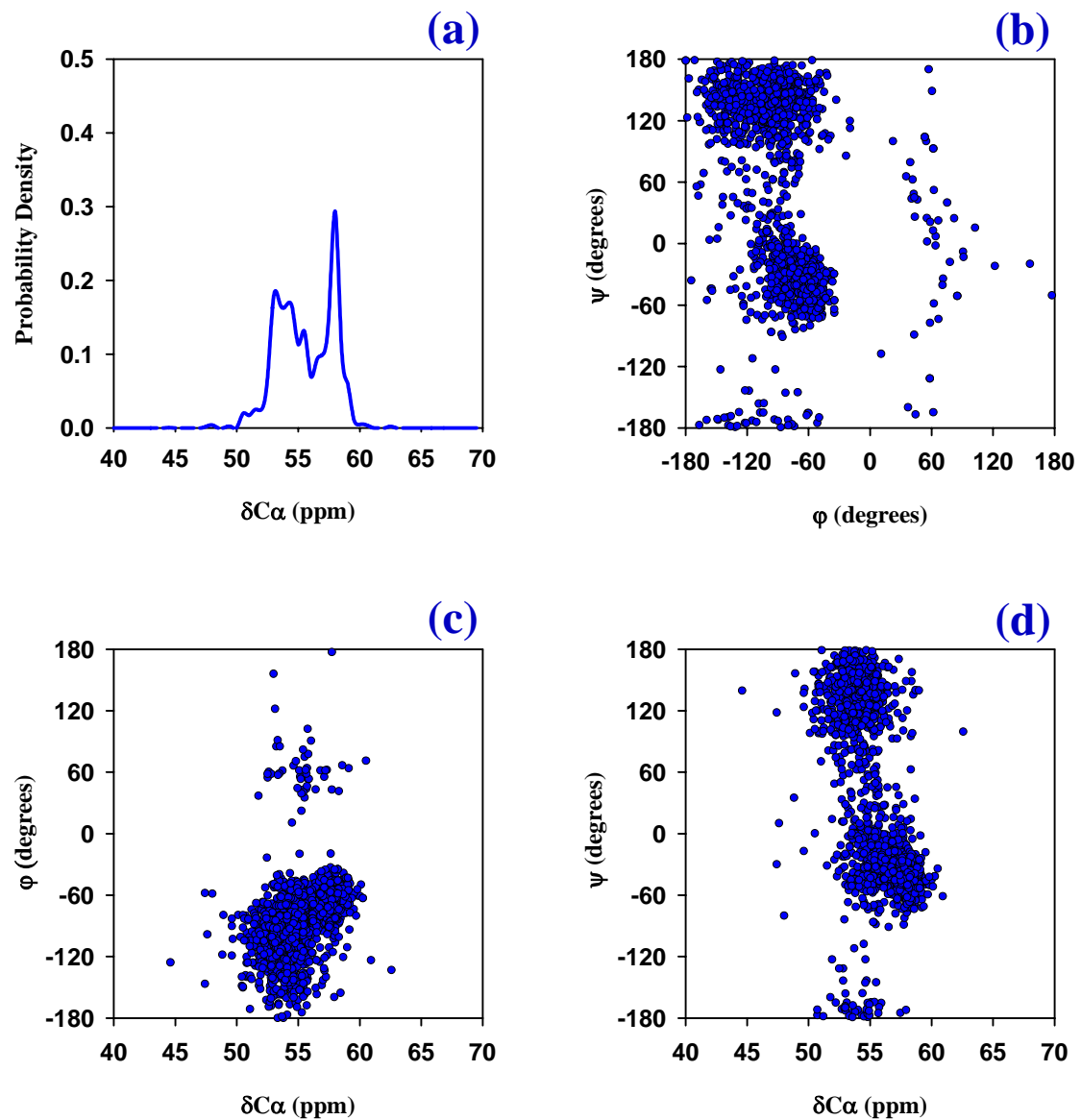


Figure 3.11 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for leucine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Lysine

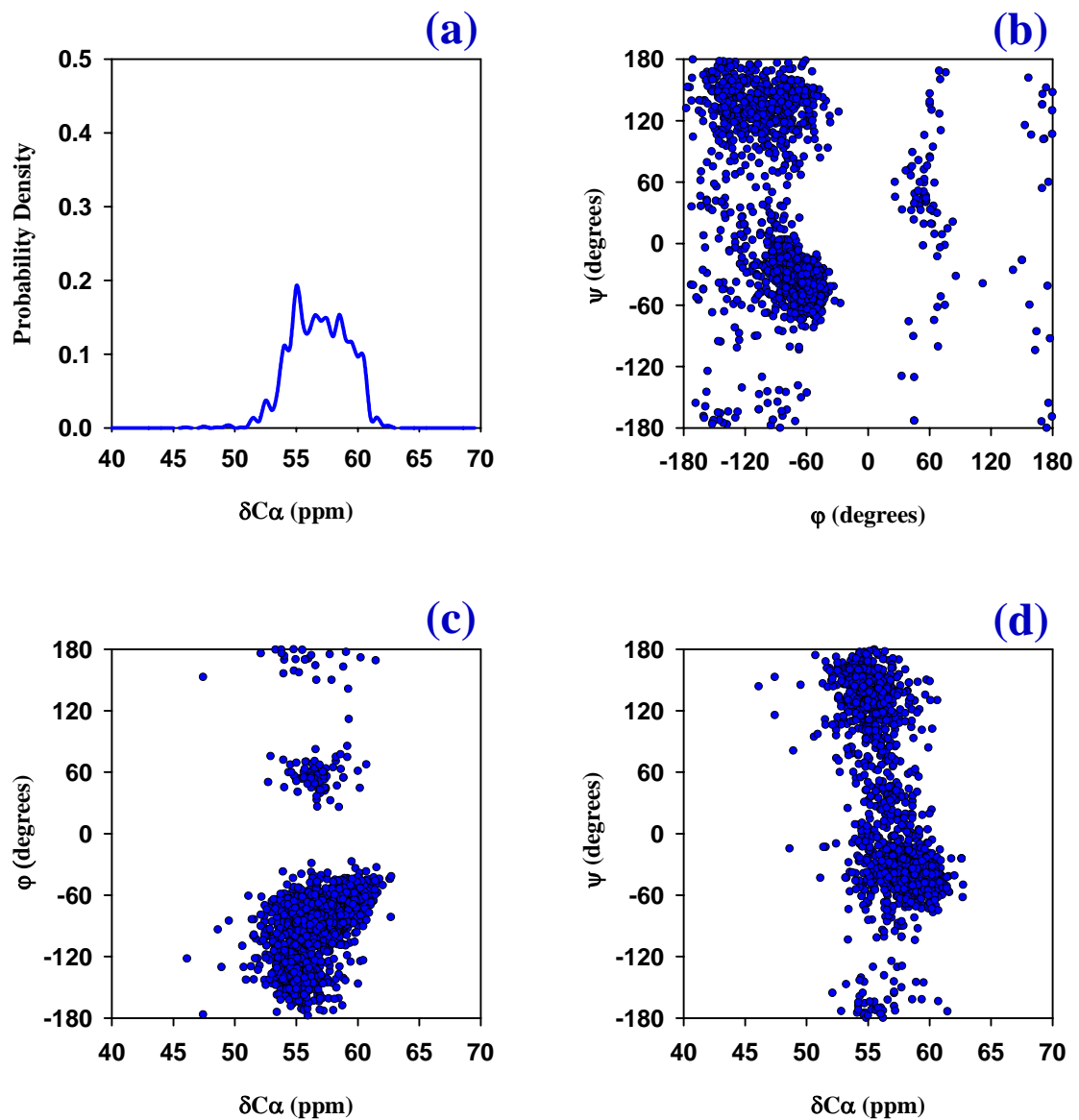


Figure 3.12 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for lysine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Methionine

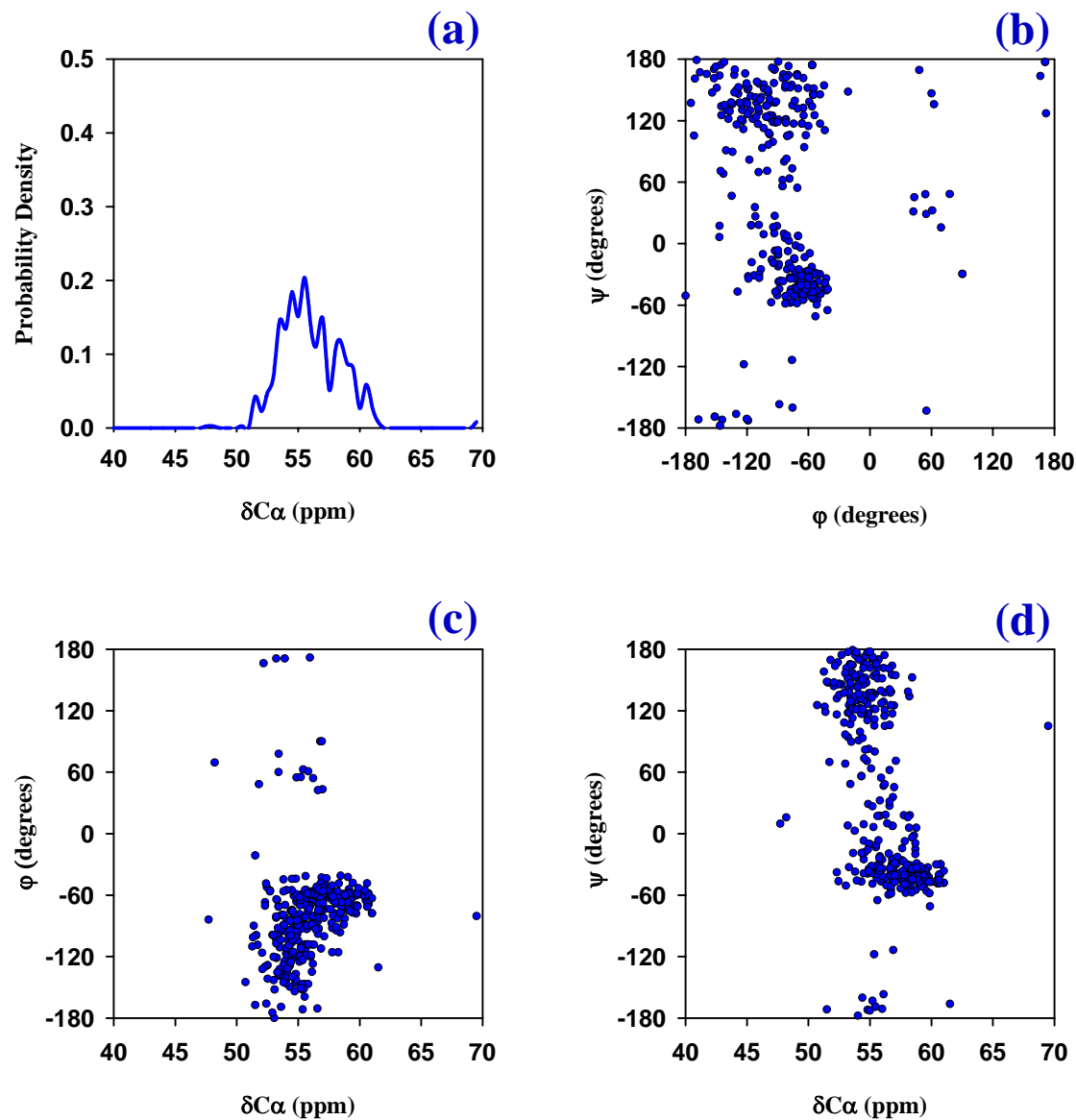


Figure 3.13 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for methionine.

(a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.

(b) Ramachandran plot of all torsion angles in the data set.

(c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.

(d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Phenylalanine

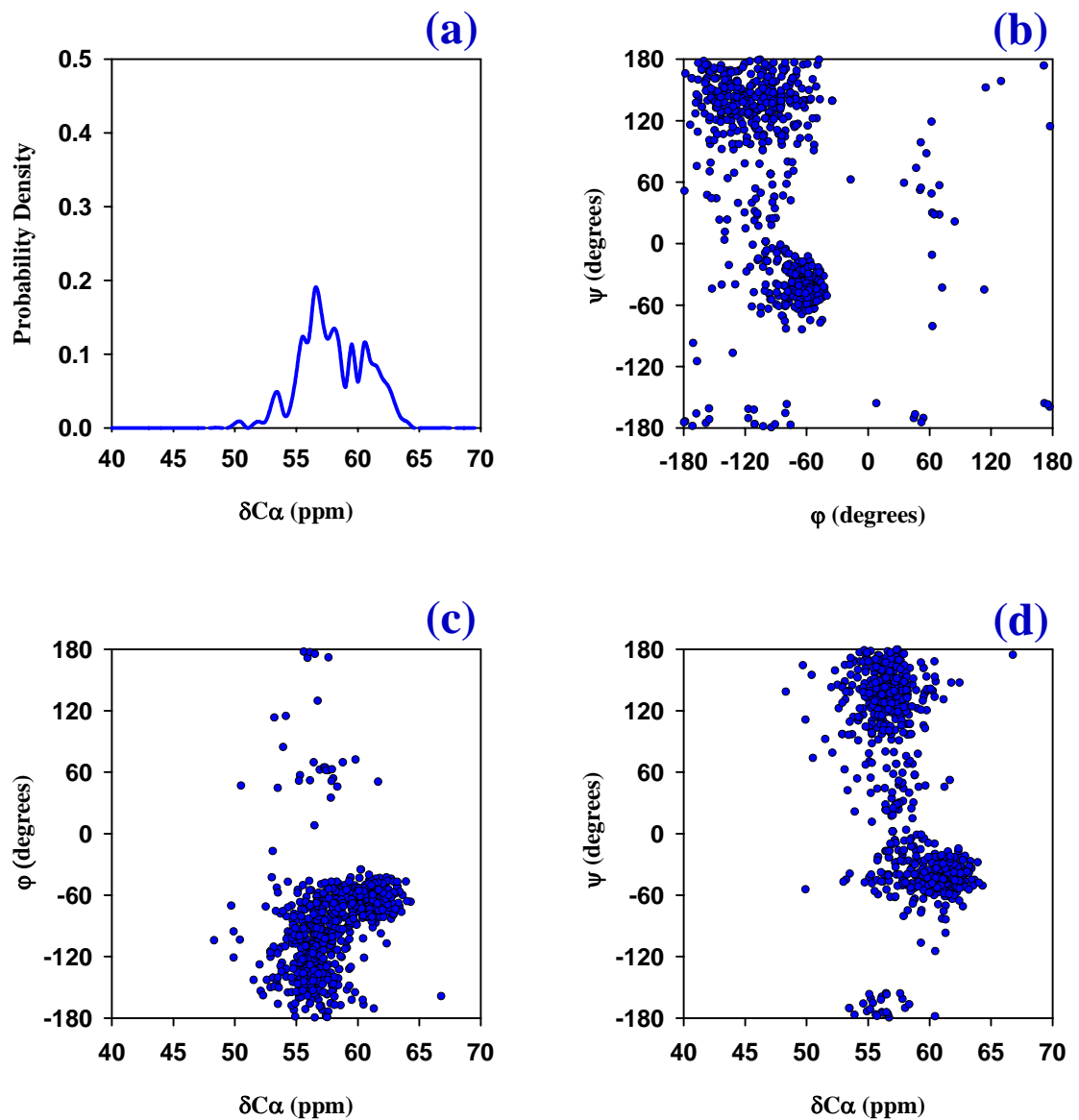


Figure 3.14 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for phenylalanine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Proline

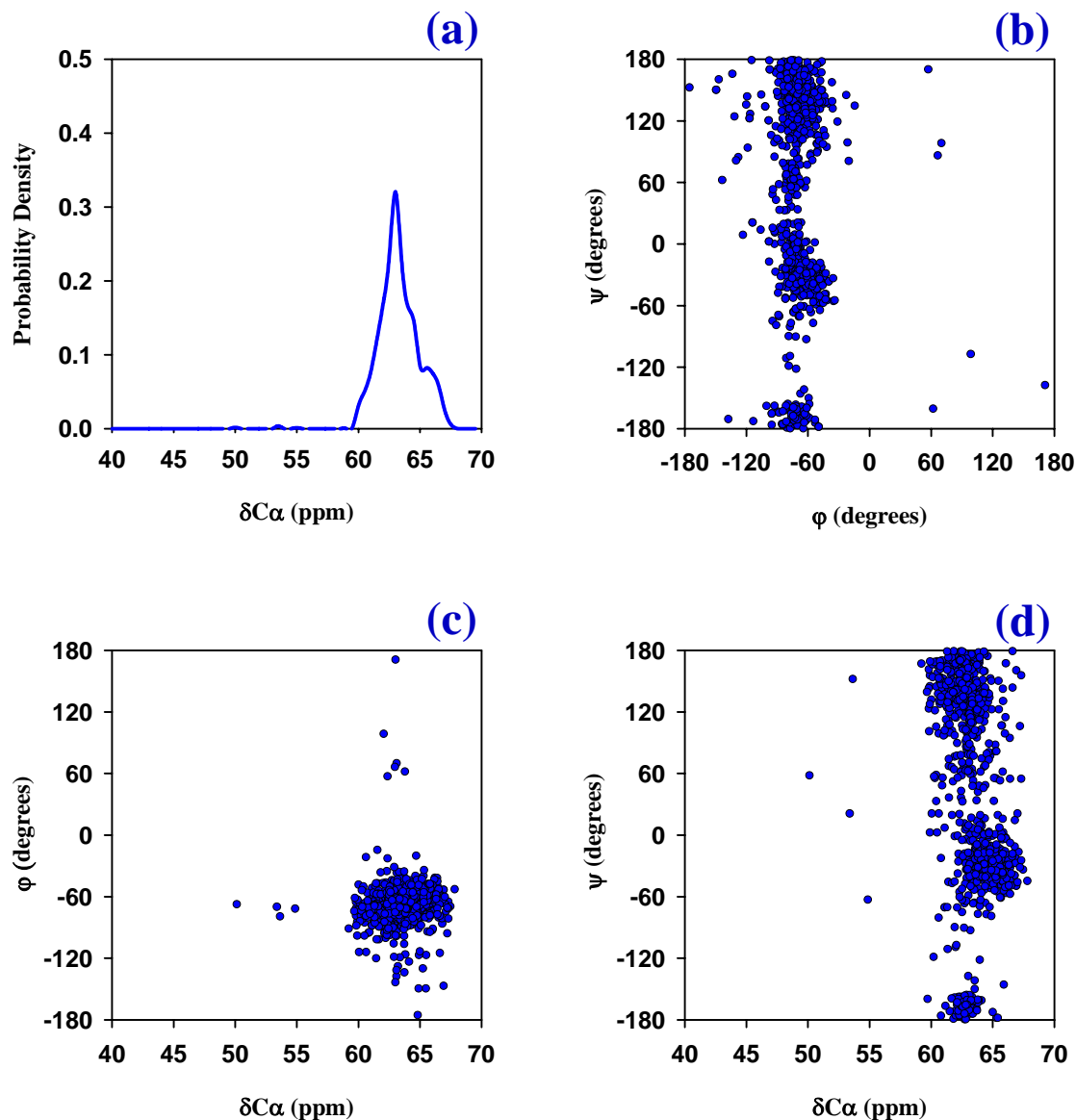


Figure 3.15 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for proline.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Serine

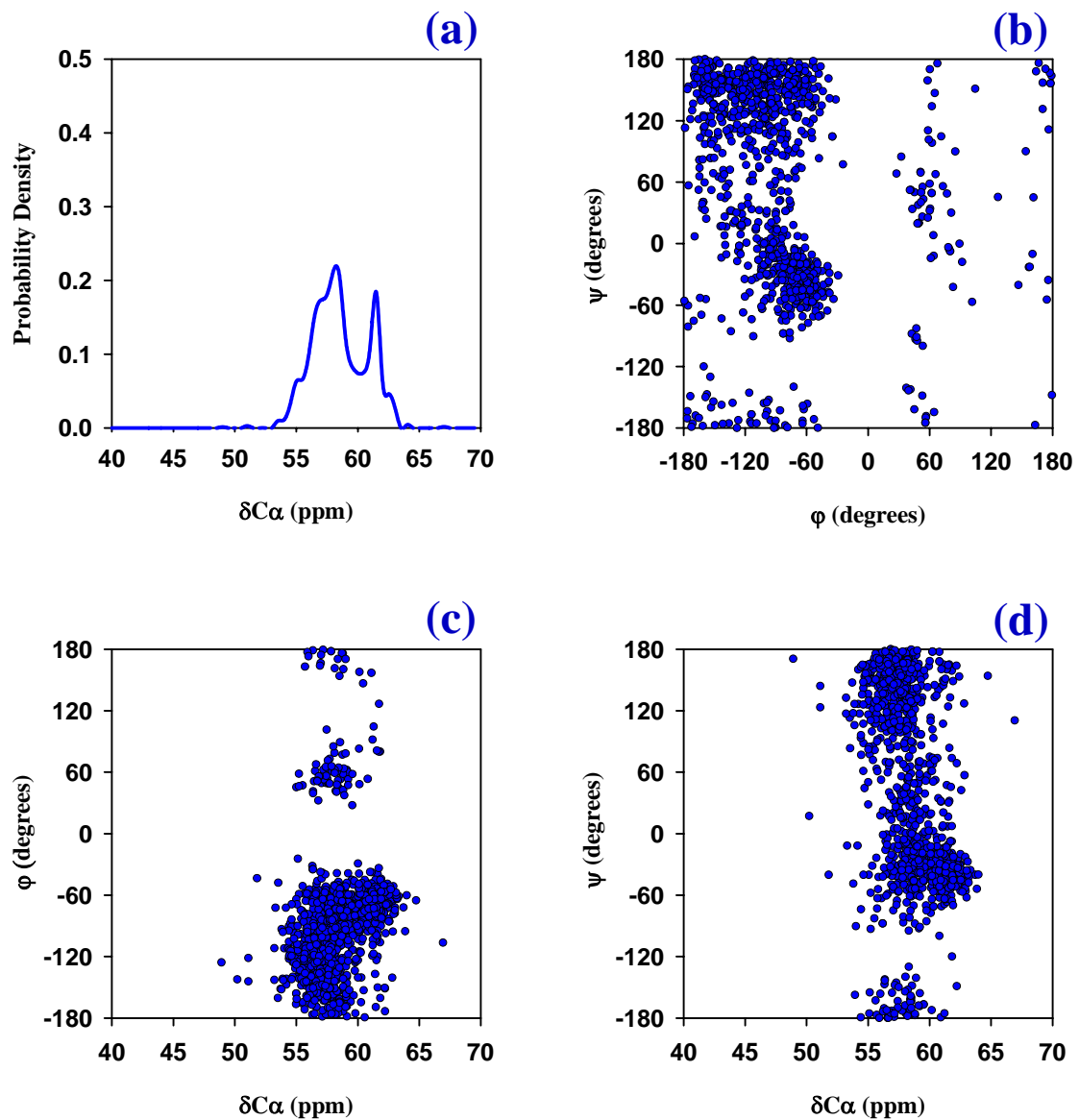


Figure 3.16 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for serine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Threonine

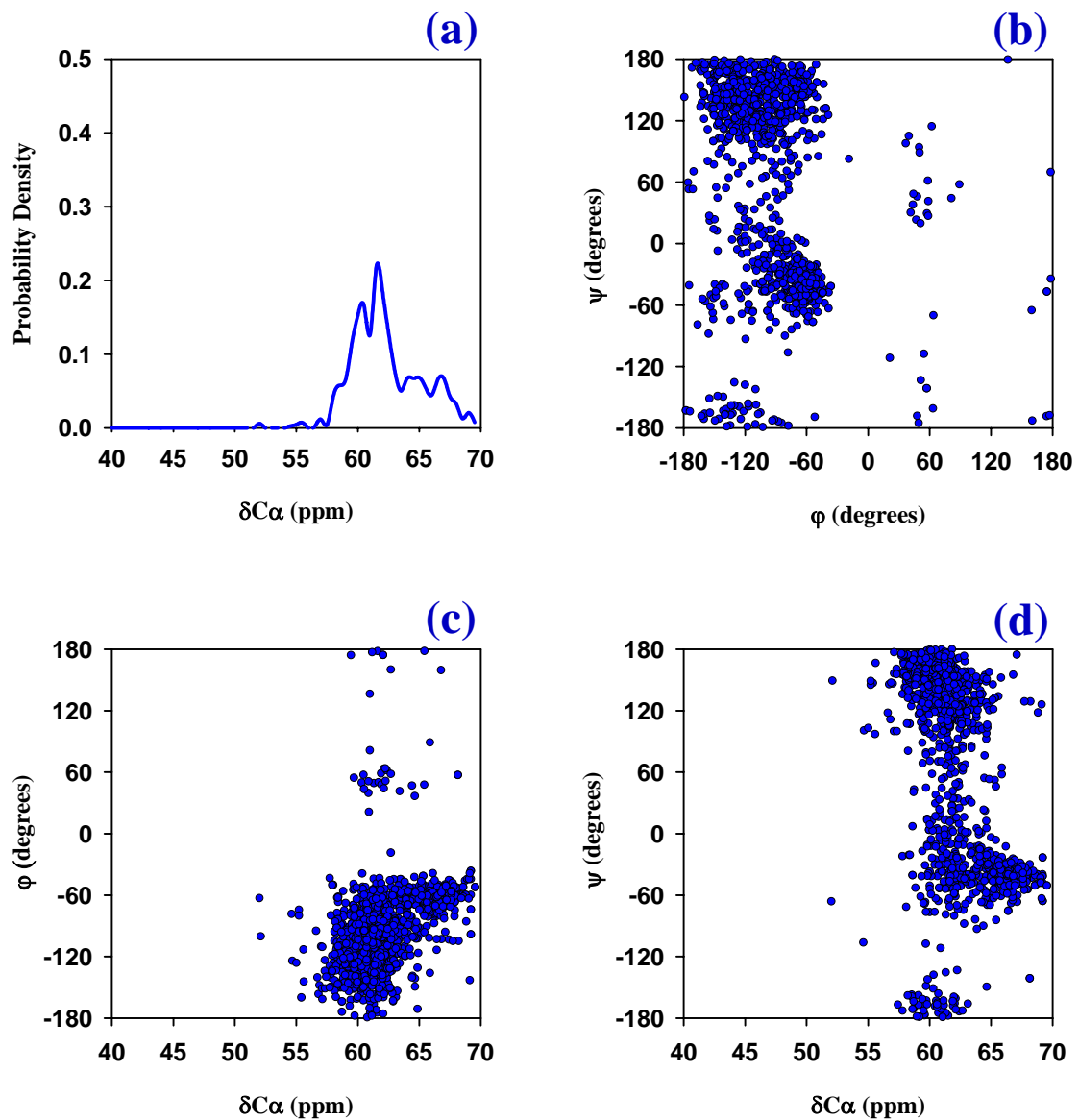


Figure 3.17 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for threonine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Tryptophan

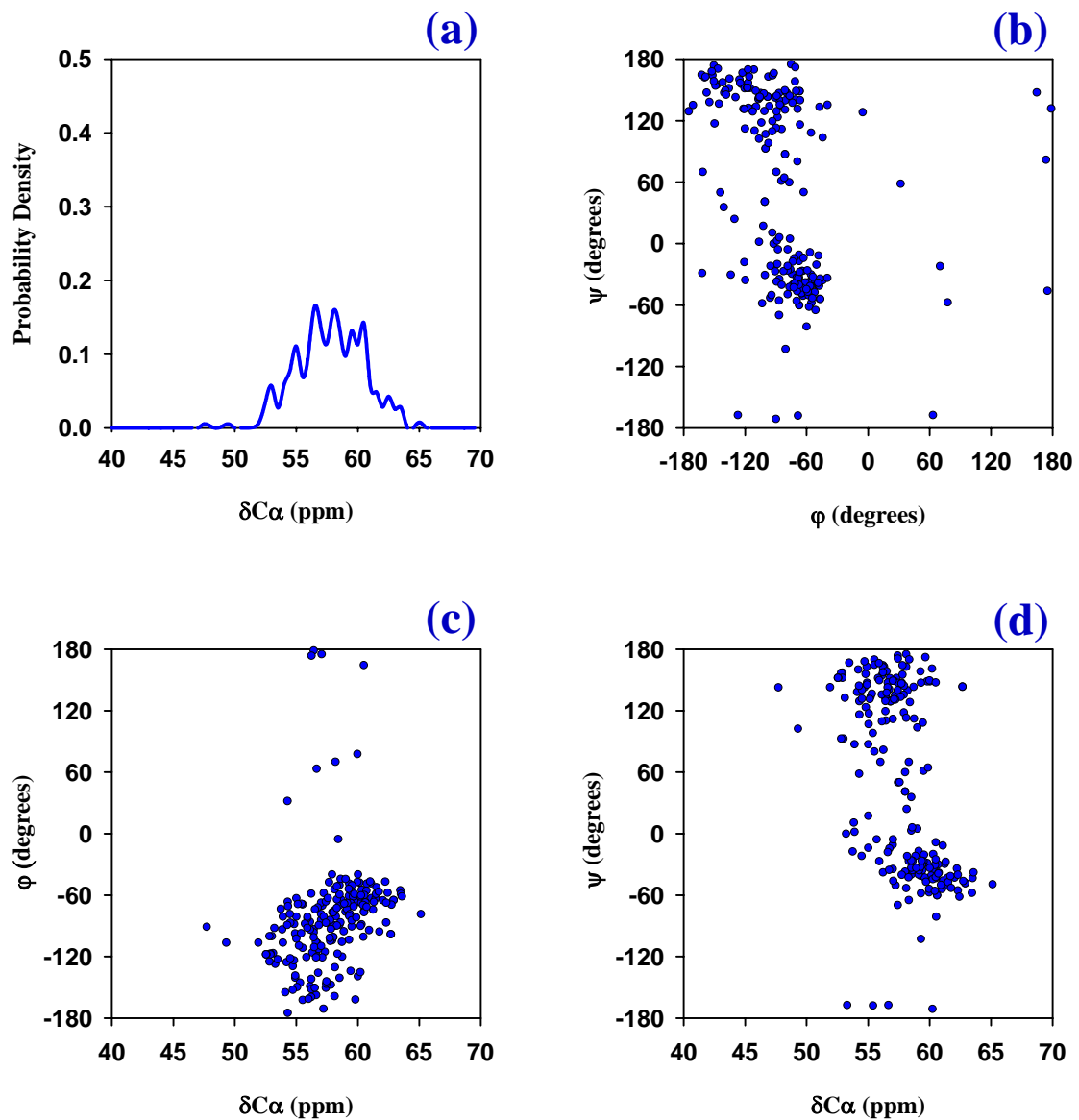


Figure 3.18 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for tryptophan.

(a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.

(b) Ramachandran plot of all torsion angles in the data set.

(c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.

(d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Tyrosine

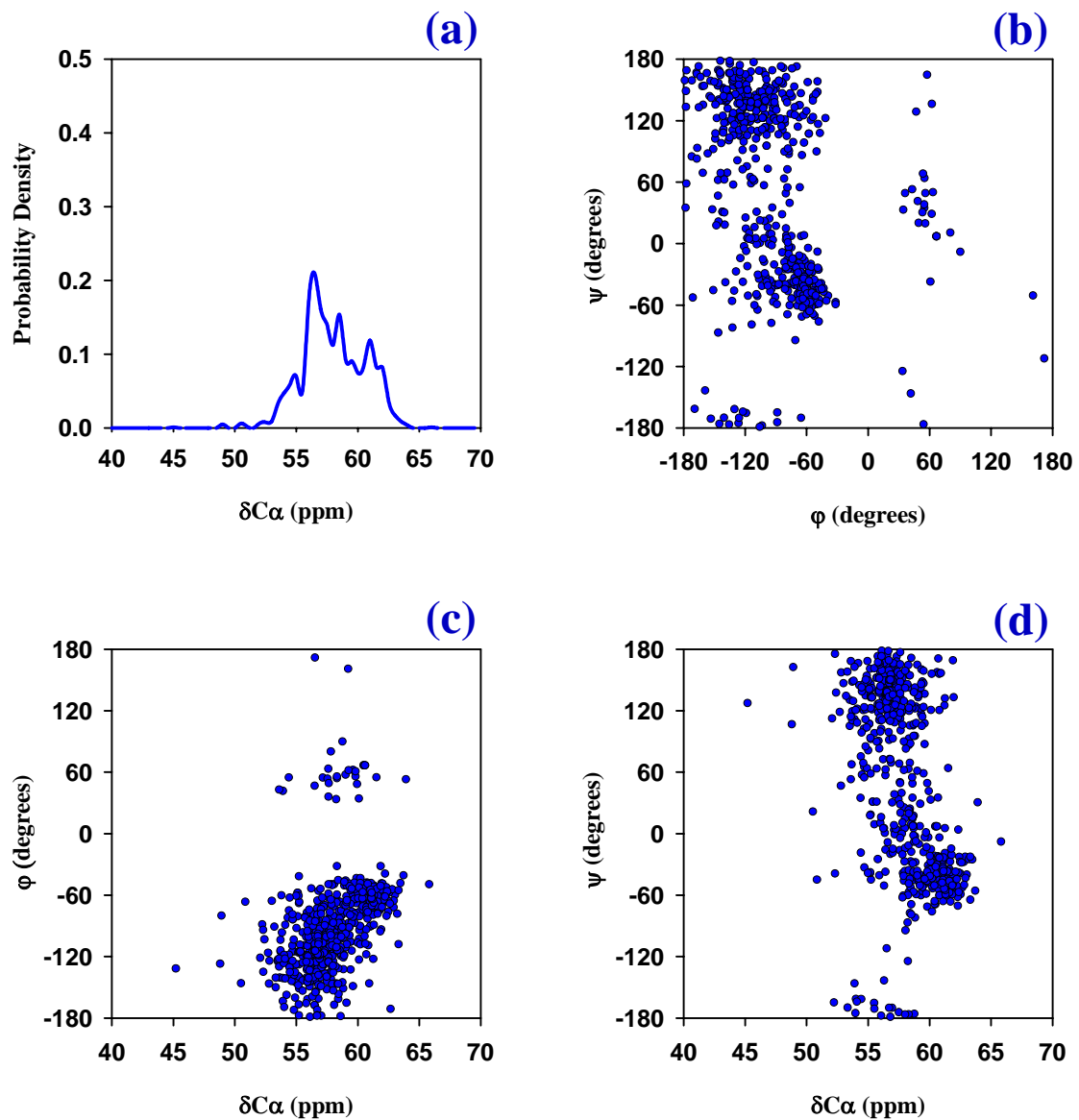


Figure 3.19 Distribution of ($\delta C\alpha$, ϕ , ψ) data obtained for tyrosine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

Valine

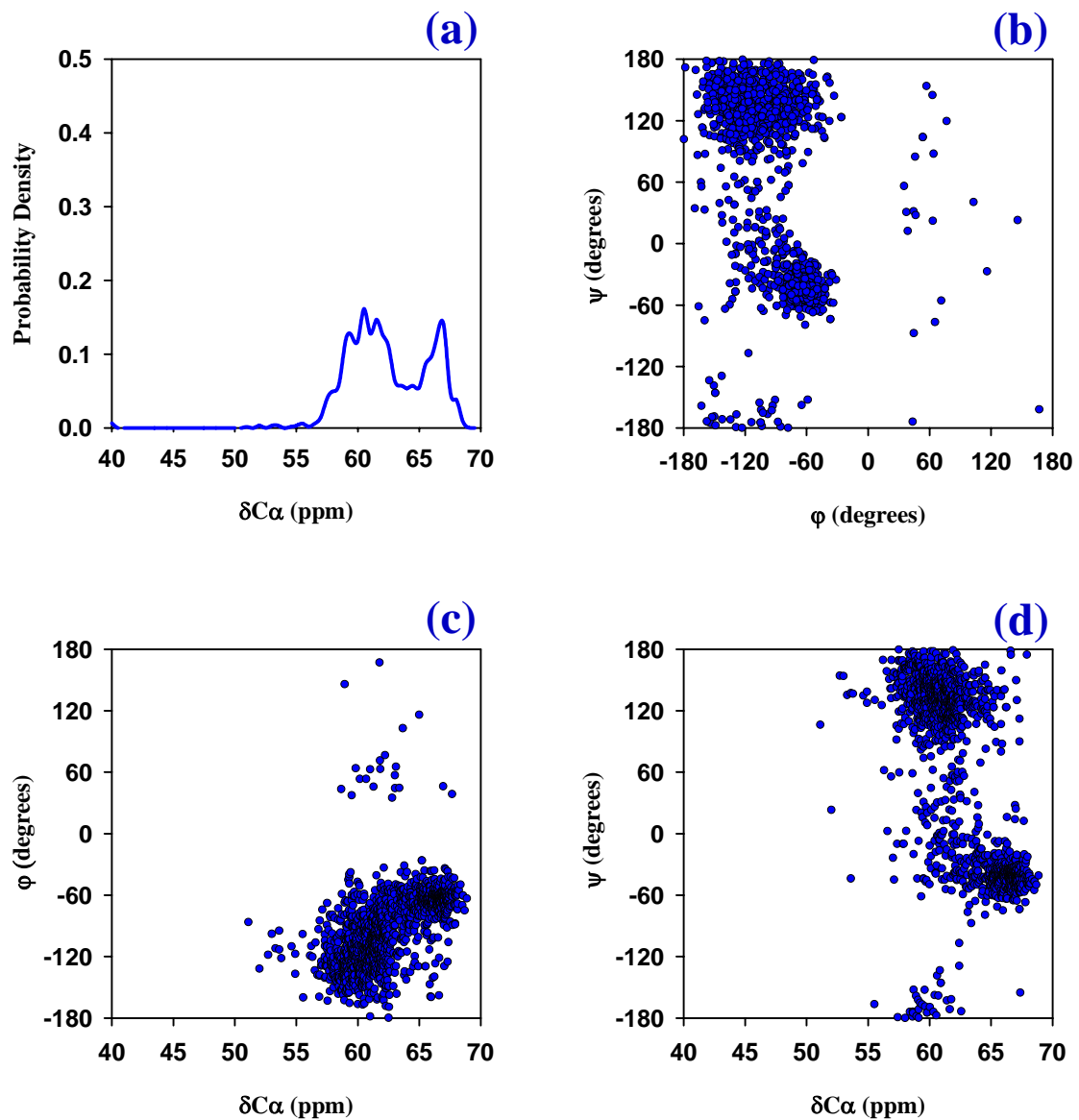


Figure 3.20 Distribution of $(\delta C\alpha, \phi, \psi)$ data obtained for valine.
 (a) Plot of an estimate of the probability density function which describes the distribution of $\delta C\alpha$ in our data set.
 (b) Ramachandran plot of all torsion angles in the data set.
 (c) Scatter plot of the distribution of ϕ with respect to $\delta C\alpha$.
 (d) Scatter plot of the distribution of ψ with respect to $\delta C\alpha$.

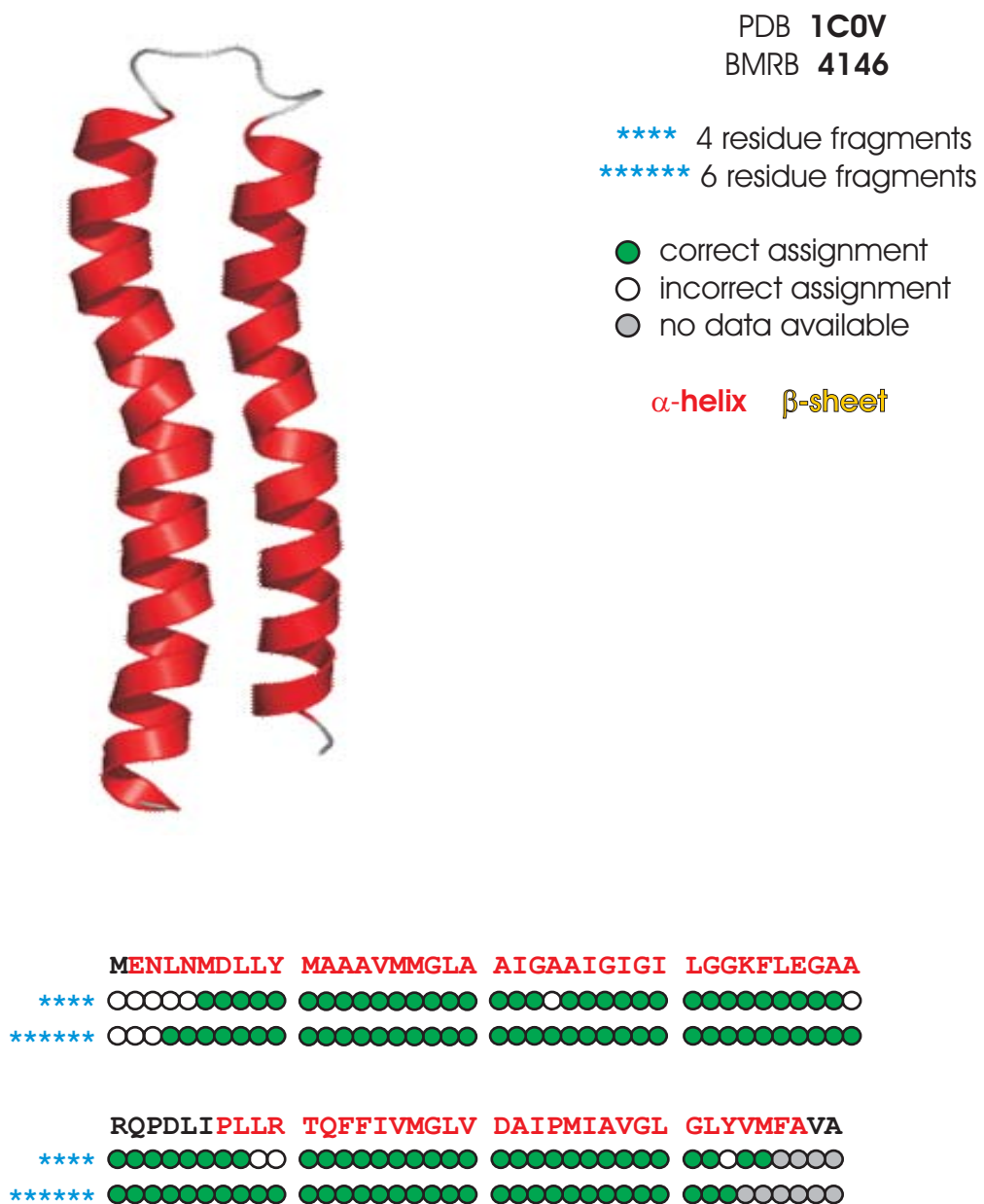


Figure 3.22 Assignment results for the protein 1C0V using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

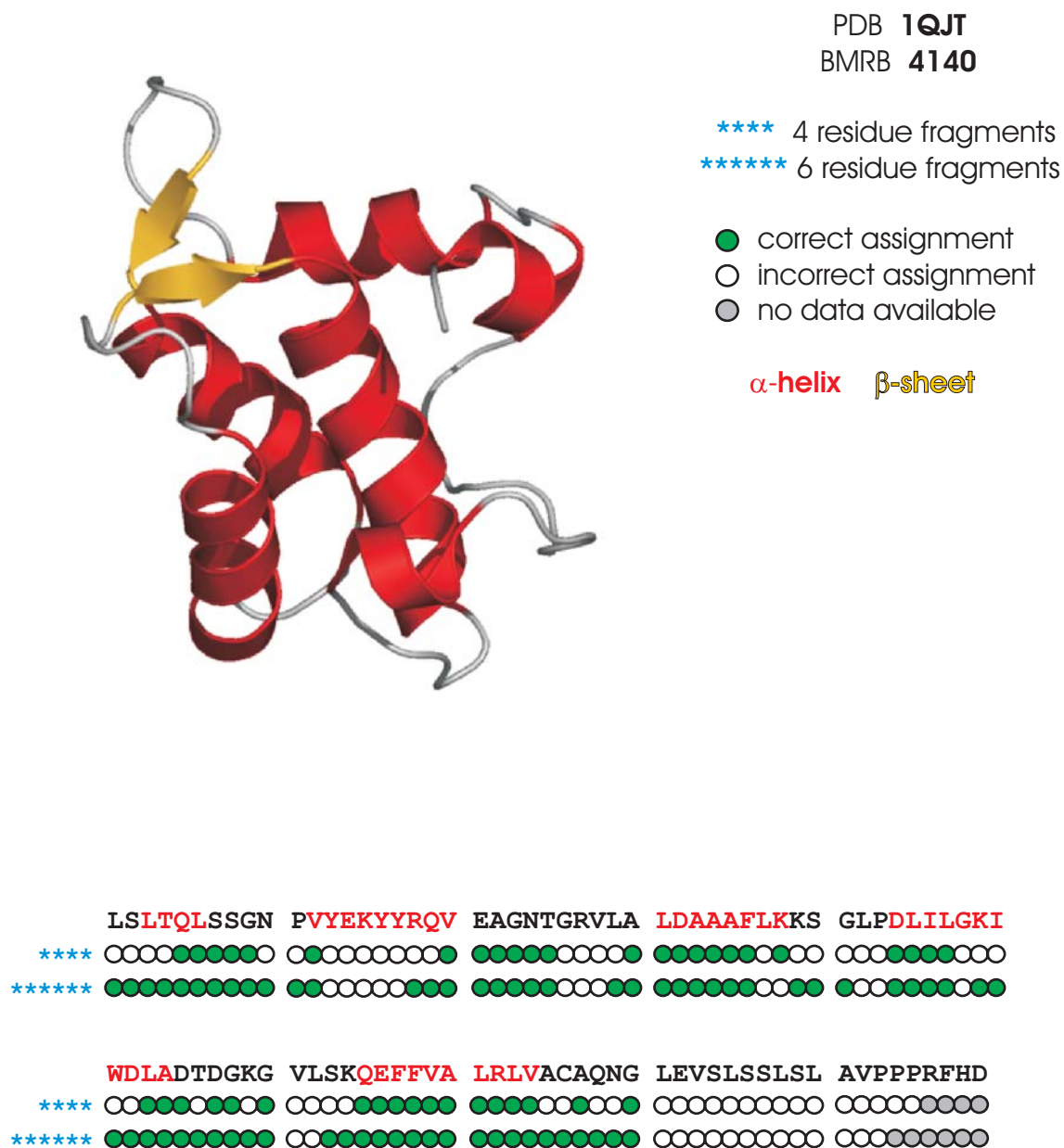


Figure 3.23 Assignment results for the protein 1QJT using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

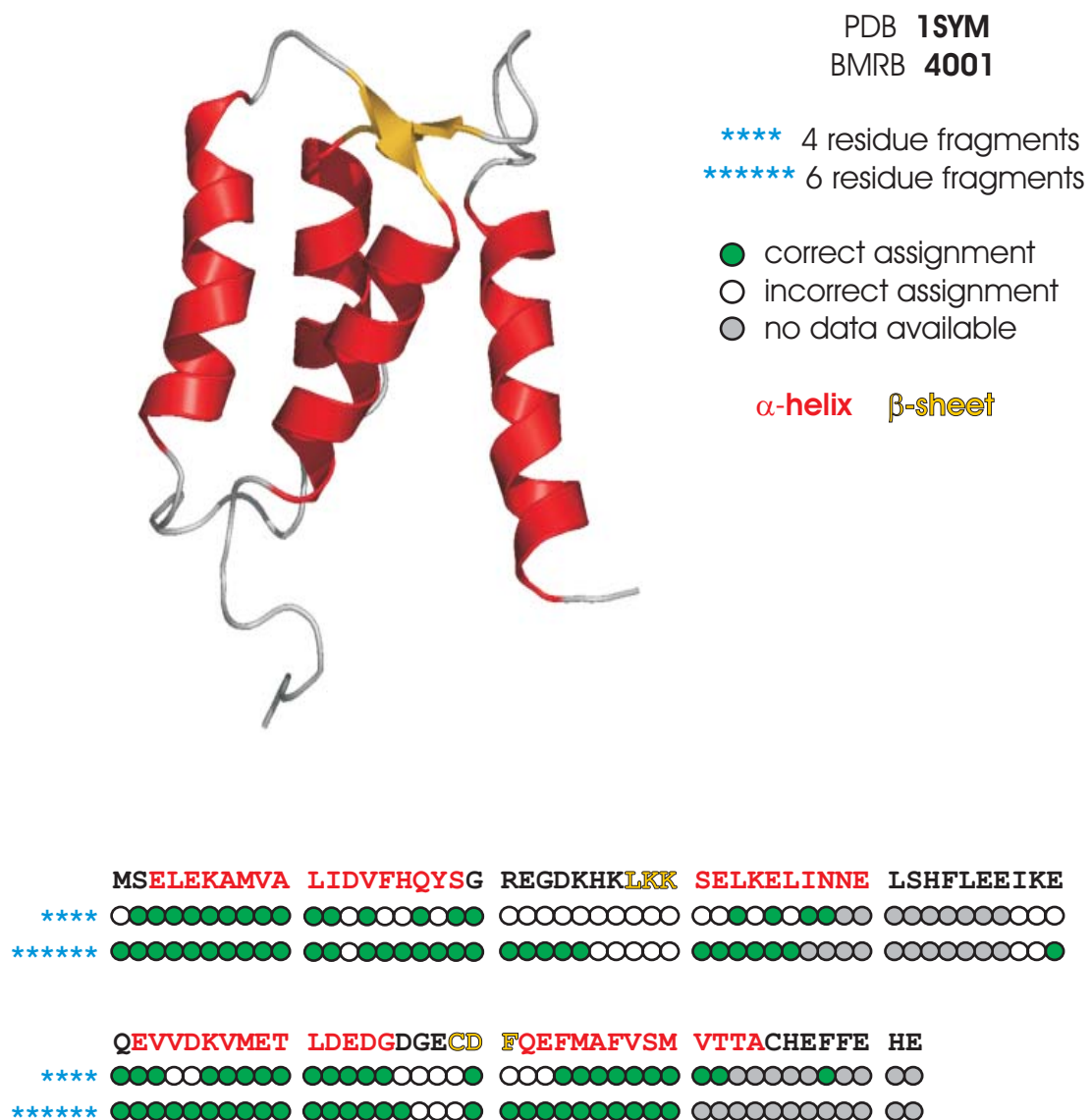


Figure 3.24 Assignment results for the protein 1SYM using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

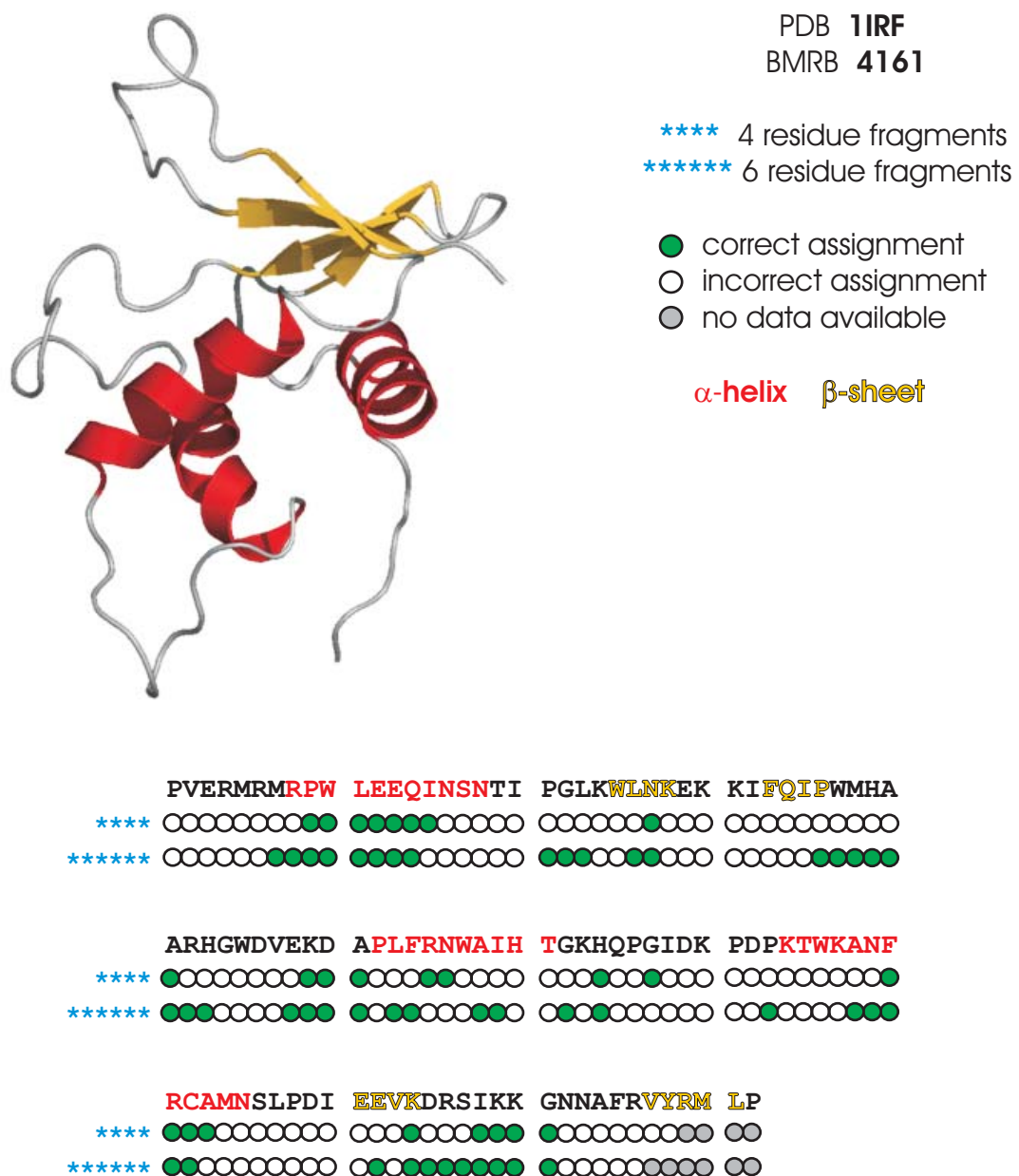


Figure 3.25 Assignment results for the protein 1IRF using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

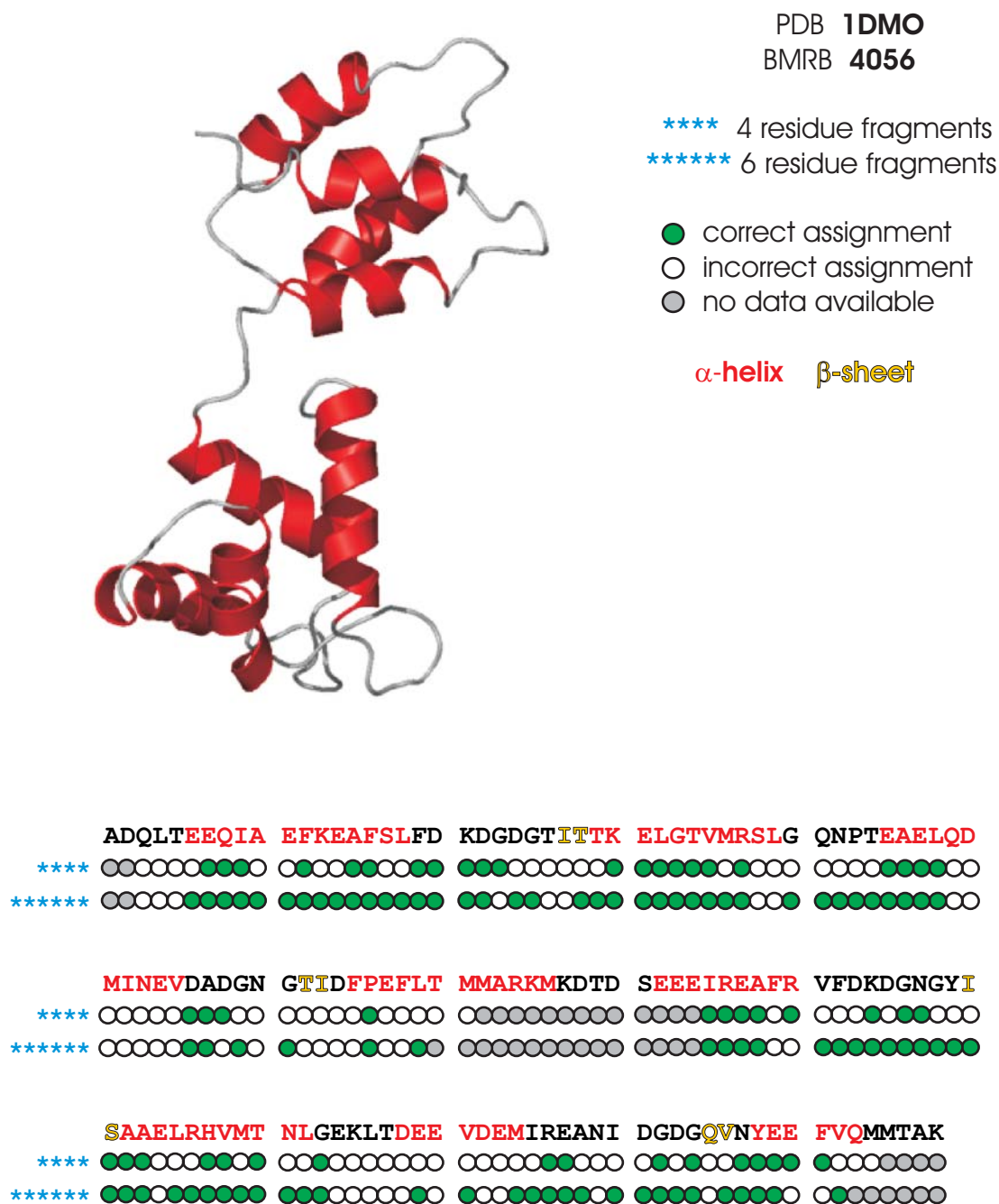


Figure 3.26 Assignment results for the protein 1DMO using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

PDB **1CDC**
BMRB **4109**

**** 4 residue fragments

***** 6 residue fragments

- correct assignment
- incorrect assignment
- no data available

α -helix β -sheet

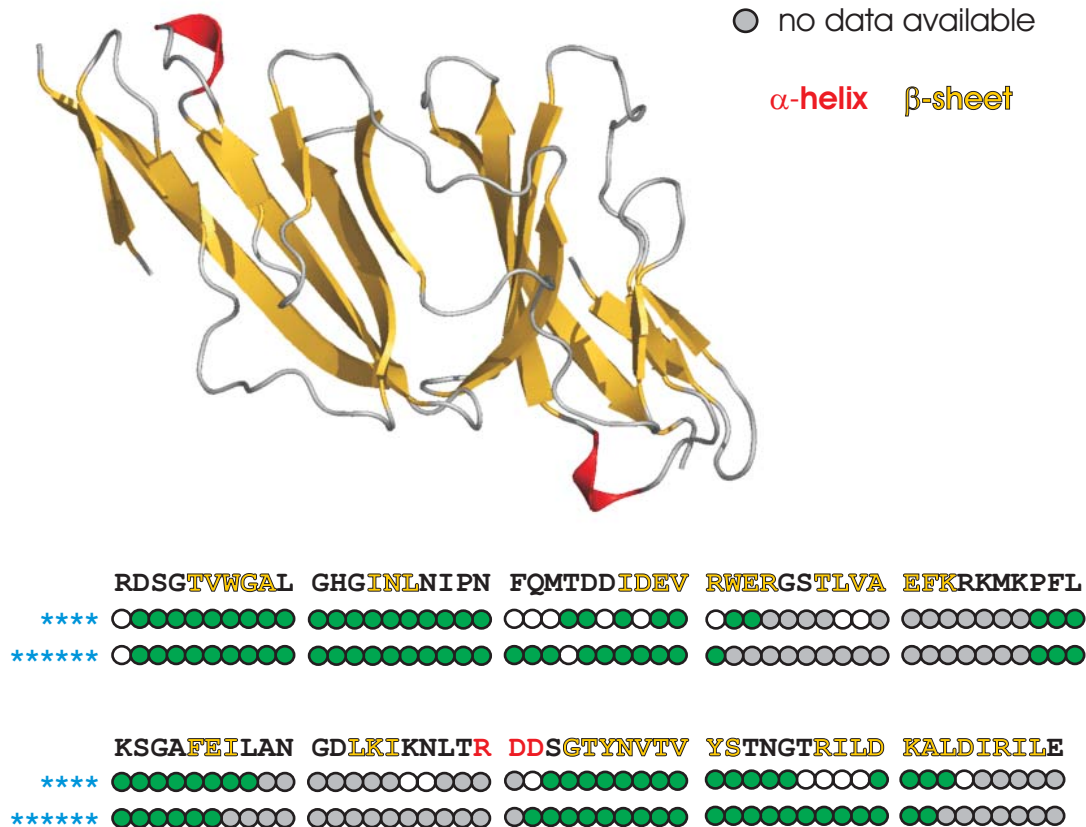


Figure 3.27 Assignment results for the protein 1CDC using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

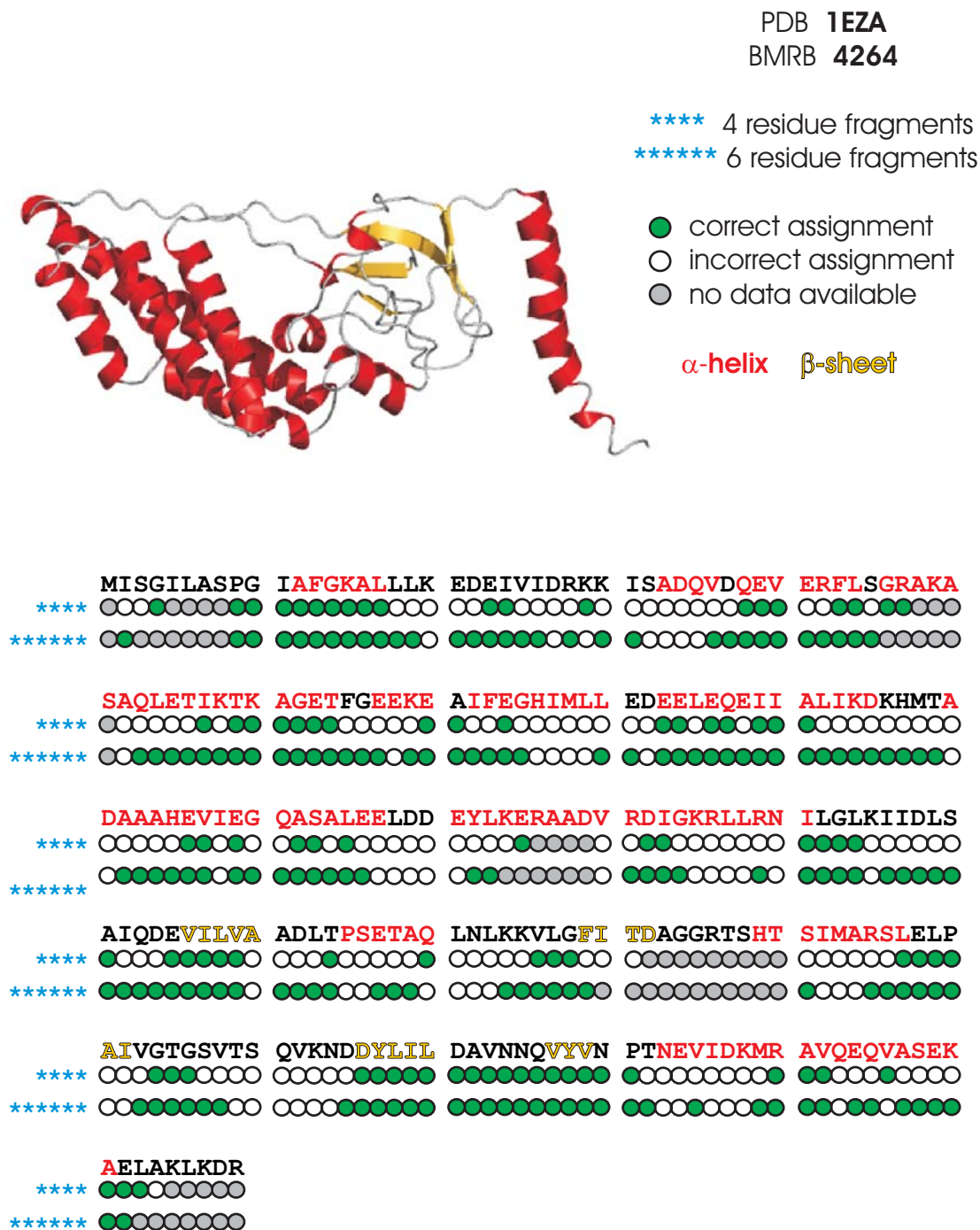


Figure 3.28 Assignment results for the protein 1EZA using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

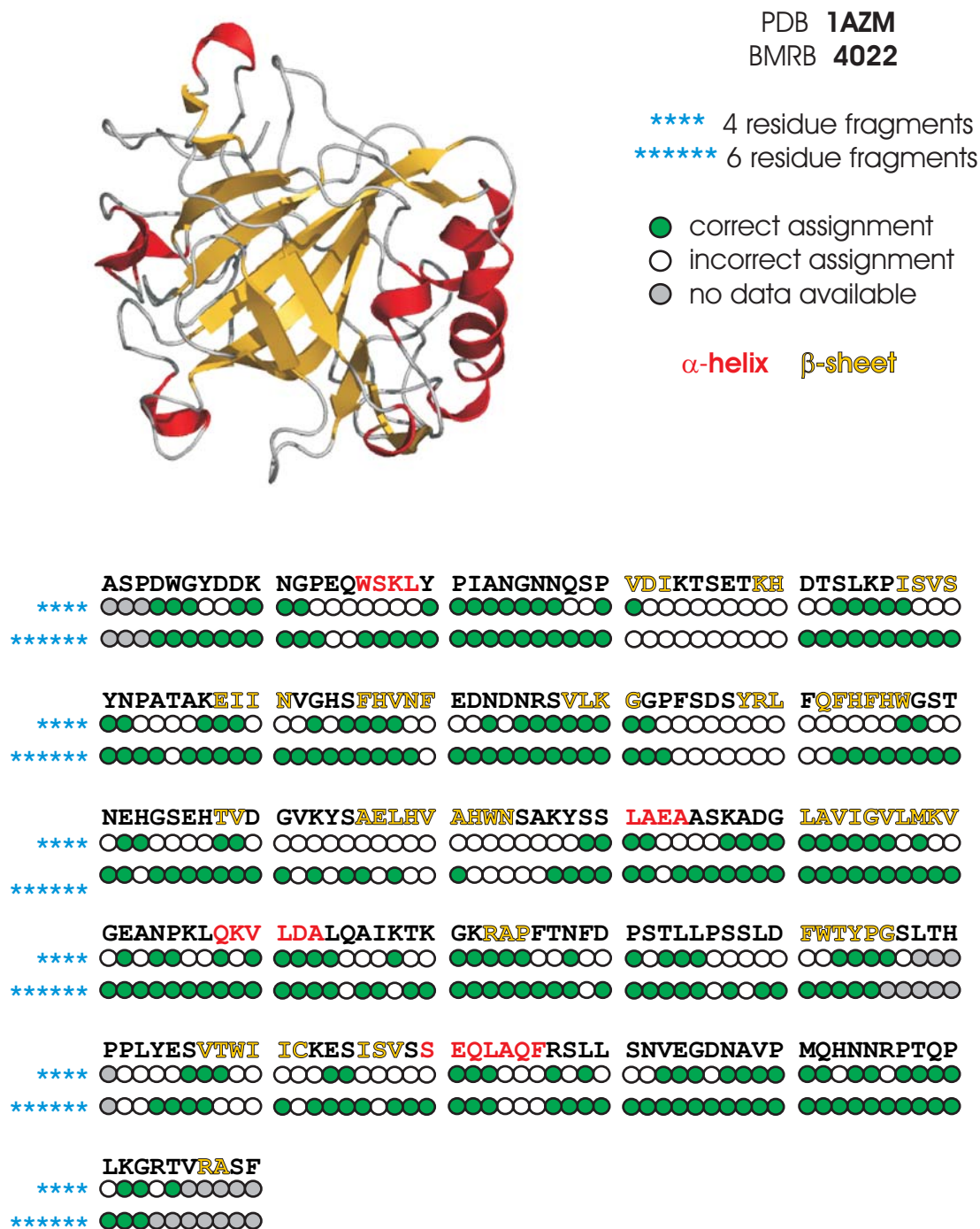


Figure 3.29 Assignment results for the protein 1AZM using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

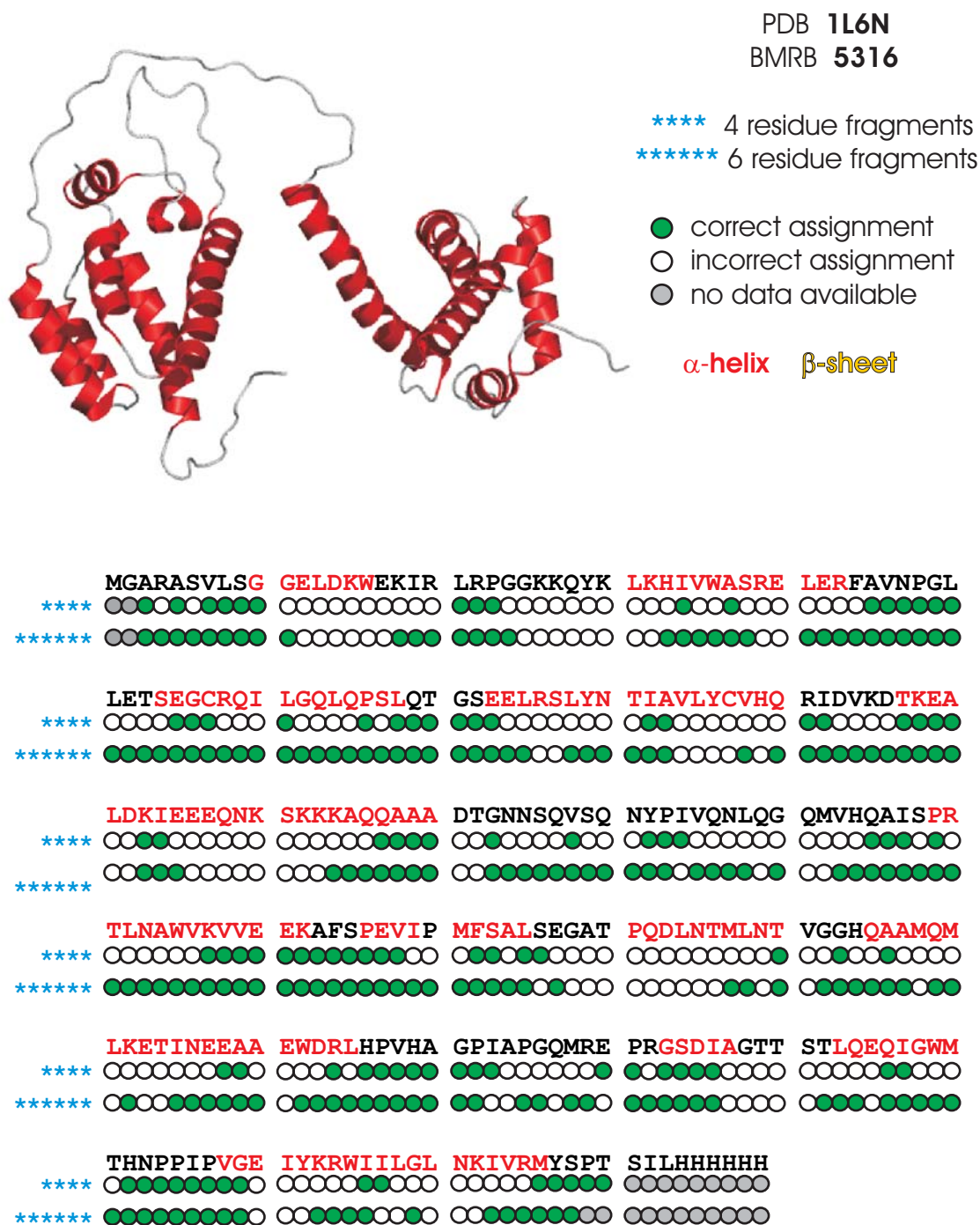


Figure 3.30 Assignment results for the protein 1L6N using fragment lengths of 4 or 6. In the structure β -sheets are golden and α -helices are red. The sequence is color coded to match the structure. The circles below the sequence indicate the position of the first residue in the fragment which should be assigned to that position. Correct assignments are shown as green circles while incorrect assignments are shown as uncolored circles. If no data was available for a particular fragment it is indicated as a gray circle.

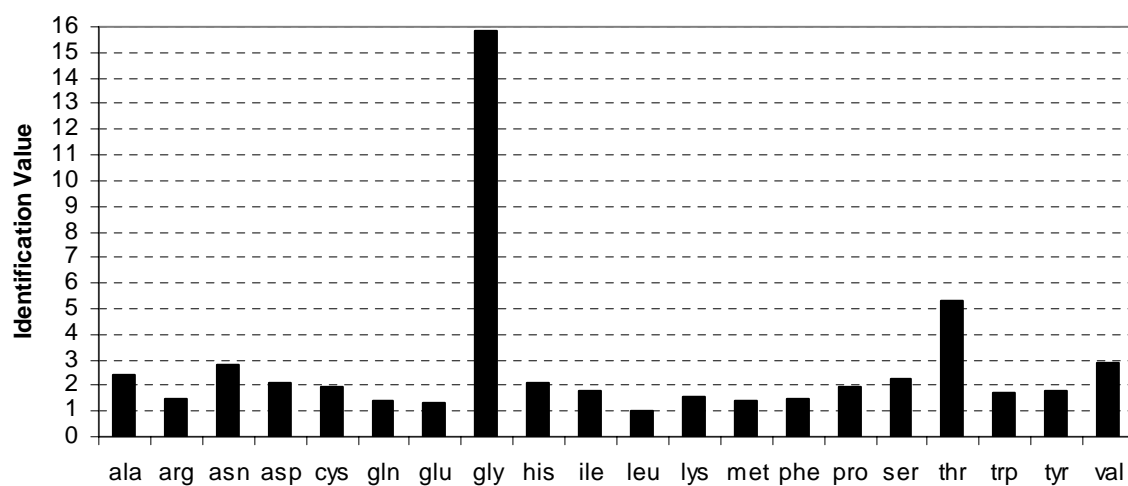


Figure 3.31. Value of $^{13}\text{C}\alpha$, ϕ , ψ in identifying an isolated single amino acid. All values are relative to the lowest value (leucine) and are multiples of that value.

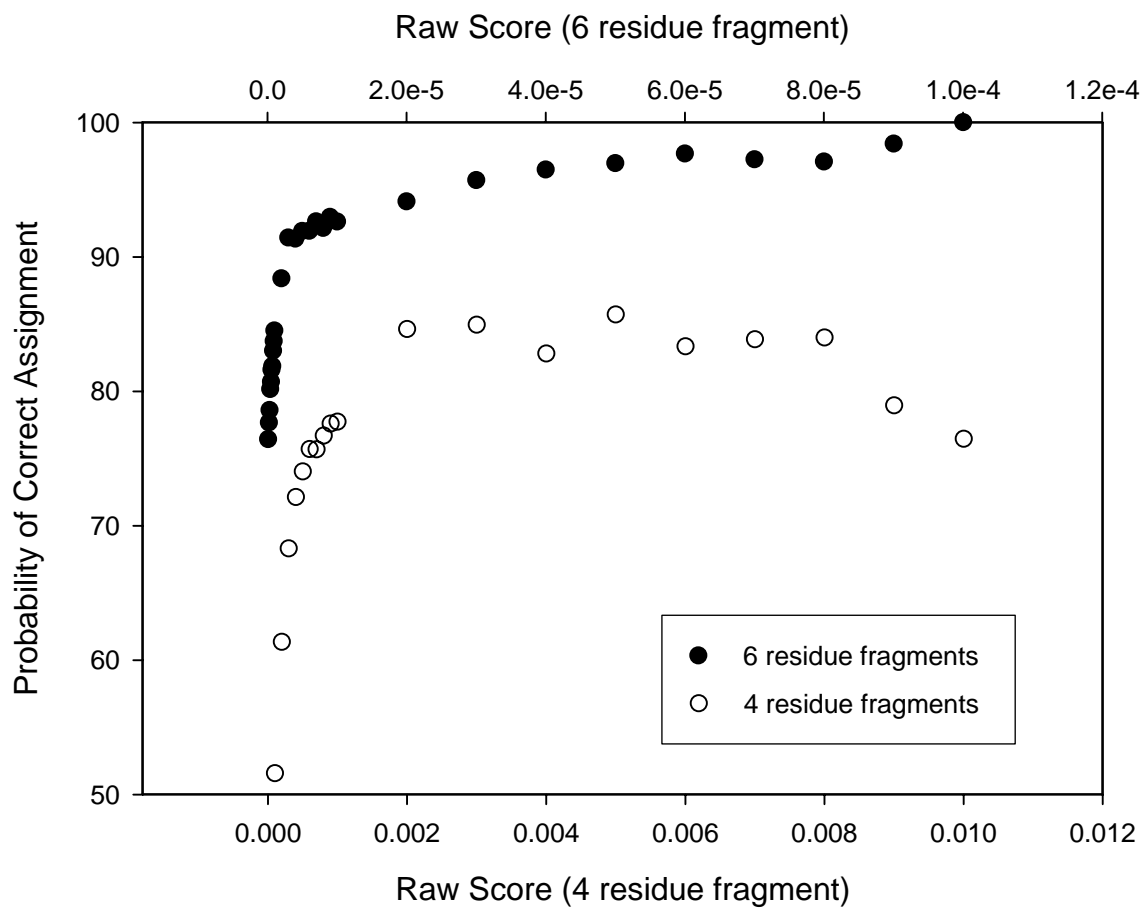


Figure 3.32. Correct assignment probability as a function of raw score for four residue fragments (bottom scale) and six residue fragments (top scale).

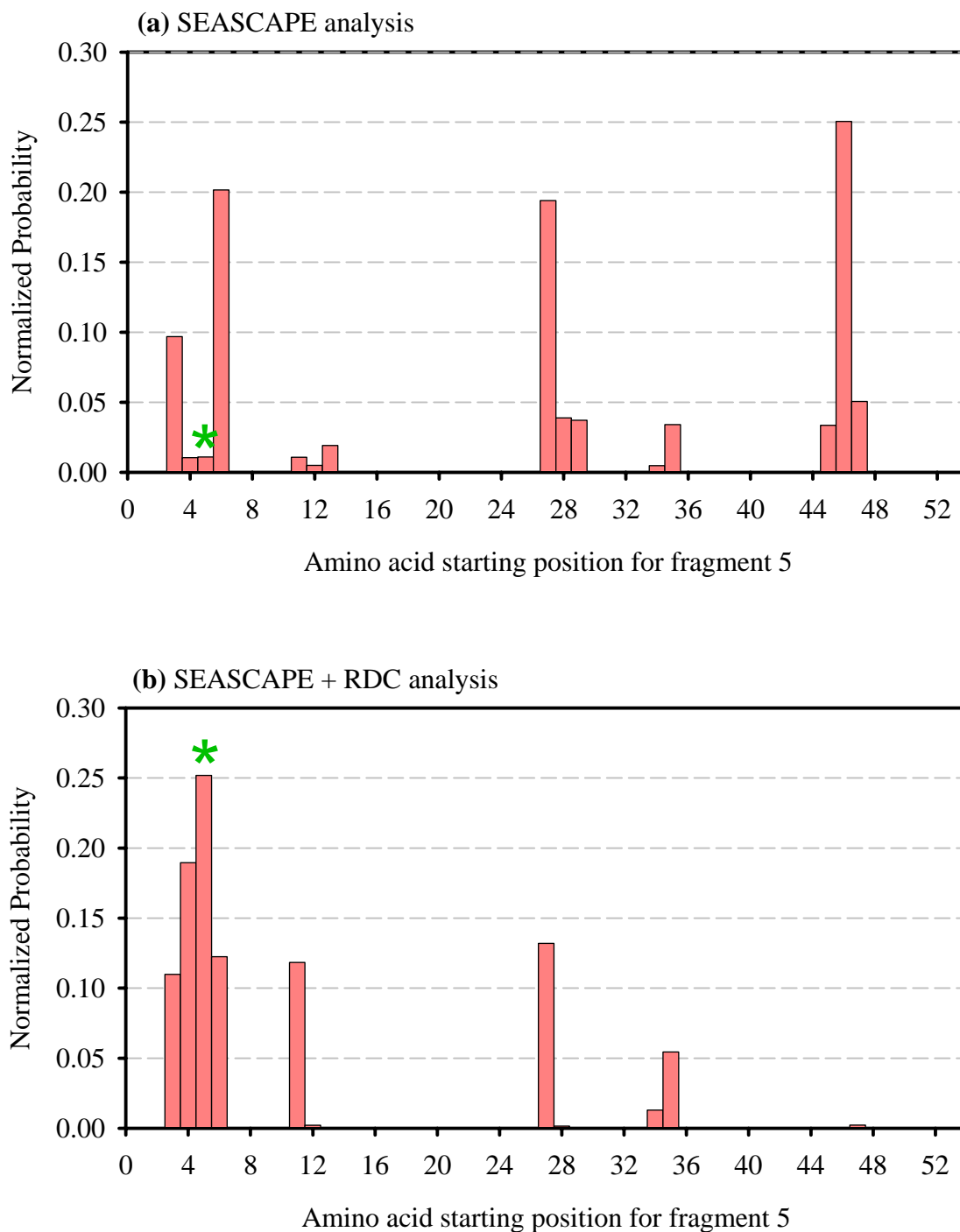


Figure 3.33 RDC assisted assignment in SEASCAPE. Results from application of SEASCAPE (a) or SEASCAPE + RDC analysis (b) to a four residue fragment whose correct position begins at amino acid 5. Both plots have been normalized to aid comparison. The correct placement of the fragment is indicated by *.

CHAPTER 4

CONCLUSIONS

Through this work we have demonstrated a new assignment strategy based on the availability of local backbone geometry and limited chemical shift connectivity data. Although still in its infancy, this methodology is a step towards accelerated assignment of protein NMR spectra. We anticipate applications where high throughput structure determination is an issue and where assignment of resonances for proteins of known structure is a research objective.

4.1 The Current State of SEASCAPE

As expected, the likelihood of correct assignment increases with increased connectivity and success varies somewhat depending on amino acid type. A detailed examination of the results indicates that fragments containing glycine are most often identified correctly. This is not surprising given that the $^{13}\text{C}\alpha$ for glycine is the furthest upfield of the amino acids and is narrowly distributed. Histidine and tryptophan have so far been the two amino acids least likely to be correctly identified, but as with other amino acids, their probabilities for correct assignment rise with an increase in the fragment length in which they are found.

The data in Table 3.1 illustrate the reduction of performance when using $^3J_{\text{HNHA}}$ in the place of φ, ψ . Although this result is expected, the performance of the algorithm with

the use of $^3J_{\text{HNHA}}$ is still useful. Preliminary results indicate that the addition of residual dipolar coupling analysis to SEASCAPE would increase the performance to even higher levels than the chemical shift version of SEASCAPE alone. In addition, SEASCAPE was able to distinguish between correct and incorrect connectivity for a few fragments.

4.2 Data and Statistical Analysis

For this initial study it was not necessary to obtain a complete set of data from the combined BMRB and PDB data sets. Now that the method has been shown to be effective, obtaining a more complete set of data could be beneficial. Both databases have grown in size since the initial set was obtained and the BMRB now includes homology information in its files. Instead of using many small commands and scripts to manipulate the data, we could create and use relational databases to manage the ever increasing data sets. Automating this portion of the work would also have the benefit of keeping the data up to date and allowing new procedures to be implemented as the need arises.

Additional statistical analyses would provide a better means of determining which results are more meaningful and to what degree. Many if not all bioinformatics programs provide this kind of information routinely to their users. On the development side, it would allow us to examine the distribution of our data to determine if some portion were under represented or skewed in some way. It would also give us a way to evaluate new procedures and shorten development time.

4.3 Incorporation of SEASCAPE into New and Existing NMR Programs

Although the program could be kept self contained, its real potential is as an addition to another program. Due to the popularity of the processing packages which are freely available from Frank Delaglio (NMRPipe; Delaglio, 1995) and Bruce Johnson (NMRView; Johnson, 1994) an obvious choice would be to make SEASCAPE available as an add-on. For structural genomics use, it is likely that this program will be incorporated by the developers of REDCAT (Valafar and Prestegard, 2003) into an integrated protein structure determination package. Both of these scenarios could be implemented using C++ programs as modules to do the computational work and Tcl/Tk as the graphical interface.

4.4 Future Modifications to SEASCAPE

As has already been demonstrated, incorporation of other data such as RDCs could enhance SEASCAPE's assignment capabilities. However it would be necessary to keep in mind that the main objective is to decrease the need for more experiments and simplify the assignment process. Still, providing the end user with a greater range of data which may be analyzed, such as additional chemical shifts or RDC data that are readily available, would ultimately make the program more versatile.

The program has been written for and applied to the sequential assignment of a single fragment. Of course assignments of multiple fragments in the same protein are not independent. Once one assignment is made the segment of the sequence to which it is assigned should be eliminated from further consideration. There is no reason that the program couldn't then run again with a new set of restrictions. When applied recursively,

the program may be able to assign a full set of fragments or at least give a few probable assignment sets with statistical analysis. The preliminary work done in this area suggests that one obstacle is that many times a correct position is not the most probable. However, the score for the correct position is not much lower than the highest probability and is usually captured in the top several selections calculated. This would require alternate assignments during the iterative exploring procedure. One possibility for handling this complexity is to use simultaneous solution methods such as those implemented by Brüschweiler in his attempt at using RDCs to aid assignment (Hus et al., 2002).

4.5 Conclusion

Although protein structure determination by NMR spectroscopy is not trivial or completely routine it currently takes longer than it should. Part of the reason for this inefficiency is that an inordinate amount of time is spent assigning resonances in the spectra. Even experienced researchers in the pharmaceutical industry spend two or three months to determine a structure by NMR. Researchers are working to make the methods in use today more powerful and user friendly but there is still much to be done. Certainly there are overlaps and interdependencies between applications and in many cases it is possible to synergistically combine many of the methods. However methodologies in common use today are only beginning to take advantage of the flood of data becoming available. Producing a structure in a week still seems like a dream to some but it's not that far from the realm of possibility now. It is my hope that this work is a step in that direction.

REFERENCES

- Atreya, H.S., Chary, K.V.R. and Govil, G. (2002) *Curr. Sci.*, **83**, 1372-1376.
- Bax, A. and Ikura, M. (1994) *J. Biomol. NMR*, **1**, 99-104.
- Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) *Nucleic Acids Res.*, **28**, 235-242.
- Cavanagh, J., Fairbrother, W.J., Palmer III, A.G. and Skelton, N.J. (1996) *Protein NMR Spectroscopy: principles and practice*, Academic Press, San Diego, CA
- Coggins, B.E. and Zhou, P. (2003) *J. Biomol. NMR*, **26**, 93-111.
- Delaglio, F., Grzesiek, S., Vüister, G.W., Zhu, G., Pfeifer, J. and Bax, A. (1995) *J. Biomol. NMR*, **6**, 277-293.
- Doreleijers, J.F., Mading, S., Maziuk, D., Sojourner, K., Yin, L., Zhu, J., Markley, J.L., Ulrich, E.L. (2003) *J. Biomol. NMR*, **26**, 139-146.
- Furui, J., Uegaki, K., Yamazaki, T., Shirakawa, M., Swindells, M.B., Harada, H., Taniguchi, T. and Kyogoku, Y. (1998) *Structure with Folding & Design*, **6**, 491-500.
- Grzesiek, S. and Bax, A. (1992a) *J. Am. Chem. Soc.*, **114**, 6291-6293.
- Grzesiek, S. and Bax, A. (1992b) *J. Magn. Reson.*, **99**, 201-207.
- Hus, J.C, Prompers, J.J., and R. Brüschweiler (2002) *J. Magn. Reson.*, **157**, 119-123.
- Johnson, B. and Blevins, R.A. (1994) *J. Biomol. NMR*, **4**, 603-614.
- Kay, L.E., Ikura, M., Tschudin, R. and Bax, A. (1990) *J. Magn. Reson.*, **89**, 496-514.
- Moseley, H.N.B. and Montelione, G.T. (1999) *Curr. Opin. Struct. Biol.*, **9**, 635-642.
- Neal, S., Nip, A.M., Zhang, H. and Wishart, D.S. (2003) *J. Biomol. NMR*, **26**, 215-240.
- Pardi, A., Billeter, M. and Wuthrich, K. (1984) *Journal of Molecular Biology*, **180**, 741-751.
- Prestegard, J.H. and Kishore, A. (2001) *Curr. Opin. Chem. Biol.*, **5**, 584-590.
- Schwalbe, H., Carlomagno, T., Hennig, M., Junker, J., Reif, B., Richter, C. and Griesinger, C. (2001) *Meth. Enzymol.*, **338**, 35-81.

- Silverman, B.W. (1986) Density estimation for statistics and data analysis, Chapman and Hall, New York, NY
- Spera, S. and Bax, A. (1991) *J. Am. Chem. Soc.*, **113**, 5490-5492.
- Stockman, B.J. and Dalvit, C. (2002) *Prog. Nucl. Magn. Reson. Spectrosc.*, **41**, 187-231.
- Tian, F., Valafar, H. and Prestegard, J.H. (2001) *Journal of the American Chemical Society*, **123**, 11791-11796.
- Tolman, J.R. and Prestegard, J.H. (1996) *J. Magn. Reson.*, **112**, 245-252.
- Valafar, H. and Prestegard, J.H. (2003) submitted
- Vüister, G.W. and Bax, A. (1993) *J. Am. Chem. Soc.*, **115**, 7772-7777.
- Wagner, G., Kumar, A., Wuthrich, K. (1981) *Eur. J. Biochem.*, **114**, 375-384.
- Wang, Y.X., Marquardt, J.L., Wingfield, P., Stahl, S.J., Huang, S., Torchia, D. and Bax, A. (1998) *J. Am. Chem. Soc.*, **120**, 7385-7386.
- Weisemann, R., Ruterjans, H., Schwalbe, H., Schleucher J., Bermel, W. and Griesinger, C. (1994) *J. Biomol. NMR*, **4**, 231-240.
- Wishart, D.S. and Case, D.A. (2001) *Meth. Enzymol.*, **338**, 3-34.
- Wishart, D.S., Watson, M.S., Boyko, R.F. and Sykes, B.D. (1997) *J. Biomol. NMR*, **10**, 329-336.
- Word, J.M. Ph.D. Thesis, Duke University, Durham, N.C., 2000
- Wüthrich, K. (1986) NMR of Proteins and Nucleic Acids, Wiley, New York.
- Zimmerman, D.E., Kulikowski, C.A., Huang, Y., Feng, W., Tashiro, M., Shimotakahara, S., Chien, C., Powers, R., and Monteloin, G.T. (1997) *J. Mol. Biol.*, **269**, 592-610.

APPENDIX A

BMRB/PDB SEQUENCE ALIGNMENT PROGRAM

Program name: seqalign.pl

Language: Perl

```
===== BEGIN CODE =====

#!/usr/bin/perl

open(BMRBSEQLIST,"bmrbbseqlist.txt") || die "Can't open bmrbbseqlist.txt";
print "Opened bmrbbseqlist.txt\n";

while(<BMRBSEQLIST>)
{
    $bmrbbseq = $_;
    chomp($bmrbbseq);

    $bmrbbfilebeg = $bmrbbseq;
    chop($bmrbbfilebeg);
    chop($bmrbbfilebeg);

    $bmrbbfile = $bmrbbfilebeg . "tr";

    print "bmrbbfile is : $bmrbbfile\n";

    $found = 0;

    open(BMRBPDB,"bmrbbpdblist.txt") || die "Can't open bmrbbpdblist.txt";

    while($found == 0) # find the PDB for the str file
    {
        $bmrbbpdblistline = <BMRBPDB>;
        chomp($bmrbbpdblistline);
        @bmrbbpdpline = split(" ", $bmrbbpdblistline);

        if(@bmrbbpdpline[1] eq $bmrbbfile)
        {
            $pdbname = @bmrbbpdpline[0];
            $pdbname =~ tr/A-Z/a-z/;
            $pdbname = substr($pdbname,0,4);
            $pdbseq = $pdbname . ".seq";
            $found = 1;
            print "bmrbb: $bmrbbfile rcsb: $pdbname\n";
        }

        if(@bmrbbpdpline[1] eq "no_pdb.str")
        {
            print "No pdb found.\n";

            $found = 2;
            $pdbseq = "no_pdb";
        }
    }
}
```

```

    } # matching PDB file has been found or does not exist

    close(BMRBPDB);

# current BMRB sequence is : $bmrseq
# matching PDB sequence is : $pdbseq

    if($found == 1)
    {
        # get the number of amino acids in the pdb file so we don't
        # accidentally run off the end of the file later (in align subroutine)
        open(PDBSEQ, "./rcsb/$pdbseq") || die "Can't open ./rcsb/$pdbseq\n";
        $numaa = 0;
        while(<PDBSEQ>)
        {
            ++$numaa;
        }
        close(PDBSEQ);
    print "Number of aa in $pdbseq : $numaa\n";
    # end getting num aa

    open(BMRBSEQ, "./bmr/$bmrseq") || die "Can't open ./bmr/$bmrseq\n";
    open(PDBSEQ, "./rcsb/$pdbseq") || die "Can't open ./rcsb/$pdbseq\n";

    $bmrseqline = <BMRBSEQ>;
    chomp($bmrseqline);
    @bmrseqlist = split(" ", $bmrseqline);
    $bmrseqaa = @bmrseqlist[1];

    $pdbseqline = <PDBSEQ>;
    chomp($pdbseqline);
    @pdbseqlist = split(" ", $pdbseqline);
    $pdbseqaa = @pdbseqlist[1];

    $numcorrect = 0;
    $matchstart = 1;

    $startmatch = &align($bmrseqaa, $pdbseqaa);

    close(BMRBSEQ);
    close(PDBSEQ);

    print "The $bmrseq alignment starts at $pdbseq residue : $startmatch \n";

    if($startmatch > 0)
    {
        $seqalignfile = &makealignfile($bmrseq, $pdbseq, $startmatch);

        print "$bmrseq and $pdbseq alignment is given in $seqalignfile\n";
    }

    else
    {
        $nopdbalignbeg = $bmrseq;
        chop($nopdbalignbeg);
        chop($nopdbalignbeg);
        chop($nopdbalignbeg);
        $nopdbalign = $nopdbalignbeg . "align";

        open(NOPDB, ">./alignments/$nopdbalign") || die "Can't open non-alignment file :
./alignments/$nopdbalign\n";
        print NOPDB "NO ALIGNMENT POSSIBLE\n";
        close(NOPDB);
        print "$bmrseq and $pdbseq alignment was NOT POSSIBLE\n";
    }

}

else
{

```

```

    $nopdbalignbeg = $bmrbsubseq;
    chop($nopdbalignbeg);
    chop($nopdbalignbeg);
    chop($nopdbalignbeg);
    $nopdbalign = $nopdbalignbeg . "align";

    open(NOPDB,">./alignments/$nopdbalign") || die "Can't open non-alignment file :
./alignments/$nopdbalign\n";
    print NOPDB "NO PDB FOR ALIGNMENT\n";
    close(NOPDB);

    print "The file $bmrbsubseq has NO PDB !!\n";
}

}

close(BMRBSEQLIST);

#####
# ALIGN subroutine
#####

sub align
{
print "Starting align subroutine\n";
    local($bmrbsubseqaalocal,$pdbseqaalocal) = @_;

    $aabmrb = $bmrbsubseqaalocal;
    $aarcsb = $pdbseqaalocal;
print "aabmrb : $aabmrb ";
print "aarcsb : $aarcsb\n";

    $endofseq = $numaa - 10;
    if($matchstart > $endofseq) { return -1;}

print "matchstart : $matchstart\n";
print "    numaa : $numaa\n";
    while($aabmrb eq "skip" || $aarcsb eq "skip")
    {
        if($aarcsb eq "skip" || $numcorrect == 0)
        {
            ++$matchstart;
        }

        $nextbmrbline = <BMRBSEQ>;
        chomp($nextbmrbline);
        @bmrbline = split(" ", $nextbmrbline);
        $aabmrb = @bmrbline[1];
print "skip aabmrb : $aabmrb ";
        $nextpdbline = <PDBSEQ>;
        chomp($nextpdbline);
        @pdbline = split(" ", $nextpdbline);
        $aarcsb = @pdbline[1];
print "aarcsb : $aarcsb\n";

    }
print "bmr : $aabmrb , rcsb : $aarcsb - no skip\n";

    if($aabmrb eq $aarcsb)
    {
        ++$numcorrect;
print "numcorrect : $numcorrect\n";
        if($numcorrect == 5)
        {
            return $matchstart;
        }
    }
    else
    {
        $nextpdbline = <PDBSEQ>;
        # read next RCSB
    }
}
# BEGIN EQ
#
#
#
#
#
#
#
#
#
#

```



```

        chomp($nextpdbline);          #      amino acid  #
        @pdbline = split(" ", $nextpdbline); #      #
        $nextaarcsb = @pdbline[1];      #      #
        #                                #
        $nextbmrbline = <BMRBSEQ>;      # read next BMRB #
        chomp($nextbmrbline);          #      amino acid  #
        @bmrbline = split(" ", $nextbmrbline); #      #
        $nextaabmrb = @bmrbline[1];     #      #
        print "Call align nextaabmrb : $aabmrb nextaarcsb : @pdbline[0] $aarcsb\n";
        #
        &align($nextaabmrb, $nextaarcsb); #      #
    }                                     #      #
}                                         # END EQ

else                                     # BEGIN NOT EQUAL
{
    ++$matchstart;                      #
    close(BMRBSEQ);                     #
    $numcorrect = 0;                     #
    open(BMRBSEQ, "./bmr/$bmrseq") || die "Can't open ./bmr/$bmrseq\n";
    #
    $firstbmrbline = <BMRBSEQ>;          ##read BMRB
    chomp($firstbmrbline);              ##      amino acid
    @bmrbline = split(" ", $firstbmrbline); ##
    $bmrbaa = @bmrbline[1];             ##
    #
    $nextpdbline = <PDBSEQ>;             ##read next RCSB
    chomp($nextpdbline);                ##      amino acid
    @pdbline = split(" ", $nextpdbline); ##
    $nextpdbaa = @pdbline[1];           ##
    #
    print "Call align bmr : $bmrbaa nextpdbaa : @pdbline[0] $nextpdbaa\n";
    &align($bmrbaa, $nextpdbaa);         #
    #
    # END NOT EQUAL
}

}

#=====
# MAKEALIGNFILE subroutine
#=====

sub makealignfile
{
    local($bmrseqfile, $pdbseqfile, $beginmatch) = @_ ;

    open(BMRBSEQ, "./bmr/$bmrseqfile") || die "Can't open BMRB sequence file :
$bmrseqfile\n";
    open(RCSBSEQ, "./rcsb/$pdbseqfile") || die "Can't open RCSB sequence file :
$pdbseqfile\n";

    $alignfilebeg = $bmrseqfile;
    chop($alignfilebeg);
    chop($alignfilebeg);
    chop($alignfilebeg);
    $alignfile = $alignfilebeg . "align";

    open(ALIGN, ">./alignments/$alignfile") || die "Can't open alignment file :
./alignments/$alignfile\n";
    print ALIGN "rcsbfile $pdbseqfile\n";

    $linenum = 1;
    $bmrfirstline = <BMRBSEQ>;
    chomp($bmrfirstline);
    @strline = split(" ", $bmrfirstline);
    $bmrfirstnum = @strline[0];
    close(BMRBSEQ);
    open(BMRBSEQ, "./bmr/$bmrseqfile") || die "Can't open BMRB sequence file :
$bmrseqfile\n";

```

```

$bmrbnum = $bmrbfirstnum - $beginmatch + 1;

while($linenum < $beginmatch)
{
    $pdblne = <RCSBSEQ>;

    $pdblinetemp = $pdblne;
    chomp($pdblinetemp);
    @pdblinesplit = split(" ", $pdblinetemp);
    $pdblnefirst = @pdblinesplit;

    $pdblnefirst =~ s/[a-zA-Z:]/g;
    $pdblneprev = $pdblnefirst;

    print ALIGN "$bmrbnum  skip  $pdblne";      # <cr> is still in $pdblne

    ++$linenum;
    ++$bmrbnum;
}

while(($bmrbline = <BMRBSEQ>) > 0)
{
    chomp($bmrbline);
    $pdblne = <RCSBSEQ>;

    if($pdblne eq "")
    {
        $pdblne = "000  skip\n";
    }

    print ALIGN "$bmrbline  $pdblne";      # <cr> is still in $pdblne
}

close(BMRBSEQ);
close(RCSBSEQ);
close(ALIGN);

return $alignfile;
}

```

===== END CODE =====

The input files are sequences derived from matched BMRB and PDB files. An example output file follows.

Example output file: bmr5.align

```
===== BEGIN OUTPUT =====
rcsbfile lahl.seq
1  GLY      1  GLY
2  VAL      2  VAL
3  SER      3  SER
4  CYS      4  CYS
5  LEU      5  LEU
6  CYS      6  CYS
7  ASP      7  ASP
8  SER      8  SER
9  skip     9  ASP
10 skip    10  GLY
11 skip    11  PRO
12 SER     12  SER
13 VAL     13  VAL
14 ARG     14  ARG
15 skip    15  GLY
16 skip    16  ASN
17 THR     17  THR
18 skip    18  LEU
19 skip    19  SER
20 skip    20  GLY
21 THR     21  THR
22 LEU     22  LEU
23 TRP     23  TRP
24 skip    24  LEU
25 TYR     25  TYR
26 PRO     26  PRO
27 skip    27  SER
28 GLY     28  GLY
29 CYS     29  CYS
30 skip    30  PRO
31 SER     31  SER
32 GLY     32  GLY
33 TRP     33  TRP
34 HIS     34  HIS
35 ASN     35  ASN
36 CYS     36  CYS
37 LYS     37  LYS
38 ALA     38  ALA
39 HIS     39  HIS
40 skip    40  GLY
41 skip    41  PRO
42 THR     42  THR
43 ILE     43  ILE
44 GLY     44  GLY
45 TRP     45  TRP
46 CYS     46  CYS
47 CYS     47  CYS
48 LYS     48  LYS
===== END OUTPUT =====
```

APPENDIX B

BMRB/PDB DATA EXTRACTION PROGRAM

Program name: csshipsi.pl

Language: Perl

```
===== BEGIN CODE =====
#!/usr/bin/perl

# 1. set aa to one of the amino acids (aa)
# 2. open aaCAChemicalshifts.txt file
# 3. parse line
# 4. open bmrxxx.align file
# 5. open rcsbyyy.seq file
# 6. find bmrxxx aa number listed in aaCA file
# 7. find rcsbyyy matching aa num in bmrxxx.align file
# 8. open yyy.phi and yyy.psi files
# 9. find rcsbyyy aa num in phi and psi
# 10. write cs, phi, psi (if all exist)
# 11. close yyy.phi, yyy.psi, rcsbyyy.seq, and bmrxxx.align
# 12. repeat from 3 until end of file
# 13. repeat from 1 for each aa (1 to 20)
#
#
#
# VARIABLES
# -----
# @aaCSfile[$i] - array of filenames of amino acid chemical shifts ($i = 0-> 19)
#
@aaCSfile =
("alaCAChemshifts.txt", "argCAChemshifts.txt", "asnCAChemshifts.txt", "aspCAChemshifts.txt",
"cysCAChemshifts.txt", "glnCAChemshifts.txt", "gluCAChemshifts.txt", "glyCAChemshifts.txt",
"hisCAChemshifts.txt", "ileCAChemshifts.txt", "leuCAChemshifts.txt", "lysCAChemshifts.txt",
"metCAChemshifts.txt", "pheCAChemshifts.txt", "proCAChemshifts.txt", "serCAChemshifts.txt",
"thrCAChemshifts.txt", "trpCAChemshifts.txt", "tyrCAChemshifts.txt", "valCAChemshifts.txt");

for($i=0;$i<20;$i++)
{
    open(CSFILE, "./str/@aaCSfile[$i]") || die "Can't open ./str/@aaCSfile[$i]\n";

    $aa = substr(@aaCSfile[$i],0,3);
    $aa =~ tr/a-z/A-Z/;
    print "$aa $i\n";

    $whilecounter = 0;

    while(<CSFILE>)
    {
        $csline = $_;
        @cslinebit = split(" ", $csline);

        open(CHECK, "processlist.txt") || die "Can't open processlist.txt\n";
```

```

$aligned = "search";

$bmrballign = substr(@cslinebit[0],0,-4) . "align";

while($aligned eq "search")
{
    $processed = <CHECK>;
    chomp($processed);

    if($processed eq $bmrballign)
    {
        $aligned = "yes";
        close(CHECK);
    }

    if($processed eq "noalign.align")
    {
        $aligned = "no";
        close(CHECK);
    }
}

if(@cslinebit[3] eq $aa && $aligned eq "yes")
{
    open(BMRBALLIGN,"./seq/alignments/$bmrballign") || die "Can't open
./seq/alignments/$bmrballign\n";
    print "opened $bmrballign\n";

    $pdbnameline = <BMRBALLIGN>;
    print "pdbnameline : $pdbnameline\n";
    @linesplit = split(" ", $pdbnameline);
    $pdbname = substr(@linesplit[1],0,-3);

    $bmrbresnum = -100000;
    print "bmrbresnum = $bmrbresnum : cslinebit2 = @cslinebit[2]\n";
    while($bmrbresnum < @cslinebit[2])
    {
        $bmrblineline = <BMRBALLIGN>;
        @linesplit = split(" ", $bmrblineline);
        $bmrbresnum = @linesplit[0];
        $rcsbresnum = @linesplit[2];
        $bmrplus = $bmrbresnum + 1;
    }
    print "bmrbresnum = $bmrbresnum\n";
    if($bmrbresnum == @cslinebit[2])
    {
        $phifile = $pdbname . "phi";
        open(PHI,"./phi/$phifile") || die "Can't open ./phi/$phifile\n";
        print "opened $phifile\n";
        print "beginning search for PHI\n";
        $search4phi = "yes";

        while($search4phi eq "yes")          ##### finding PHI
        {
            $phileraw = <PHI>;
            chomp($phileraw);
            $residuenum = substr($phileraw,2,4);
            @resnum = split(" ", $residuenum);
            $phiresnum = @resnum[0];

            if($phiresnum == $rcsbresnum)
            {
                $search4phi = "success";
                $phivalue = substr($phileraw,33,9);
                print "phi found : $phivalue\n";
            }

            if($phiresnum > $rcsbresnum)
            {

```

```

        $search4phi = "fail";                                #
        print "phi NOT found : failure\n";                  #
    }                                                         #
} #####
close(PHI);

$psifile = $pdbname . "psi";
open(PHI, "./psi/$psifile") || die "Can't open ./psi/$psifile\n";
print "beginning search for PSI --- phi : $search4phi\n";
$search4psi = "yes";

while($search4psi eq "yes" && $search4phi eq "success") ## finding PSI
{
    $psilineraw = <PSI>;                                     #
    chomp($psilineraw);                                     #
    $residuenum = substr($psilineraw,2,4);                  #
    @resnum = split(" ", $residuenum);                      #
    $psiresnum = @resnum[0];                                #
    #
    if($psiresnum == $rcsbresnum)                           #
    {                                                         #
        $search4psi = "success";                             #
        $psivalue = substr($psilineraw,33,9);              #
        print "psi found : $psivalue\n";                   #
    }                                                         #
    #
    if($psiresnum > $rcsbresnum)                             #
    {                                                         #
        $search4psi = "fail";                                #
        print "psi NOT found : failure\n";                  #
    }                                                         #
} #####
close(PHI);

if($search4phi eq "success" && $search4psi eq "success")
{
    print "SUCCESS!! Now let's print out the data.\n";
    $dataout = substr(@aaCSfile[$i],0,5) . "phipsi.txt";
    print "file is : $dataout\n";
    open(OUT, ">>$dataout") || die "Can't open $dataout\n";

    $chemshift = @cslinebit[6];

    print OUT "$chemshift $phivalue $psivalue\n";
    print "DATA: $chemshift , $phivalue , $psivalue\n";
    close(OUT);
}
else
{
    print "No success. What's the problem?\n";
}
} # IF statement end

} # outermost IF statement

close(BMRBALIGN);

++$whilecounter;
print "while counter : $whilecounter\n";

} # outermost WHILE loop

close(CSFILE);

} # outermost FOR loop

```

Example output file (partial): alaCAphipsi.txt

```
===== BEGIN PARTIAL OUTPUT =====
49.95   -112.67    125.34
50.26   -114.94    147.65
53.06    -70.00    -25.14
50.92    -76.28    154.45
49.95   -112.67    125.34
50.26   -114.94    147.65
53.06    -70.00    -25.14
50.92    -76.28    154.45
47.2    -114.85    154.36
53.1     -62.24    -23.45
47.3    -103.39    116.38
50     -149.91    167.78
50.8     -80.18    131.56
54.3     -63.75    -38.46
53.5     -70.27    -32.93
49.9     -77.97    147.97
49.2    -156.72    167.81
48.3    -119.60    108.75
53.8     -55.60    -44.88
48.8    -166.84    157.43
49.6     -73.62    171.33
53.4     -61.86    -41.41
54.2     -61.10    -40.07
55.1     -72.59    -38.88
55.1     -68.39    -29.20
52.36    -60.00    -38.30
55.1     -69.15    -39.72
55.1     -69.15    -39.72
48.8     -79.15    164.06
48.4    -148.24    155.61
50.7    -156.16    150.40
53.0     -53.16    -29.15
48.2    -120.74    153.46
48.5    -119.39    133.33
49.7     -73.39    -11.44
53.7     -52.33    -25.23
53.8     -68.95     -8.81
===== END PARTIAL OUTPUT =====
```

APPENDIX C

THE SEASCAPE PROGRAM

Program name: seascape.pl

Language: Perl

```
===== BEGIN CODE =====
#!/usr/bin/perl

#
#           @ARGV[0]   @ARGV[1]
# usage: seascape.pl sequencefile linkagefile
#                   [expdat]
#
# sequencefile contains one letter abbreviations
#
#
#

open(LINKS,"@ARGV[1]") || die "Can't open @ARGV[1]\n";

$linkslines = <LINKS>;

@splitline = split("\t",$linkslines);
$numlinked = @splitline[1];
$header = @splitline[0];

open(SEQ,"@ARGV[0]") || die "Can't open @ARGV[0]\n";

$numres = <SEQ>;
chomp($numres);

print "$numres : number of residues in the sequence file\n";
print "=====\n\n";

#####
# read in all densities for all amino acids

@aaaname = ("ala","arg","asn","asp","cys","gln","glu","gly","his","ile",
            "leu","lys","met","phe","pro","ser","thr","trp","tyr","val");

@aaletter = ("A","R","N","D","C","Q","E","G","H","I",
            "L","K","M","F","P","S","T","W","Y","V");

#####
# read in (C-alpha, phi, psi) densities

for($i=0;$i<20;++$i)
{
    print "@aaaname[$i] @aaletter[$i]\tCa, phi, psi\n";

    $aafilename = @aaaname[$i] . "Cajydensity.txt";

    open(DENSITY,"../densities/$aafilename") || die "Can't open aa density file: $aafilename\n";
```



```

$j = 0;

$aalet = @aaletter[$i];
$aaoord = ord($aalet);

while(<DENSITY>)
{
    $aadenline = $_;
    chomp($aadenline);

    @aaline = split(" ", $aadenline);

    $aa3density[$aaoord][$j] = @aaline[0]; # density for aa=@aaname[$i], line ($j-1)

    ++$j;
}

close(DENSITY);
}

#####
#
# for each fragment, calc. density
#

while(@splitline[0] ne "END") #### beginning of every header section
{

    print "$header : (beginning of a linkage segment)\n";

    $numpos = $numres - $numlinked + 1; # number of possible matches

    print "$header : $numpos : number of possible matches\n\n";

    for($i=1;$i<=$numlinked;++$i) # enter sequence into the 1st segment
    {
        $linksline = <LINKS>;
        @splitline = split(" ", $linksline);

        $expdat[$i][0] = @splitline[0]; # type of experimental data
        $expdat[$i][1] = @splitline[1]; # C-alpha
        $expdat[$i][2] = @splitline[2]; # phi
        $expdat[$i][3] = @splitline[3]; # psi

        #    print "calpha $i\t$expdata[$i][1]\n";

        $segment[1][$i] = <SEQ>;
        chomp($segment[1][$i]);
        #print "segment 1 $i $segment[1][$i]\n";
    }

    for($i=2;$i<=$numpos;++$i) # enter sequence into all subsequent segments
    {
        for($j=1;$j<$numlinked;++$j)
        {
            $segment[$i][$j] = $segment[$i-1][$j+1];
            #    print "segment $i $j $segment[$i][$j]\n";
        }

        $seginput = <SEQ>;
        chomp($seginput);
        @segbits = split(" ", $seginput);
    }
}

```

```

        $segment[$i][$j] = @segbits[0];
        chomp($segment[$i][$numlinked]);
#print "segment $i $numlinked $segment[$i][$numlinked]\n";
    }

    ##### all possible match segments have been prepared

#    print "$header : all possible match segments have been prepared\n";

    close(SEQ);

##### NEW PART #####

# now calc product of densities for each segment

##### I loop

for($i=1;$i<=$numpos;$i++) # increment position along sequence
{
    $density[$i] = 1; # density for each fragment ($i)

    ##### J loop

    for($j=1;$j<=$numlinked;++$j) # for each residue in the segment
    {
        $csnum = int(($expdat[$j][1] - 39.75)/0.5);
        $phinum = int(($expdat[$j][2] + 182.5)/5);
        $psinum = int(($expdat[$j][3] + 182.5)/5);

#print "cs: $csnum\tphi: $phinum\tpsi: $psinum\n";
        #density line number is calculated by: csnum*73*73 + phinum*73 + psinum
        # 61 chemical shifts (40 ->70, every 0.5 ppm; not really relevant)
        # 73 phi values (-180 -> 180, every 5 degrees)
        # 73 psi values (-180 -> 180, every 5 degrees)
        # 0 (zero) is considered the first line

        $dlinenum = ($csnum * 5329) + ($phinum * 73) + $psinum;

        $segord = ord($segment[$i][$j]);
        $density[$i] *= $aa3density[$segord][$dlinenum];
#        print "density : $segord\t$dlinenum\t$aa3density[$segord][$dlinenum]\n";
    }

    ##### J loop end

} ##### I loop end

$densitymax = 0;

for($i=1;$i<=$numpos;++$i)
{
    if($densitymax < $density[$i])
    {
        $densitymax = $density[$i];
        $imax = $i;
    }
}

$PDout = "prob." . @ARGV[1];

open(OUT,">>$PDout") || die "Can't open $PDout\n";

$id = substr($header,3);

```

```

chomp($id);

print OUT "$imax\t$densitymax\t";

print "\n*****\n$header MAX DENSITY :
$imax\t$densitymax\n*****\n";

for($i=1;$i<=$numpos;++$i)
{
    print OUT "$i\t$density[$i]\t";
}

print OUT "\n";

close(OUT);

open(SEQ,"@ARGV[0]") || die "Can't open @ARGV[0]\n";

$numres = <SEQ>;
chomp($numres);

$linkslines = <LINKS>;
chomp($linkslines);

@splitline = split("\t",$linkslines);
$numlinked = @splitline[1];
$oldheader = $header;
$header = @splitline[0];

if(@splitline[0] ne "END")
{
    print "$oldheader : end of this sequence. new sequence is : @splitline[0]\n\n";
    print "-----\n";
    print "\n\n";
}
else
{
    print "\n\n\n\n      END OF ANALYSIS\n";
}

}

close(SEQ);
close(LINKS);

===== END CODE =====

```

There are two input files created by the user. One is the sequence of the protein and the other is the fragment data. An example of each follows.

Example input protein sequence file: 1m2y.seq

```
===== BEGIN INPUT =====  
53  
A  
K  
Y  
V  
C  
K  
I  
C  
G  
Y  
I  
Y  
D  
E  
D  
A  
G  
D  
P  
D  
N  
G  
V  
S  
P  
G  
T  
K  
F  
E  
E  
V  
P  
D  
D  
W  
V  
C  
P  
I  
C  
G  
A  
P  
K  
S  
E  
F  
E  
K  
L  
E  
D  
END  
===== END INPUT =====
```

Example input fragment data: 1m2y_linkfrags4.txt

```
===== BEGIN INPUT =====
seg3Y  4
3      56.63   -125.1  135.1
3      58.28   -110.2  140.1
3      59.79   -75.1   64.8
3      59.08   -14.7   -9.9
seg10Y 4
3      60.44   -70.1   170.1
3      59.3    -105.2   99.8
3      57.57   -84.9   129.7
3      51.5    -109.8   75.2
seg35D  4
3      53.15   -99.8    -15
3      59.65   -69.8   164.9
3      56.95   -164.8  165.1
3      57.12   -55.2    85
END
===== END INPUT =====
```

One example line (wrapped due to length) from the output file:
prob.1m2y_linkfrags_4.txt

```
===== BEGIN OUTPUT LINE=====
3      6.17212471048479e-05  1      1.3754887656276e-07  2      1.51757979345732e-05
3      6.17212471048479e-05  4      1.98186090320336e-06  5
8.76259553326437e-06  6      4.80072245213234e-08  7      4.16546835194224e-09
8      1.10914231549504e-09  9      1.81113034492704e-08  10
4.89313229940425e-05  11     8.28682457024744e-06  12     3.37897796933881e-06
13     1.54318683277866e-07  14     7.36397334650422e-11  15
4.72568949249739e-11  16     9.97837838763985e-12  17     2.37659254753391e-09
18     1.17223735181676e-07  19     4.90822806555007e-12  20
1.07882183681902e-11  21     1.43705557662283e-09  22     1.62293259732707e-09
23     4.73493231576062e-08  24     3.5639252018855e-10  25
1.72441419619372e-10  26     8.12838889552617e-09  27     1.06081117126841e-05
28     3.19794830170004e-05  29     9.80282792258308e-07  30
8.74484720193339e-07  31     3.34801446611457e-05  32     5.42220708933835e-07
33     1.78959645066988e-08  34     1.32157446908946e-07  35
6.11933662324585e-06  36     2.86581734848282e-06  37     5.20803998836807e-06
38     4.31828861486842e-06  39     1.27509251237443e-09  40
7.99063201360425e-10  41     6.18652038936987e-12  42     6.76311243385296e-10
43     5.72595126765158e-07  44     5.21534669187057e-07  45
1.86119020504599e-05  46     2.22514175964921e-05  47     2.31987546687587e-05
48     1.07673717214942e-05  49     5.7276381013558e-06  50
2.70221546089026e-06
===== END OUTPUT LINE =====
```