10-30-2014

Reproducible Protein NMR Data Analysis

Matthew Fenwick
University of Connecticut, mfenwick@student.uchc.edu

Reproducible Protein NMR Data Analysis

Matthew Fenwick, PhD

University of Connecticut, 2014

Nuclear Magnetic Resonance (NMR) spectroscopy is a technique for studying biological molecules such as proteins at the atomic level. The information obtained from NMR is used to identify binding partners, locate active sites and binding pockets, and obtain structural and dynamics information which can be used in drug design. In order to study molecules using NMR, an NMR spectrometer is used to collect free-induction decay (FID) data sets from a pure, high-concentration sample of the molecule(s) of interest. In subsequent analysis, the FID data is processed to frequency-domain spectra, which are then analyzed to find peaks and assign the peaks to specific nuclei in the molecule, in a process known as chemical shift assignment. The typical process makes use of automated tools to speed up simple and tedious tasks where possible, but relies upon manual analysis for complicated and difficult cases. Spectroscopists use a deductive strategy of iteratively applying previously identified rules to make analyses of specific cases. Ambiguous cases are noted and deferred, or the highest probability interpretation is made. Following chemical shift assignment, Nuclear Overhauser Effect (NOE) spectra are peak picked and assigned, and finally a structure is calculated and refined. During the analysis process, large amounts of data and meta data are generated.

However, much of this is not recorded and thus does not show up in archives such as the Biological Magnetic Resonance Data Bank (BMRB). This raises serious reproducibility concerns, since the data and meta data describing how the analysis was carried out are lost. These concerns lead to practical issues, including how to collaborate when data is missing, how to efficiently identify and correct errors, and how to augment previous analysis with new data. The growing problems caused by irreproducibility in science have been noted recently. The main contribution of this project is a definition of reproducibility within protein NMR, a strategy for rendering NMR analysis reproducible, a software implementation to enable reproducible analysis, a means for sharing reproducible data sets through a public archive, and a data set analyzed using fully reproducible means.

Reproducible Protein NMR Data Analysis

Matthew Fenwick

B.S., University of Oklahoma, 2009

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of Doctor of Philosophy

at the

University of Connecticut

2014

APPROVAL PAGE

Doctor of Philosophy Dissertation

Reproducible Protein NMR Data Analysis

Presented by

Matthew Fenwick B.Sc

Major Advisor

Michael Gryk

Associate Advisor

Mark Maciejewski

Associate Advisor

Dmitry Korzhnev

Associate Advisor

Jeffrey Hoch

University of Connecticut

2014

Acknowledgements

I would like to acknowledge Dr. Gryk for his gentle guidance on the road to infinity, and his lab members for their unwavering belief in the power of data integration.

I would also like to thank my advisory committee, Dr. Maciejewski, Dr. Korzhnev, and Dr. Hoch for their commitment to and investment in my education.

Thank you to Eldon Ulrich and the folks at the BMRB and NMRFAM for being awesome, making NMR awesome, and freely sharing their work and time to try and make NMR more awesome.

Thank you to the students and faculty of the department and the program.

Thank you to Ms. Bonnemaison.

A lot of time and effort was expended by a lot of people in helping the author of this document, whether emotionally, academically, technically, financially, morally, physically, spiritually, incidentally, accidentally, or intellectually. Thank you everybody!

Contents

1 Introduction **1**

1.1 Protein NMR 1

1.2 Scientific methods and reproducibility 2

1.3 Irreproducibility of computational and manual analysis in NMR 6

1.4 The significance of reproducibility to NMR 7

1.5 An approach for reproducible analysis 8

1.6 Reproducible NMR data set 8

1.7 Software for practical reproducibility 9

1.8 CONNJUR is free and open source 10

1.9 Scope and significance 11

1.10 Figures 13

2 NMR Spectroscopy of Proteins **15**

2.1 Protein NMR 15

3 A Data Overview **24**

3.1 Global prior knowledge 25

3.2 Local prior knowledge 27
3.3 Analysis 30
3.4 Desired knowledge 32
3.5 Tables 34
3.6 Figures 43

4 A Process Overview **58**

4.1 Data collection 58
4.2 Spectral processing 59
4.3 Spectral analysis 61
4.4 Structure determination 70
4.5 Discussion 73
4.6 Conclusions 75
4.7 Tables 77
4.8 Figures 78

5 An Approach for Reproducible Analysis **83**

5.1 NMR data analysis is irreproducible 83
5.2 Concrete data sets 85
5.3 Missing data and its role in analysis 86
5.4 A model for reproducible NMR 93
5.5 An implementation of the model 99
5.6 Applying reproducible analysis: using the model 99
5.7 Future directions 100

5.8 Discussion 101

5.9 Figures 103

6 Reproducible NMR Data Set **104**

6.1 Methods and materials 104

6.2 The analysis process 105

6.3 NMR-STAR deposition to the BMRB 110

6.4 Assessing the difficulties encountered 112

6.5 Alternative implementations 115

6.6 Discussion 117

6.7 Tables 118

6.8 Figures 122

7 Sparky Extension for Reproducible Spectral Analysis **125**

7.1 Getting started with Sparky 126

7.2 Concepts 127

7.3 Project setup 129

7.4 Capturing a snapshot 130

7.5 Reproducible backbone assignment tutorial 131

7.6 Pitfalls 136

7.7 Creating an NMR-STAR file 136

7.8 Tables 137

7.9 Figures 138

8 Software for Practical, Reproducible Analysis **141**

8.1 NMR-STAR library 141
8.2 CONNJUR ST 143
8.3 CONNJUR WB 146
8.4 Sample scheduler 148
8.5 Discussion and conclusions 151
8.6 Tables 155
8.7 Figures 157

9 Conclusions **166**

9.1 The future of NMR as an experimental technique 166
9.2 Reproducibility: challenge, opportunity 168
9.3 The future of NMR software 169
9.4 Final thoughts 170

Appendices **171**

A List of Publications **172**

A.1 Reproducible protein NMR data analysis (in progress) 172
A.2 CONNJUR Workflow Builder (in progress) 172
A.3 A bioinformatics sandbox 173
A.4 Accessing archived NMR data 173
A.5 Random phase detection 174
A.6 Software architecture for effective NMR data processing 174
A.7 CONNJUR Spectrum Translator 174

B Library of Deductive Reasons **175**

C Assignment: Considerations, Notations, Strategy **185**

C.1 Typing of H-N-rooted GSSs 185

C.2 Graph patterns of pulse sequences 186

C.3 Partial ambiguities in resonance typing 187

C.4 Tables 188

C.5 Figures 192

References **197**

List of Tables

3.1 The atomic nuclei in alanine, in a protein chain. 34

3.2 Alanine bond lengths, calculated from first principles. 35

3.3 Estimates of alanine bond angles. 35

3.4 Gyromagnetic ratios of biologically important nuclei. 35

3.5 The covalent groups that appear in the NHSQC experiment. 36

3.6 The covalent groups that appear in the HNCO experiment. 36

3.7 The covalent groups that appear in the HNCACB experiment. 37

3.8 The covalent groups that appear in the HBHA(CO)NH experiment. 38

3.9 The covalent groups that appear in the C(CO)NH-TOCSY experiment. . . . 39

3.10 The covalent groups that appear in the HC(CO)NH-TOCSY experiment. . . 40

3.11 Ambiguities in stereospecific assignments. 41

3.12 The number of times nuclei typically appear in pulse sequences. 42

4.1 Connections between various data types. 77

6.1 Quality of the NHSQC automated peak pick. 118

6.2 Quantification of the completeness of the signals in the NHSQC. 119

6.3 Spectra used in Samp3 analysis. 120

6.4 Quantification of the completeness and unambiguity of the backbone GSSs. 121

7.1 Important Sparky keyboard shortcuts. 137

8.1 The point generators of the sample scheduler. 155

8.2 The quadrature generators of the sample scheduler for each point. 155

8.3 The point selectors of the sample scheduler. 156

8.4 The modifiers of the sample scheduler. 156

8.5 The forced point selectors of the sample scheduler. 156

8.6 The formatters of the sample scheduler. 156

C.1 GSS types used in typing of H-N-rooted GSSs. 188

C.2 Key used to model various common pulse sequences as patterns on graphs. 189

C.3 Common pulse sequences and their graph patterns using the key given in Table C.2. 190

C.4 Notation for partial resonance typings. 191

List of Figures

1.1 A scientific method as a sequential, cyclic pipeline. 13

1.2 A scientific method in which each step interacts with every other. 14

3.1 General NMR and molecular knowledge. 43

3.2 Experimentally determined knowledge: obtained before analysis. 44

3.3 Knowledge obtained during analysis. 45

3.4 Information about the actual physical properties of a molecule. 46

3.5 A 1-dimensional cross-section of Gaussian peaks. 47

3.6 A uniform sample schedule. 48

3.7 A non-uniform sample schedule. 48

3.8 Restraints are used to build a structural model. 49

3.9 Torsion angles provide structural information. 50

3.10 A frequency-domain NHSQC spectrum. 51

3.11 A peak picked NHSQC spectrum. 52

3.12 The nuclei correlated by an NHSQC. 52

3.13 The nuclei correlated by an HNCACB. 53

3.14 The nuclei correlated by a CBCA(CO)NH. 53

3.15 The nuclei correlated by an HNCO. 54

3.16 The nuclei correlated by an HN(CA)CO. 54

3.17 The nuclei correlated by an HBHA(CO)NH. 55

3.18 The nuclei correlated by an H(CCO)NH-TOCSY. 55

3.19 The nuclei correlated by a C(CO)NH-TOCSY. 56

3.20 The nuclei correlated by an HCCH-TOCSY. 57

4.1 An overview of the NMR process for protein structure determination. . . . 78

4.2 The connections between various data types. 79

4.3 Matching peaks between an NHSQC and an HNCACB spectrum. 80

4.4 Overlap of carbon resonances in an HNCACB spectrum. 81

4.5 Assignment of a GSS chain to residues 81

4.6 Breakdown of the NMR process by reproducibility. 82

5.1 A relational model of a snapshot and deductions. 103

6.1 An asparagine sidechain in the NHSQC and HNCACB spectra. 122

6.2 The secondary structure prediction according to TALOS+. 123

6.3 The final structure as seen in jmol. 123

6.4 A false negative – a true signal missed by the automated peak picker. 124

6.5 An extraneous GSS. It was not assigned to any specific residue. 124

7.1 The Sparky interface, showing how to activate the reproducibility extension. 138

7.2 Two widgets provided by the reproducibility extension. 139

7.3 The Sparky data model. 140

8.1 A code snippet of NMRPyStar. 157

8.2 NMRPyStar produces a parse tree as output. 158

8.3 The parse tree can be used to extract key NMR information. 159

8.4 The results of a query run against the parse tree. 160

8.5 WB's data model, created with MySQLWorkbench. 161

8.6 A data model of sample schedules, created with MySQLWorkbench. 162

8.7 The parameter file input for a sample schedule. 162

8.8 A sample schedule generated from the parameter file. 163

8.9 A graphical view of a sample schedule. 164

8.10 A sample schedule for which the first points along each axis are collected. . 164

8.11 The pointspread function of the sample schedule, calculated using a real-only FFT. 165

8.12 The pointspread function shows a noticable difference. 165

C.1 The portion of a GSS observable from an HNCACB experiment. 192

C.2 An alternative naming scheme which accounts for ambiguity. 193

C.3 A backbone GSS in HNCACB. 194

C.4 The extent of a sidechain Q or N GSS in an HNCACB. 195

C.5 A GSS in an HNCACB, assigned to a Q sidechain. 195

C.6 A sidechain R gss in an HNCACB. 196

List of Acronyms

AMBER	Associated Model Building with Energy Refinement
BMRB	Biological Magnetic Resonance Data Bank
CCPN	Collaborative Computing Project for NMR
CHSQC	Carbon HSQC
CONNJUR	CONNecticut Joint University Research
ELN	Electronic Laboratory Notebook
(F)FT	(Fast) Fourier Transform
FID	Free Induction Decay
GLP	Good Laboratory Practices
GMP	Good Manufacturing Practices
GUI	Graphical User Interface
GSS	Generalized Spin System
HSQC	Heteronuclear Single-Quantum Coherence
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
LIMS	Laboratory Information Management System

NESG	NorthEast Structural Genomics consortium
NHSQC	Nitrogen HSQC
NMR	Nuclear Magnetic Resonance
NOE	Nuclear Overhauser Effect
NOESY	Nuclear Overhauser Effect SpectroscopY
PDB	Protein Data Bank
PPM	Parts Per Million
RDBMS	Relational DataBase Management System
RDC	Residual Dipolar Coupling
RNMRTK	Rowland NMR Toolkit
RF	Radio Frequency
ST	Spectrum Translator
STAR	Self-defining Text Archival and Retrieval
TOCSY	TOtal Correlated SpectroscopY
VCS	Version Control System
WB	Workflow Builder
XML	eXtensible Markup Language

Chapter 1

Introduction

There is no other species on the Earth that does science. It is, so far, entirely a human invention, evolved by natural selection in the cerebral cortex for one simple reason: it works.

- Carl Sagan

1.1 Protein NMR

NMR (Nuclear Magnetic Resonance) spectroscopy is an experimental technique for studying proteins and other biological molecules at atomic resolution. In comparison to other techniques for high-resolution characterization of biological molecules, NMR's significance as an experimental technique stems from its ability to collect detailed molecular data, from which further analysis can derive not only structural information but also dynamics, activity, and interactions with other molecules. The importance of NMR spectroscopy to the structural biology community has steadily increased, as measured by the number of biologically relevant molecules that have been studied using the technique, and the corresponding data deposited into publicly available databases – since 1990, the structures of nearly 9,000 proteins that

were solved using NMR have been deposited in the Protein Data Bank (PDB) [1], a facility for the archival and sharing of protein-related data, and NMR data is available for over 10,000 proteins in the BioMagnetic Resonance Bank (BMRB) [2], a facility for the archival and sharing of specifically NMR-derived data. The data collected using NMR techniques is important to the field of drug design [3, 4, 5], as it facilitates identification of potential binding partners based on surfaces as well as actual binding partners based on chemical shifts, and understanding biological processes.

1.2 Scientific methods and reproducibility

The term "science" can refer to both the enterprise of knowledge acquisition through empirical means, and the body of knowledge acquired through such means [6]. The core of science is the notion of reproducibility [7, 8] – that claims can be independently tested and verified. Reproducible knowledge can be obtained using a set of general techniques collectively known as "the scientific method". These techniques share many characteristics, most notably:

- data collection: observation of a natural phenomenon, in which experiments are performed and the results quantified and recorded
- analysis, in which the data collected in an experiment is processed to gain information, knowledge, and understanding
- experimental design: inventing and codifying a procedure for data collection which tests specific variables of a system, while preventing other variables from confounding the results
- hypotheses, which synthesize and organize the information, knowledge, and under-

standing gained in order to both explain the observed results and predict the outcomes of further experiments

Reproducible research is the combination of a scientific result, along with the data, experiments, and analyses used to obtain that result, in sufficient detail such that an independent researcher could, in principle, perform the same procedure and reach the same result.

Two example orderings of these basic steps are shown in Figure 1.1, in which a rigid structure is imposed as a sequential pipeline, and Figure 1.2, which is a more flexible method because it allows any step to influence any other step. In general, as both examples illustrate, the core of science is a method of iterative experimental designing, data collection, analysis, and hypothesis formulation, which leads to new experiments, data and analysis, and so on, resulting in the acquisition of knowledge about the natural world.

A basic property of scientific methods is that they enable reproducibility. In order for scientific results to be reproducible, all components of the experimental design, data collection and data analysis steps must be reproducible. These include:

- Experimental design. Our understanding of scientific methods, the significance of reproducibility, and means of achieving reproducibility have evolved along with our experimental methods. Lab notebooks are a standard tool for recording information. Their advantages are well-known [9]: not only do they enable flexible and persistent storage of information, they also enhance continuity within research groups as personnel, techniques, and reagents change over time by explicitly recording key information. Experimental designs must be recorded in sufficient detail to convey the key points to other scientists. It may also acknowledge the possibility of confounding variables, and take measures to control for them. The appropriate level of detail is neither too much

nor too little: it should be repeatedly executable such that similar results are obtained – within expected deviations due to experimental error and variables outside the control of the experiment [6]. The long history of usage of lab notebooks has created a rich culture of their use; scientists have a well-developed understanding of what should, and what should not, be recorded.

- Experimental execution. Lab notebooks and Laboratory Information Management Systems (LIMS) are also important for recording the details of experiments, including any problems such as contamination. Recent efforts have applied computational power to create Electronic Lab Notebooks (ELNs) [10, 9, 11]. Previous efforts led to results such as Good Laboratory Practices (GLPs) and Good Manufacturing Practices (GMPs), intended to ensure quality and reproducibility within industry research and manufacturing efforts [12, 13]. When carrying out an experiment, the actual procedure followed should be recorded in sufficient detail, including any deviations from the given experimental design and justifications for those changes. Biased sampling should also be recognized if possible [14, 15]. Journal publications are used to convey information about experimental designs and results to other scientists. Scientists thus have a firm notion of experimental reproducibility.

- Computational. As computers continue to play an ever-growing role in science, scientists have noted the problems that indiscriminate computational use poses for reproducibility [16, 17]. By the early 1990's, researchers began defining reproducible computational analysis [18], and describing strategies for achieving it. A key point is that all tools, scripts, platforms and code are available, including the exact versions used [19, 20, 21]. In addition, all parameterizations [22], as well as input and output

data should be recorded.

- Analysis. A further concern of reproducibility is with non-computational analysis; failing to recognize analytical bias and to use appropriate statistical measures has long been a source of irreproducibility in scientific endeavors [23]. This has also been previously noted in [24] in [25, 26], and was the focus of a recent Nature special (`http://www.nature.com/nature/focus/reproducibility/`). To ensure reproducibility of analysis, appropriate, necessary, and sufficient statistical measures should be used [27, 28]. If bias is present in the experimental data, its sources should be recognized and accounted for [29, 30]. Additionally, manual changes made during analysis should be recorded.

Reproducibility is important to science for several reasons [31]. First, it provides a means for measuring the quality and usefulness of a study or claim. Second, reproducibility facilitates knowledge transfer between peers, which enables fellow researchers to build on the foundation provided by a study, whether by extending the experimental design and data collection, applying the experiment in a different context, or applying additional analyses to existing data. Third, reproducibility promotes collaboration between fellow scientists by enabling sharing of data [32], information, knowledge and experimental designs. Fourth, reproducibility reduces time wastage due to inability to replicate results [24, 33, 34, 35].

1.3 Irreproducibility of computational and manual analysis in NMR

Much progress has been made within NMR in the areas of experimental reproducibility and their dissemination. In addition, many software tools exist for structure validation [36, 37, 38]. Tools such as ShiftX and SHIFTX2 [39, 40], which use final results to back-predict intermediate data, also provide consistency checks which are useful for evaluating reproducibility. The AQUA [37] program was also used to evaluate structure quality across the entire BMRB [41]. The BMRB's efforts to model, archive, and disseminate NMR-obtained data and results are critical to efficient sharing of techniques and results between NMR spectroscopists. Projects such as RECOORD [42] have produced interesting results, recalculating the structures of 500 deposited proteins using new techniques and data, and getting improved results, although the specific reasons for why the final results are better are not yet clear.

However, computational results are often irreproducible due to missing code and data; additionally, strategies for computational reproducibility covering all uses of computation are not yet in place despite efforts such as [16, 17]. Sequential, deductive processes pose different challenges for reproducibility, as strategies for efficiently capturing the full set of information in a meaningful and easy-to-understand way continue to be elusive. Integration of manual analysis with computational analysis poses further challenges for reproducibility.

Thus, this work will focus on the irreproducibility of computational and manual analysis in the field of NMR protein structure determination. The context of NMR will be explored from a data-centric standpoint, as well as the process of analysis and its interaction with

the data. This will be used to demonstrate that NMR data analysis, both computational and manual, is not reproducible because of several broad categories of data which are not explicitly recorded during the analysis process and are lost.

1.4 The significance of reproducibility to NMR

Irreproducible NMR spectral analysis and structure determination causes several problems. The value and quality of irreproducible NMR analyses are difficult or impossible to judge; irreproducibility limits the ability to transfer knowledge and techniques (for interpretation of spectra, resonance assignment, stereospecific resonance assignments, etc.) effectively between scientists, as well as preventing close collaborations during data analysis and leading to time wastage as irreproducible results are discovered and following up on them is found to be impossible. In addition, irreproducibility renders error detection and correction difficult, because the data that would show when, why, and how an error occurred would be lost. It also causes the teaching of analysis methods to students and other newcomers to be difficult due to implicit, missing data; by capturing and making explicit these data, a more complete picture of the process can be discussed and shown. Finally, irreproducible data may be less amenable to future reinterpretation; reinterpreting data is necessary when augmenting a data set with additional results, which may fill in missing pieces, but may also show the original analysis to be in error. In short, reproducibility of NMR spectral analysis and structure determination will lead to better quality results.

1.5 An approach for reproducible analysis

One possible approach for rendering analysis reproducible will be discussed. The limitations of current practices in capturing insufficient data are covered, and a data model for the additional information is presented. Then, a general strategy for using the data model is described. Additionally, this section will discuss how to effectively put this strategy into practice, covering common roadblocks and problems as well as tips and suggestions.

1.6 Reproducible NMR data set

In order to prove the utility of the reproducibility approach and annotation model in practice, it was applied to a full-scale protein structure determination process. Starting with time-domain data of the Samp3 protein, the structure determination process was carried out from start to finish, including peak picking, sequence-specific assignment, Nuclear Overhauser Effect Spectroscopy (NOESY) analysis and structure calculation. Intermediate snapshots were captured and appropriately annotated. This data set has been deposited to the BMRB with id 25258.

The data model used was based on those of the Collaborative Computing Project for NMR (CCPN) [43] and of the BMRB [2], with several extensions as previously noted to enable reproducibility. A library of NMR phenomena and their use as deductive inference rules was constructed. These rules were applied for snapshot annotation.

Whereas a single implementation of the previously described strategy is here described, in principle the approach is platform-agnostic and therefore could be implemented and used by other research groups, or added into an existing tool as an extension.

1.7 Software for practical reproducibility

Software tooling comprises a significant portion of enabling practical reproducibility. High-quality tools can make reproducibility easy, pleasant, and safe (in the sense of not error-prone), without placing additional unreasonable time, effort, and education demands on potential users. This section will explore a suite of software designed to enable and support reproducibility of NMR analysis in various ways.

First, a Sparky reproducibility extension has been developed. This extension facilitates reproducible data capture during the spectral analysis stage. Sparky [44] is a popular tool for analysis of NMR data. A major strength of Sparky is its extensibility through user-defined Python modules, which can be added without recompiling the program. This extension allows simple data snapshotting, annotation, and capture of extraneous results, without disrupting the standard Sparky user experience. The increase in workload is minimal due to Sparky's keyboard accelerators.

Second, a library for reading and writing of NMR Self-defining Text Archival and Retrieval (NMR-STAR) files was implemented [45]. NMR-STAR is the standard format used by the BMRB for the deposition and archival of NMR data [46, 47, 48]. By storing data analysis in NMR-STAR files, users gain the benefits of data integration with the BMRB – analysis results can be uploaded and thereby shared with fellow researchers. The approach taken in implementing and using this parser represents a radical departure from standard NMR software techniques; the approach ensures that the software will remain easily usable and maintainable as NMR data expands and matures.

Third, two tools for working with time- and frequency-domain data are presented: CONNJUR Spectrum Translator (ST) and CONNJUR Workflow Builder (WB) [49, 50]. ST

translates between various formats of time- and frequency-domain spectra; such a tool is necessary because of the input and output requirements of many spectral processing tools. WB provides a high-level interface to spectral processing and stores the parameterization, functions, and intermediate data in a central, relational database. This means that the stage is reproducible.

Fourth, a sample scheduler has been implemented [51]. This tool facilitates the creation of non-uniform sample schedules, which are used to collect time-domain data which are non-uniformly sampled in the indirect dimension(s). Non-uniform sampling can help decrease the amount of data collection time required, and also help in avoiding the penalties imposed by the necessity of sampling past the Rovnyak limit [52], which is 1.3 times the transverse relaxation rate, R_2. A novel data model of sample schedules covering non-uniformity not only in the time dimensions but also in the number of transients per Free Induction Decay (FID) and the quadrature was implemented. Several common algorithms were gathered from descriptions in the literature and implemented. This project facilitates reproducibility of sample schedule creation by capturing the parameterization.

1.8 CONNJUR is free and open source

All software developed by the CONNJUR team is released under standard open source licenses and is freely available on our website. The open source movement first became popular as a means for users to get control and legal rights over software that they had purchased. This level of ownership is important because it enables users to freely and perpetually use, share, inspect, fix, maintain, and improve their software. Such considerations become important in view of the rapid changes that scientific fields, including NMR, undergo:

new datatypes, analyses, statistical measures, and protocols are developed, requiring updates to old software or entirely new software to be written from scratch. Open source software provides additional value in the context of reproducibility: in order to replicate a study, one must have access to the exact same computational tools that the original study used [19]. This involves both physical access – in the sense of being able to get a program loaded onto a computer – as well as licensing issues: whether the second group has the right to use the software in the exact same way as in the original study.

Our belief is that open source software can help mitigate these and other problems, as well as aid the field in more effectively dealing with its nascent software problem, by leading to adoption of a community development model and increased sharing, reducing the barriers to future progress in the field. We can only hope that other research groups place as much value as we do on open source licenses, and that adoption of open source development models will increase.

1.9 Scope and significance

Reproducibility is a key enabler of the success of the scientific approach to acquiring knowledge. This work inspects reproducibility in the field of NMR, defines the requirements for data analysis that must be met in order to achieve reproducibility, and identifies where current practices fall short. To remedy this situation, a strategy for reproducible data analysis is presented. This strategy is made practical by means of a formal model, support from the BMRB [2], and a software implementation.

Reproducibility of analysis offers additional potential benefits. By making the data involved in the process explicit, it facilitates communication and reveals bias. Both advantages

can lead to major improvements in the quality of NMR analysis. Improved communication helps the transfer of knowledge between scientists, which is useful for building on established protocols and techniques, as well as for teaching and learning. Revealing sources of bias highlights the deficiencies of current practices, indicating areas which would most benefit from improvements. Therefore, improving the reproducibility will lead to improvements in quality as well.

1.10 Figures

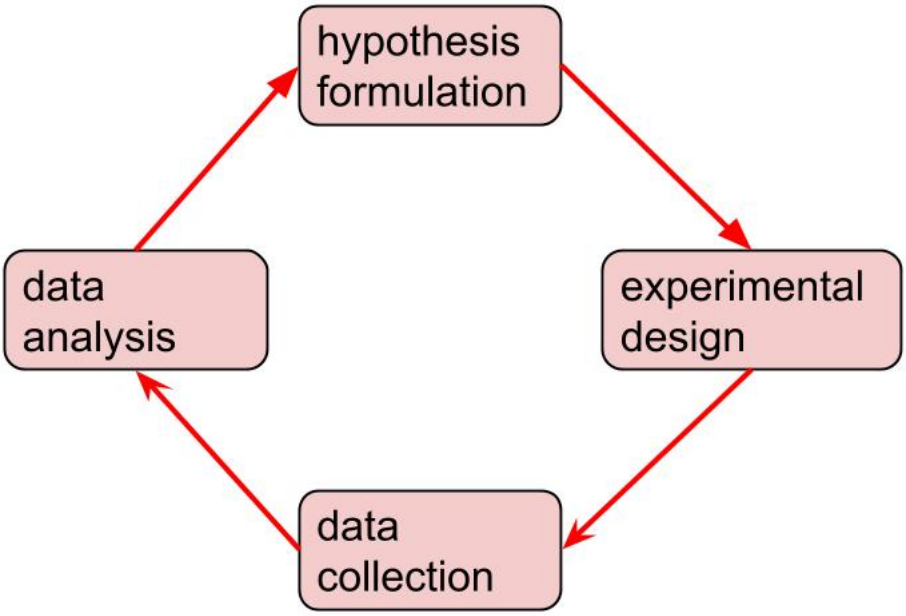

Figure 1.1: A scientific method as a sequential, cyclic pipeline.

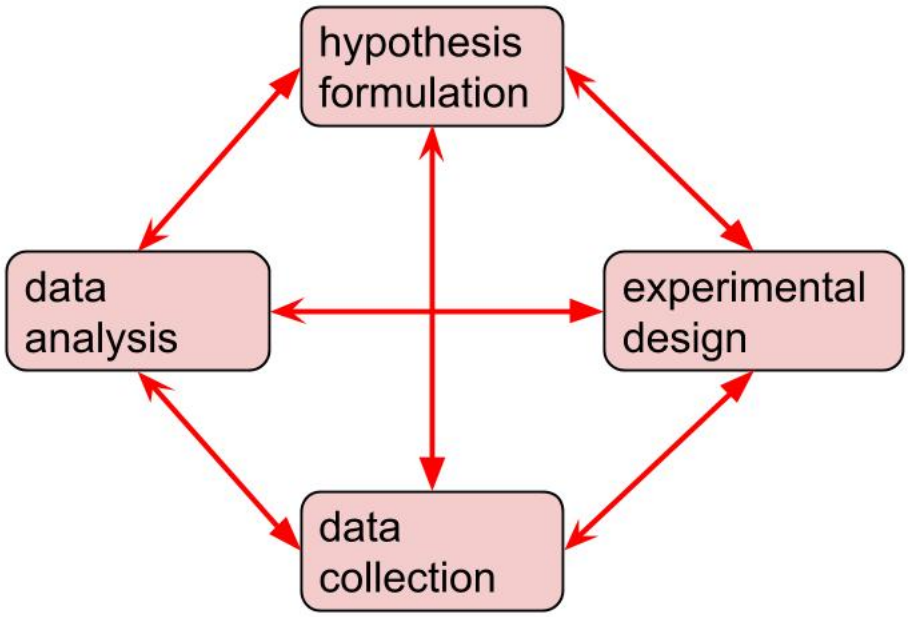

Figure 1.2: A scientific method in which each step interacts with every other.

Chapter 2

NMR Spectroscopy of Proteins

Nature's imagination far surpasses our own.

- Richard Feynman

2.1 Protein NMR

Sample preparation

The first step of an NMR analysis process is to obtain a sample of the protein of interest. NMR is a relatively insensitive technique, and so it is important to get a sample with relatively high concentration compared to other structural biology techniques, often in the millimolar range. One technique for obtaining such a high concentration of sample protein is to express the protein in bacteria. Bacteria are relatively cheap and easy to grow, and their food sources can be regulated to provide NMR-friendly isotopes such as ^{13}C or ^{15}N if desired, by controlling the growth media. The protein is produced by transforming bacteria through adding a plasmid, and then inducing transcription of the plasmid followed by translation. Finally, the protein is

isolated and purified [53].

Data collection

Assuming the purified, high-concentration protein sample does not aggregate or denature, NMR experiments are run by inserting a tube of the sample into a spectrometer, running sequences of radio-frequency pulses which selectively excite nuclei within the sample, and observing the results. The data are a superposition of decaying sinusoids of different frequencies and amplitudes; the frequencies correspond to the precession at the Larmor frequency of the various nuclei [54].

Various pulse sequences are used to probe specific functional groups within the protein. There are two major categories of pulse sequences, based on the nature of the interactions they exploit: the first group exploits scalar couplings which exist between covalently-bonded nuclei, and are thus called "through-bond" experiments [55]. The second group exploits cross-relaxation between proton pairs that are dipole-dipole coupled; these protons are spatially close but do not have to be covalently bonded, and are thus called "through-space" experiments [56]. The data produced by these two experiments are used differently.

Two important parameters of data collection are the dwell time and the number of points collected. The dwell time determines the range of frequencies that can be distinguished. The number of points collected is related to the digital resolution (the ability to distinguish between nearby, but distinct, frequencies). The acquisition time, which is the product of the dwell time and the number of points, is related to the resolution.

Maximizing sensitivity is helpful for later data analysis. Sensitivity depends on the gyromagnetic ratios of nuclei and the strength of the external magnetic field due to the

magnitude of the difference between the high- and low-energy states of a spin-1/2 nucleus according to the Boltzmann distribution. Running an experiment multiple times and summing the results is a means of increasing sensitivity as the signal increases faster than the noise, assuming random distribution of the noise [57].

Referencing ensures that results are comparable from multiple spectrometers. While the absolute Larmor frequencies of nuclei vary depending on the magnetic field of the spectrometer, the normalized values, when compared to a known material are consistent [58]. These are known as chemical shift values, and are reported in parts per million of deviation. A small amount of a known material is placed in the sample to provide referencing.

Spectral processing

The time-domain experimental data are converted to frequency-domain spectral data. A decaying sinusoid in the time-domain becomes a peak with a finite and non-zero linewidth at the corresponding frequency in the frequency-domain. The Fourier Transform [59] is a standard method for converting between time- and frequency-domains, and is often used on NMR data. Through appropriate use of scaling and normalization, the frequency axis is converted to chemical shifts.

The Nyquist theorem [60, 61] places bounds on the dwell time with respect to the final spectral width. A poor choice of dwell time can lead to spectral aliasing, in which peaks appear in unexpected spectral regions because their frequencies are outside the range supported by the chosen dwell time. Two factors confound resolution: coincidental closeness of resonances frequencies, and experimental quality. In general, larger proteins, which have more atoms than smaller proteins and therefore more nuclei as well, have more resonance

frequencies in close proximity to each other, increasing the probability of overlap. To increase resolution, data points are collected to longer times. Care must be taken to avoid decreasing the sensitivity; non-uniform sampling and Maximum Entropy reconstruction offer one means of so doing [52, 62].

A peak in a through-bond spectrum and in a through-space spectrum have different meanings. In a through-bond spectrum, a peak indicates the observation of several resonating nuclei connected by a small number of covalent bonds. However, a peak in a through-space NOESY spectrum indicates the presence of two protons within approximately 5 Angstroms of each other [63].

Peak picking

Peak picking is the process of identifying signals in an NMR spectrum using peaks as a proxy. The position of the multi-dimensional peak indicates the chemical shifts of the nuclei giving rise to it, and the amplitude may have significance in some but not all experiments.

Peak picking would be if easy if several conditions were met by the spectra:

- all expected signals appear
- all signals are easily distinguishable from noise and artifacts
- all signals are well dispersed from all others
- no unexpected signals appear

However, in practice, none of these conditions are met [64]. Thus, expected signals are missing, some signals are close to the noise level, some noise appears to be signal, some

artifacts appear, signals overlap to greater or lesser extents, and some unexpected signals appear, perhaps due to contamination or multiple conformations [65].

Therefore, accurate peak picking must deal with these problems, in order to identify all the true signals, none of the false signals, and to correctly characterize the position and volume of the true signals. Initial peak picking is often performed using a computational tool, but there typically is some level of manual intervention in order to correct mistakes and other problems [66].

Chemical shift assignment

Nuclei in NMR experiments resonate at characteristic frequencies; chemical shift assignment is the process of drawing the correspondence between the resonance frequencies identified from picked peaks and the nuclei in the protein of interest. Assignment is typically accomplished using a set of through-bond spectra, many of which are based around H-N groups and nearby nuclei, and others which obtain the chemical shifts of the aliphatic sidechains (both carbons and protons) and still others for the aromatic portions of sidechains [67, 68, 69, 70, 71].

Assignment proceeds through two key intermediate data types: generalized spin systems (GSSs) and resonances [2, 43]. Resonances are the NMR incarnations of nuclei: a nucleus resonates at a characteristic frequencies across spectra. GSSs are similar to NMR incarnations of amino acid residues, but may span multiple residues and are therefore networks of covalently-bonded resonances [72].

Peaks are assembled into GSSs by exploiting the redundancy between several experiments: resonances appear in multiple spectra, at the same characteristic frequency giving rise to

peaks (signals), and this is used to match these peaks into the same GSSs and resonances. Peaks can also be matched within a single spectrum into the same GSS, depending on the nature of the experiment [43].

GSSs and resonances are assigned an amino acid type or a nucleus type, respectively. As can be found in the BMRB, there is large variation in average chemical shifts depending on amino acid type and nucleus type, especially for serine/threonine, glycine, and alanine residues, for which the CB, CA, and CB nucleis' chemical shifts are essentially unique [73].

A second type of overlap is used in chemical shift assignment: due to the similar J-couplings of the CA both of the previous and same residue to the backbone nitrogen, it is possible to correlate a backbone amide group with both adjacent sidechains. While the J-coupling to the CA is typically larger than the J-coupling to the CA(i-1), the relative ratio is usually within a factor of two. The practical implication is that the correlations to both CA nuclei can be collected in the same experiment. Thus, each sidechain may be correlated with two backbone amide groups, causing their resonances to appear in conjunction with two other groups. As the nuclei resonate at a characteristic frequency, this can be used to identify a sequential connection between the two GSSs based on backbone amide groups. Once a sufficiently long chain is built using these sequential connections and the (possibly incomplete) GSS typings, the chains may be assigned to specific residues in the protein sequence. Combined with resonance typing, chemical shift assignments may be obtained [67, 68, 69].

However, chemical shift assignment is complicated by missing, overlapped, and extraneous signals, as well as ambiguities in GSS typings, resonance typings, sequential GSS connectivity, and sequence-specific GSS-residue assignments [64, 66]. The ambiguities

in typings are caused by the non-uniqueness of average chemical shifts for most residue types (apart from glycine, alanine, serine, and threonine) [2]. The ambiguities in sequential assignments are caused by degenerate chemical shifts across multiple residues, as well as by missing and extraneous signals, and those in sequence-specific assignments are caused by non-uniqueness of the match between GSS typing of a sequential chain and the primary sequence of the protein.

Accurate and complete chemical shift assignment requires nearly complete and correct peak picking, as well as the presence in the spectra of nearly complete expected signals, well-dispersed such that there is little to no overlap [73, 66]. It is often helpful to use a computational tool to quickly assign most of the chemical shifts, but later to make manual interventions to fix mistakes and assign any missed resonances [65].

NOESY assignment

NOESY spectra provide distance restraints between proton pairs. In order to make use of their latent structural information, the peaks must be assigned to resonances and thereby to nuclei. This is done with the help of the chemical shift assignments: NOESY peak cross-sections are matched to nuclei based on similarity of the cross-section's chemical shift to that of the resonance assigned to the nuclei.

However, NOESY data are heavily ambiguous, because there are typically several resonances with chemical shift values close enough to match a single NOESY peak cross-section. There are several strategies for mitigating this problem. One is to collect 3D or 4D NOESY experiments, in which the additional dimensions correlate covalently-bonded ^{13}C or ^{15}N nuclei to the protons involved in the NOE interaction [74]. This approach greatly reduces

the ambiguity. Furthermore, characteristic peak patterns are expected, such as intra-residue NOESYs between protons of that residue, as well as NOESY peaks between protons of sequential residues. Another approach is selective labeling [75].

Accurate and complete NOESY interpretation requires nearly complete chemical shift assignment. Furthermore, some manual intervention in NOESY assignment may be necessary to correct and validate troublesome cases, or to prevent automated assignment programs from making mistakes [73, 66].

NOESY data may be assigned manually, but are often assigned computationally as well, or with a combination of the approaches. The CYANA and ARIA structure calculation programs include facilities for automated NOESY assignment [76, 77]. A third tactic for dealing with ambiguous NOESY data is to iteratively reduce the ambiguity using network-anchoring approaches, that use initial structure estimates to drive further unambiguous NOESY assignment in a self-consistent cycle [76, 77].

Structure calculation

There are several other types of structural information obtained through NMR besides the proton-proton distance restraints provided by NOESY spectra. Using a program such as TALOS+ [78], backbone torsion angles can be predicted. TALOS+ uses the chemical shift assignments of backbone nuclei in conjunction with a database search to make its predictions. 3-J-coupling constants can be related to dihedral torsion angles through the Karplus equation [79, 80]. Residual dipolar couplings (RDCs) provide information about internuclear orientations.

These data are synthesized into a structural model by programs including CYANA,

ARIA, and XPLOR-NIH. CYANA is useful for quickly obtaining coarse structure estimates. XPLOR-NIH is able to provide more detailed structural models, but may take far longer to calculate a structure [81, 76].

Deposition

The BMRB is the main repository for information derived using NMR spectroscopy. A BMRB deposition may be prepared that includes chemical shift assignments, peaks, peak assignments, binary spectral and time-domain data, sample preparation protocol, and various other relevant data. The PDB is the main repository for structural data. BMRB depositions may be linked to PDB depositions [2, 1].

Chapter 3

A Data Overview

With insufficient data it is easy to go wrong.

- Carl Sagan

NMR data can be broadly grouped into four categories based on when it is known. First, Figure 3.1, there is information that is known without performing any NMR experiments: this includes information about the molecule as well as general NMR and molecular knowledge. Second, Figure 3.2, is data that is collected using the NMR spectrometer, and is known before analysis begins. Third, Figure 3.3, is the information generated during analysis, and fourth, Figure 3.4, is the final goal of an NMR study, information about the actual structure of the molecule of interest.

This chapter will present the data generated during the NMR process. Later chapters will focus in on subsets of the data, as well as show how the data is used during analysis.

3.1 Global prior knowledge

Knowledge of NMR and molecules that is available before any experiments are performed.

Molecule

Primary sequence

The primary sequence of amino acids of the protein is known. For example, the sequence of Ubiquitin is MQIFVKTLTG KTITLEVEPS DTIENVKAKI QDKEGIPPDQ QRLIFAGKQL EDGRTLSDYN IQKESTLHLV LRLRGG according to Swiss-Prot.

Amino acids

The atoms contained in each amino acid (see Table 3.1), bond lengths between atom pairs (see Table 3.2), and bond angles (see Table 3.3) are known [82].

NMR

Nuclei

The gyromagnetic ratio of nuclei (see Table 3.4) and pulse sequences. The gyromagnetic ratio and the magnetic field strength determine the approximate frequency range at which nuclei appear in spectra.

Time-domain data

Resonances appear as decaying sinusoids, and the full data set is a sum of many of those sinusoids at various frequencies and intensities.

Through-bond experiments

Time-domain free induction decay (FID)s are collected using pulse sequences [83] designed to target specific nuclei by exploiting the coupling constants and characteristic chemical shifts of specific nuclei and functional groups. The collected FIDs are sums of decaying sinusoids.

The pulse sequence determines which covalently bound groups will appear in experiments (see Tables 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10).

Through-space experiments

Nuclear Overhauser Effect (NOE) [84] experiments transfer magnetization between spatially near proton pairs, and do not require a network of covalent bonds. Each true peak indicates a pair of protons within approximately 5 Angstroms of each other. This is different from through-bond correlation spectra, in which peaks indicate the nuclei of covalently bonded atoms; NOESY spectra depend not on covalent bonds but rather on spatial proximity. Thus, each NOESY peak contains some information about the actual three-dimensional structure of a molecule, although this information is not used until the correspondence between peak cross section and an atom's nucleus is determined.

Resonance

A particular nucleus is expected to resonate at approximately the same frequency in all spectra (although factors such as temperature and the Bloch-Siegert shift can produce differences).

Frequency-domain spectra

The Fourier Transform of a decaying sinusoid produces a frequency-domain spectrum with a peak at a frequency matching the oscillation frequency of the sinusoid. Since NMR time-domain data consists of multiple decaying sinusoids caused by nuclei resonating at characteristic chemical shifts, the frequency domain spectrum will contain peaks for every oscillating sinusoid present in the time-domain data.

Spectra contain peaks, which are characterized to obtain volume and peak cross section attributes; peak cross sections are characterized to obtain position and width attributes (see Figure 3.5).

Chemical shift statistics

Statistics from previously analyzed molecules are maintained by the BMRB [2], which show clear trends of average chemical shifts based on both amino acid type and nucleus (see Table 3.1 for the chemical shift statistics of alanine).

Chemical shift values are correlated to secondary structure, as described by secondary chemical shift statistics [85]. Chemical shifts are also correlated to three-dimensional structure, as shown by CHESHIRE [86] and CS-ROSETTA [87].

3.2 Local prior knowledge

Knowledge which is available after performing NMR experiments, but before performing analysis.

Sample

Preparation procedure

The procedure used to express and purify a sample of interest, including growth medium and conditions, expression organism, and buffer optimization.

Isotope labeling

The specific H, N, and C labeling pattern. While ^{1}H is the most abundant isotopye of hydrogen, and convenient for NMR experiments, ^{13}C is the most abundant isotope of carbon and is not NMR-active; ^{2}H is not as easy to observe because of its nuclear spin value of 1 (as opposed to the proton's spin value of 1/2), but can improve the quality of experiments observing nearby protons because, in comparison to protons, nearby nuclei are afforded fewer relaxation pathways. Labeling patterns with specific properties have been implemented to take advantage of these different properties, as in SAIL-FLYA [88]. See Table 3.4.

Sample contents

The components present in the sample as well as their concentrations. Samples can degrade and aggregate over time, and may be affected by the NMR experiments.

NMR

Spectrometer

Operating characteristics of the spectrometer, such as its field strength.

Time-domain data

The experimental conditions in which each time-domain data set is collected, its sampling schedule, and the FIDs themselves.

For each time-domain data set, the delay between data points, number of points collected, and total delay between the first and last point (or acquisition time, which may be calculated from the first two parameters).

The sample schedule gives rise to frequency-domain artifacts according to its point spread function, and also determines which means can be used to process the time-domain data to frequency-domain. Two examples of different types of sample schedule are shown in Figure 3.6 and Figure 3.7.

Spectra

The set of frequency-domain spectra and the spectral processing workflow used to construct them. Key attributes are the spectral width, or range of different frequencies that can be distinguished, and resolution, or the ability to distinguish nearby but distinct signals in the frequency domain [89]. See Figure 3.10 for an example Nitrogen Heteronuclear Single-Quantum Coherence (HSQC) spectrum.

Stereospecific ambiguities

Given the set of spectra collected, the atoms in the molecule, and the isotope labeling scheme, there may be stereospecifically ambiguous assignments. See Table 3.11 for a list of potential ambiguities.

3.3 Analysis

Knowledge which is obtained while analyzing the experimentally collected data.

This includes peak picking (see Figure 3.11 for a peak picked spectrum), the assembly of peaks into GSSs, GSS and resonance typing, sequential GSS assignments, and sequence-specific GSS assignments. In general, the correspondence between resonances, which appear in NMR spectra as peak cross sections, and atomic nuclei is not known. The correspondence between resonances and peak cross sections is also not known, and is difficult to determine in the presence of ambiguity.

GSSs [90, 91, 92, 93, 94] and resonances [43] correspond to residues and nuclei. They are key to data analysis because they link the data collected in NMR experiments to the molecular details of the sample. Although these concepts had been used to a limited extent by earlier programs such as XEasy and Sparky [95, 44], more recent work has treated GSSs and resonances as explicit, first-class members of data analysis [43, 2]. These definitions are based on the BMRB and CCPN data models, the complete documentation of the NMR-STAR data dictionary may be found online at `http://www.bmrb.wisc.edu/dictionary/` and `http://www.ccpn.ac.uk/software/extras/datamodelfolder/datamodel`.

Peak

A feature of a spectrum that corresponds to a group of covalently bound resonances (in a through-bond spectrum) or to a pair of nearby protons (in a NOESY spectrum). A peak has one cross section for each dimension of the spectrum in which it appears; each cross section corresponds to a resonance at a characteristic frequency.

Resonance

A resonance is an NMR-visible signal which corresponds to a nucleus [43]. In general, a nucleus resonates at a single characteristic frequency based on its local environment; a resonance will be found at the same frequency across multiple experiments. This phenomenon is used to aid in identification, although it is confounded by degeneracy as well as proteins with multiple conformations.

A resonance is used to link a peak cross section to a nucleus, for a chemical shift assignment, by means of a spin system and a residue. Each peak cross section is assigned a resonance, and the resonances are assigned to spin systems, with the semantics that they are covalently bound.

Generic spin system

A generic spin system (GSS) is an NMR-visible group of peaks, typically across multiple spectra, which corresponds to a group of covalently bonded resonances; it is similar to a residue. The key to GSSs is a set of multi-dimensional pulse sequences designed to correlate resonances within GSSs [96, 68, 67, 69], which are based on an N-H group (due to its sensitivity and chemical shift dispersion) and correlate additional nearby nuclei through covalent bonds.

These pulse sequences use several types of overlap. First, each includes the N-H group. Second, due to the similar scalar couplings between N and the CA and CA(i-1) nuclei, it is possible to simultaneously correlate an N-H group with both nearby CA nuclei; this means that each CA may be correlated with two N-H groups, or in other words, may be a part of two H-N-rooted GSSs. Third, due to the scalar coupling between N and CO(i-1), correlations to

C*(i-1) appear in two pulse sequences of a pair (e.g. HNCACB and CBCA(CO)NH), while C* appear in only the HNCACB. See Figure 3.12, Figure 3.13, Figure 3.14, Figure 3.15, Figure 3.16, Figure 3.19, Figure 3.20, Figure 3.18, and Figure 3.17 for an illustration of the correlated nuclei. Table 3.12 tabulates the number of distinct measurements of each nucleus that can be obtained from common pulse sequences under ideal conditions.

These characteristics lead to a definition of a GSS: a root resonance or resonances, typically an amide H-N group, and additional covalently-bonded resonances. The precise extent of a GSS is in principle determined by the available NMR experiments [68, 67, 69]. In practice, a backbone GSS often is initially comprised of a backbone H-N, CO, CA, CB, CO(i-1), CA(i-1), and CB(i-1).

Sidechain GSSs

The standard pulse sequences can also excite sidechain resonances, if the chemical shifts of the resonances and the coupling constants between those resonances are similar to those of the targeted backbone resonances. Typically, sidechain GSSs are observed for tryptophan, asparagine, glutamine, and arginine residues in H-N-rooted pulse sequences. The potential GSS types for common pulse sequence are given in Tables 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10.

3.4 Desired knowledge

Restraints

Through the analysis process, structural constraints are obtained. These constraints include H-H interatomic distances obtained from NOESY spectra (see Figure 3.8), residual dipolar couplings (RDCs) (see Figure 3.8) which give orientation constraints between two nuclei,

and torsion angles (see Figure 3.9) which constrain the orientation of three adjacent bonds. In structural models, constraints that are not satisfied. are known as "violations".

Structure

The three-dimensional coordinates of the atoms, or correspondingly, the three-bond torsion angles. Either of these describes the molecular structure.

3.5 Tables

Atom name	Average chemical shift (PPM) of nucleus
C	187.20
O	–
N	123.32
H	8.19
HA	4.25
CA	53.16
HB1	1.35
HB2	1.35
HB3	1.35
CB	19.06

Table 3.1: The atomic nuclei in alanine, in a protein chain. Average chemical shift statistics are taken from the BMRB.

Atom 1	Atom 2	length (Angstroms)
C	N	1.45
C	C	1.51
C	O	1.25
N	H	1.03
C	H	1.1
O	H	1.1

Table 3.2: Alanine bond lengths, calculated from first principles in [82].

Atom 1	Atom 2	Atom 3	estimated angle (degrees)
C	CA	CB	128
C	CA	N	112
N	CA	CB	110
O	C	CA	120

Table 3.3: Estimates of alanine bond angles. The first two are calculated from first priniciples in [82]; the latter two are based on number of atoms bonded to the central atom.

Nucleus	gyromagnetic ratio (MHz / Tesla)
^{1}H	42.576
^{13}C	10.705
^{15}N	-4.316
^{19}F	40.052
^{31}P	17.235

Table 3.4: Gyromagnetic ratios of biologically important nuclei.

Covalently-bound group	Amino acid sequence
H-N	[^P]
HE-NE	R
HD21-ND2	N
HD22-ND2	N
HE21-NE2	Q
HE22-NE2	Q
HE1-NE1	W

Table 3.5: The covalent groups that appear in the NHSQC experiment.

Covalently-bound group	Amino acid sequence
H-N-C(i-1)	.[^P]
HE-NE-CZ	R
HD21-ND2-CG	N
HD22-ND2-CG	N
HE21-NE2-CD	Q
HE22-NE2-CD	Q
HE1-NE1-CE1	W

Table 3.6: The covalent groups that appear in the HNCO experiment.

Covalently-bound group	Amino acid sequence
H-N-CA	.[^P]
H-N-CA(i-1)	.[^P]
H-N-CB	.[^PG]
H-N-CB(i-1)	[^G][^P]
HE-NE-CD	R
HD21-ND2-CB	N
HD21-ND2-CA	N
HD22-ND2-CB	N
HD22-ND2-CA	N
HE21-NE2-CG	Q
HE21-NE2-CB	Q
HE22-NE2-CG	Q
HE22-NE2-CB	Q

Table 3.7: The covalent groups that appear in the HNCACB experiment.

Covalently-bound group	Amino acid sequence
H-N-HA(i-1)	[^G][^P]
H-N-HA2(i-1)	G[^P]
H-N-HA3(i-1)	G[^P]
H-N-HB(i-1)	[ITV][^P]
H-N-QB(i-1)	A[^P]
H-N-HB2(i-1)	[PRNDCQEHLKMFSWY][^P]
H-N-HB3(i-1)	[PRNDCQEHLKMFSWY][^P]

Table 3.8: The covalent groups that appear in the HBHA(CO)NH experiment.

Covalently-bound group	Amino acid sequence
H-N-C*(i-1), * in (A)	G[^P]
H-N-C*(i-1), * in (A, B)	[HDSNCAFYW][^P]
H-N-C*(i-1), * in (A, B, G)	[EQM][^P]
H-N-C*(i-1), * in (A, B, G2)	T[^P]
H-N-C*(i-1), * in (A, B, G, D)	[RP][^P]
H-N-C*(i-1), * in (A, B, G1, G2)	V[^P]
H-N-C*(i-1), * in (A, B, G, D, E)	K[^P]
H-N-C*(i-1), * in (A, B, G1, G2, D1)	I[^P]
H-N-C*(i-1), * in (A, B, G, D1, D2)	L[^P]
HD21-ND2-C*, * in (B, A)	N (sidechain)
HD22-ND2-C*, * in (B, A)	N (sidechain)
HE21-NE2-C*, * in (G, B, A)	Q (sidechain)
HE22-NE2-C*, * in (G, B, A)	Q (sidechain)

Table 3.9: The covalent groups that appear in the C(CO)NH-TOCSY experiment.

H-N-H*(i-1), * in A2, A3	G[^P]
H-N-H*(i-1), * in A, B2, B3	[HDSNCFYW][^P]
H-N-*(i-1), * in HA, QB	A[^P]
H-N-*(i-1), * in HA, HB, QG2	T[^P]
H-N-H*(i-1), * in A, B2, B3, G2, G3	[EQM][^P]
H-N-H*(i-1), * in A, B2, B3, G2, G3, D2, D3	[RP][^P]
H-N-*(i-1), * in HA, HB, QG1, QG2	V[^P]
H-N-H*(i-1), * in A, B2, B3, G3, G3, D2, D3, E2, E3	K[^P]
H-N-*(i-1), * in HA, HB, HG12, HG13, QG2, QD1	I[^P]
H-N-*(i-1), * in HA, HB2, HB3, HG, QD1, QD2	L[^P]
HD21-ND2-*, * in (HB3, HB2, HA)	N (sidechain)
HD22-ND2-*, * in (HB3, HB2, HA)	N (sidechain)
HE21-NE2-*, * in (HG3, HG2, HB3, HB2, HA)	Q (sidechain)
HE22-NE2-*, * in (HG3, HG2, HB3, HB2, HA)	Q (sidechain)

Table 3.10: The covalent groups that appear in the HC(CO)NH-TOCSY experiment.

Ambiguity type	Atomic nuclei	Amino acid types
3 nuclei, 1 peak	QB	A
3 nuclei, 1 peak	QG1	I
3 nuclei, 1 peak	QG2	[TI]
3 nuclei, 1 peak	QE	M
2 nuclei, 2 peaks	HA2/HA3	G
2 nuclei, 2 peaks	HB2/HB3	[RHKDESNQCPLMFYW]
2 nuclei, 2 peaks	HG2/HG3	[RKEQPM]
2 nuclei, 2 peaks	HG12/HG13	I
2 nuclei, 2 peaks	HD2/HD3	[RKP]
2 nuclei, 2 peaks	HD21/HD22	N
2 nuclei, 2 peaks	HE2/HE3	K
2 nuclei, 2 peaks	HE21/HE22	Q
2 nuclei, 2 peaks	CG1/CG2	V
2 nuclei, 2 peaks	CD1/CD2	L
2 nuclei, 2 peaks or 2 nuclei, 1 peak	HD1/HD2	[YF]
2 nuclei, 2 peaks or 2 nuclei, 1 peak	HE1/HE2	[YF]
2 groups of 3 nuclei, 2 peaks	QG1/QG2	V
2 groups of 3 nuclei, 2 peaks	QD1/QD2	L

Table 3.11: Ambiguities in stereospecific assignments.

	NHSQC	HNCO	HN(CA)CO	HNCACB	CBCA(CO)NH
H	1	1	2	4	2
N	1	1	2	4	2
CO	0	1	0	0	0
CO(i-1)	0	1	1	0	0
CA	0	0	0	1	0
CA(i-1)	0	0	0	1	1
CB	0	0	0	1	0
CB(i-1)	0	0	0	1	1

Table 3.12: The number of times nuclei typically appear in pulse sequences.

3.6 Figures

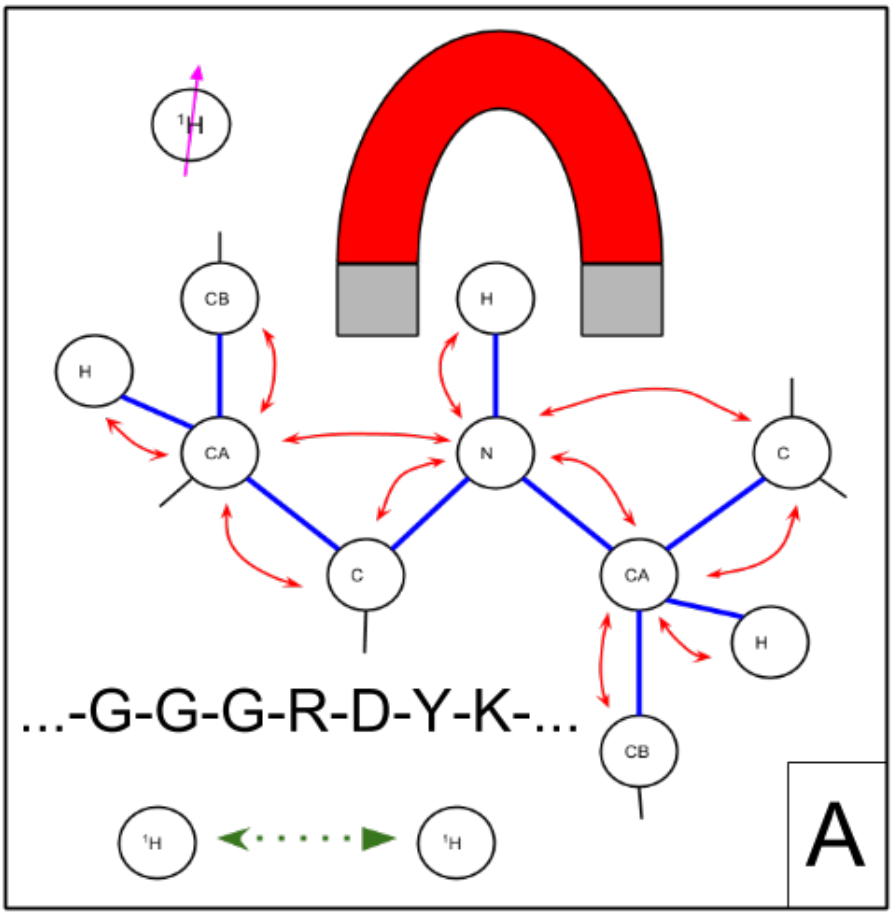

Figure 3.1: General NMR and molecular knowledge, including NMR phenomenon such as through-bond and through-space interactions, primary sequence, and gyromagnetic ratios.

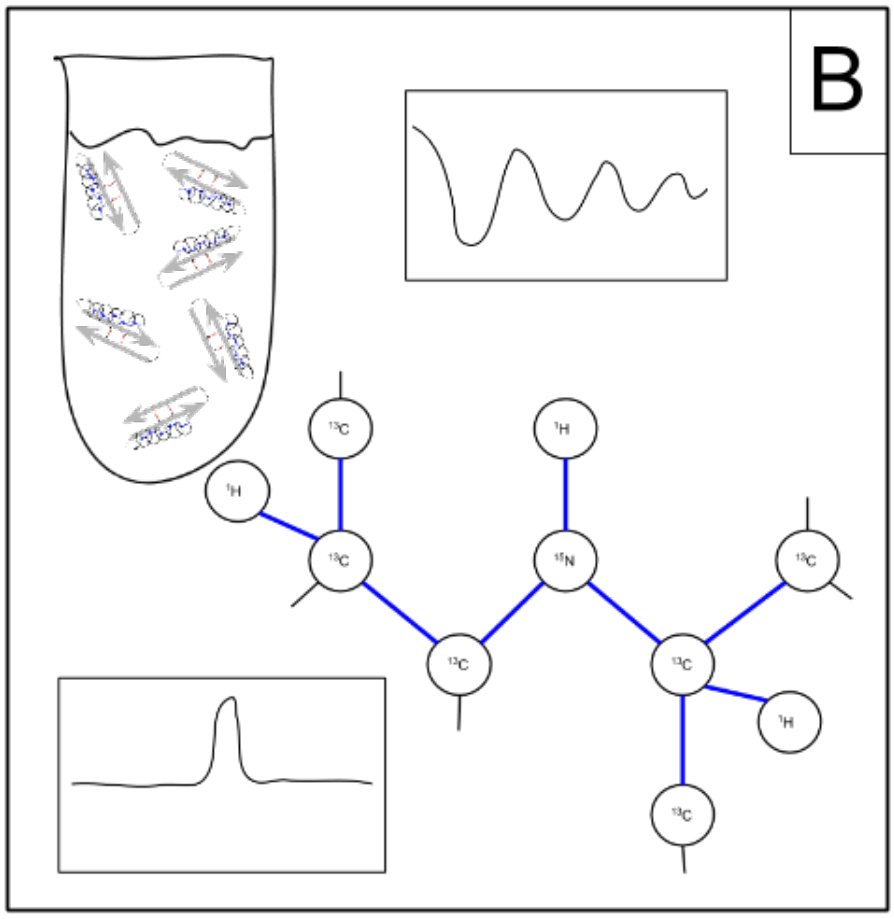

Figure 3.2: Experimentally determined knowledge: obtained before analysis, including sample preparation and conditions, isotopic labeling, and time-domain and frequency-domain data.

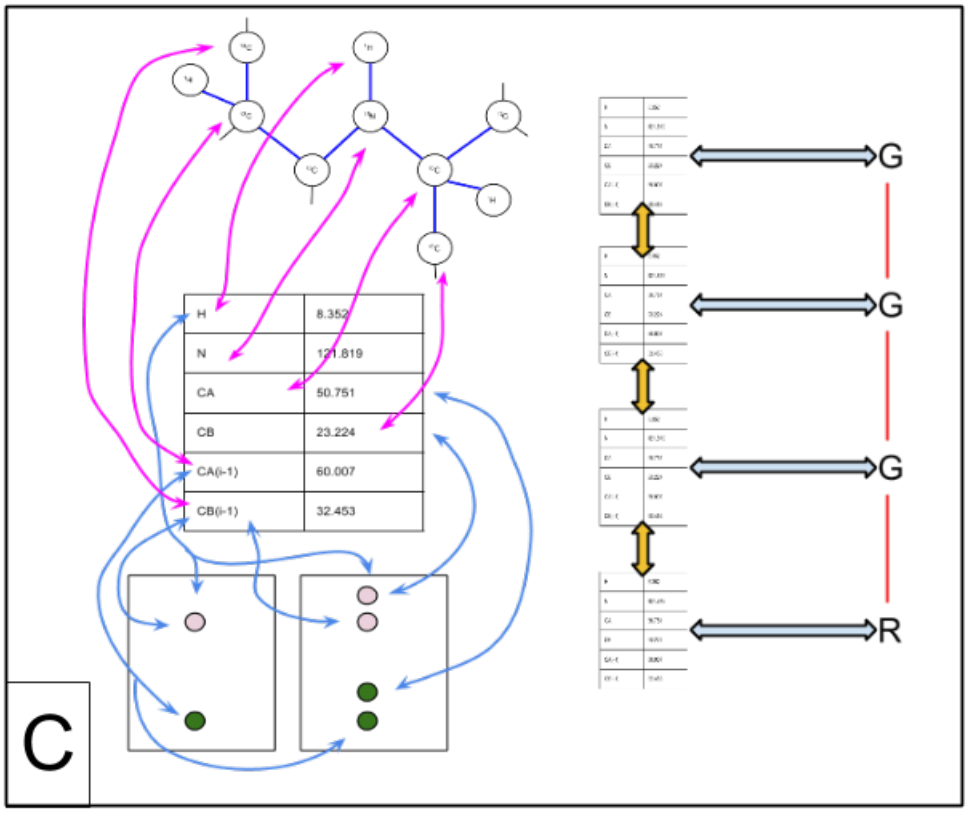

Figure 3.3: Knowledge obtained during analysis, including peaks, GSSs, resonances, sequential and sequence-specific assignments, and chemical shifts.

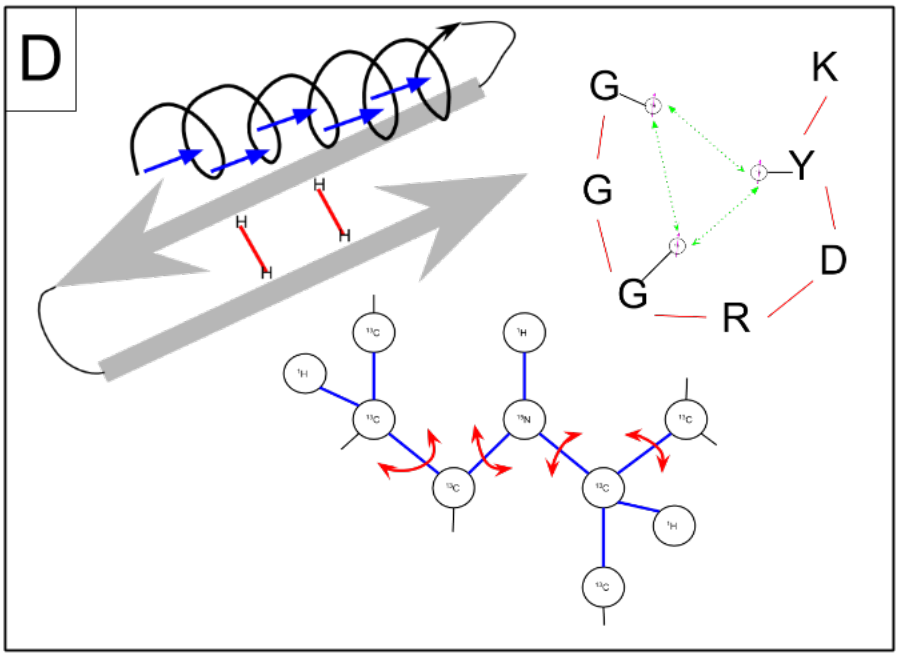

Figure 3.4: Information about the actual physical properties of a molecule is derived from structural and angle restraints.

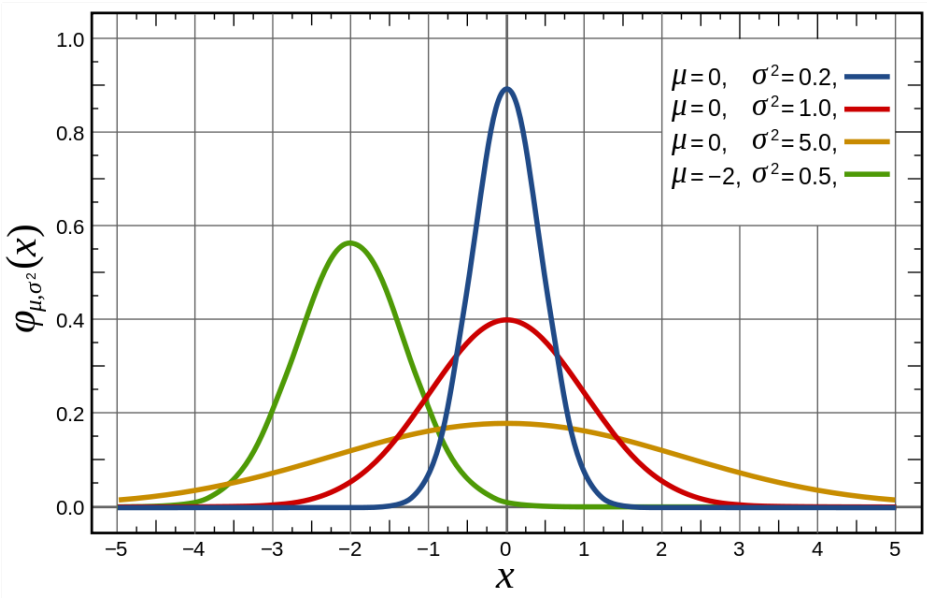

Figure 3.5: A 1-dimensional cross-section of Gaussian peaks. Each peak cross section may be characterized by its position, as well as its width. The peak as a whole has an intensity. This image is in the public domain and was accessed at `http://en.wikipedia.org/wiki/File:Normal_Distribution_PDF.svg`.

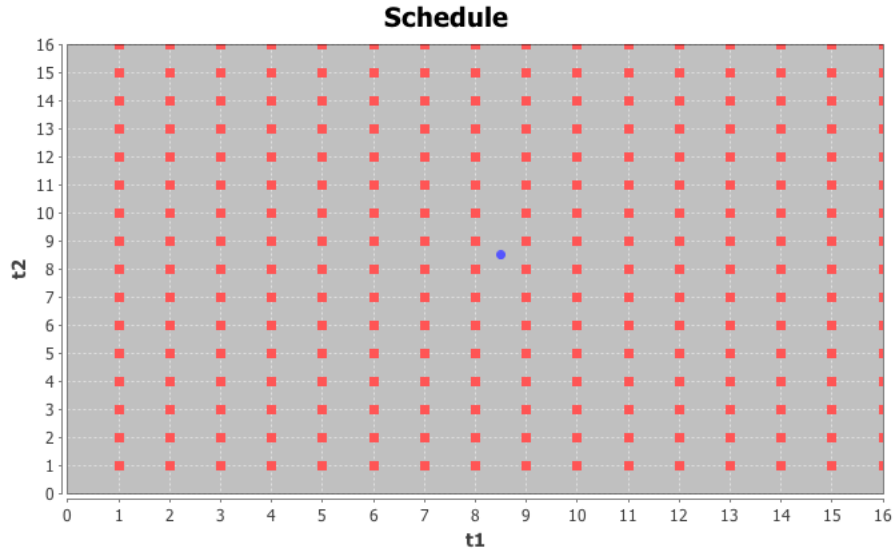

Figure 3.6: A uniform sample schedule. The gaps between the points are constant. The two axes represent the variable time delays in the two indirect dimensions of a three-dimensional experiment.

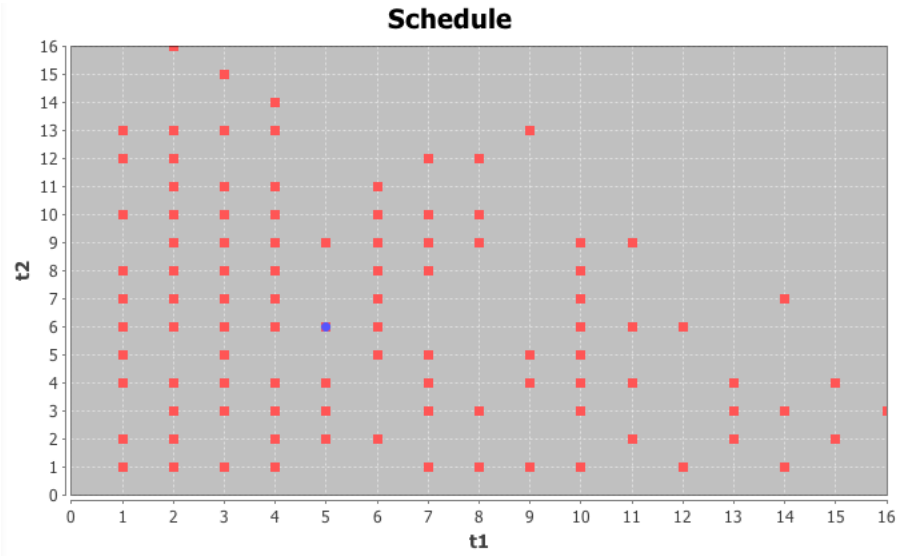

Figure 3.7: A non-uniform sample schedule. The gaps between the points are not constant. The two axes represent the variable time delays in the two indirect dimensions of a three-dimensional experiment.

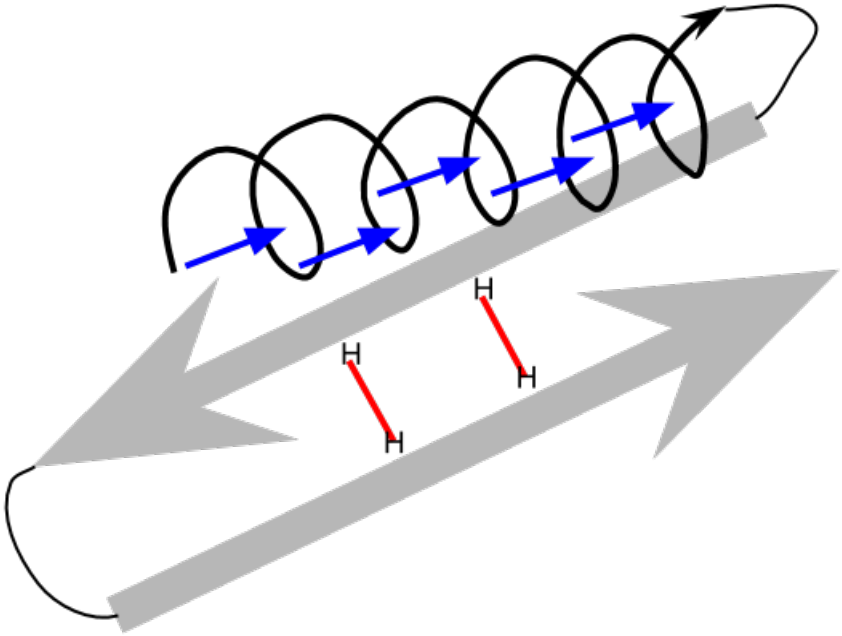

Figure 3.8: Restraints are used to build a structural model. Residual dipolar couplings give orientation constraints, and NOESY spectra give interatomic (H-H) distance constraints.

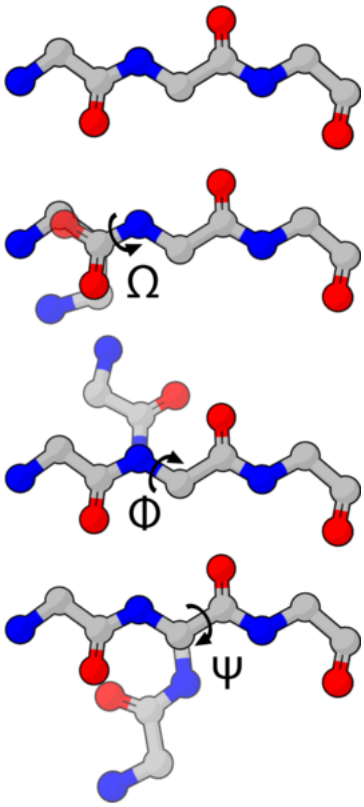

Figure 3.9: Torsion angles provide structural information. A torsion angle is described by the positions of four atoms, and is the angle between two planes. For protein structures, two key torsion angles are between the backbone atoms of each residue. This image is in the public domain, and was accessed from `http://en.wikipedia.org/wiki/File:Peptide_angles.png`.

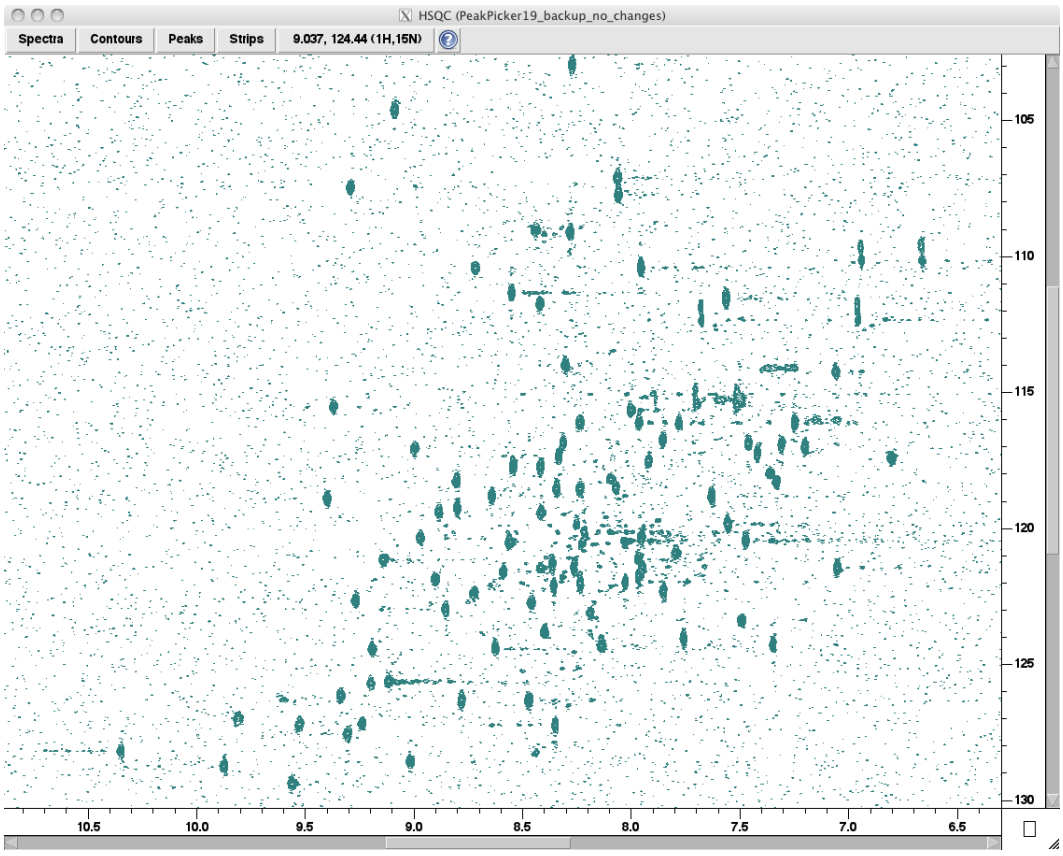

Figure 3.10: A frequency-domain NHSQC spectrum. The x- and y-axes are nitrogen and proton, respectively.

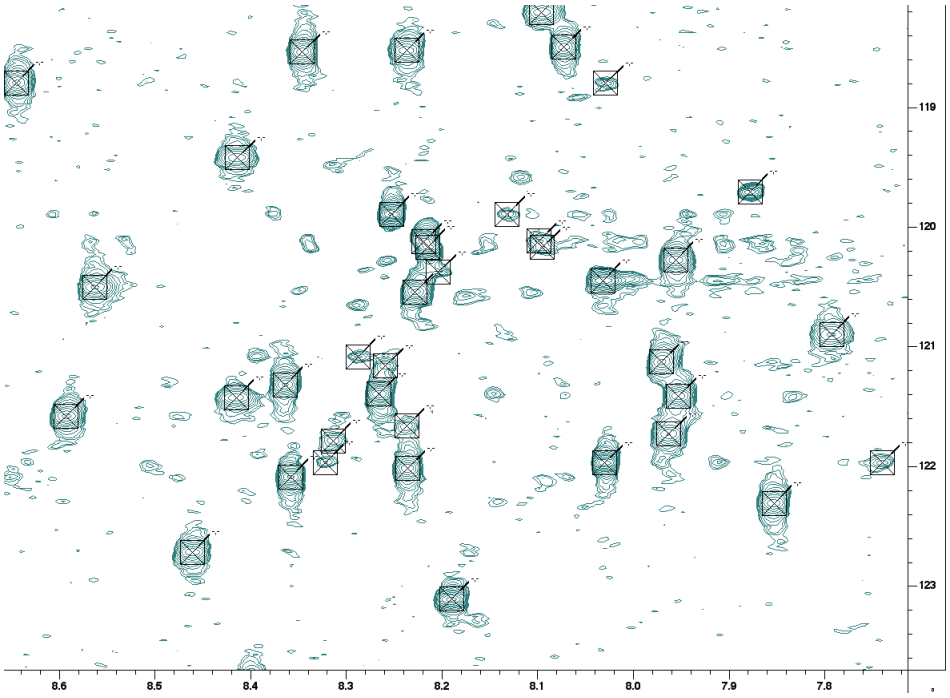

Figure 3.11: A peak picked NHSQC spectrum. Peaks are indicated by squares and crosses.

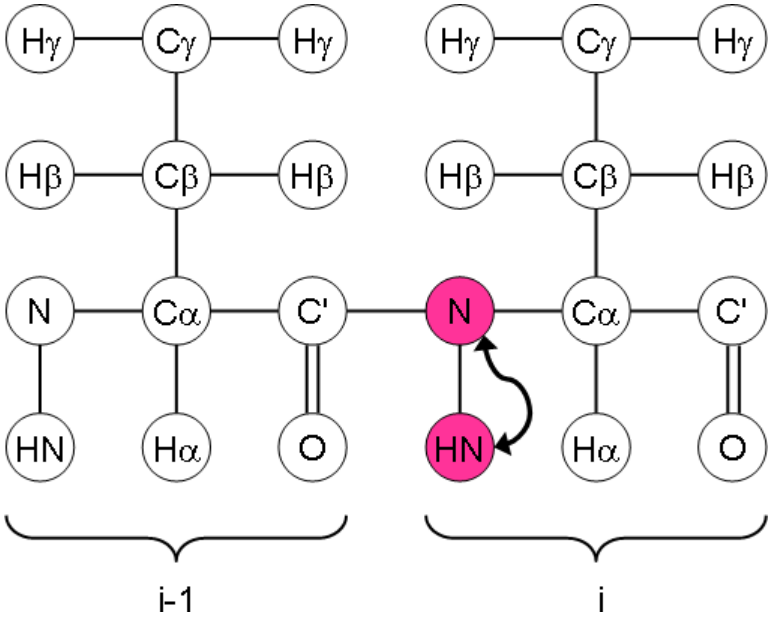

Figure 3.12: The nuclei correlated by an NHSQC. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

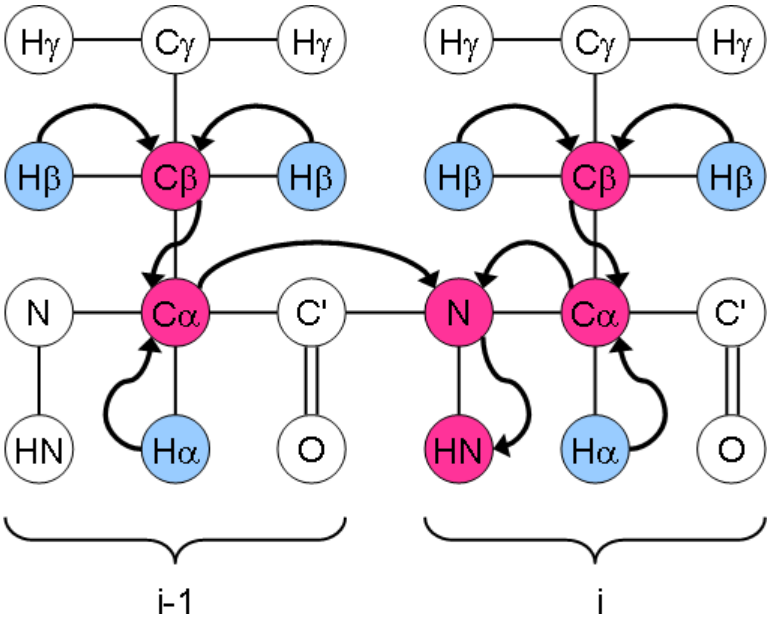

Figure 3.13: The nuclei correlated by an HNCACB. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

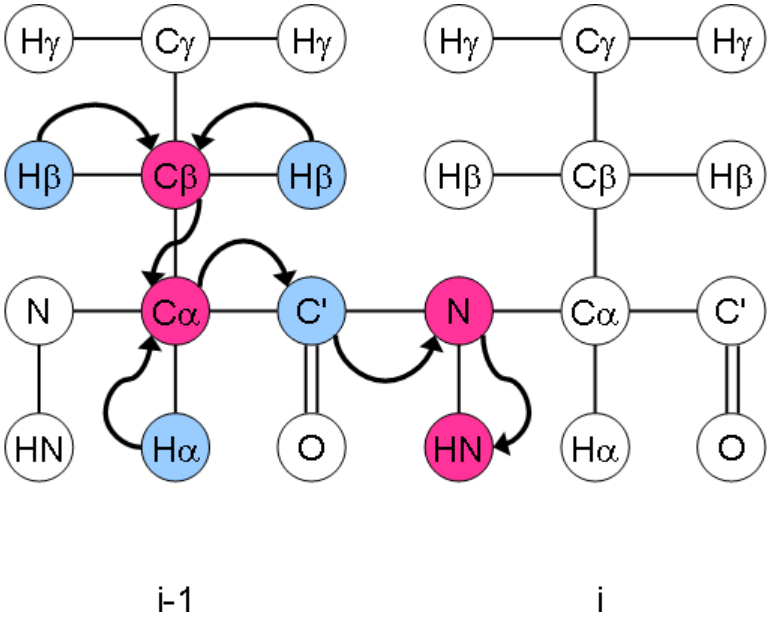

Figure 3.14: The nuclei correlated by a CBCA(CO)NH. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

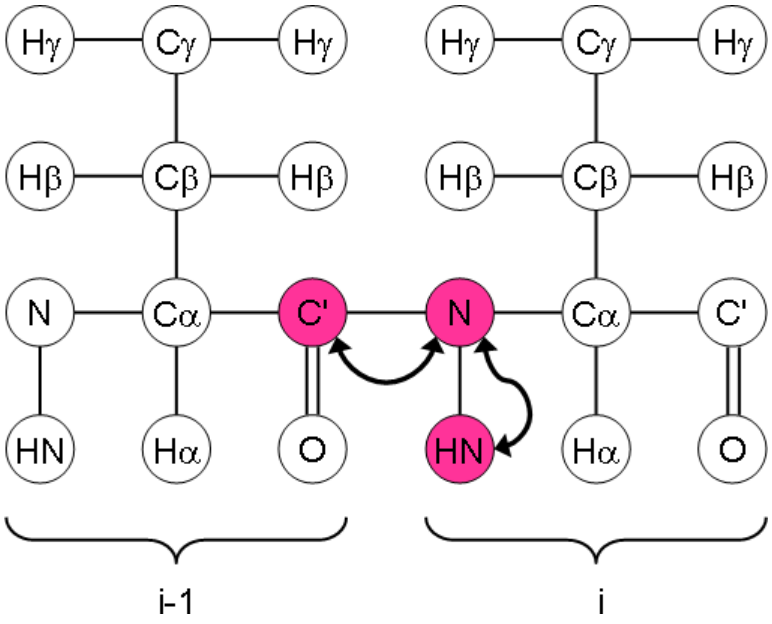

Figure 3.15: The nuclei correlated by an HNCO. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

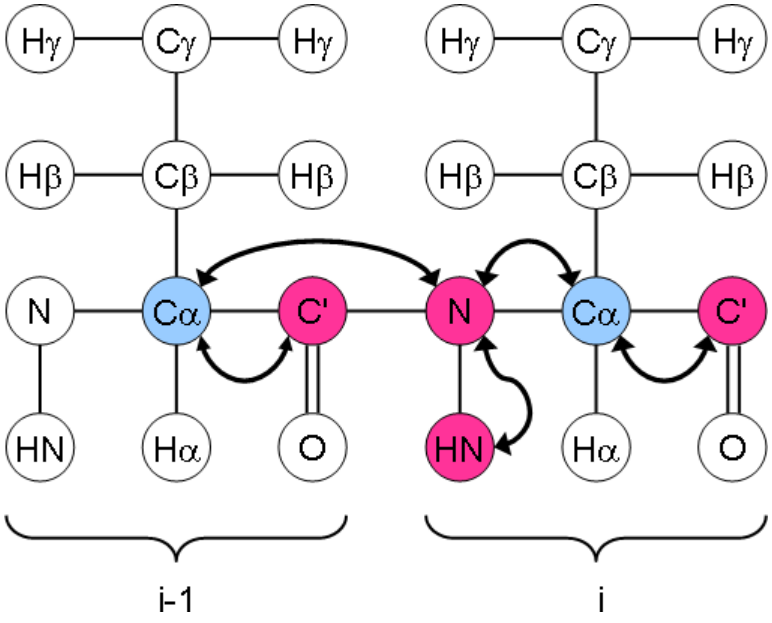

Figure 3.16: The nuclei correlated by an HN(CA)CO. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

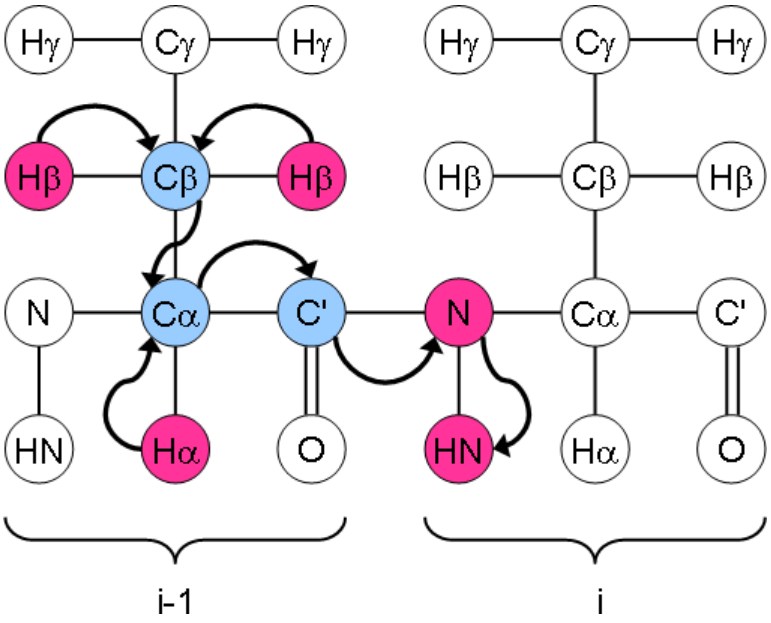

Figure 3.17: The nuclei correlated by an HBHA(CO)NH. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

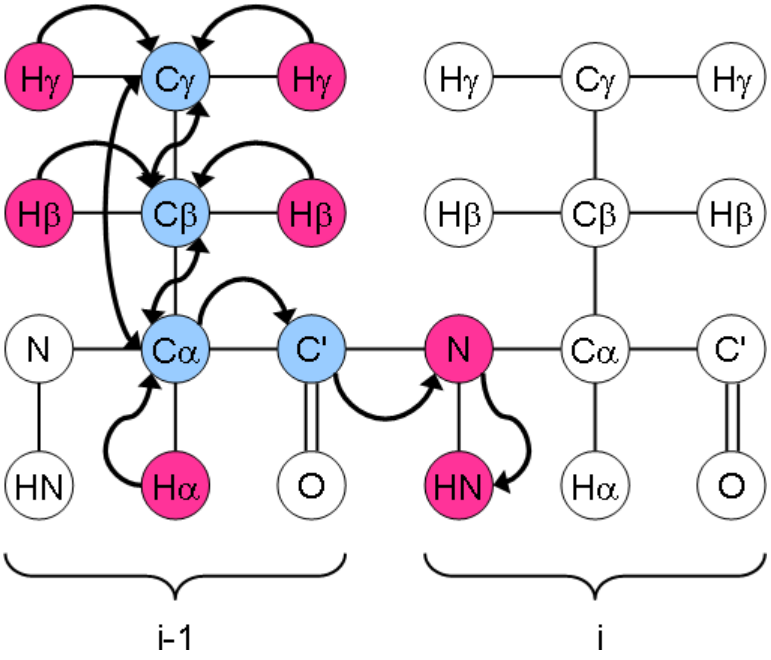

Figure 3.18: The nuclei correlated by an H(CCO)NH-TOCSY. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

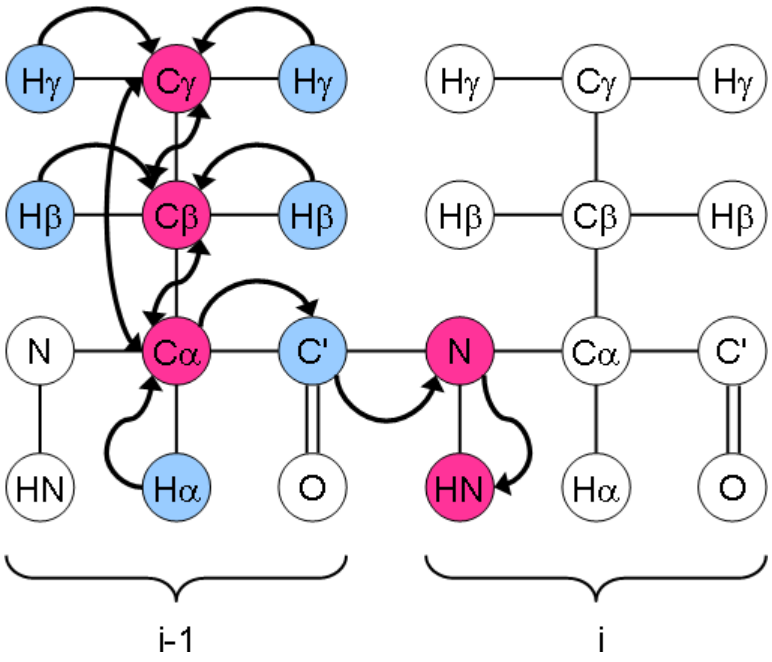

Figure 3.19: The nuclei correlated by a C(CO)NH-TOCSY. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

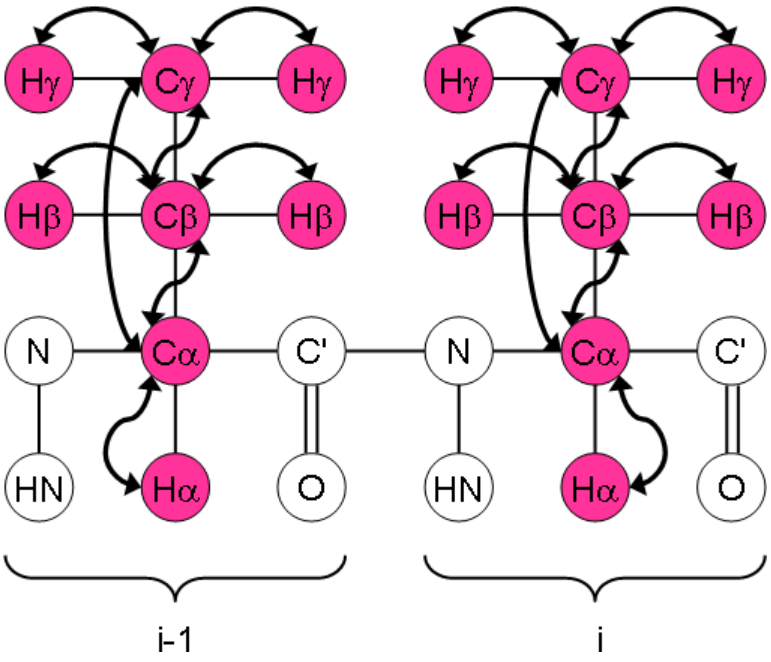

Figure 3.20: The nuclei correlated by an HCCH-TOCSY. This figure is reproduced from `http://www.protein-nmr.org.uk/` with the permission of Victoria Higman.

Chapter 4

A Process Overview

If you can't describe what you are doing as a process, you don't know what you're doing.

- W. Edwards Deming

This chapter will describe the NMR data analysis process in detail, including the roles of computational and manual analysis, their interaction with the data types, and tools used in the process. In order to study proteins in solution using NMR, a multi-step process is employed to collect and analyze data, as shown in Figure 4.1, which breaks the process down into a series of independent stages [66]. Figure 4.2 shows a view of the process in the context of the data discussed in the previous chapter.

4.1 Data collection

See panel A of Figure 4.1.

Sample preparation

The protein or molecule of interest is isolated and a solution obtained. The preparation procedure employed will determine isotope labeling and concentration.

Time-domain data acquisition

The solution is placed in an NMR spectrometer and an array of pulse sequences are used to collect time-domain data.

The collection of data suitable for Fourier Transform processing, described in the next section, requires uniformly collected data in each dimension.

Sensitivity, which determines the ability to discern true signals from noise in the frequency domain, places constraints on data collection [52]. Non-uniform sampling approaches [51] avoid these tradeoffs by collecting more data points where the signal-to-noise ratio is high, but maintain resolution by still collecting some data points where the signal-to-noise is low.

4.2 Spectral processing

In Figure 4.1 panel B, spectral processing operates on these FID data sets. They are converted to frequency-domain spectra using tools such as NMRPipe [97] and the Rowland NMR ToolKit [98]. Functions such as zero-fills, Fourier transforms, phase shifts, apodizations, and linear predictions are applied to the data as a processing pipeline. These functions are used to ensure that the spectra are amenable to further analysis, by optimizing peak size and shape and minimizing processing artifacts.

The raw data collected from an NMR spectrometer is referred to as time-domain data. In a typical NMR experiment, these data represent the sum of multiple decaying sinusoids.

These FIDs are converted to frequency-domain spectra which are used in further analysis. The goal of this phase is to construct a frequency spectrum which indicates the resonance frequencies of the nuclei that were observed in the experiment. A common tool for such a transformation is the Fourier Transform, which is able to convert a uniformly collected data set into a frequency spectrum. An example of an NMR spectrum of N-H groups is shown in Figure 3.10. Due to relaxation decreasing the amplitude of an NMR signal over time, peaks have an intrinsic linewidth in the frequency spectrum. Related approaches include multidimensional decomposition [99] and maximum entropy reconstruction [100]. When FIDs are non-uniformly collected, these processing methods are required.

Considerations include minimization of processing artifacts, signal-to-noise ratio, accounting for water lines, avoiding rolling baselines and baseline offsets, linewidth and shape, phasing, and apodization. Multiple software packages exist for carrying out this conversion, such as NMRPipe and RNMRTK [97, 98]. These packages include functions for processing the data in specific ways to guarantee desirable qualities. A typical procedure for spectral processing involves the sequential application of multiple functions from one of these packages. At each stage, the input is a data set and associated meta data, which includes information such as spectral width, dwell time, and number of points. Each function may require the setting of one or more parameters in order to proceed. Thus, in addition to the final frequency-domain spectrum, the process generates several intermediate data sets, several intermediate meta data sets, and the sequence of functions used along with their parameterizations. A previous program developed by our lab, known as CONNJUR WB, has enabled the convenient collection of necessary meta data during spectral processing [50].

4.3 Spectral analysis

In the spectral analysis stage, Figure 4.1 panel C, the overall goal is to identify the chemical shifts of individual nuclei. The spectra may be analyzed using a tool such as XEasy [95], Sparky [44], NMRViewJ [101], or CCPN Analysis [43].

In each spectrum, peak picking is performed, and true signal peaks must be identified and separated from peaks caused by noise and artifacts. Additionally, signal peaks caused by contaminants must be identified. Next, GSSs are identified and constructed [43]. The connectivity of resonances in a GSS is exploited in through-bond experiments. GSSs then must be assigned connectivities to other GSSs through overlap of mutual resonances, amino acid types, and finally specific residues of the sample of interest. Resonances must also be assigned to specific nuclei [43], with the final result being that specific nuclei in the sample of interest are assigned chemical shift values. Currently, 100% assignments are not achievable due to several factors such as data quality, ambiguity, missing resonances, and metal ions [66]. Between 80% and 95% completion may be required [64] for successful analysis at later stages.

Peak picking

Peak picking is the process of identifying and characterizing the peaks in a spectrum. The goal is to identify all true signal peaks, while recognizing and separating false peaks.

To a first approximation, a peak is identified by a local maximum in the frequency spectrum. However, not all local maxima are necessarily true peaks: noise and artifacts give rise to false peaks. Nor do all peaks show up as local maxima if they are weak and close to the noise level, which causes them to be nearly indistinguishable from the noise and baseline;

this may be due to sample instability such as aggregation or precipitation, or low sample concentration [102, 103, 104, 105, 106, 107] [64, 73, 108, 65]. Each peak cross section has a non-zero width; adjacent signals may give rise to overlapping peaks, distorting measurement of their attributes and possibly also leading to disappearance of a local maximum. Figure 3.11 shows a portion of a peak picked NHSQC spectrum; note the overlap, and that some – but not all – low-intensity spectral features have been identified as peaks.

Given these inherent issues, a general strategy for peak picking is described in [106, 102] and summarized here. First, the noise level is estimated and points below it are discarded. Next, of the remaining spectral regions, isolated areas are picked as peaks. Finally, overlap is resolved by lineshape matching, the peaks are picked and their attributes measured. An optional additional step is filtering based on symmetry and linewidth [106, 102].

Correct peaks are important because they form the basis for the construction of GSSs, the assignment of chemical shifts to nuclei, and the interpretation of NOESY spectra which give rise to distance restraints as a preliminary to structure calculation [66]. Incorrect peak identification or position can result in misinterpretation of NOESY spectra, which could lead to false distance restraints between atoms which are in fact very far apart in the actual protein structure.

Estimates of the amount of false positive and false negative peaks picked by computational tools range from low (10-40%) to high (70-135%) [107]. The quality of the results generally depends on characteristics of the spectrum, especially the signal/noise ratio, and resolution, as well as characteristics of the molecule including T_2 (which has an effect on peak width) and number of nuclei – more nuclei give rise to more peaks, and therefore a higher chance of overlap.

Since none of these approaches yields perfect results [66], manual intervention during peak picking is important for obtaining results of sufficiently high quality [73]; many peak picking programs allow and encourage semi-automated interaction in order to clear up troublesome spectral features. Manual intervention is often accomplished based on knowledge outside of the spectrum: existence, position, and shape of peaks in other spectra, knowledge of the solvent, characteristic artifactual patterns caused by a specific processing scheme, knowledge of the local dynamics of a small region of the protein. [64, 73, 108, 65] Because of this, peak picking can often not be completely finished until later analysis has been accomplished.

GSS and resonance construction

As many pulse sequences are specificially designed to exploit the strong backbone H-N coupling and correlate additional nearby nuclei, H-N groups appear in many through-bond spectra and are given a privileged position in analysis: the signals which H-N groups give rise to appear at matching chemical shifts across multiple spectra. This H-N matching enables grouping of peaks into GSSs; given multiple spectra which include N-H dimensions, peaks with matching N and H chemical shifts are determined to belong to the same GSS. Table 3.12 shows that H-N chemical shifts are captured in multiple spectra, and sometimes multiple times within a single spectrum. At a later stage, GSSs are often augmented with additional sidechain resonances.

In addition to GSSs of backbone resonances, H-N-rooted GSSs typically are present for asparagine and glutamine sidechains, and smaller GSSs of tryptophan sidechains appear. Arginine sidechains may also give rise to an H-N-rooted GSS under certain experimental

conditions.

The difficulty in constructing these GSSs correctly and unambiguously stems from the issues inherent in NMR data. First, the success of the standard suite of experiments rooted in H-N – NHSQC, HNCO, HNCACB, etc. – depends on [93]:

1. good dispersion, i.e. no overlap, otherwise it is difficult to determine which peaks belong with which H-N-rooted spin system.

2. the H-N chemical shifts being nearly identical across all spectra. This may not be the case if there are variations in the sample or the temperature. The Bloch-Siegert shift and experimental error also can have an effect on chemical shift.

3. nuclei appearing at a single chemical shift. If there are multiple conformations or chemical heterogeneity [93], a nucleus may appear at multiple chemical shifts and appear to be two different resonances.

4. the presence of an H-N group – proline is a notable exception, and so it does not show up in experiments which rely on the presence of an H-N group

5. extraneous peaks which do not seem to fit into a spin system, or peaks which do not seem to match peaks in other spectra

6. accurate (or at least consistent) spectral referencing. Misreferenced spectra will cause the same nucleus to show up at different chemical shifts across multiple spectra.

7. quality of peak picking [93, 109]: chemical shifts, lineshapes, as well as the numbers of false positives, false negatives, extraneous peaks

Computational approaches for GSS construction tend to require manual assistance in some cases, such as AutoAssign and Mars [93, 109]. Incorrect or incomplete GSSs will have negative effects on the quality of later analysis; several assignment tools assume that manual intervention will verify and, if necessary, correct the GSSs [64]; this allows the tools to be conservative in their predictions [93]. However, it may not be possible to unambiguously and completely construct GSSs until the results of later analysis are available: some approaches use NOESY peaks and assignments as well as structure results to verify and correct GSSs [93].

Figure 4.3 shows an example of matching peaks between two spectra. The quality of the match – how closely the chemical shifts line up, as well as the lack of overlapping peaks – means the peaks are easily identified as members of the same GSS.

Resonance typing

Before assigning a resonance to a specific nucleus, the type of a resonance may be assigned. In an HNCO experiment, this is typically straightforward, because for each backbone spin system, the H dimension always corresponds to the backbone H, the N dimension always corresponds to the backbone N, and the C dimension always corresponds to the backbone C(i-1). However, the situation is more complicated in an HNCACB experiment, as there are generally four choices of type assignment for the C dimension: CA, CB, CA(i-1), and CB(i-1). Thus, the resonance given by the C dimension of each peak must be assigned one of these choices. Reasons for choosing a specific assignment include peak sign, as well as chemical shift compared to statistics available in the BMRB. In addition, the overlap between experiment pairs such as the HNCACB and CBCA(CO)NH facilitates resonance

typing: while the CA(i-1) and CB(i-1) are expected to appear in both experiments at the same chemical shift for a given backbone H-N root, the CA and CB are expected to appear only in the HNCACB spectrum.

GSS typing

Correspondingly, GSSs are also assigned amino acid types. This phase interacts strongly with the assignment of nuclei to resonances, in that the possible nuclei to which a resonance may be assigned depends on the amino acid type, and the expected chemical shift ranges for various types depends on amino acid type as well. For instance, GSSs assigned to the glycine amino acid type should not have a CB; and the CB resonance's chemical shift of a GSS assigned to alanine is expected to be very different from all other CB chemical shifts. Backbone amino acid types may be split into several categories [90] based on BMRB statistics for CA and CB chemical shifts [2]:

1. Ala
2. Gly
3. Pro
4. Ser, Thr
5. Val, Met, Lys, His, Arg, Glu, Gln, Trp, Cys
6. Asp, Asn, Ile, Leu, Phe, Tyr

However, GSS typings are complicated by several factors. First, GSS typing requires correct and complete GSS construction. Second, correctly assembled GSSs may include overlapped

or extraneous peaks, expected peaks (based on a spectrum's typical results) may also be missing. Third, most GSSs can not be uniquely typed based solely on CA and CB chemical shifts, as groups 5 and 6 (above) as well as 4 are ambiguous. Fourth, sidechain GSSs must be identified and separated.

Sequential GSS assignment

Sequential GSS assignments exploit the previously mentioned overlap of pulse sequences such as the HNCACB and HN(CA)CO. Sequential GSSs are expected to have CA/CA(i-1), CB/CB(i-1), and CO/CO(i-1) resonances at identical chemical shifts. This duplication enables sequential assignment of GSSs. There is substantial interaction between nucleus-resonance assignment and sequential GSS assignment: assigning two GSSs sequentially implies the CB vs CB(i-1), CA vs CA(i-1), and C vs C(i-1) type assignments of the resonances in both GSSs; knowing the resonance typings of two spin systems can prevent their sequential assignment (if, for example, the matching resonances are both CB(i-1)); and not knowing the resonance typing implies that the sequential GSS assignment may be invalid. Sequential GSS assignment is complicated by:

1. missing peaks, possibly caused by local dynamics, which reduce the number of overlapping resonances between potential sequential GSSs, and can also disrupt resonance typing

2. extraneous peaks, which may be false positives or caused by multiple conformations of the protein, causing incorrect matches

3. degeneracy of chemical shifts: given two GSSs with identical CA(i-1) and CB(i-1) resonances, as well as a third GSS with matching CA and CB resonances, it is

impossible to unambiguously assign sequentially solely on the basis of chemical shift matching between the two GSSs [93].

An example of GSS overlap is shown in Figure 4.4. Green peaks are CA, and purple peaks are CB; note that one each of green and purple peaks match between the two GSSs.

Sequence-specific GSS assignment

Since backbone GSSs are H-N-rooted, a GSS is assigned to a backbone-amide; this implies the assignment of resonances to nuclei as well, based on matching of resonance typing. When a typical GSS is assigned to a residue, the H, N, C, C(i-1), CA, CA(i-1), CB, and CB(i-1) nuclei will be assigned resonances as well. Sequence-specific assignment interacts heavily with sequential GSS assignment, because the protein sequence must be compatible with the GSS sequence, where 'compatible' means that the amino acid types of the GSS match those of the protein sequence. Note that full assignment of amino acid type to GSS is not a prerequisite for GSS-residue assignment; in fact, GSS-residue assignment may lead to GSS-amino acid type assignment for sequentially connected GSSs. GSS-residue assignment is facilitated by long chains of sequential GSSs in which some of the GSSs are typed as serine, threonine, glycine, or alanine. The longer a GSS chain, the fewer places it might possibly fit into the protein sequence [90]. Also, as sequence-specific assignment proceeds, the number of unassigned GSSs and residues decreases; the result is that initially ambiguous assignments become unambiguous as choices are removed. Conversely, complications arise from incomplete sequential GSS assignments resulting in short, ambiguous chains. The presence of prolines generally ends chains due to the lack of a backbone H-N group. Missing GSSs also terminate chains. Relatively few Ser, Thr, Gly, and Ala residues means the number

of unambiguous anchor points will be lower.

Figure 4.5 shows an example of sequence-specific GSS assignment. Although not all of the GSSs have been typed, the presence of a glycine and serine in the GSS chain reduces the possible assignments to residues. Additionally, the length of the GSS chain helps reduce the ambiguity compared to shorter GSS chains.

Sidechain: spin system and resonance assignment

The next group of experiments collects chemical shifts of sidechain atomic nuclei. These experiments include the HBHA(CO)NH [70] in Figure 3.17, the C(CO)NH-TOCSY [71] in Figure 3.19, the HC(CO)NH-TOCSY [110] in Figure 3.18, and the HCCH-TOCSY [111] in Figure 3.20. The purpose of these experiments is to obtain the chemical shift values of sidechain resonances of protons, since proton frequencies are necessary in order to interpret NOESY spectra. To interpret these spectra, the peaks must be assigned to GSSs and the new resonances typed. While several of these experiments are also rooted in backbone H-N groups, facilitating the addition of peaks to the correct GSS, others – such as the HCCH-TOCSY – are not. These are analyzed by the matching of resonance chemical shifts with those from other experiments targeting sidechains. Resonance typing can generally be made with reference to compiled BMRB statistics. Complications in this phase include: stereospecificity – nuclei such as HA2 and HA3 may give rise to different chemical shifts, but resolving the correspondence may be impossible without further data; overlap – especially in the HCCH-TOCSY where sidechains of the same amino acid type but different residue may have many closing matching chemical shifts; overlap between resonances within the same GSS, especially in Leu and Ile; missing and extraneous data; and the difficulty of both

obtaining and unambiguously interpreting aromatic data. New approaches for sidechain data collection and assignment have recently been developed [112, 113] which seek to address these issues by reducing ambiguity of chemical shifts.

Alternative approach: probabilistic assignment

The previously described approach views analysis as a pipeline: input is transformed into output, which becomes the input for the next stage, and so on. PINE [107] removes the pipeline constraint by connecting each stage to each other and allowing information to flow freely; this enables statistical weighting of interpretation as well as dependencies such as peak picking on GSS construction (a dependency which is not possible in the pipeline approach). PINE does not remove the need for manual intervention; it is still assumed that some level of intervention is necessary to obtain the best results [107].

4.4 Structure determination

In the final stage, Figure 4.1 panel D, the chemical shift assignments are used to interpret the NOESY experiments and a structure is calculated and refined.

NOESY peak picking and assignment

NOESY peaks provide structural restraints if it can be determined which protons gave rise to the peak. Analysis of NOESY spectra therefore requires chemical shift assignments of nuclei to determine the protons involved in a peak. NOESY spectra are processed and peak picked, similarly to through-bond spectra, and resonance assignments of peaks made. Considerations used to analyze NOESY spectra include: symmetry – a peak is expected to correspond to a

matching peak with the frequencies of the two ^{1}H dimensions swapped; patterns based on known proximity of nuclei from the primary sequence giving rise to many short-distance NOE peaks; and network anchoring. Complications include overlap caused by degenerate chemical shifts of protons, leading to ambiguous interpretations of peak assignments; this can be greatly mitigated by the use of an extra dimension: ^{15}N- or ^{13}C-edited NOESY spectra reduce the ambiguity, as well as incorrect or incomplete chemical shift assignments.

NOESY assignment may be done automatically by programs such as CYANA and ARIA [76, 77]. NOESY peak picking may be automated as well by programs such as MUNIN [103, 104].

An alternative approach is taken by ABACUS [72], which uses Monte Carlo probabilistic methods for assignment, NOE assignment, and structure calculation. A key difference of the ABACUS approach is the reduced dependency on the quality and analysis of through-bond experiments; through-bond experiments are used mainly to assemble peaks into GSSs, but sequential connectivities are obtained from NOESY experiments. A more detailed explanation may be found at `http://www.nmr2.buffalo.edu/nesg.wiki/Resonance_Assignment/Abacus/Introduction_to_ABACUS`. Importantly, input to ABACUS – correct NOESY peak picking and GSS construction – must be complete and accurate.

Structure calculation

The resonance assignments of the NOESY data are interpreted to obtain distance restraints, which are then used to calculate coarse-grained three-dimensional structures. The structures may then be refined and fine-tuned using a computational tool such as Assisted Model

Building with Energy Refinement (AMBER) [114]. Unambiguous resonance assignment of NOESY data purely on the basis of chemical shift assignments may often be impossible or impractical, due to degenerate chemical shifts and to non-stereospecific assignments. While these ambiguities can often be resolved through the collection of additional NMR data, the expense involved in doing so may often make it more practical to attempt to resolve the ambiguities through a structure determination program such as CYANA [76].

CYANA is able to calculate a three-dimensional structure from NOESY peaks, chemical shift assignments, and distance restraints [76, 77] using an iterative approach to NOESY peak assignment and building structural models. It also requires secondary structure information in the form of torsion angle restraints as input; as chemical shift values are correlated to secondary structure, as described by secondary chemical shift statistics [85], secondary structure can be calculated from chemical shift assignments using a program such as TALOS+ [78]. As of version 3.0, CYANA is able to make use of RDCs during structure calculation as well.

Chemical shift assignments may also be used to calculate potential structures. CS-ROSETTA [87] uses chemical shift assignments of nuclei in backbone GSSs and produces a set of structures. The general method is to compare the chemical shifts with those of proteins of known chemical shift and structure, and then to select structure fragments from those known proteins. By assembling these fragments into a complete structure, a full model is constructed. Although CS-ROSETTA is computationally intensive, it still offers a massive potential time savings by eliminating the need to collect and analyse NOESY spectra.

Additional programs may be used to build, manipulate, and refine structures. AMBER [114] is a set of force fields which facilitate simulation of molecular dynamics. The force

fields are parameterizable and describe potential energy; when applied to a molecule they provide a description of the molecule's potential energy. Major components of the force fields are contributed by atomic bonds, electron orbitals, bond torsions, van der Waals interactions, and electrostatics. XPLOR-NIH [81] is a powerful structural calculation and refinement program that is capable of incorporating torsion angle restraints, J coupling restraints, isotope effects, ^{13}C secondary shifts, proton chemical shift restraints, RDCS, and NOEs. These data are modeled by means of a set of energy terms which the program attempts to minimize. XPLOR-NIH is also capable of modeling explicit solvent molecules in order to determine their effect on the structure, although this approach is superceded by the potential energy term model to a certain extent [81].

4.5 Discussion

The inherent NMR issues of ambiguous, missing, and extraneous data cause problems throughout the entire analysis process. Correctly dealing with these issues is difficult, but absolutely critical in order to obtain high-quality results [64, 73, 108, 65]. As yet, computational tools are not able to deal perfectly with these issues, due to one or more of several basic limitations:

1. they require high-quality input in order to function correctly: SAGA [90], ABACUS [72], Mars [109], AutoAssign [94], EZ-ASSIGN [91], PINE [107], and CYANA [76]; this input is generally assumed to have been manually prepared in order to meet the stringent quality requirements of completeness and absence of extraneous results

2. even with high-quality input data, tools are not able to produce perfect results

3. tools perform differently in different contexts, although performance generally decreases as protein size increases and spectral quality decreases

4. manual verification and correction of the results is assumed, even for tools that claim to be fully automated [64, 73, 108, 65]

A key limitation of many analysis tools is the fixed input data. While this simplifies the use of the tool in a simple pipeline, it may also lead to reduced quality of results and explain the necessity of manual intervention: while the input data that a tool handles is restricted, manual interventions can make use of any additional information required to make specific deductions. Thus, PINE and related efforts are an exciting effort to loosen these incidental restrictions. Initial results are promising, and show a marked improvement, although manual intervention is still assumed to be necessary in order to obtain the best results [107]. Further tools such as SHIFTX2 and CHESHIRE [40, 86], which calculate chemical shifts from structure, bring additional information to bear, helping to validate assignments. Table 4.1 and Figure 4.2 show some of the key connections between various data types, putting these tools in the overall context of NMR data analysis.

Another exciting development is the rise of probabilistic methods [90, 107]. These methods reflect the reality that the confidence of a specific interpretation depends on the exact state of the data; in other words, an assignment which is 50% confident given only an HNCA spectrum may become 90% confident if an HN(CO)CA spectrum is added. The significance of this confidence level is that it enables easy tracking of ambiguous and/or low-confidence interpretations – i.e. those that stand to benefit from collecting additional data sets. By including confidence values on all assignments, an understanding of the troublesome areas is facilitated. This helps to reduce the cost of cascading errors – if the uncertainty is

tracked as a confidence level, further interpretations based on a highly uncertain datum will also receive low confidence levels. In addition, confidence levels are an alternative to the inherent balance between completeness and correctness – it is no longer necessary to sacrifice one for the other [94, 107].

4.6 Conclusions

The massive amount of data involved in a structure determination process – often on the order of gigabytes – necessitates the use of computational tools for data management as well as efficiency of analysis. To address specific problems in the NMR analysis process, many software implementations of useful data processing algorithms have been created, distributed, and maintained in recent years (`http://nmrbox.org/NMRbox.org/Registry.html`, `http://nmrwiki.org/wiki/index.php?title=Category:Software`, `http://bmrb.wisc.edu/tools/prog_corner.shtml`). Additionally, several groups have accelerated the process by producing software tools spanning and integrating multiple steps to decrease the necessity for time-consuming human intervention [72]. This allows automated or semi-automated structure determination for small proteins. Other groups have built integrated pipelines, using one specific tool for each step [65, 88], and allowing manual intervention at traditionally difficult stages. Many recent methods re-envision structure determination as an iterative process, where the results of a later stage may require the researcher to re-evaluate or re-perform an earlier stage [76]; this has been applied to interpretation of NOE-derived restraints [77]. Altogether, the structure determination process can often take several months [66].

In general, while computational tools are able to deliver results relatively quickly com-

pared to manual analysis, they may not be able to produce more accurate results, especially when the input data is low-quality, irregular, or otherwise problematic; this can result in false positives and negatives [64]. This is a problem at every stage of spectral analysis.

This has the consequence that NMR structure determination data analysis processes cannot be fully automated if high-quality results are required. An effective solution to this problem combines the strengths of the automated and manual approaches, in a semi-automated fashion: computational tools are used to quickly perform the majority of analyses such as peak picking and GSS construction, and manual analysis is used to clear up the relatively small number of cases involving ambiguities and errors caused by problematic or unclear data. Thus, some amount of manual analysis may be required at all stages of the data analysis process [73, 64].

Manual analysis therefore plays a critical role in NMR data analysis, due to the inherent issues of analysis which complicate automated tools, and to the ability to bring sufficient context to bear to solve difficult cases. Manual intervention is assumed to be necessary by most tools, even automated ones, to ensure the completeness and correctness of results. However, despite the importance that manual intervention plays in analysis, the specific modifications made and their reasons for – which may be quite complicated – are not captured [73]. Thus, the meta data of manual intervention is lost, and analysis is irreproducible. Figure 4.6 shows which data are and are not required for deposition by the BMRB, indicating data missing from final results that prevents reproducibility.

4.7 Tables

Data	Example
FID -> spectrum	NMRPipe
Spectrum -> peaks	Sparky peak picker
Peaks -> GSSs	Manual analysis
Shifts -> sequence	ShiftY
Shifts -> secondary structure	TALOS+
Shifts -> tertiary structure	CS-ROSETTA, CHESHIRE
Tertiary structure -> shifts	SPARTA+, ShiftX
Shifts -> NOESY assignments	CYANA
NOESY assignments -> structure	CYANA
NOEs, shifts, j-couplings -> structure	XPLOR-NIH

Table 4.1: Connections between various data types.

4.8 Figures

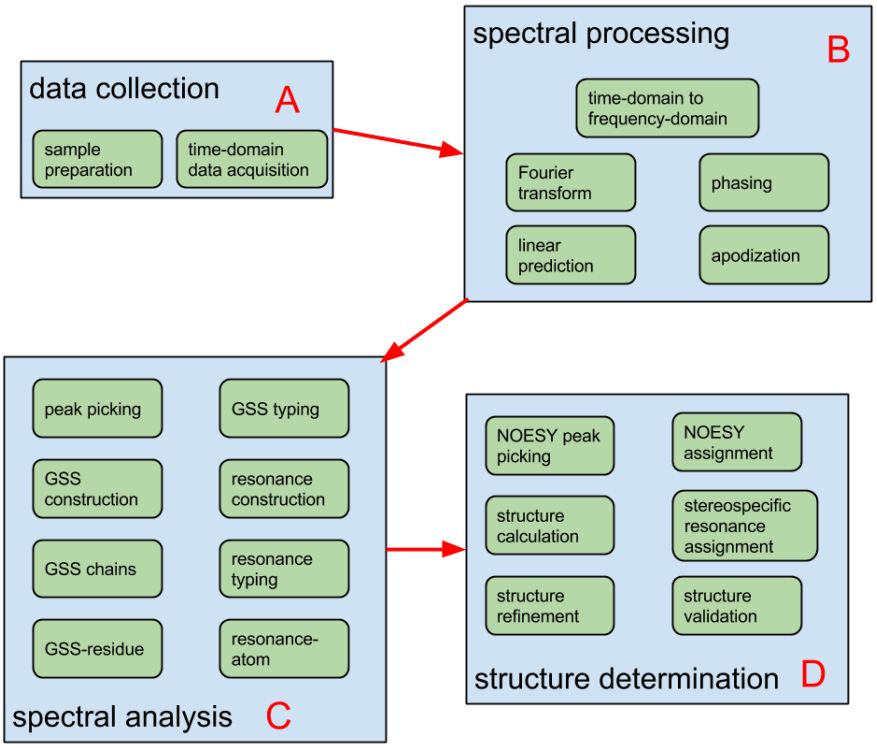

Figure 4.1: An overview of the NMR process for protein structure determination.

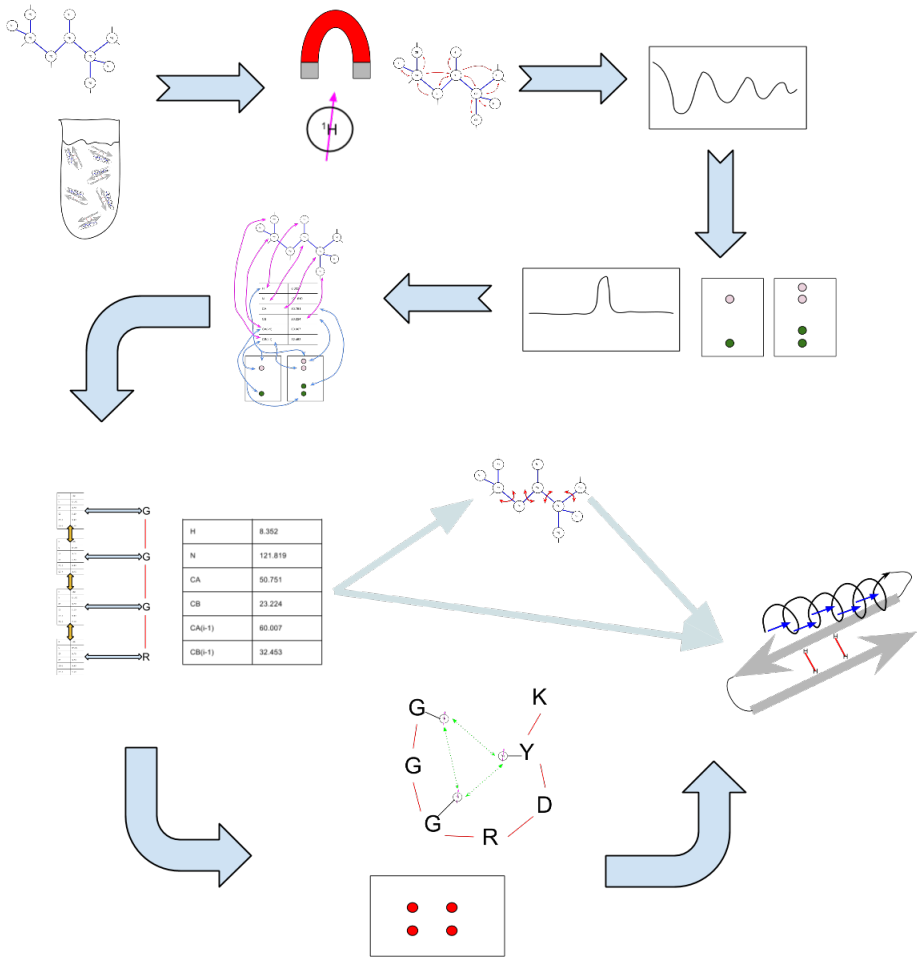

Figure 4.2: The connections between various data types.

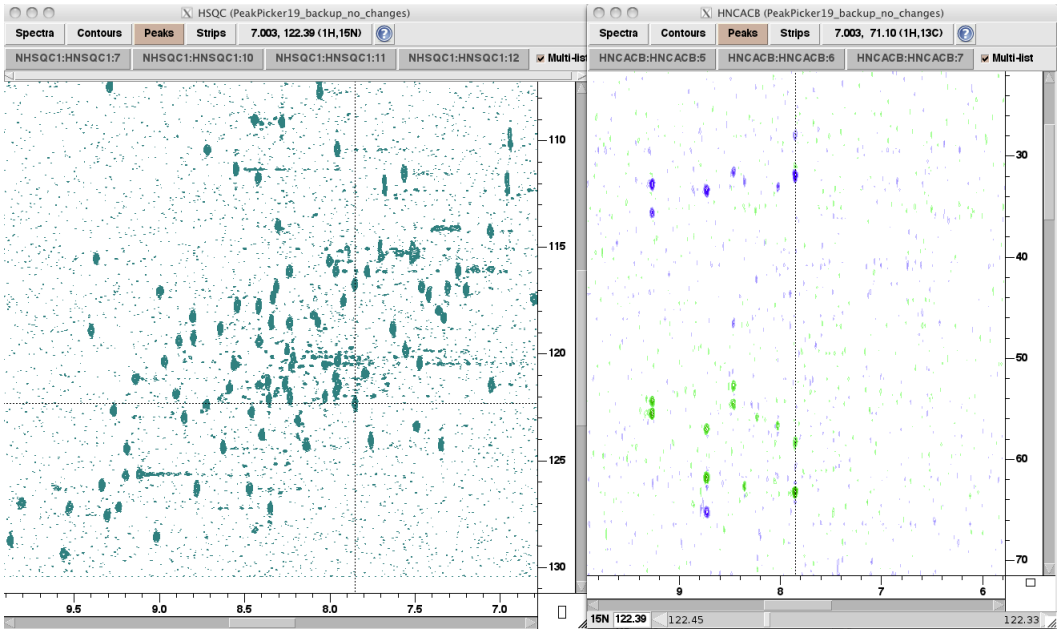

Figure 4.3: Matching peaks between an NHSQC and an HNCACB spectrum. This likely indicates that the peaks belong to the same GSS.

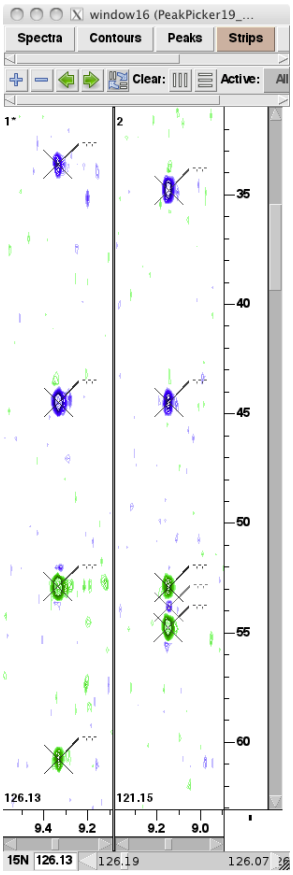

Figure 4.4: Overlap of carbon resonances in an HNCACB spectrum.

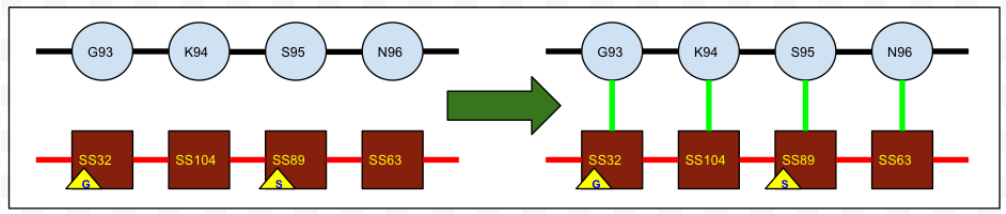

Figure 4.5: Assignment of a GSS chain to residues. The circles are residues, black lines are peptide bonds, squares are GSSs, red lines are sequential GSS assignments, and green lines are GSS-residue assignments.

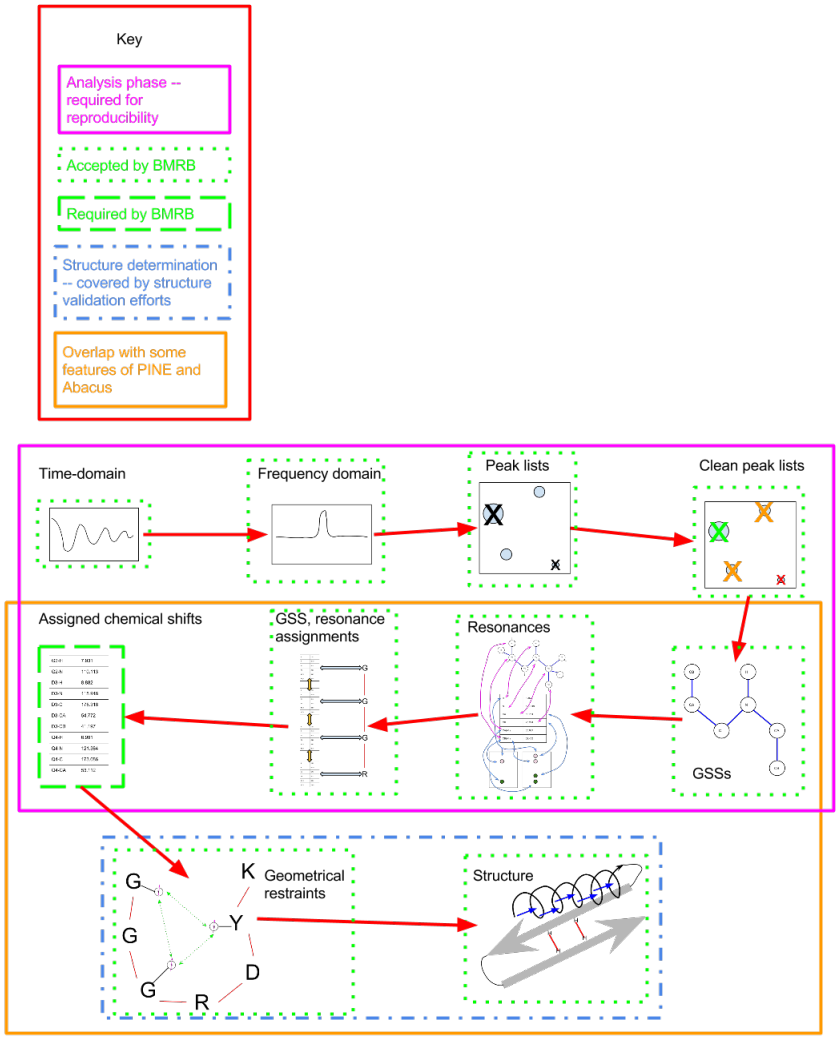

Figure 4.6: Breakdown of the NMR process by reproducibility. The BMRB currently requires deposition of some data types, and accepts others. However, these data are incomplete.

Chapter 5

An Approach for Reproducible Analysis

If you're doing an experiment, you should report everything that you think might make it invalid - not only what you think is right about it; other causes that could possibly explain your results; and things you thought of that you've eliminated by some other experiment, and how they worked - to make sure the other fellow can tell they have been eliminated.

- Richard Feynman

5.1 NMR data analysis is irreproducible

Successful achievement of a reproducible NMR study requires reproducibility at each stage of the process. First, the protocol for expressing, purifying, and preparing the sample of interest for experiments inside the NMR spectrometer must be reproducible, as well as the exact experimental conditions, spectrometer, pulse sequences and collection times

used to collect the time-domain data must be captured. Second, the software, platform, functions, and parameterizations for spectral processing stage must be captured. Third, both the computational results of peak picking, GSS construction, GSS assignment, and resonance assignment as well as any manual changes, along with the associated deductive process of reasoning, must be captured. Fourth, analysis and assignment of NOESY spectra, structure calculation, stereospecific resonance assignment and structure refinement must be captured. This last stage may also include computational as well as manual analysis components.

This work will focus on reproducibility of the third and fourth stages, spectral analysis and structure determination. Unfortunately, according to the definition of reproducible NMR given above, these stages are irreproducible.

Much work has been done to capture additional data from the assignment process. The CCPNMR effort, including the significant projects of CCPN Analysis and the CCPN data model [43], captures a significant portion of final data. Other significant efforts include SPINS [115], Sesame [116], and the NorthEast Structural Genomics consortium (NESG) [117]. However, much of the previous work in this area has focused on project management rather than reproducibility. While SPINS is effective at capturing intermediate and final results, it is only intended for use on primary data files – it is not designed to capture the key meta data mentioned above. Sesame is capable of project management; however, it is intended for use in high-throughput studies. ELNs [9], while effective at capturing experimental design and data, are not designed to support in-depth analysis, nor are they intended to interface with NMR tools.

While the need for integration of automation with occasional manual validation and editing has been recognized [115], no current systems are capable of combining these features

with full meta data capture. What is still missing is an approach and tooling for collecting all the primary data and meta data of NMR spectral analysis and structure determination. Much time and effort is expended in these stages, but the data is not recorded. The result is that the final data sets deposited into the BMRB are incomplete. NMR data analysis is irreproducible because insufficient information is captured, recorded, or disseminated during the analysis process. This chapter will outline the data involved in NMR, and then describe in more detail the missing data, its role and importance, then a model for capturing that data, and finally a strategy for using the model during the analysis process.

5.2 Concrete data sets

The data involved in the NMR process was covered abstractly in Chapter 3. During execution of the NMR process, these data are embodied concretely as structured files, read to and written from file and databases on local and remote hosts. Final data sets are deposited to the BMRB and are then shared as NMR-STAR files. Since files are used as the means of transfer and sharing of NMR information, it is necessary to define the meaning and extent of a data set in terms of the number, format, and content of the data files involved.

In the data collection stage, the extent of a data set includes the FIDs stored as structured binary files as well as the meta data describing various parameters of how the data was recorded and saved. The scope of a data set is expanded during spectral reconstruction to include binary spectral data, spectral meta data, and the sequence of functions used to convert the data from time-domain to frequency-domain, which often takes the form of a shell script file invoking processing tools.

During spectral analysis, a data set includes peak lists and chemical shift assignments.

Depending on which analysis tool one uses, these data are saved in various formats along with additional parameters and settings which describe the configuration and state of the tool itself (such as contour levels and peak ornament colors on a spectral display). When integrating analysis with tools such as TALOS+ and CS-ROSETTA [78, 87], specially formatted files may be temporarily generated to transmit information to those tools, such as chemical shift assignments in a specific format. The protein sequence is often stored as a textual file containing the protein's amino acids.

Once the structure determination stage is reached, the data set includes torsion angle restraints, structure files, lists of restraint violations, NOESY peak lists, RDC lists, and scripts for running and analyzing results from tools such as CYANA.

5.3 Missing data and its role in analysis

The key deficiencies causing irreproducibility are missing primary data and meta data which are modeled neither in the CCPN data model nor in the NMR-STAR data dictionary, and are not archived and disseminated. In other words, these data form an important piece of the analysis process, but are not captured as concrete data in the data set. Spectral analysis, including peak picking, GSS construction, GSS and resonance typing, sequential GSS assignment, and sequence-specific GSS assignment is accomplished using a step-by-step process of deductive reasoning which is often augmented by computational tools. The computational results may be subject to manual validation, correction, and extension [66]. This section will explore the various types of data involved.

Deductive process of reasoning

Manual modifications are performed using a process similar to deductive reasoning. It follows a general pattern:

1. identification: a feature of the data is identified as amenable to interpretation. For example, the feature may be a false negative (such as a signal peak misclassified as noise by the automated peak picker), a false positive (such as an artifactual peak misclassified as signal), or an ambiguity (such as overlapped GSSs, causing clustering algorithms to fail).

2. pattern recognition: the spectroscopist identifies a potential method for interpreting the feature based on his/her domain knowledge of NMR and experience with interpretation of previous data sets. For example, such methods may take the form of deductive rules: if <the data matches a certain pattern>, then <it could be interpreted a certain way>.

3. application of the rule to the data feature. The chosen rule is applied, and the result of the interpretation is included back into the data set. The result may now be used to drive further deductions.

4. repeat – go to step 1 to identify features for further interpretations This method is a form of iterative, sequential deduction. The key components are the ordered series of steps, the state of the data before and after each step, and the deductive rules used to make interpretations at each step. In addition, it should be noted that the final data set can not be regenerated using automated tools alone if there are any manual modifications made to tool output.

The information describing the deductive process of reasoning employed in manually interpreting a feature of the primary data is not captured, although it is important because it provides the explanation of why something was done.

Each deductive reason is based on NMR knowledge of how to interpret a data feature. In general, a deductive rule requires an input, produces an output, and has an intuitive justification for its action.

Application of these rules provides a rationale for manual modifications. A rationale justifies the correctness of a change and indicates why it was made. Therefore, capturing the deductive rules employed enables verification of the modifications. It also facilitates knowledge transfer, both in the contexts of collaboration and teaching, by providing a meaningful annotation, with reference to the domain of NMR, for actions taken.

Intermediate results: the data set is the deductive context

When the output of a computational tool, whether peak picking, GSS construction, or sequence-specific GSS assignment, is modified to correct mistakes, a discrepancy is introduced between the output of the tool given the input and a suitable parameterization, and the final data set. Thus, the final results are not the output of a single replicable step, but rather of a series of steps of refinement and modification. An alternative viewpoint is that a new data set is implicitly generated from the previous one during the analysis process after every modification, whether manual or automated. The application of a deductive rule to modify the data set is sensitive to the current state of the data set – in other words, the validity of the use of a given rule, as well as its effect, depends on the exact state of the data set. Therefore, it is important to record the data context when applying a deductive rule. The intermediate

results rectify the discrepancy between automated tool output and the final data set. However, this data is not recorded.

With respect to reproducibility, the importance of capturing intermediates is due to the dependence of deductions on context: without knowing the context, it is impossible to evaluate the correctness, confidence, and alternative interpretations of a deduction. In standard approaches to analysis, the contexts are not captured; they are implicit. By making the contexts explicit, it becomes possible not only to fully recapitulate the process of analysis, but also to employ error detection and correction strategies by analysis of deductions and their contexts. An interesting side-effect of capturing contexts is that analysis can be restarted in the middle, by selecting an appropriate context and applying a different deduction.

Furthermore, during manual analysis and modification of results, when the state of the data is continually being modified and improved, the analyses which may be made are dependent upon the data context. For example, assignment of a GSS to a residue may allow a further unambiguous assignment of a different GSS to a residue (an assignment which previously would have been ambiguous) by eliminating one of two assignment possibilities based on matching amino acid type. Implicit in the sequence of data sets are logical dependencies of derived data upon features of the previous data set: the context of each deduction is important, because the exact context determines what deductions may be made and the confidence level of each deduction.

Extraneous results

Standard approaches treat all peaks as true signal, with no provision for storing peaks determined to be processing artifacts or noise. Such spurious peaks are simply deleted and do

not show up in the final results. This is a problem because the fact that a peak was found, and later interpreted as noise is not present in the final data set. The same problem applies to GSSs that are found but can not be assigned to any residue of the sample of interest, or are believed to correspond to atomic nuclei of a contaminant. Such GSSs should be represented in the final data set.

During analysis, some portion of the positive results are not of direct interest to the final answer. Not only peaks, but also resonances and GSSs are included. The positive results include both false positives, caused by noise or artifacts, and true positives, caused by contaminants.

Although not of direct interest, such extraneous results play a role in the process of analysis because they provide part of the context of analysis. Changing the context affects which rules apply and what deductions are made. Therefore, as a part of the context, extraneous results matter during analysis. If incorrectly identified or left unrecognized, extraneous features can lead to incorrect peak picking, chemical shift assignments, GSSs, and GSS assignments.

A further benefit of capturing extraneous results is the ability to distinguish between identifying a data feature and interpreting it. In other words, peaks picked during peak picking are treated as positives, this is the "identification" phase; in the later "interpretation" phase, these peaks are separated into false positives and true positives. This allows rectification of the discrepancies between uncorrected computational results and the final, deposited results as well as marking potentially suspicious results for future perusal. Picking a peak, then interpreting it as extraneous and discarding it is typically not reported in final data sets, despite containing information about the spectrum. There is a balance between false positives

and false negatives [107]; false negatives are more undesirable [107, 90, 73], and capturing extraneous results helps to avoid this tradeoff: by reducing the cost of a false positive, tools are free to focus on avoiding false negatives.

The process of separating positives into false and true is prone to introducing bias; by keeping and reporting the initial results, such bias can be estimated. This is not possible if the extraneous results are not reported, and also allows the feature identification phase to proceed without bias, since error correction will be applied at a later stage. By providing additional context, it may be possible to estimate the quality of an analysis, where errors may be most likely found in the borderline cases; it will also help assigning confidence levels to datums by not. Additional quality measures enabled include the number of peaks found by the peak picker, the number of false positives, the number of peaks assigned to GSSs, and the number of GSSs assigned to residues. It may also be possible to estimate contamination, incompleteness and overcompleteness, overfitting, and consistency.

Notes: incompletions, uncertainties, ambiguities

Odd, ambiguous, abnormal, or otherwise unexpected situations occur during analysis [118]. As notes indicate the deficiencies and potential problems present in a data set, they are valuable to future scientists as they highlight a data set's flaws and how it can be improved.

Due to the difficulties inherent in data analysis, situations are reached in which the interpretation of a specific feature is problematic:

- uncertain or impossible. The evidence for a particular deduction is not solid.
- ambiguous. Multiple interpretations of a feature are consistent with the data and satisfy the constraints. It is not possible to choose between them.

- inconsistent. The data set is in an inconsistent state, or a deduction would leave it in an inconsistent state.

A simple example is non-stereospecific sidechain proton assignments: a residue such as a histidine or lysine which has two beta protons will often give rise to two resonances, one for each beta proton; however, without additional information, it is impossible to assign a resonance to a specific nucleus. A related example is the two delta and epsilon protons in phenylalanine and tyrosine aromatic sidechains; the two resonances, even if distinguishable, can not be uniquely assigned to nuclei. In both cases, the ambiguity is resolvable through the use of additional information; however, before that additional information is provided, it is useful to be able to store what is known – that there are two peaks, each of which corresponds to one nuclei, but exactly which is unsure – as an indication to future analysis that a problem has been identified but not yet solved.

Correctly identifying and characterizing peaks in the presence of significant amounts of overlap is difficult [66]. The number, position, and intensity of peaks become distorted by the overlap. In such a case, it may not initially be possible to fully and correctly resolve the overlap (although later information from additional spectra, such as a higher-dimensional spectrum in which the additional dimension removes the overlap, may resolve the problem); a note explaining that overlap is suspected and that the characterizations may be in error points this out.

Building unambiguous and complete sequential GSS assignments is complicated when multiple GSSs have the same or nearly the same chemical shift values for resonances which are or potentially may be assigned to CA, CB, CO, or the corresponding (i-1) nuclei. Leaving a note in the data set describing what the ambiguity is ensures that this information is not lost,

and is clearly marked for re-analysis when more data becomes available.

5.4 A model for reproducible NMR

A data model is a means of specifying the structure of information [119]. This information may be used as inputs and outputs for computational tools, or it may be archived and available for reference use. Data models are useful because they provide a formal specification of the structure, which enables unambiguous, correct, and automated use of data. Data models are abstract specifications; concrete implementations are used in programs

This section will cover a data model for reproducibility. Once a data model exists, it can be implemented as part of a software program that facilitates reproducible data analysis. The core of this data model is formed by the BMRB [2] and CCPN [43] data models. These models are then extended with several additional data types and properties in order to enable reproducibility.

Deductive reasoning

When a data feature is interpreted, a deductive rule is used to provide the result, given the input. In order to support the capture of this data, a model both for the application of a rule to a data set, and for the rules themselves, was created. The rules are modeled as an extensible library of commonly used deductive reasons. Modeling the rules as an enumerated library enables quick and easy use. During analysis, one or more rules are applied to make a deduction. This is shown in Figure 5.1.

Rule-based systems have previously been applied in computational fields [120, 121], including medical diagnostic, industrial fault detection, and e-mail spam filtering. Important

characteristics include the system's ability to support and describe probabilistic reasoning, learning and tutoring, future extensions, explanations of past analyses, and performance evaluations [120]. The content of the deductive rule library is based on established practices during data analysis [66, 68, 67, 69, 70, 102, 95, 44, 43]. It is presented in Appendix B.

Intermediate results

The general outline of the solution is a model of the process of data analysis, consisting of a sequence of snapshots of the data set, taken at carefully chosen moments during analysis, which show the full process of analysis by capturing all changes. Each snapshot after the first contains a link to the previous snapshot, as well as a set of data differences. The differences between snapshots explicitly show how the analysis changes over time. Associated with each snapshot is a small amount of meta data to help describe it, including a timestamp, author information, and a deductive annotation.

The strategy is based on that used by Version Control System (VCS) software tools [122, 123], which are commonly applied for managing source code of software projects [124, 125, 126]. These tools were originally implemented in order to manage the change in source code over time, while retaining the ability to easily inspect past states of the code. It was found that application of such tools led to large increases in productivity, robustness, correctness, and reduced faults [127].

The goal is to effectively capture intermediate NMR data sets, such that the process of analysis is clear and understandable. Capturing meaningful intermediate snapshots is challenging; it is not sufficient to capture them indiscriminately. If snapshots are captured too infrequently, the situtation is not significantly different from current practices: the analysis

process will not be reproducible. If too many snapshots are captured, reconstructing the logical dependencies will not be possible; in addition, the valuable information may be difficult to identify compared to the large amounts of useless information. A third potential problem is collecting snapshots indiscriminately, such that they do not correspond to the actual process employed. This, too, prevents later use of the intermediate data because the process has not been correctly captured.

Therefore, there are several principles of intermediate data collection which must be observed in order to create a useful data set. These principles help to ensure that snapshots are created neither too often nor too rarely, and that they are useful for future perusal:

- time. Snapshots should be taken often enough that all modifications are captured. For example, when peaks are initially picked by an automated tool, and then modified (perhaps sorting them into signal, noise, and artifact classifications) by manual adjustment, a snapshot must be taken immediately after the automated peak picker is run, and before any modifications are made. When additional modifications are made, it is again necessary to take another snapshot before these changes, in order to capture the previous state of the data set – which is otherwise lost if this is not done.

- content. Each snapshot should have a clear and simple focus on analyzing a single feature or performing a single type of interpretation. For example, a snapshot should not include changes both to resonance typing and to peak lists if those changes are not inter-dependent.

- cohesion. Similarly to the previous point, changes which are inter-dependent belong in the same snapshot. For example, when assigning GSSs sequentially, assume there

are two potentially matching GSSs based on CA and CB resonances, which however have not been specifically assigned i/i-1. If one GSS is determined to be the first, and the other the second, then the CA(i), CA(i-1), CB(i), and CB(i-1) assignments of the resonances in both spin systems are determined. These changes all naturally belong in a single snapshot, since they are logically co-dependent.

- logical dependencies must be recoverable. The previous two points enable recovery of deductive, logical dependencies. The sequential process of deduction is the core of manual analysis, and therefore it is important to capture it clearly. This means that the dependencies must be reconstructable from the sequence of intermediate data sets.

The goal of capturing intermediate data sets is to facilitate reproducibility by modeling and saving the process of analysis. A system which does so by capturing a sequence of snapshots of the data set at intermediate timepoints meets the requirements for reproducibility. First, such a system is able to correctly recapitulate the changes over time due to manual and automated analysis. Second, by capturing the full context of each modification, the logical and temporal dependencies between various features of the data set are trackable.

Extraneous results

Our approach is to allow any number of peaks and GSSs, and to augment them with additional data fields which distinguish between signal, noise, contaminants, etc. This allows one to make a critical distinction between: 1) finding/recording a peak based purely on characteristics of the spectrum such as volume, height, relative height compared to noise, lineshape, and linewidth, and 2) interpreting a peak as signal, noise, etc. (and the same for GSSs). Even peaks and GSSs for which no analysis is made can be kept in the data set without encumbering

assignment of true peaks and GSSs.

To model extraneous results, the BMRB and CCPN models [2, 43] were extended to support additional fields which distinguished between extraneous and primary data. This applies to peaks, resonances, and GSSs. When using this model, one never directly deletes a peak, resonance, or GSS, but rather must mark it as extraneous by modifying its associated category from 'signal' to 'artifact', 'noise', or 'contaminant'.

For example, while using an interactive spectral analysis tool such as Sparky or CCPN Analysis [44, 43] for peak picking a spectrum, it is common to run the automated, built-in peak picker and then to manually correct the results by deleting some peaks and adding new ones. The reproducible approach works differently; peaks are not deleted. If the category of each of the peaks initially picked by the automated tool is 'signal', then the user must correct all of the categories for peaks determined to be signal or noise; note that these peaks are not deleted from the list. They remain in the list but with a different category tag that differentiates them from signal peaks.

A further category of extraneous data is peaks from amino acid sidechains; it is presented in Tables 3.5, 3.6, 3.7, 3.8, 3.9, and 3.10. While most of the peaks in these experiments correspond to backbone covalently bound groups, and are used for backbone sequential assignments, many sidechain GSSs are visible as well. Although these peaks are often ignored, they do contain useful information. They also can confound analysis if they are not properly recognized as sidechain peaks. Lastly, their presence is surprising to newcomers to the analysis process, as they are typically not explicitly recognized as part of the standard experiments.

Notes

The key idea is, given the inevitability of such problems during data analysis, to create facilities for explicitly recognizing, discussing, and handling such problems [128]. Several strategies for such an approach are covered in [118] including deferral of the problem while flagging it for later follow-up.

To model notes, the BMRB and CCPN models [2, 43] were extended to support an additional data type: a note. A note can refer to one or more other feature of the data set, and also includes a textual description of the nature of the problem, as well as an indication of how the problem might be resolved (although that is optional). As the purpose of a note is to explicitly indicate known deficiencies, incompletions, or uncertainties in the data set, wherever and whatever they may be, this approach is able to do so.

Enabling the representation of such data has similarities to the probabilistic approach applied by PINE [107] to great effect. PINE deals with the innate uncertainty of data analysis by resolving the tradeoff between false positives and false negatives through association of a probabilistic confidence metric with each feature interpretation; low confidence values are used as evidence that an interpretation is suspicious and needs additional verification or data. In a complementary approach, capturing notes of analysis issues also resolves the tradeoff for manual analysis, by enabling the association of an explanatory or warning message with suspected low-quality deductions. In addition, the message may contain more information than a scalar: it may necessarily refer to multiple conflicting pieces of the data set in the case of a contradiction.

5.5 An implementation of the model

A software implementation of this model was created as an extension to the popular assignment program Sparky [44]. The design, implementation, and use of this program is covered in detail in Section 7.

5.6 Applying reproducible analysis: using the model

This section presents some general advice for how to use the reproducibility model effectively in practice. It provides tips and suggestions, as well as covering common problems and how they can be avoided.

- one snapshot, one focus. Keeping each snapshot focused on dealing with a single issue helps the process of analysis to remain understandable. This is because it makes the logical dependencies more obvious; when a single snapshot contains many unrelated things, or is extensive enough that part of the snapshot depends on other parts, then it is no longer clear what the logical relationships are. Keeping snapshots small and focused alleviates this issue.

- level of detail. It is not necessary to exhaustively annotate every last single change; clearly, such an approach would be problematic because it would require far too much time and effort on the part of its users. Rather, the value of this reproducible approach is to clearly indicate major issues and modifications. The more important and the more time and brain power went into making a deduction, the more annotation it typically deserves. In other words, a complicated deduction requires a complicated justification. On the other hand, if multiple peaks are quickly and straightforwardly identified as

artifactual with a minimum of effort, a small annotation is needed; the deduction does not become clearer with additional annotation.

- apply the correct rule(s).

- record uncertainty and resolution. When in doubt or difficulty during analysis, record all information pertaining to the issues, whether as a note or extraneous data. Even if the problem is easily or quickly solved, describing it creates a record of that problem which is valuable for later inspection. Trends over such a record help to indicate more large-scale problems, as well as illuminating troublesome spots for collaborators and learners.

5.7 Future directions

The deductive library is not complete as it does not include deductions for every single possible analysis or interpretation which may be performed on NMR data. Examples include residual dipolar couplings, hydrogen bonds, and pseudocontact shift restraints. However, the library is extensible: it can be easily augmented with new deductive reasons. This is important because even if the library were complete today, it would likely still need to be extended in the near future to deal with new types of analysis and new data types. Thus, the strength of the approach that has been presented here, and its corresponding model, is that the approach is orthogonal to the specific datatype under analysis, which allows the library to be extended as necessary.

A further source of incompleteness is that of tools such as SPARTA+, CHESHIRE, Shifty, CS-ROSETTA, and SHIFTX2 [129, 86, 40, 87, 130]. These tools enable new interconnections

between various stages of the analysis process (see Table 4.1 and Figure 4.2), creating possibilities for skipping steps or going backwards in the process in order to verify that results are consistent with expectations. Effective use of these tools employs several useful deductive rules, similarly to manual analysis. Again, as the deductive library is extensible, it would be straightforward to add deductive reasons for incorporating the results of these tools into the analysis process.

5.8 Discussion

By collecting reproducible data sets, the true information content used in NMR spectroscopy is made explicit and visible. This is analogous to how lab notebooks are intended to be used in wet-lab work: as a means of recording the crucial details describing how an experiment was done, so that the procedure can be shared with and improved upon by others. A key difference, however, is that while lab notebooks have been in use for several centuries, the culture of reproducibility of digital analysis is still in its infancy: we do not yet have much experience with the what, how, and why of reproducibility in electronic media.

The first step is to define the lost data and a model for it. Then the model must be applied in practice, and its correct use taught. By extending standard existing models, the barrier to entry is greatly reduced, and instead of requiring an abrupt and drastic change in the workflows of those already using the standard BMRB and CCPN models [2, 43], the change to reproducible analysis can be incremental and gradual. This should help adoption.

Not only will such data sets make the process explicit, they will also help make biases explicit. It is possible that different research groups and different analysis techniques have different innate biases; it is quite likely that such biases will become obvious through the

collection of these full data sets. Each bias will represent an opportunity for learning and for improving the quality of analysis.

A natural question to ask of the approach is whether it is able to deal with the various strategies employed in practice by NMR spectroscopists around the world. Is the library of deductive reasoning sufficient for all use cases? While it is not possible to prove that the approach is universal, it is not necessary to do so. Rather, it is important to identify the principles of NMR data analysis, and embody them into the library. Furthermore, this risk of non-universality has been mitigated by the extensibility of the library: although as much was included as was reasonably possible to identify, if a deficiency is found in the library, it can easily be rectified by extending it with an additional deductive reason.

5.9 Figures

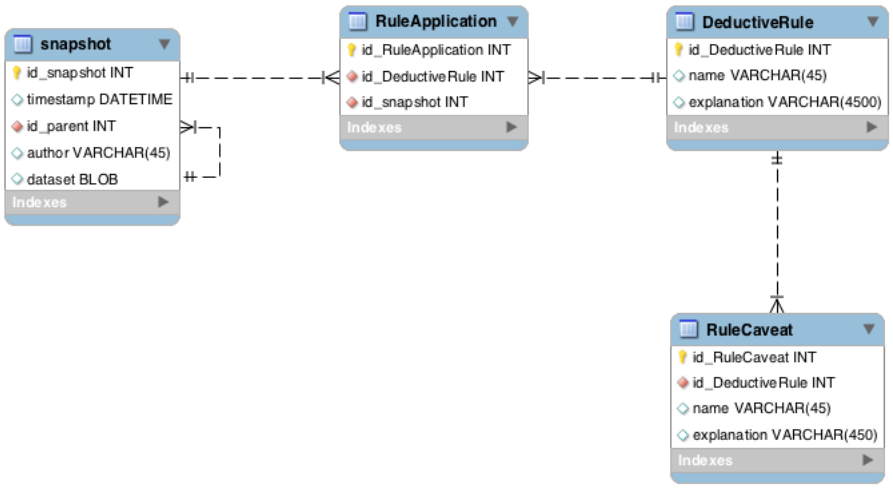

Figure 5.1: A relational model of a snapshot and deductions.

Chapter 6

Reproducible NMR Data Set

Our responsibility is to do what we can,

learn what we can, improve the solutions, and pass them on.

- Richard Feynman

In order to prove the validity of the approach described in the previous chapter, it was applied to typical NMR data in order to solve a protein structure. The process was carried out in full, and sufficient data was captured during the process in order to render it reproducible. The data set was deposited to the BMRB with id 25258.

6.1 Methods and materials

Time-domain data of Samp3, a Ubiquitin-like protein, were kindly provided by Dr. Mark Maciejewski. The data were processed to frequency-domain spectra using NMRPipe, and then analyzed first in CCPN Analysis [43] and later in Sparky [44]. A preliminary analysis process used CCPN Analysis, along with git and several Python utilities to create and manage

the reproducible history. The final analysis used Sparky, the model and approach described in the previous chapter, and the extension described in the following chapter. CYANA was used to assign NOEs, disambiguate stereospecific assignments, and calculate structure bundles.

6.2 The analysis process

Starting from time-domain data sets, I carried out the standard data analysis process using a subset of the CCPN data model [43], with the additional property that the process itself was captured as a series of annotated snapshots, and extraneous results were recorded as well. The full data set including complete annotated snapshot history may be found at `https://github.com/connjur/samp3` (a previous version using the CCPN Analysis program may additionally be found at `https://github.com/connjur/PeakPicker`). A summary of the process is described here.

Initial peak picking and GSS construction

First, the NHSQC and HNCO spectra were peak picked using an automated peak picker; see Table 6.3 for a complete list of spectra used. Then GSSs were built based on NHSQC peaks and matching HNCO peaks according to the H and N dimensions. Peaks either in the NHSQC or HNCO which did not match peaks in the other spectra were verified manually, and many were labeled as extraneous and not included in GSSs.

Several sidechain spin systems, including asparagine, glutamine, tryptophan, and arginine, were identified based on either peaks with matching N and C chemical shifts but different H shifts, spectral position, or peak intensity combined with splitting. An example of an asparagine sidechain is shown in Figure 6.1. Several factors allow these peaks to be grouped

into a single GSS, and that GSS to be typed as an asparagine:

- two peaks in the NHSQC with matching N frequencies, but different H frequencies, in the top right of the spectrum. These are caused by the two non-equivalent protons on the sidechain N.
- two additional, less intense peaks in the NHSQC matching the two H frequencies but at a lower PPM in the N dimension. These are caused by a small fraction of the sample which has one D and one H.
- two peaks in the HNCACB matching each of the D/H peaks in the NHSQC, and matching each other as well. These are caused by the CB and CA; however, the signs of the peaks are swapped from what they should be for a backbone GSS, since the CB is now closer to the H-N group than the CA.
- characteristic chemical shifts which match a CB/CA of asparagine, and do not match a CG/CB of glutamine.

GSS and resonance typing

Analysis continued with peak picking of the HNCACB and C(CO)NH-TOCSY spectra. The inclusion of the H and N dimensions allowed these peaks to be added into GSSs based on the matching of these two chemical shift values to the root peaks of existing GSSs. There was some H/N overlap which complicated the task of unambiguous peak-GSS assignment, but it was mostly resolved through the extra carbon dimensions, which allowed peaks to be identified separately and with more precise chemical shift values.

The overlap between the two spectra – peaks of sidechain i-1 carbons, relative to a back-

bone amide group, appeared in both spectra, while only the i peaks appeared in the HNCACB – allowed for assignment of resonance type of HNCACB peaks and some C(CO)NH-TOCSY peaks. Additionally, for GSSs with more than two i-1 carbon nuclei, the C(CO)NH-TOCSY resonance typings were simplified by the removal of CA and CB assignment possibilities due to overlap with the HNCACB.

The carbon chemical shifts provided sufficient information for GSS typing in many cases. The number of peaks in C(CO)NH-TOCSY strips also provided information about the typing of the previous GSS, the use of which is described in the following section.

Sequential and residue-specific GSS assignment

GSSs were built into sequential chains based on overlap of CA(i) and CB(i) peaks of one GSS with CA(i-1) and CB(i-1) peaks of another. It was critical that the overlap be unambiguous; ambiguous overlaps were deferred until later when additional information would provide disambiguation. Simultaneously, chains were assigned to residue sequences based on GSS typing and chain order matching the residue typing and chain order. GSS chains were often anchored on a glycine, serine, threonine, or alanine residue due to their unique carbon chemical shifts.

The whole process was heavily interdependent for several reasons. First, by assigning a GSS chain to specific residues, the typing for the GSSs both immediately preceding, and immediately following the first and last GSSs, respectively, of the chain were implied; this narrowed the search for further GSSs to those either with the appropriate typing, or an unassigned typing but with chemical shifts that matched the average BMRB statistics sufficiently well. It was also necessary that these GSSs were not already assigned as part of a GSS chain

because ambiguous assignments were disqualified. Second, assigning a GSS chain to residues reduced the number of unassigned residues, and therefore the number of possibilities for other unassigned chains as well. Several times, assigning a chain led immediately to the assignment of another chain by the process of elimination. Third, assigning GSSs sequentially reduced the number of sites in the sequence that could potentially accomodate the chain, both because of the chain-ending constraints imposed by proline residues as well as the matching of GSS typing to residue typing.

Sidechain: GSS augmentation and resonance typing

The C(CO)NH-TOCSY provided information on the chemical shifts of aliphatic sidechain carbons, as covered in the previous section. The HBHA(CO)NH, HC(CO)NH-TOCSY, and HCCH-TOCSY provided the chemical shifts of aliphatic protons, and corroborated the aliphatic carbon assignments.

First, these spectra were peak picked by an automated peak picker. These initial peak picks were corrected in a coarse validation step, and the peaks assembled into GSSs based on matching chemical shifts. For the HC(CO)NH-TOCSY and HBHA(CO)NH spectra, this was based on matching of backbone H and N resonances and was mostly unambiguous. The HCCH-TOCSY, lacking these resonances, was based on matching to sidechain H and C resonances, and was slightly more difficult to interpret due to ambiguities. Average BMRB statistics as well as peak patterns based on prochiral methylene groups were used for resonance typing.

Aromatic assignment

The two final through-bond experiments, the hbCBcgcdHD and hbCBcgcdceHE, were used to find the HD and HE resonances of aromatic sidechains and place them into the correct GSS. The spectra were again peak picked with an automated peak picker, and then assigned by matching the C dimension with the previously analyzed CB chemical shifts: the GSS with a matching CB(i) resonance was the appropriate one.

NOE assignment and structure calculation

The four NOESY spectra were peak picked with an automated peak picker. TALOS+ [78] was run on the chemical shift assignments of the H, HA, CA, CB, CO, and N nuclei in order to generate Phi and Psi torsion angle constraints for backbone conformation. The TALOS+ results can be seen in Figure 6.2.

CYANA was then run by providing it the NOESY peak lists, chemical shift assignments, torsion angle constraints, and Samp3 sequence. The script used for running CYANA is shown here:

```
# NOESY peak lists in XEASY format
peaks       := aromatic_gnoesy_chsqc.pks,aromatic_noesy.pks,
               D2O_noesy_chsqc.pks,noesy_nhsqc.pks
# names of chemical shift lists
prot        := shifts.txt
# additional (non-NOE) restraints
restraints  := aco.aco
# shift tolerances: H, H', C/N', C/N
```

```
tolerance   := 0.04, 0.03, 0.45
# number of initial, final structures
structures  := 100,20
# number of torsion angle dynamics steps
steps       := 10000
# random number generator seed
randomseed  := 434726

nproc=8

noeassign peaks=$peaks prot=$prot autoaco
```

After running CYANA once, the output indicated issues with stereospecificity of certain assignments (methylene protons, as well as leucine and valine CG groups). These assignments were corrected according to CYANA's feedback, and CYANA re-run with the corrected stereospecific assignments. The final structure is shown in Figure 6.3 as seen in Jmol [131].

6.3 NMR-STAR deposition to the BMRB

The standard means for sharing NMR-derived data is the BMRB [2], which uses the NMR-STAR file format. In order to enable archival of reproducible data sets, we have collaborated with the BMRB to extend the NMR-STAR data dictionary, so that reproducible data sets may be collected and deposited in the NMR-STAR format. The NMR-STAR data dictionary, which may be found at `http://www.bmrb.wisc.edu/dictionary/`, catalogs the names, structure, intended use, and definitions of the data types handled by the BMRB.

The git repository may be extracted and converted to a single NMR-STAR file; code for doing this may be found at `https://raw.githubusercontent.com/CONNJUR/Samp3-extractor/master/samp3_extractor.sh`:

```
filepath="sparky_data.json"
myids=( $(git log --format="%H" $filepath) )


for (( i=0,j=1; i < ${#myids[@]}; i++,j++ ))
    do
        name="temp/a$j.txt"
        sha=${myids[$i]}
        git show $sha:$filepath > $name
    done
```

This is a Bash shell script which uses the git commands "log" and "show" to first locate identifiers for each version of the JSON-formatted dump file, then to read the contents of those versions from the git database, producing new files, which are stored in the "temp/" directory. A Python program is then used to load the file contents into memory, and perform a semantic diff algorithm which recognizes differences between file versions. The final output is in the extended NMR-STAR format.

The BMRB accession ID is 25258, and includes the full Sparky analysis, TALOS+ and CYANA input and output, time and frequency domain data, and a git repository of the analysis including annotated snapshots.

6.4 Assessing the difficulties encountered

NMR data analysis must deal with the inherent issues of false positives, false negatives, ambiguity, and extraneous data. Presented here is a quantification of these problems which were encountered during several phases of analysis.

NHSQC peak pick

The NHSQC peak pick is a critical component of analysis because it is used to find the GSS roots, off of which may be based the restricted peak picks of the three-dimensional triple-resonance experiments. Table 6.1 gives an overview of the results. The precision of the peak pick was 99 / 105 = 94.3%; the 6 false positives were likely caused by sinc wiggles from an extremely intense peak, as they matched the ^{1}H chemical shift of a signal peak; no matching signals were found in the three-dimensional spectra. The recall was 99 / 118 = 83.9%, and the F1-score was 2 * precision * recall / (precision + recall) = 88.8%. See Figure 6.4 for an example of a false negative.

Completeness of detected NMR signals in NHSQC

Whether a signal is detectable in NMR experiments is not always predictable directly from the primary sequence: it may depend on the structure and dynamics of the molecule. Table 6.2 presents the numbers of expected signals based solely of off the primary sequence, and of found signals, based off of the chemical shift assignments. An extraneous GSS was found that could not be fit into the sequential assignments (See Figure 6.5).

Overlap in the NHSQC

Overlap presents difficulties during both sequential and NOESY assignment because individual signals are not identifiable and assignable. In the NHSQC, 2 signals from two different aspartate residues were heavily overlapped; although it was clear from the NHSQC that there were two separate peaks, it was not possible to determine their centers; furthermore, the correlated CO as well as CA/CB (8 separate resonances were expected, but only 2 were detected) resonances in the three-dimensional experiments were all overlapped as well. These GSSs appear to come from a group of 4 consecutive Aspartate residues.

A second region of NHSQC overlapped was caused by 5 peaks from 3 separate GSSs: one backbone, one arginine sidechain contributing 2 signals, and one asparagine sidechain contributing 2 signals. The signals essentially were not distinguishable based on the NHSQC alone, and were initially believed to be a single GSS, but additional data from the HNCO, HNCACB, and CCONH yielded more signals than could be expected from a single GSS and enable resolution of the overlap.

Quality of backbone GSSs

The completeness and unambiguity of the assigned backbone GSSs were estimated based on the presence of all expected resonances and the uniqueness of the chemical shifts of resonances within a GSS; the results are presented in Table 6.4. This gives an indication of the difficulty of uniquely assigning sequential connectivities, because every instance of CA(i/i-1) overlap is initially indistinguishable from two other cases: 1) a missing resonance, and 2) which of the resonances is the CA and which is the CA(i-1).

Ambiguity of CA and CB chemical shift assignments

The amount of ambiguity in the final chemical shift assignments was quantified by means of a 0.2 PPM threshold for CA and CB chemical shift assignments: given any two residues for which the CA/CB chemical shifts were assigned (CA only for glycine), check whether both corresponding shift pairs are withing 0.2 PPM. The significance of this is that CA and CB chemical shifts are used to build sequential connectivities in some assignment strategies, and ambiguous assignments confound its simplicity – ambiguities usually have to be resolved with reference to the primary sequence or to data from additional spectra.

Six ambiguous groups were found based on CA and CB chemical shifts:

- Glycines: residues 2, 3, 30, 94, and 105
- Glycines: residues 57, 60, 66, 78, and 106
- Aspartates: residues 8, 9, 10, and 11
- Arginine20 and Glutamine31
- Arginine26 and Glutamate27
- Proline102 and Valine103

The ambiguities from the first three groups were resolved by sequential connectivities and the primary sequence; the last three were resolved using additional data from the CCONH-TOCSY which provided GSS typing by means of the additional ^{13}C aliphatic peaks. Note that in the last case, the CA and CB of Proline102 are only visible through the N-H group of the following Valine.

6.5 Alternative implementations

While the approach taken was to generate a single NMR-STAR file from the git repo for reasons of compatibility with existing programs and data archival and retrieval facilities, there are other solutions to this problem. This section will inspect the strengths and weaknesses of an alternative solution which was taken during preliminary stages of this project.

The approach centered around using the open source VCS tool git [124]. A VCS tool enables the history of filesystem trees, typically source code trees of software projects, to be captured in a series of successive snapshots. Git, as a distributed VCS, includes a more robust model of snapshot history, which was critical in the success of its application to NMR data analysis.

Using git, multiple snapshots of a JavaScript Object Notation (JSON) or NMR-STAR flat text file were taken during the analysis process. Each snapshot was annotated with a JSON formatted string which provided the deductive reasoning behind the change, as well as a timestamp, author, and parent commit. This approach was initially employed due to git's intended use aligning very well with the domain problem.

The solution was quite easy, flexible, and robust in practice. Git is a mature, popular tool and therefore it was easy to learn to use, and contains many features for creating, tracking and querying the history. Here is a code snippet that shows how to create and annotate a new snapshot using git; all commands are run from a standard shell:

```
# prepare two files for snapshotting
#    peaks and parameters
$ git add nhsqc_peaks.str peak_picker_parameters.str
```

```
# create a snapshot, and annotate with a structured reason
$ git commit -m "{\"reason":
                  "automated Sparky peak picker\"}"

# display the history of commits in the project
$ git log
```

The git approach was the inspiration for the solution eventually employed. There is an inherent tradeoff between the two approaches: in the git approach, the full data set is spread across multiple snapshots of files, and the meta data is also separate from the primary data. In the NMR-STAR based approach, all the data, primary and meta, is in a single file. While the git approach makes it easier to query a single snapshot in time, it is more difficult to query across multiple snapshots, or meta data. The single file approach makes it more difficult to query a single snapshot (other than the most recent one), but easier to query across the complete data set including meta data. These differences are merely incidental, and not fundamental, as the same data is represented and stored in both approaches.

Git includes more features that were not used in our implementation. One such feature is branches, which allow data analysis to fork and rejoin. This could potentially be useful for interpreting ambiguous or unclear features in multiple ways, because it would allow an explicit record of the choice point. It could also be useful for error correction, to show the cascading effect of an incorrect analysis early in the process.

6.6 Discussion

Applying the reproducibility approach proves that it is viable. It can be used to effectively analyze NMR data. The approach I have applied is not the only one which could possibly be employed; rather, it is one of several. The principles embodied in the approach are to make data explicit – whether that data is the context of a deduction, or the deduction itself, or the motivation behind a deduction – and to capture that data efficiently. In this respect, it is similar to recording the process of analysis in a lab notebook. The overal proof of the effectiveness of the approach is that the relevant information is explicitly captured and available for later inspection. This ground-breaking work also shows how to use the approach, including possible pitfalls; the final, reproducible data set is valuable in that it shows what can be achieved through a reproducible approach, as well as creating a platform for future work towards enhanced reproducibility.

6.7 Tables

	Not signal	Signal	Total
Not picked	–	19	19
Picked	6	99	105
Total	6	118	124

Table 6.1: Quality of the NHSQC automated peak pick. The 6 peaks which were determined to not be signals all appeared to be sinc wiggles off of an extremely intense backbone glycine peak. Of the 19 unpicked signals, many were low-intensity.

	Expected	Found	Missing
Backbone	102 = 106 - 4	100	2
sidechain Q/N	12 = 3 * (2 + 2)	10	2
sidechain W	1	1	0
sidechain R	14 = 7 * 2	6	8
Total	129	117	12
Extraneous	0	1	–

Table 6.2: Quantification of the completeness of the signals in the NHSQC. Out of 106 residues, 3 are prolines and thus lacking a backbone N-H group; also, the first residue is not detectable. Each Q/N sidechain has two protons, giving rise to two signals; partial deuteration gives rise to a second signal for each proton, for a total of 4 signals; there are three Q/N sidechains. The two missing signals were the partial deuteration signals from N77. Each arginine sidechain, if outside the decoupling band, produces two signals. Only three were observed; the sidechains of R4, R26, R61, and R79 were not observed.

Name	Dim 1	Dim 2	Dim 3	Is NOESY?
NHSQC	H	N		No
HNCO	H	N	C	No
HNCACB	H	N	C	No
C(CO)NH-TOCSY	H	N	C	No
HBHA(CO)NH	H	N	H	No
HC(CO)NH-TOCSY	H	N	H	No
HCCH-TOCSY	H	C	H	No
hbCBcgcdHD	H	C		No
hbCBcgcdceHE	H	C		No
NOESY-NHSQC	H	N	H	Yes
NOESY-CHSQC (D_2O)	H	C	H	Yes
Aromatic NOESY	H	H		Yes
Aromatic GNOESY-CHSQC	H	C	H	Yes

Table 6.3: Spectra used in Samp3 analysis. The first four spectra were used for sequential assignment, the next three for aliphatic sidechain assignment, the next two for connecting aromatic sidechain protons to the aliphatic backbone, and the last four (all NOESYs) for obtaining distance restraints.

GSS quality	Number
All CA/CB resonances present	76
CA overlap	7
CB overlap	6
CA and CB overlap	5
CA(i-1) and CB(i-1) overlap	1
Missing 1 resonance	4
Missing 2 resonances	1
Total	100

Table 6.4: Quantification of the completeness and unambiguity of the backbone GSSs, based on the presence of the CA, CA(i-1), CB, and CB(i-1) resonances correlated with each backbone amide group. Backbone glycine GSSs do not have a CB resonance; this was taken into account.

6.8 Figures

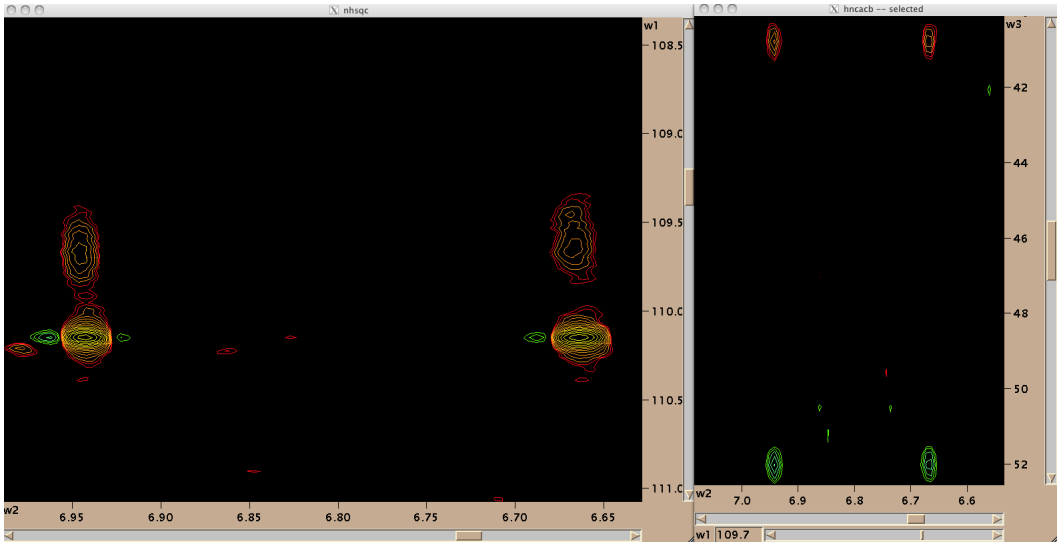

Figure 6.1: An asparagine sidechain in the NHSQC and HNCACB spectra.

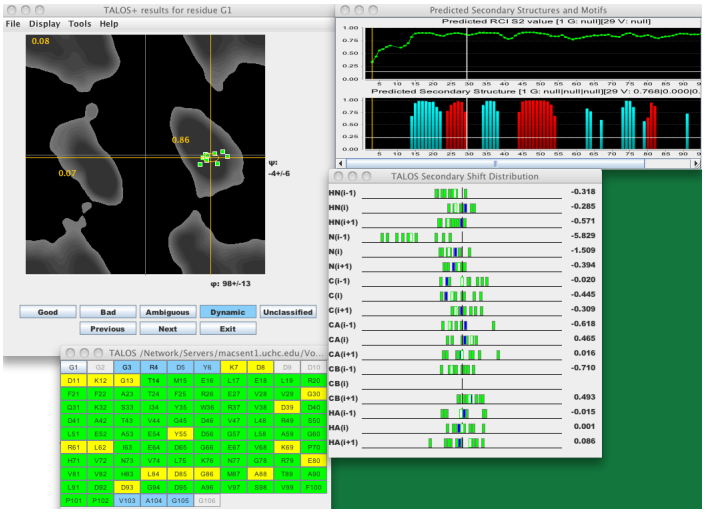

Figure 6.2: The torsion angles and secondary structure prediction according to TALOS+, along with a confidence interval. These results can be shown in a Ramachandran plot.

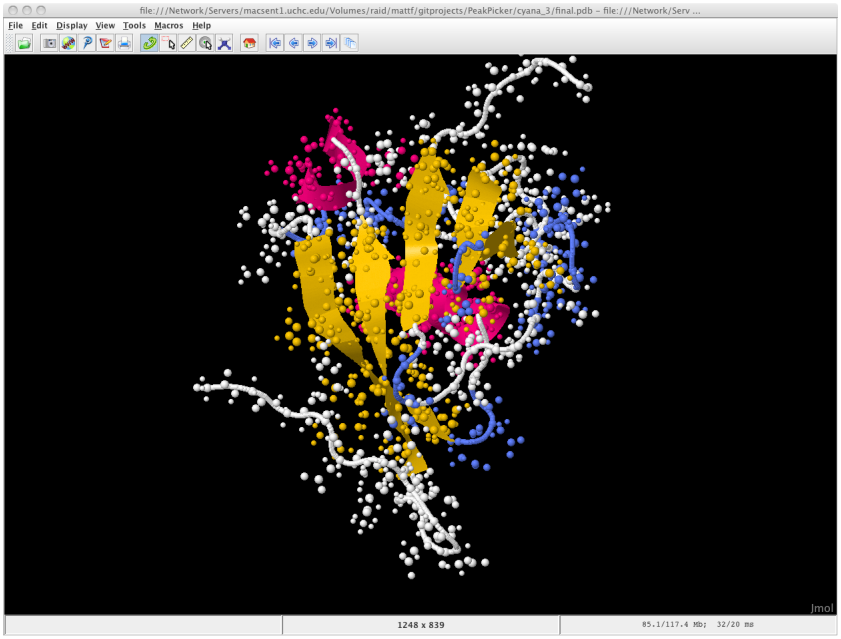

Figure 6.3: The final structure as seen in jmol. Secondary structure is visible: beta-sheets as parallel and anti-parallel arrows, and alpha-helices as coiled ribbons.

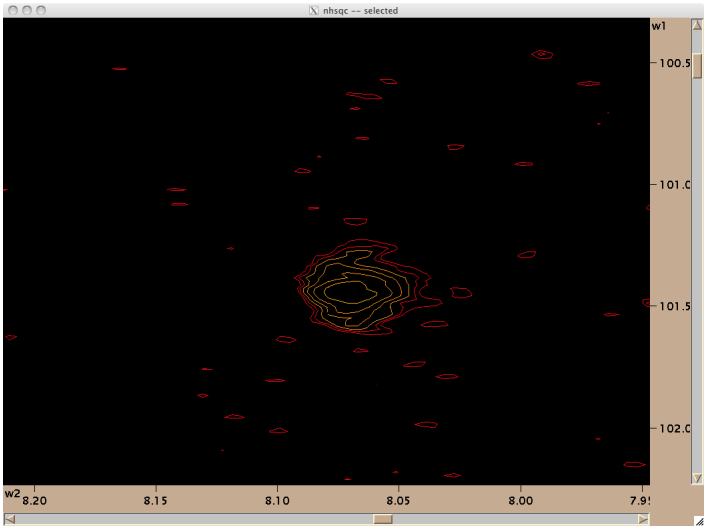

Figure 6.4: A false negative – a true signal missed by the automated peak picker. Its intensity compared to the spectral noise level, as well as peaks in additional spectra whose frequencies in multiple cross sections match this peak, indicate that it is a true peak.

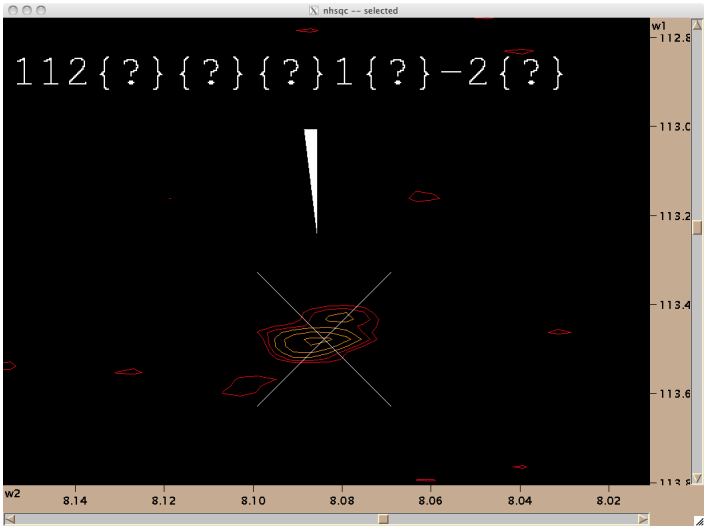

Figure 6.5: An extraneous GSS. It was not assigned to any specific residue.

Chapter 7

Sparky Extension for Reproducible Spectral Analysis

The best way to predict the future is to invent it.

- Alan Kay

Sparky [44] is a popular program for interactive peak picking, GSS construction, and chemical shift assignment. Sparky is implemented with a C++ core, and Python extensions. It is designed with extensibility in mind, with a convenient Python interface through which the core data model can be accessed. The extensions are also able to augment the user interface with additional controls, as well as script common operations, and provide extra algorithms for analysis. Since Python is a full-featured programming language, it is also possible to interact with the filesystem, loading and dumping data if necessary, as well as calling additional third-party tools.

The extension is intended to help the spectroscopist to capture the missing data of analysis: intermediate primary data, extraneous data, deductive meta data, and notes. The general

approach is to augment Sparky's data model and user interface with new functionality.

7.1 Getting started with Sparky

Getting Sparky

A Sparky version including the reproducibility extension can be found at `https://github.com/connjur/SparkyExtensions/releases`. Simply choose the latest version of the correct platform, download it, untar and unzip it, and run the sparky executable (in Contents/Resources/bin in the Mac version, and bin/ in the Linux version).

Dependencies

The reproducibility extension requires a working git installation in order to capture snapshots. git is a freely available tool for version control, and may be found at `http://git-scm.com/downloads`. git works with local files – no setup of remote hosts is required. A git client (a list of which may be found at `http://git-scm.com/downloads/guis`) is a useful tool to help view and manage a git repository. A simple, cross-platform git client that I have used successfully is "giteye", which may be found at `http://www.collab.net/giteyeapp`.

Sparky manual

Sparky manuals, which cover the complete use of the program and its many features in depth, may be found at `http://www.cgl.ucsf.edu/home/sparky/manual/` and `http://pine.nmrfam.wisc.edu/PINE-SPARKY/`.

Sparky keyboard shortcuts

While all Sparky functionality can be accessed through its point-and-click menu interface, it is faster to use the keyboard shortcuts, especially for common tasks. The most useful of these shortcuts are covered in Table 7.1.

Sparky extensions

Extensions are accessible through the "extensions" pull-down menu, as shown in Figure 7.1. The reproducibility extension is also accessible through the "re" shortcut (see Figure 7.2), and the "rg" shortcut open the group editor.

7.2 Concepts

Automated algorithms do not usually produce perfect results in NMR analysis. Automated peak picking, GSS construction, and GSS-residue assignment need manual validation and correction. These manual modifications are inherently difficult, tedious, and error-prone because they are the most complicated to correctly analyze. However, a correct and complete interpretation is necessary to obtain a high-quality final result. The main goal of this extension is to facilitate reproducibility by capturing the entire analysis process, including manual modifications.

The process of manual analysis is composed of a series of discrete steps. At each step, a deductive rule is applied to the data set, producing a modified new data set. Thus, for each rule, knowing the context is important: it allows one to determine how appropriate the application of a specific rule is, as well as what the results should be, and to determine whether the results are consistent with expectations.

GSS and resonance

Sparky's data model is shown in Figure 7.3. It includes entities for spectra, peaks, resonances, GSS, nuclei, and molecules, among others. Sparky does not natively distinguish between a GSS (generic spin system) and a residue, nor between a resonance and a nucleus. GSSs and resonances have been implemented in the extension, based on the CCPN model [43].

Extraneous data

In standard analysis, false positive peaks (picked as true peaks initially, but later determined to be noise or artifactual) and false positive spin systems (not assigned to any residue) are typically deleted and/or ignored. In this extension, **data is never deleted**. Such results are kept, since they have valuable information content, even if they are not used for the immediate purpose of the analysis process (be it structure or dynamics determination). These results are kept by explicitly marking them as not of interest, and placing them in a different category.

Snapshots

Snapshots of intermediate states during analysis are used to enable rehashing of the deductive process. In general, by capturing a snapshot of the data set each time a deductive rule is applied, the context of each manual modification is captured.

An additional benefit of capturing snapshots is that they allow one to go backward in time. This is useful for understanding what happened and why, but is also useful for fixing mistakes and accidents, which usually can not be undone from within Sparky. If a datum is accidentally changed, it may not even be noticed, and it is impossible to easily restore. Capturing snapshots trivially solves both of these problems.

Deductive reasoning

Capturing the deductive rules applied during the process of manual assignment provides semantic information about what is being done and why. This makes it more meaningful to examine the sequence of snapshots of an analysis process and determine the context.

7.3 Project setup

From the terminal, create an empty directory and "cd" into it. Intialize it as an empty git repository and create the Sparky directory structure with the following terminal commands:

```
$ git init
$ mkdir Projects/
$ mkdir Save/
$ mkdir Data/
```

Make sure to save all spectra files in the "Data/" directory. When in Sparky, save the project in the "Projects/" directory. The Sparky .save files will automatically be written into the "Save/" directory.

Now start Sparky and take care of tedious initial project setup. This includes:

- opening each of the spectra that will be used (using the "fo" shortcut)
- setting the contour levels ("ct"). Some nice default setting are "15 levels" for both positive and negative contours, and "red-yellow" color for positives, along with "blue-green" for negatives
- setting the ornament and label sizes ("ot" and "oz")

- setting the visible depth
- getting the axes in the correct order ("xx" and "xr")
- setting the aspect of each spectrum appropriately ("yt")
- syncing axes of identical nuclei. For example, the ^{1}H axes of an NHSQC, HNCACB, HN(CO)CACB, HNCO, and C(CO)NH-TOCSY should be synced; similarly, all their ^{15}N axes should be synced as well. However, the ^{13}C axis of an HNCACB and an HNCO should not be synced, although the ^{13}C axis of an HNCACB and a C(CO)NH-TOCSY should by synced

Now it is time to take a snapshot of this configuration.

7.4 Capturing a snapshot

Whenever a snapshot is desired, this procedure must be followed:

- save the project ("js")
- type in the appropriate deductive reason in the "deductive reason used" text box, or whatever annotation accurately describes the purpose of the snapshot
- click the "make snapshot" button

This will use git to create a new snapshot. At any time, the contents of the git repo can be examined using the terminal and the file browser. Git supports many features for examining and manipulating history. These may all be accessed from the terminal or from a git client program. Snapshots should be captured immediately after running an automated tool and

before making any manual modifications. During manual analysis, snapshots should be captured after each focused sub-goal is completed.

7.5 Reproducible backbone assignment tutorial

A a nice tutorial and dataset in Sparky format may be found at `http://www.nmr.chem.uu.nl/~abonvin/tutorials/Assignment-Data/assignment.html`. It includes the NHSQC, HNCACB, and HN(CO)CACB spectra of a small protein, providing enough data to carry out sequential backbone assignments. This section will show how to reproducibly make these chemical shift assignments.

NHSQC peak picking

Use standard Sparky facilities to peak pick your NHSQC spectrum. Set the contour levels high enough that few noise and artifact peaks will be picked, but low enough that most true signal peaks will be picked.

NHSQC cleanup: signal/noise/artifact identification

Some of the peaks Sparky found will turn out to be noise or artifact peaks. **Do not delete peaks, ever, for any reason!** Instead, when peaks are identified as such, select them and press the "Set selected peaks to noise" or "Set selected peaks to artifact" button, as necessary. Peaks must not be assigned when setting them to noise or artifact.

Sparky will not find all true peaks. Whenever a true signal is found unpicked, simply pick the peak manually by switching the cursor mode to "find/add peak" and using "pc" to center the peaks after they have been picked.

Overlapped peaks are also a problem. They may cause too few peaks to be picked, or peaks to be picked in a slightly wrong position. If these errors can be rectified manually, simply add peaks and move others as necessary.

Either a restricted peak pick (using NHSQC peaks) or a standard peak pick of the full dimensions should be used to peak pick all other N-H-rooted spectra (such as the HNCO, HNCACB, CCONH-TOCSY, etc.).

GSS initialization

It is convenient to create one GSS for each NHSQC peak that is or may be a signal peak. Select the NHSQC peaks which will be used as GSS roots. All signal peaks can be selected using the "select signal peaks" drop-down. Then press the "create new group for peak" button.

When an NHSQC peak is used to initialize a GSS, the GSS will start off with two resonances: one for the H, and one for the N.

GSS construction

Open the peak-GSS assignment dialog by pressing the "Open peak-GSS dialog" at the bottom of the reproducibility window.

Now set up the parameters by choosing the spectra from which peaks will be used as GSS roots (typically the NHSQC) and that which has the peaks to be assigned. Make sure the desired dimensions are matched and set the tolerances appropriately; I typically use 0.2 PPM for a ^{15}N axis and 0.02 PPM for a ^{1}H axis but this can vary based on the alignment between spectra. Finally, select all the peaks in the "from" spectrum that should be used; you can select all signal peaks using the "select signal peaks" drop-down.

In GSS construction, for each peak in the "from" spectrum, all peaks in the "to" spectrum within the tolerances will be assigned to the same GSS as the "from" peak. Peaks in the "to" spectrum that match 0 "from" peaks will not be assigned to any GSS; those that match more than 1 peak will also not be assigned, but a warning will be generated in the shell, allowing manual resolution.

Matching requires that some subset of the spectral dimensions match. Using an NHSQC and an HNCACB, the H and the N dimensions match. The resonance assignments of the peak cross sections will be carried over for matching dimensions, and new resonances will be created for the C dimension.

Peak cross section to resonance assignment

A resonance is assigned to each cross section of each peak which is in a GSS. If a peak cross section has a unique chemical shift value within a GSS, it will be the only one assigned to that resonance; if multiple cross sections share the same or similar chemical shift value, they will be assigned to the same resonance. This reflects the NMR property that a nucleus resonates at a characteristic frequency across spectra.

Peak cross section to resonance assignment can fail in two cases, and these can be resolved by manually modifying the assignments to resonances. First, when the chemical shifts between the peak cross sections do not match within the tolerances, multiple resonances are created. These can be merged using the built-in Sparky assignment tools by changing the assignment of one peak cross section to match the other. Second, in the case of overlap, a single resonance is created when there should actually be multiple resonances. This can also be resolved using the built-in Sparky assignment tools, by simply changing the assignment of

one peak dimention to a fresh resonance id.

Changing peak-GSS assignments

There are two major cases for moving a peak from one GSS to another. First, for GSS types such as Q and N sidechains, there are usually multiple peaks in the NHSQC (because of the two protons). Select the appropriate peaks, choose a GSS id (typically the lowest of the GSS ids of the selected peaks, just for convenience and consistency) and press the "set groups of selected peaks" button. Second, peaks may be assigned to the wrong GSS. Select the incorrect peaks, type in the desired GSS id, and press the "set groups of selected peaks" button.

Resonance typing

The types of resonances are assigned using BMRB statistics, GSS typing, peak characteristics, and the pulse sequences of the spectra in which they appear. For some spectra, such as the NHSQC, there are relatively few choices: for backbone GSSs, the resonance types are always amide-H and amide-N. However, other spectra have more choices: for the HNCACB, resonances in backbone GSSs in the C dimension can be CA, CA(i-1), CB, or CB(i-1).

There are two ways to assign resonance types. First, each resonance assigned to a peak's cross sections may be assigned at once. The possibilities for these resonance typings are called "peaktypes" and depend on the pulse sequence of the spectrum in which the peak is found. This can be done using "assign peaktype" drop-down menu. Select the correct spectrum, which will bring up a list of possible peaktypes which may be found in that spectrum. Make sure to set the dimension order properly.

The second way is to assign resonance types individually. This can be done using the

keyboard shortcut "rg" to bring up a group/resonance editor (see Figure 7.2). Clicking on a group will allow for editing of group assignments, while clicking on a resonance allows for resonance typing.

Ambiguous resonance typings are also supported. Common examples include CA/CA(i-1), HA2/HA3 (glycine), HB2/HB3 (many amino acids), and QB (alanine).

GSS typing, sequential, and sequence-specific assignment

GSSs are assigned types based on BMRB statistics, the presence or absence of characteristics resonance types, and possibly also based on the assignments of linked GSSs. GSS types for H-N-rooted GSSs include backbone types for each of the 20 standard amino acids, as well as sidechain types for Q, N, R, W, and K.

Sequential GSS assignments are identified based on overlapping compatible resonances between GSSs: for example, a GSS with a CA at 58.32 PPM and a CB at 30.27 PPM, and a second GSS with a CA(i-1) at 58.21 and a CB(i-1) at 30.41: these GSSs have overlapping compatible resonances and can be sequentially assigned. It is not necessary that resonance types are unambiguously assigned before doing sequential GSS assignments: if it is unclear whether a resonance is a CA or CA(i-1), this may be assigned simultaneously as the sequential assignment is made, if it is consistent with the overlap from the other GSS.

Additionally, sequential GSS assignment can lead to the picking of additional peaks, or the resolution of GSS overlap, if such picking and/or resolution leads to consistent and compatible overlap with another GSS. GSSs are assigned to residues on the basis of GSS typing, sequential GSS assignments, and the primary sequence.

GSS assignment is accomplished using the "group editor" dialog by clicking on the

appropriate group. This will bring up a dialog which allows typing, sequential assignment, and sequence-specific assignment.

7.6 Pitfalls

Sparky's data model was extended to support reproducible analysis. However, it is possible to circumvent the extension's model and assign peaks and resonances differently. If care is not taken when doing so, the data may be incompatible with the extension.

7.7 Creating an NMR-STAR file

While the data is stored as standard Sparky-formatted files in a git repository during analysis for convenience, after analysis is complete, the project may be converted to a single NMR-STAR file, containing the complete history of the project. This conversion may be accomplished using the code found at `https://github.com/CONNJUR/Samp3-extractor`, which extracts each snapshot version from the git repository, parses the snapshot into a data structure, calculates semantic diffs between successive snapshots, and finally generates an NMR-STAR file containing the data.

7.8 Tables

shortcut	effect
ct	open contour dialog
yt	open spectrum dialog
pc	center peak
aD	delete peak assignment
re	open reproducibility dialog
rg	open group editor dialog
ot	ornament settings
xr	roll axes
xx	transpose axes
jo	open project
jc	close project
js	save project
fo	open spectrum
py	open Python interpreter

Table 7.1: Important Sparky keyboard shortcuts.

7.9 Figures

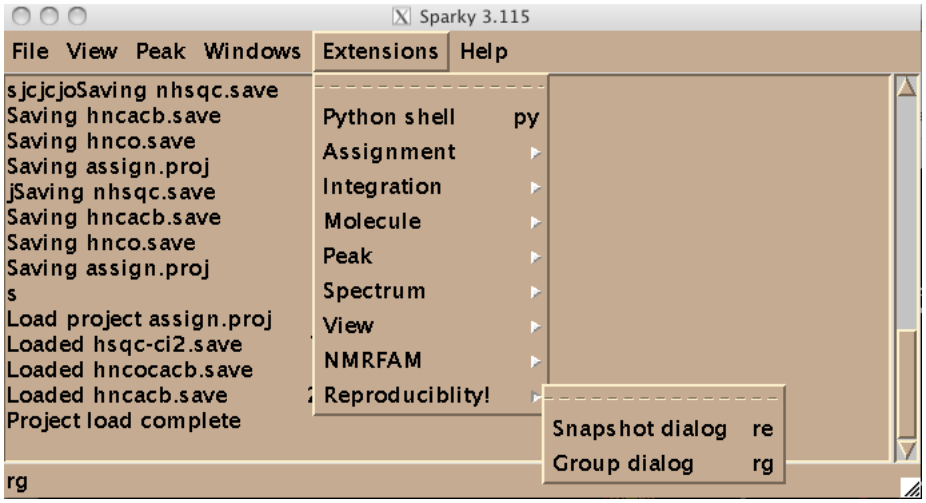

Figure 7.1: The Sparky interface, showing how to activate the reproducibility extension.

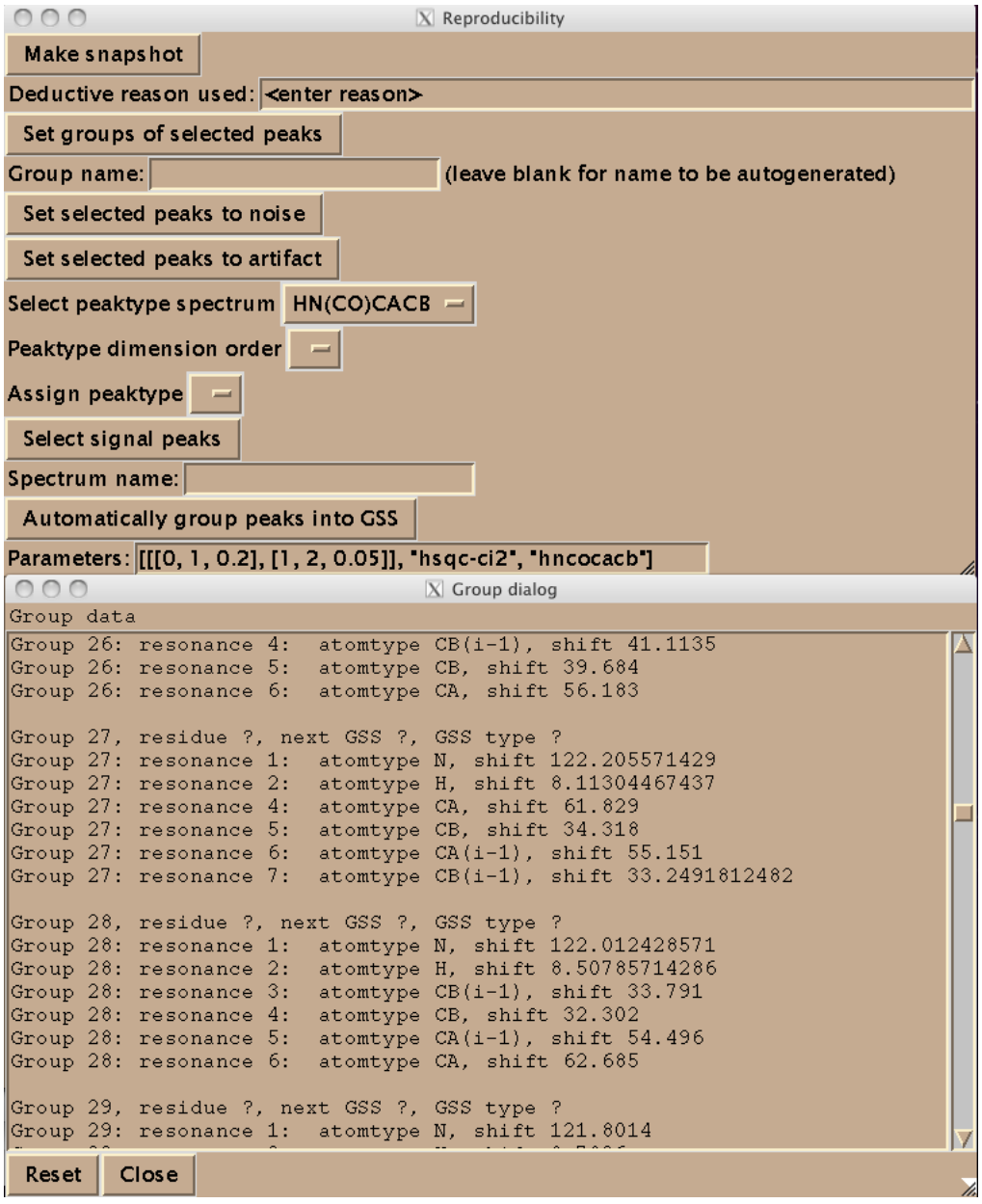

Figure 7.2: Two widgets provided by the reproducibility extension. The first provides core functionality for making snapshots, annotating snapshots, and creating and building GSSs, as well as assigning peaktypes. The second provides functionality for displaying, assigning and merging GSSs and resonances.

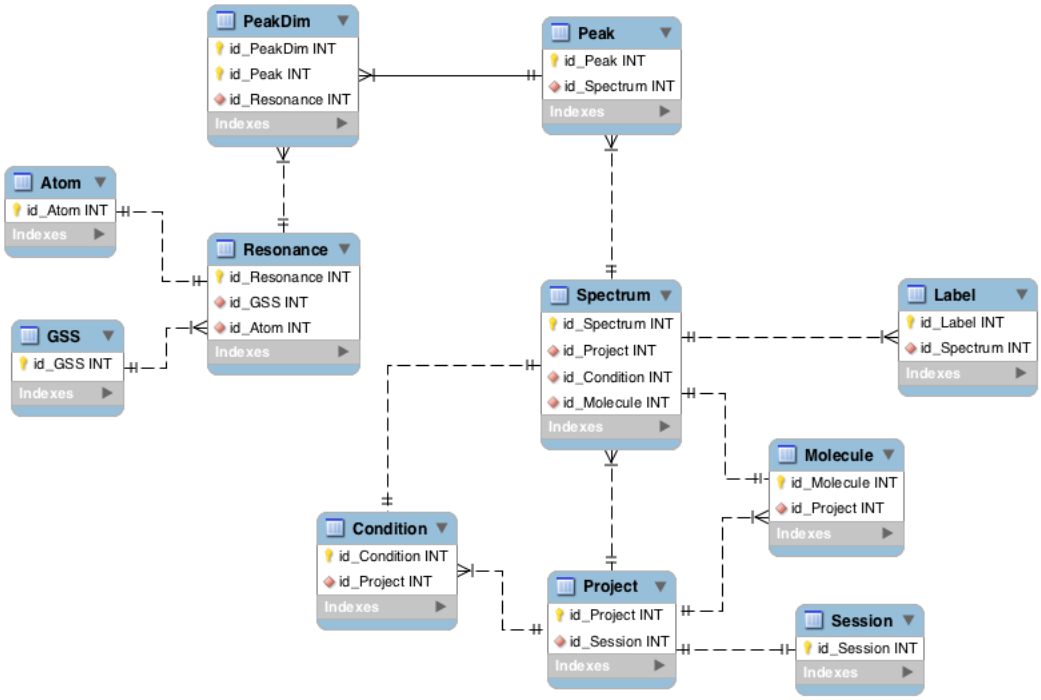

Figure 7.3: The Sparky data model showing the key relationships. These data are available from within Sparky extensions. The model was created in MySQLWorkbench.

Chapter 8

Software for Practical, Reproducible Analysis

When we had no computers, we had no programming problem either. When we had a few computers, we had a mild programming problem. Confronted with machines a million times as powerful, we are faced with a gigantic programming problem.

- Edsger Dijkstra

Software plays an indispensable role in NMR data analysis, allowing us to effectively manage and process massive data sets. This chapter will present several software projects that facilitate reproducible analysis.

8.1 NMR-STAR library

NMR-STAR is a file format used by the BMRB [2] for archival of NMR data. As such data is useful for further studies, and archiving data in the BMRB is the primary means of

dissemination, it is important to be able to work with NMR-STAR files.

This library provides a robust interface for working with NMR-STAR files; it allows the creation of NMR-STAR files as well as extraction of data from existing files.

To handle these files, a library was implemented both in Java [45] and in Python. This library provides capabilities both for reading and for writing NMR-STAR files. Although several tools for dealing with NMR-STAR files had already been implemented [43, 2], there are several attributes of this library which set it apart:

- error reporting of illegal input. When malformed input is encountered, a useful, location-specific error is reported which includes sufficient information to quickly pinpoint and diagnose the problem.

- complete, standards-compliant NMR-STAR syntax definition.

- open source under the MIT license. This allows other interested developers to inspect the source code to gain ideas, use the library in new applications, and modify and extend the library to fix problems or add new features if necessary.

- low coupling. In order to use it, the library is simply imported using standard language facilities. It does not require any external tools or dependencies, reducing the barrier to setup and installation. It does not require learning to use additional tools or languages; it takes advantage of the native facilities for abstraction and composition provided by the host language.

- high cohesion. The library provides a simple, focused interface. This means it is easy to learn and use because it only deals with parsing the concrete syntax of NMR-STAR files.

The library is freely available online (`https://pypi.python.org/pypi/NMRPyStar`, `https://github.com/CONNJUR/StarParser`).

An example of NMRPyStar in action is shown in Figure 8.1. First, it is imported into the client module. Then, using a built-in Python library to read URLs, an NMR-STAR file is downloaded from the BMRB [2]. The file is then parsed, and a parse tree, representing the structure of the file, is returned; see Figure 8.3 Figure 8.2. A parse tree is easier to query than an unstructured string. Finally, a query to extract the chemical shifts from the parse tree is executed, and the results are used to calculate the standard deviation of chemical shift assignments, grouped by residue type and resonance type; see Figure 8.4.

This examples demonstrates the library's flexibility: since it imposes no restrictions on its use, it is easily adapted to different situations; in this example, to reading NMR-STAR files downloaded programatically from the BMRB. This flexibility is enabled by its bottom-up design. The parse tree, which is the output from the parser, holds a representation of the file, and queries are easily run against it.

8.2 CONNJUR ST

The spectral reconstruction phase involves the processing of time-domain FID data to frequency-domain spectra. There are several approaches for such a transformation, including the Fourier Transform, Maximum Entropy reconstruction, and Multi-dimensional decomposition [97, 98, 99]. There are two key pieces to the data: the primary data (time-domain or frequency-domain), which is typically in a binary format; and the meta data which records important properties such as number of points, dwell time, and spectral width.

Spectral reconstruction involves the sequential application of multiple functions, in-

cluding linear prediction, zero fill, and apodization, each of which must be parameterized appropriately, for the purposes of optimizing spectral characteristics such as peak shape, line width, and signal-to-noise ratio. In general, the exact effect of each of these operations may depend on both the primary data as well as the meta data, and both may be modified during the operation.

Correct spectral reconstruction requires that the meta data is correctly handled, and reproducibility requires that all of the parameters are captured as well (if the exact software versions used are captured, then it is not necessary to capture the input and output primary and meta data from each operation, since these can be regenerated as needed as long as the initial primary and meta data are captured). Two recent tools from our lab, ST [49] and WB, facilitate reproducible spectral reconstruction.

ST was designed with the goal of translating between various formats of time- and frequency-domain data, including Bruker, Varian/Agilent, NMRPipe, and RNMRTK. It is necessary to convert between multiple formats during spectral reconstruction and analysis because different tools, each of which provides valuable functionality, require different formats for input and output. Thus, to use tools with differing format requirements, conversion will be necessary. While several tools do exist which implement specific conversions between pairs of formats, there was previously no tool able to perform a conversion between any two arbitrary formats. The result was an artificial restriction on combinations of tools, due to format constraints. In addition, attempting to remove this restriction by implementing additional tools is not a satisfactory solution, because the number of tools required – if each one performs a single conversion – grows with the square of the number of formats. Such a solution clearly requires too much time and effort for initial implementation, as well as future

maintenance effort.

ST addresses this problem by means of a common data model, which can be converted to and from any format. For each format, a single importer and a single exporter is required, which deal with conversion between the format and ST's common data model. This reduces the number of converters required to the number of formats. Thus, for translation between any of five formats, the number of conversions required is reduced from 25 to 10 – a 60% reduction in the amount of conversions. The discrepancy is even greater when larger numbers of formats are considered.

This program is implemented in the Java programming language as an open source library available from our website at `http://connjur.uchc.edu/downloads/st/`. The first major advantage of using Java is that the code, once compiled, may run on any Java Virtual Machine (JVM). JVMs have been implemented for many platforms, including Windows, Macintosh, and Linux. This enables ST to run without modification on virtually any computer. The second major advantage of Java is that as a popular programming language, there are many developers familiar with its syntax, semantics, class libraries, tooling, and deployment.

ST promotes data integration and consistency through the use of a common data model and a single, unified conversion strategy. It also automatically reads meta data, easing the burden of meta data correctness for the user, which helps to ensure that the meta data is more correct. By applying a single interface to any format conversion, the program has a smaller learning curve compared to learning multiple differing interfaces for multiple tools.

ST has recently been extended to support non-uniform time-domain data. As the Rowland NMR Toolkit format is required in order to use its implementation of Maximum Entropy

reconstruction [98], ST is an important enabler of the use of the technique, helping to make non-uniform data collection a realistic possibility for users who might otherwise face significant hurdles in tooling.

8.3 CONNJUR WB

WB is a tool for spectral reconstruction which builds on the successes of tools such as RNMRTK [98], NMRPipe [97], and ST [49] by providing additional features for data integration, reproducibility, meta data correctness, interactivity, and expressiveness of spectral processing pipelines. While WB is a standalone tool, it relies on external tools for the actual execution of spectral processing functions. This allows it to capitalize of user's familiarity with and knowledge of existing tools.

The general architecture of WB is three tiers. The first is the user-facing Graphical User Interface (GUI). The GUI is responsible for providing an intuitive, obvious, integrated, pleasant, and consistent experience to users, and for ensuring that the critical information is present and easily accessible. This layer is implemented using Java's Swing library. At the other end is the third layer, or back-end, which is responsible for data persistence, integrity, and integration. The back-end is implemented as a MySQL relational database management system (RDBMS), in which are stored both the spectral meta data and the spectra themselves (if desired); information specific to WB and its internal model of spectral reconstruction is also stored in the database. The middle layer is responsible for mediating the data exchanges between the GUI, the back-end, and third-party tools such as RNMRTK and NMRPipe.

WB allows viewing and modification of spectral meta data, which is important for ensuring correctness. By capturing the meta data, WB facilitates reproducibility of spectral

reconstruction. Similar to ST, WB is implemented in Java, allowing it to run anywhere that a JVM does; however, since it uses third-party tools which are platform-dependent, its usefulness on a platform is restricted if those additional tools do not run there. Nevertheless, WB can be used to build spectral reconstruction workflows on any platform that has a JVM. WB also allows export of spectral reconstruction data in the eXtensible Markup Language (XML) and NMR-STAR formats. The XML export facilitates sharing between peers, while the NMR-STAR exporting capabilities enables reproducible archival as well.

WB has a robust model of spectral reconstruction, and clearly and simply presents its model to the user. WB treats spectral reconstruction as a workflow composed of actors, which are analagous to functions. Each actor takes as input primary data and meta data, and produces primary data and meta data as output. The actor is responsible for presenting the information required for correct parameterization of the underlying function, and may contain significant logic and functionality. A notable example is the actor for Maximum Entropy reconstruction: estimating the noise correctly is a prerequisite for Maximum Entropy [112]; this actor is able to both estimate the noise level, as well as parameterize the RNMRTK implementation [98]. A portion of the model is shown in Figure 8.5.

A further benefit of WB's approach was noted at a workshop hosted at the University of Connecticut Health Center in June, 2012. For beginning NMR spectroscopists, the barrier to entry is rather high. Not only is a large amount of domain knowledge required, but one must also be familiar with the incidental complexity of NMR, including idiomatic expressions and implicit data. WB was observed to greatly reduce the learning curve due to its integration of necessary domain knowledge, explicit handling of relevant data, and GUI-based presentation. While such an interface may not be necessary for experts, it is certainly useful for beginners.

8.4 Sample scheduler

The creation of effective sample schedules is an important aspect of efficient, non-uniform data collection [51, 52, 112]. There are multiple strategies for data collection. The strategy used to generate a sample schedule and the exact sample schedule used to collect time-domain data has an effect on the quality of the data and on the ease of later analysis, due to properties such as artifacts (described by the point-spread function), resolution, and sensitivity (related to signal-to-noise ratio).

A tool has been implemented to reproducibly capture the parameterizations used in sample schedule creation, and is available online at `https://github.com/connjur/PyScheduler`. The tool features a collection of popular algorithms for creating non-uniform sample schedules with specific, desirable properties. The algorithms are integrated within a single uniform, consistent interface which allows all input parameters and outputs to be captured and archived. It integrates with previous work `http://sbtools.uchc.edu/nmr/sample_scheduler/`.

The tool implements a data model of sample schedules; see Figure 8.6. The model deals with non-uniform quadrature (partial component) [51], non-uniform time delays, and non-uniform numbers of transients. The latter aspect of non-uniform data collection remains a relatively unexplored domain, into which this tool provides novel data representation capabilities. In general, an N-dimensional sample schedule, which is used for collecting (N+1)-dimensional data sets, consists of a collection of N-dimensional pairs, each of which has a time delay, represented as a positive integer, as well as a quadrature phase, one of R or I. Associated with each point is the number of transients to collect, also modeled as a positive integer. Within a sample schedule, each N-dimensional pair represents a unique point in the

space of the sample schedule; for example, in a 2-dimensional schedule, the point (2,R),(4,I).

One of the main strengths of this sample scheduler is its ability to flexibly combine multiple different sampling approaches to create a schedule. This flexibility stems from its breakdown of sampling algorithms into independent pieces with standard interfaces; by selecting one piece of each category, a combinatorial number of different choices can be made, resulting in sample schedules with all imaginable kinds of properties. This avoids artificial restrictions of what kinds of sample schedules can be constructed and faciliates experimentation. The general categories of algorithms are:

- coordinate generator. Responsible for generating N-dimensional time delays. Includes algorithms for generating all combinations within finite N-dimensional bounds, Poisson gap [132] sampling, as well as others. See Table 8.1.

- quadrature generator. Responsible for generating the quadrature components. Includes algorithms for full-component generation as well as various partial-component strategies. See Table 8.2.

- point selector. Applies a filter to points based on coordinates, quadrature components, and number of transients, which can be used to create biased sampling schemes, such as exponentially-weighted decay. See Table 8.3.

- point modifier. Modifies some or all points in a sample schedule, slightly changing the coordinates of points. Includes algorithms for bursty [133] and blurred [134] modifications. See Table 8.4.

- special point generator. Forces the addition of specific points to a sample schedule, such as the first point along each axis. Such properties are useful when evaluating the

quality of spectra. See Table 8.5.

- formatter. Includes facilities for generating textual output of a sample schedule in RNMRTK [98], Bruker, JSON, and Agilent formats. See Table 8.6.

To create a sample schedule, the algorithms must be chosen and parameterized. The algorithms are implemented as functions in standard Python modules, and could therefore be imported and used as a simple library. A command-line interface is also included. The interface requires the choice of algorithms and parameters to be passed in through an appropriately formatted file. The parameters are then extracted and passed to the appropriate algorithms, the sample schedule is constructed, and the output is returned.

An example of the program in action is given in Figures 8.7 and 8.8. The first figure shows a parameter file in structured text; this is passed in to the program. The program then generates a sample schedule, and writes the schedule out as a file, a part of which is shown in the second figure (a graphical view of a sample schedule is shown in Figure 8.9; the numbers are delay times in the indirect dimensions which are shown as x- and y-coordinates in the chart). The parameter file includes parameters which apply to the schedule as a whole, as well as to specific dimensions; the output file includes a single line for each set of time increments, and includes the associated quadrature components.

This program was also integrated with an existing Java-based program written by Mark Maciejewski, Val Gorbatyuk, and Jeffrey Hoch (`http://sbtools.uchc.edu/nmr/sample_scheduler/`), in order to leverage the Java program's GUI capabilities for parameterizing sample schedule creation and displaying sample schedules and pointspread functions. Figure 8.9 shows a sample schedule created by my tool. The Java GUI is used to set the parameters, and it then invokes the Python program, and finally displays the results. Note

the increasing sparsity as time delay increases in both non-uniformly sampled dimensions. Figure 8.10 shows the pointspread function of this sample schedule, indicating artifacts that it will create in the frequency domain. This was calculated using a Fast Fourier Transform, and was also implemented in the Java program by Maciejewski, Gorbatyuk and Hoch. Figure 8.11 provides a comparison to the first sample schedule; the parameterization is identical, except that the first points along each axis are all collected (this is an example of a special point selector). This difference causes a noticable change in its pointspread function, seen in Figure 8.12.

8.5 Discussion and conclusions

All software discussed in this chapter is available under standard open source licenses (either the MIT licence or the LGPL). These licenses grant the rights to inspect the source code, use the code in a program, modify the code, and incorporate the code into a larger program. While I do not believe that it is necessary or desirable to force all scientists to publish source code under open source licenses, I do believe that there are concrete benefits to publishing source code under such licenses.

A major theme of this dissertation is reproducibility and its importance to science. Reproducibility is fostered by explicitness both in data collection and analysis, as well as in communication between scientists, of results, protocols, and analyses. Journal articles are an excellent means for communicating scientific findings, and the goal of such articles is typically to include sufficient detail that the findings can be reproduced. If the article does not include sufficient detail, authors are often perfectly willing to engage in communication with interested scientists to provide additional detail. However, in many cases, this does not apply

to the software used to produce a result. Scientific software is often treated as proprietary intellectual property which may not be shared. This leads to an interesting constrast between the treatment of source code compared to data and experiemental protocols. If it were not for free and open communication of scientific results, it is unlikely that science would have progressed as rapidly. Similarly, by not freely sharing source code, I believe that science's progress is restricted. While releasing code under open source licenses is clearly not the only way to lift such restrictions, it seems to offer clear benefits to scientists and funding agencies alike.

The choice for which programs to describe in this section was driven by the theme of reproducibility. All of the programs described in this section facilitate reproducible data analysis. The Sparky extension provides functionality for tracking intermediate, derived, and extraneous data during spectral analysis, and integrates with the NMR-STAR library for recording that data in NMR-STAR files for later deposition to the BMRB. ST, WB and the sample scheduler model and capture the meta data associated with data collection and spectral reconstruction, by making that meta data explicit, visible, and possible to correct and modify if necessary.

The use of explicit data models provides succinct, precise documentation as to the intended use of a program and the semantics of its input and output data. The design and dissemination of data models was an important part of the implementation of the software programs described in this section. A data model, expressed as an entity-relationship model or a similar format, provides an overview of the core functionality of a program through the data types which it documents.

A critical challenge that must be faced in the development of scientic software is the

apparent tradeoff between flexibility and adaptability on the one hand, and robustness and strong guarantees on the other. In the first case, the need is driven by the ever-changing nature of the scientific enterprise: as new phenomena are discovered and studied, and new techniques and strategies for analysis are invented and applied, the requirements of the supporting software naturally must be correspondingly updated. In the second case, scientists need to have at their disposal software which works correctly in order to guarantee accurate and reproducible computational analyses. Much of the conviction that a software program works correctly comes from the experience of many users applying it; it is difficult to understand whether software works correctly without using it. Over the course of my studies, I have explored an alternative approach to scientific software development which I believe can help address this issue by resolving the tradeoff.

This technique is known as "bottom-up" [135, 136] design (as opposed to "top-down"). The core of this technique of software design is to solve small problems simply and completely, and then to build bigger software – whether programs or libraries – by combining the small solutions. The opposing strategy of top-down design focuses on breaking problems down into smaller problems, and so on, until small enough problems are reached to be implemented directly as code. While both strategies can lead to successful outcomes, the key difference that has been observed is that while top-down approaches typically require the solution to be known in advance and do not react well to changing requirements, bottom-up solutions are easier to implement effectively with incomplete prior knowledge of the problem and of the desired solution, and are also better equipped for dealing with changing requirements [137, 135, 136]. Therefore, it would seem that bottom-up software is a more natural fit for science, due to the ever-changing software needs of scientists.

The tools presented here, including the NMR-STAR library, the sample scheduler, the Sparky extension, as well as ST and WB have been purposely designed with the bottom-up approach in mind. The result is a flexible tool that is readily adapted to different uses and straightforward to extend. The key realization encountered while designing these tools was the importance of avoiding coupling [138] whenever possible: by specifying as few irrelevant details as possible, a software solution gains flexibility to be applied in different ways as part of larger programs.

8.6 Tables

Halton	a sub-random sequence of points
HyperTable	a Cartesian product of points
Poisson gap	Poisson gap method
Poisson disk	Poisson disk method
Concentric shell	points on concentric shells
Spiral	points spiraling outward from the origin
Radial	spokes directed outward from the origin

Table 8.1: The point generators of the sample scheduler.

All	all quadrature components
Single random	one randomly chosen component
Just reals	the all-real component
FRSB	real in dim1, both in dim2

Table 8.2: The quadrature generators of the sample scheduler for each point.

All	select all points
Exponential	exponentially decaying bias
Random	random bias
User defined	user-defined bias expression

Table 8.3: The point selectors of the sample scheduler.

Bursty	select additional points around a point
Blurred	bump a point in each dimension
None	don't make any changes

Table 8.4: The modifiers of the sample scheduler.

None	force selection of no points
All lower bounds	all points along axes
Point block	all points less than given bounds
First point	the point with minimum time delays
Last point	the point with maximum time delays
Halton	a sub-random sequence of points

Table 8.5: The forced point selectors of the sample scheduler.

Bruker	output schedule in Bruker format
Agilent	output schedule in Agilent format
RNMRTK	output schedule in RNMRTK format
JSON	output schedule in JSON format

Table 8.6: The formatters of the sample scheduler.

8.7 Figures

```
from .. import fullparse as parser
import json
import urllib2
import numpy


def parseUrl(url):
    page = urllib2.urlopen(url)
    inputStr = page.read()
    page.close()
    return parser.parse(inputStr)

def getChemicalShifts(dataBlock, saveName='assigned_chem_shift_list_1'):
    saveShifts = dataBlock.saves[saveName]
    loopShifts = saveShifts.loops[1]

    shifts = {}
    for ix in range(len(loopShifts.rows)):
        row = loopShifts.getRowAsDict(ix)
        key = (row['Atom_chem_shift.Comp_ID'], row['Atom_chem_shift.Atom_ID'])
        if not key in shifts:
            shifts[key] = []
        shifts[key].append(float(row['Atom_chem_shift.Val']))
    return shifts

def run():
    model = parseUrl('http://rest.bmrb.wisc.edu/bmrb/NMR-STAR3/18504')
    if model.status == 'success':
        shifts = getChemicalShifts(model.value)
        many = [(k, (len(v), numpy.std(v))) for (k, v) in shifts.items()]
        devs = filter(lambda x: x[1][0] > 1, sorted(many, key=lambda x: x[0]))
        for result in sorted(devs, key=lambda x: (x[0][1], x[1][1])):
            print result
        return result
```

Figure 8.1: A code snippet of NMRPyStar accessing the BMRB. A file is downloaded, then parsed. Note that in order to use the NMRPyStar library, it need only be imported as a regular library. It does not require any external dependencies or tedious setup.

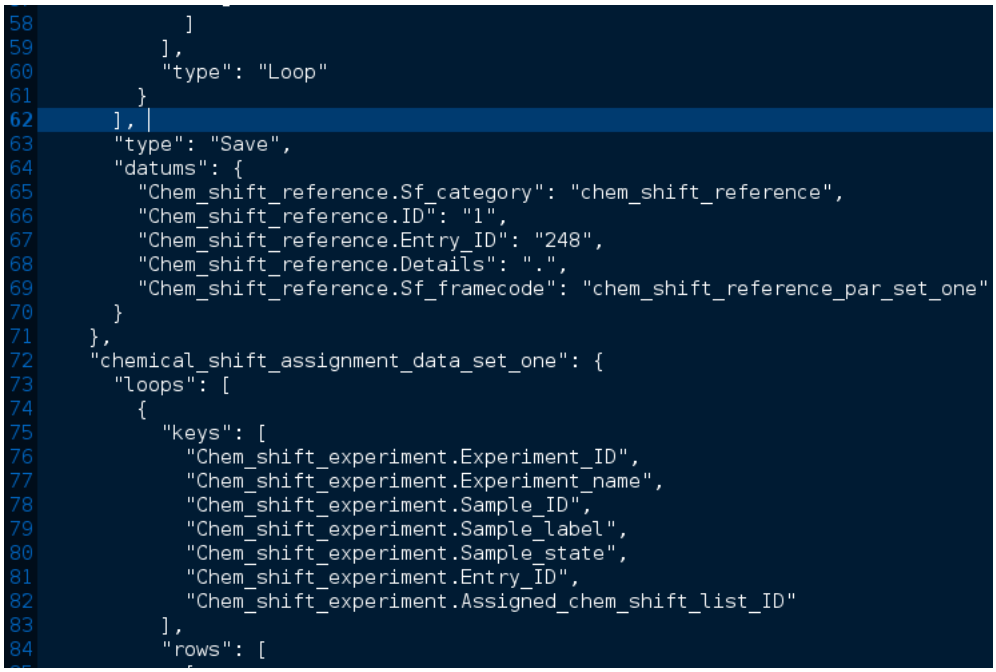

Figure 8.2: NMRPyStar produces a parse tree as output, shown here in JSON format

Figure 8.3: The parse tree can be used to extract key NMR information. It is far easier to query a structured tree than to query a flat, unstructured string.

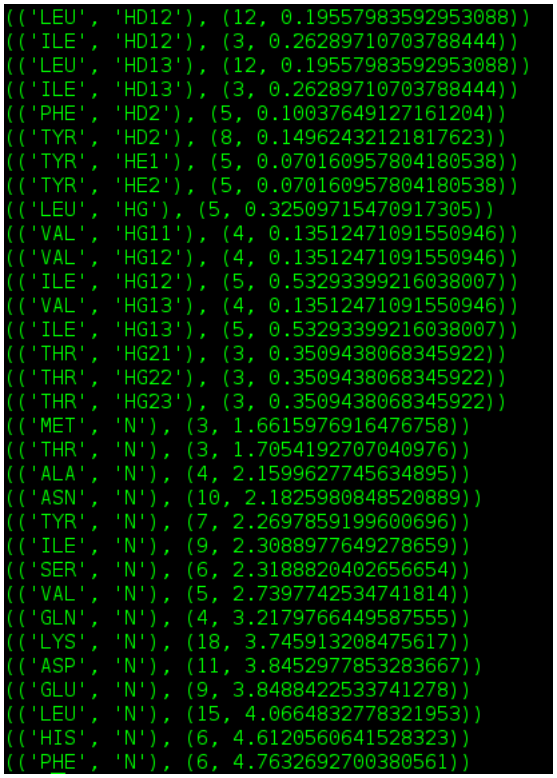

Figure 8.4: The results of a query run against the parse tree. The query groups the assigned chemical shifts by amino acid type and resonance type, and calculates the standard deviation of each group. The first column is the amino acid name. The second column is the resonance type. The third column is the number of measurements, and the fourth column is the standard deviation of the measurements.

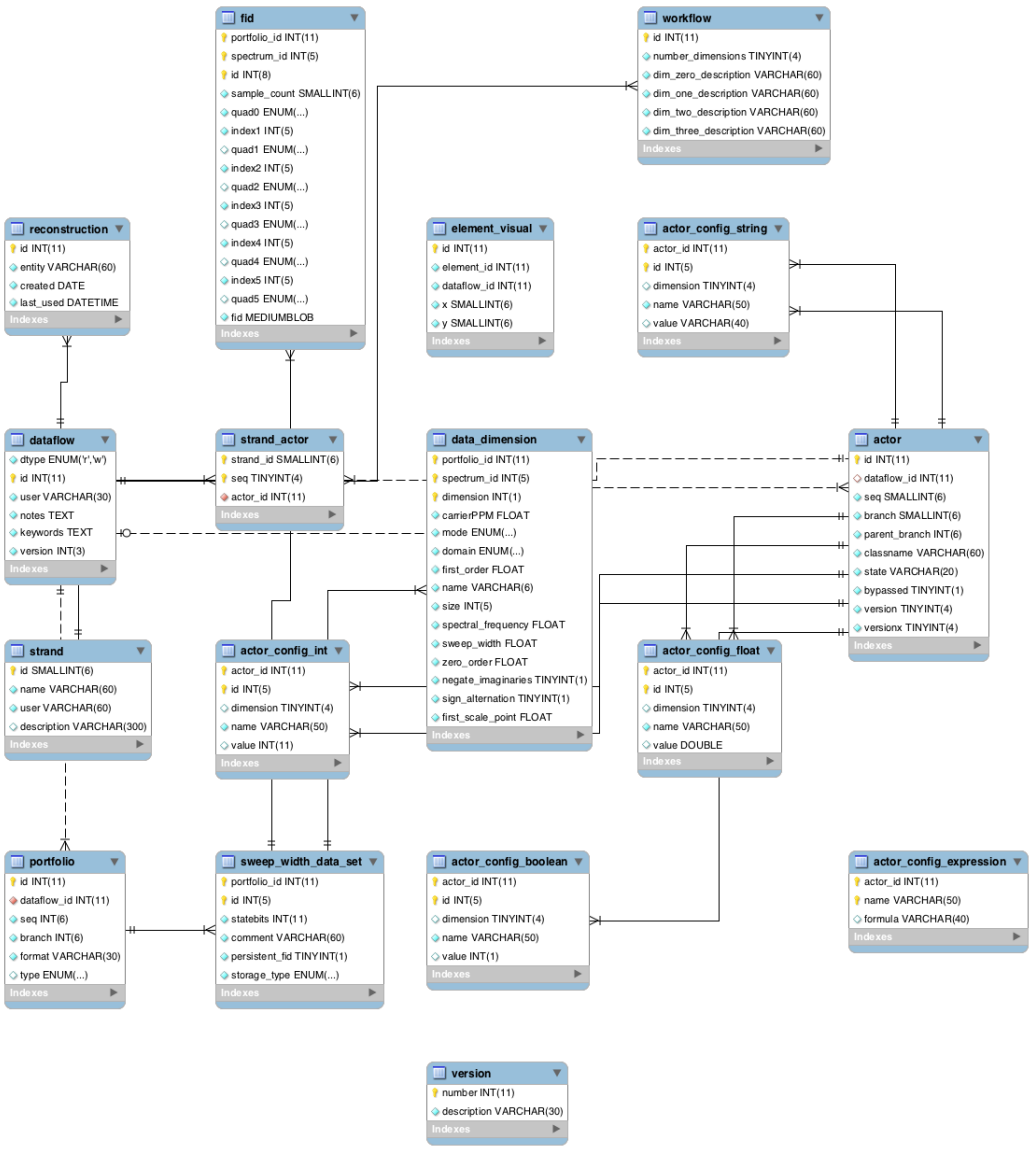

Figure 8.5: WB's data model, created with MySQLWorkbench.

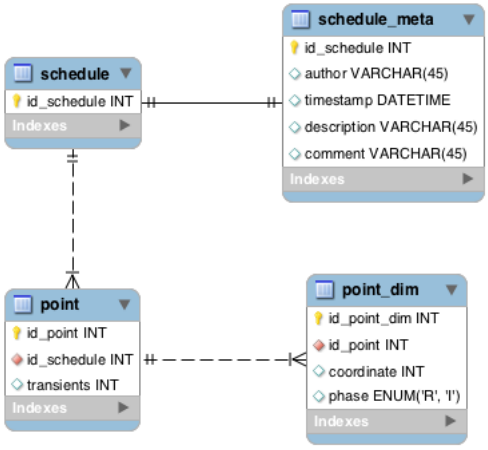

Figure 8.6: A data model of sample schedules, created with MySQLWorkbench.

```
{
    "coordinateGenerator" :     "halton",
    "numGeneratedPoints" :      300,
    "formatter" :               "toolkit",
    "pointSelector" :           "random",
    "numSelectedPoints" :       150,
    "quadratureMapper" :        "singleRandom",
    "forcedSelector" :          "none",
    "postSelectionModifier":    "blurred",
    "seed" :                    45,
    "blurWidth":                2,
    "formatQuadrature":         true,
    "dimensions": [
        {
            "range" :       [0, 128]
        },
        {
            "range" :       [0, 128]
        }
    ]
}
```

Figure 8.7: The parameter file input for a sample schedule.

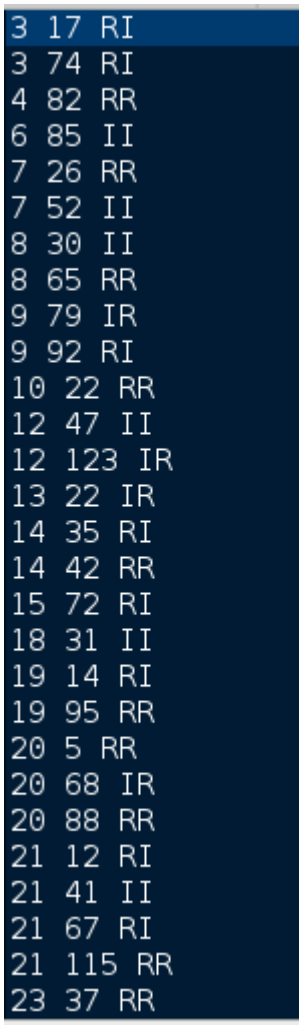

Figure 8.8: A sample schedule generated from the parameter file.

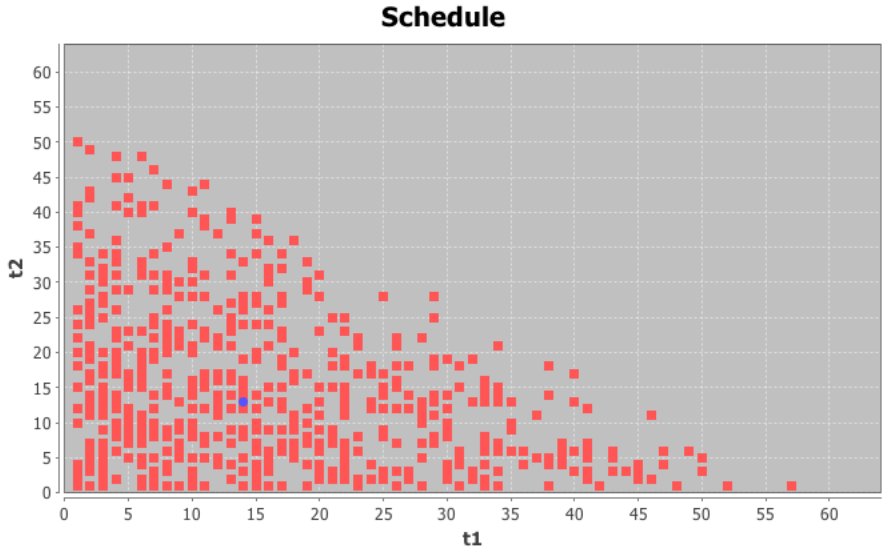

Figure 8.9: A graphical view of a sample schedule.

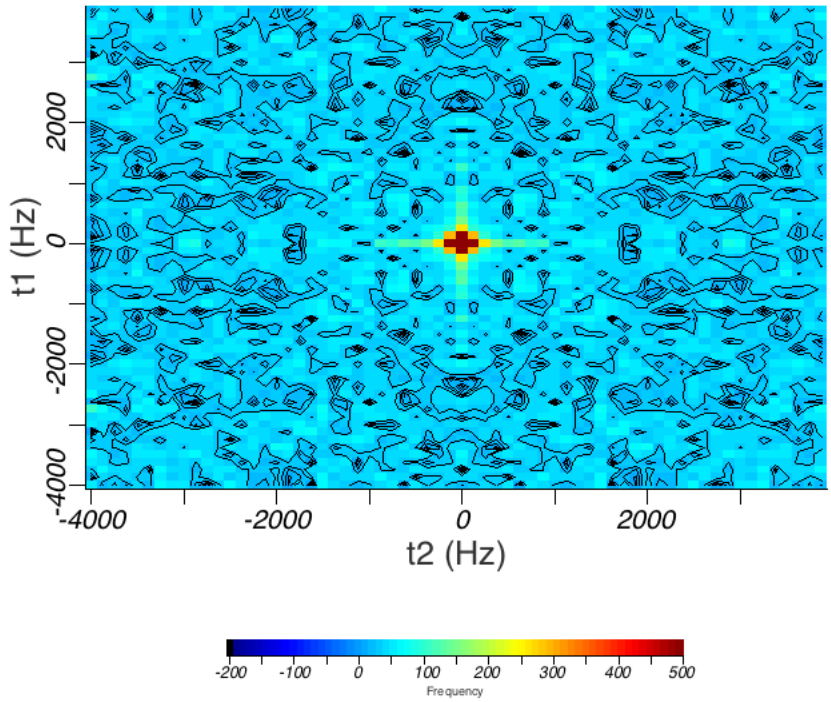

Figure 8.10: A sample schedule for which the first points along each axis are collected.

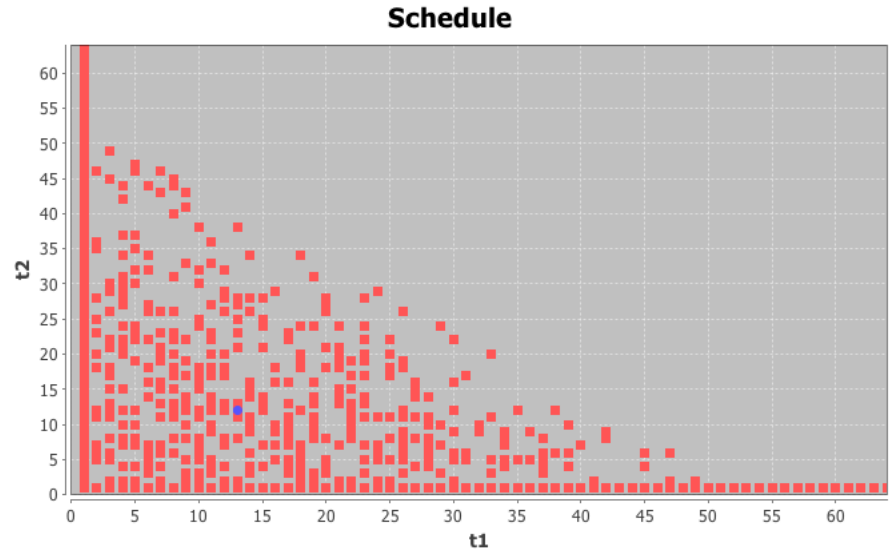

Figure 8.11: The pointspread function of the sample schedule, calculated using a real-only FFT.

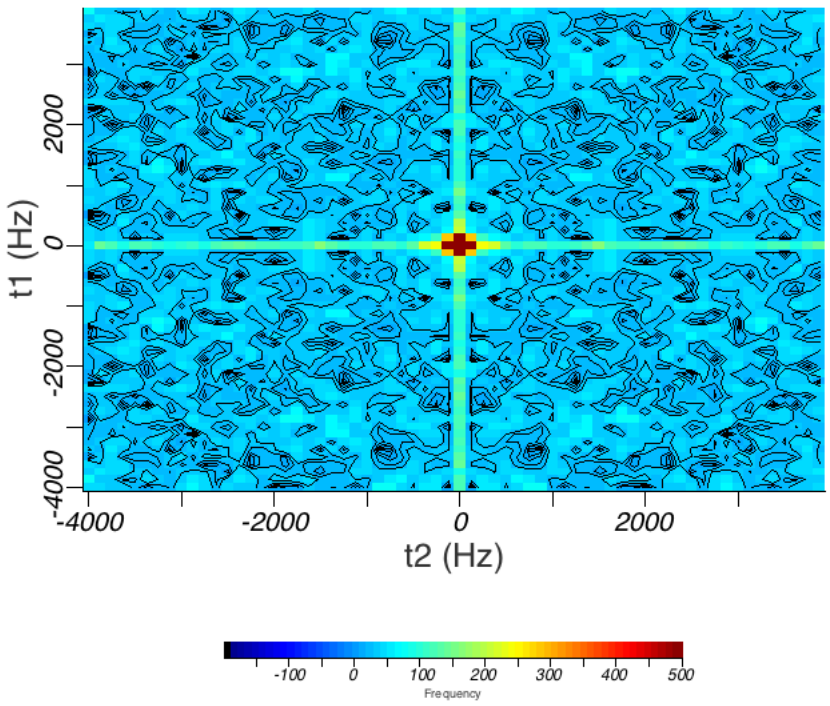

Figure 8.12: The pointspread function shows a noticable difference.

Chapter 9

Conclusions

Dealing with failure is easy: Work hard to improve. Success is also easy to handle: You've solved the wrong problem. Work hard to improve.

- Alan Perlis

9.1 The future of NMR as an experimental technique

NMR is an important technique for studying biological molecules. It is useful for structural, binding, and metabolomics studies. However, to continue to grow in usefulness, several pressing problems must be solved. These problems are varied in nature: some are inherent, and caused by the experimental phenomenon of studying nuclei in large magnets using radiofrequency pulses; others find their roots in our approach to understanding and analyzing the data.

The ability to isolate pure, high-concentration and stable samples of proteins is a prerequisite to carrying out an NMR study. However, this is not only difficult, but may be impossible in certain cases given our current techniques [139]. Worse still is that even for proteins which

can be studied effectively, it is difficult to ascertain how relevant the information obtained is to the protein's actual structure and function in the complex biological system in which we desire to understand its role. Thus, the challenge facing NMR is how to expand its reach to study additional proteins, and to study proteins in native conditions.

Large proteins also pose severe problems for effective NMR data collection. This is because as molecular size increase, peak widths generally increase as well; at the same time, the number of peaks increases because there are more atoms in the molecule. The result is decreased resolution, increased overlap, and a decreased signal-noise ratio. All of these lead to data that is more difficult or impossible to interpret. One current approach is to chop large proteins into several smaller pieces which can be studied independently; the hope is that the results gained are relevant to the full molecule. Progress is also being made in the area of improved pulse sequence design and labeling schemes [140]. Nevertheless, the problem of molecule size remains an ongoing challenge facing the field of NMR.

Effectively passing on knowledge to new students, and efficiently such that they are able to quickly make valuable contributions is a problem of a different sort. While resources such as `http://www.protein-nmr.org.uk/` and books such as [100] and [141] are incredibly valuable and helpful tools for learning, the task of learning NMR not only in breadth and in depth, but also from a practical standpoint, remains a formidable and time-consuming one. In my opinion, a major contributor to this problem is the means employed for transmission of knowledge, information, and data of the analysis of specific proteins and biological systems: a portion of the information is implicitly and transiently communicated, such that the recorded analysis lacks appropriate context. Books, websites, data archives, and journal publications are clearly effective in explicitly transmitting and disseminating

information; just as clearly, these are not intended to provide complete information and data for specific analyses. Thus, the problem facing NMR is how to identify and provide the resources necessary for efficiently and effectively introducing new persons into the field.

Related to previous problem is fostering an understanding and a means of discussing and sharing flaws in our work. I believe the ability to correctly recognize and understand problems is a prerequisite for solving them; and that making progress in the field requires such recognition and solutions. Furthermore, I believe that in a collaborative community in which the work of one individual becomes the basis for the work of another, such as the current scientific community in which we work, explicitly identifying problems and open issues is as important as making new discoveries; the cycle of first identifying, and then solving problems is natural to science. Therefore, the field of NMR must determine how to improve its communication of flaws and holes in our research, in order to facilitate the solution of them.

9.2 Reproducibility: challenge, opportunity

The major contribution of this work is substantial progress towards solving the latter two problems by means of reproducibility. I have presented approaches, models, and applications and techniques of those, all aimed toward two goals: first, to make information and its context explicit, by recording and preserving them; and second, to create a concept, means, and vocabulary for identifying and communicating issues of NMR data analysis. While these accomplishments only directly help to solve the latter two problems, I believe they will also help to deal with the former two as well – reproducibility leads to improved and robust data analysis, which is a prerequisite for dealing with the lower quality data collected from large

or unstable proteins.

9.3 The future of NMR software

Software is integrated into every phase of the NMR analysis process. Good software facilitates progress, while bad software restricts it, and so software plays a major role in determining the quality, speed, and robustness of NMR analysis. Two measures of software quality are how well a software product meets its users' current needs, and how well it is able to adapt to meet their future needs. Easily adapting to meet new challenges enables fast and cheap innovation; conversely, high barriers to adapting and building software restrict innovation and prevent improvements. While NMR software has achieved incredible results, I believe it is reaching a crossroads with respect to its adaptability.

If both flexibility and adaptability are to be achieved, a new approach to building scientific software must be adopted and a reevaluation of the costs and value of software to NMR must be made in order to capitalize on software's potential. These changes are fundamentally different from rewriting software to be more efficient, or have fewer bugs or more features. Rather, the goals must be to create software such that it is simple, obvious, and composable. Such design goals naturally lead to correct, feature-rich, adaptable, and maintainable software – but the converse is not true. Worse is that complex, non-composable software, instead of enabling us, places an upper limit on the problems that can be solved using it.

CONNJUR is an open source project which provides flexible, composable tools through data integration. While it is not expected to solve every software problem NMR spectroscopists face, I believe the software created by the CONNJUR project embodies the principles needed to create effective scientific software, through its bottom-up design and open source

licensing: the first is the best way we know of for dealing flexibly with changing requirements, and the second is the best way we know of to ensure the continuous development and availability of valuable scientific software projects.

9.4 Final thoughts

In this current climate of decreased scientific funding levels, and increased competition for precious grant dollars, it is important to consider the effect of science on society, and the effect of society on science. While good science can have a positive impact on individuals and on society as a whole, bad science can have a corresponding negative impact. I therefore hope that the goal of reproducibility is taken seriously by all scientists, not only as a means of expanding human knowledge more quickly and with less effort, but also a means to minimize the harm of bad science. I look forward to science of the highest quality.

Appendices

Appendix A

List of Publications

A.1 Reproducible protein NMR data analysis (in progress)

This paper will cover the definition of reproducibility, the approach to applying reproducibility to NMR data analysis, the library of deductive reasons used for annotating NMR analysis, the Sparky extension which facilitates practical reproducibility, the extensions to the NMR-STAR data dictionary to enable archiving and dissemination of reproducible data sets, and a reproducible data set of a protein analyzed using the Sparky extension according to the above approach, with the final data set in NMR-STAR format. This is the subject of Chapters 1 through 7.

A.2 CONNJUR Workflow Builder (in progress)

This paper [50] will cover the design, applications, and context of our new software product, WB, which is used for spectral reconstruction. This product is novel because it facilitates reproducible spectral reconstruction, meta data correctness, and is designed to seamlessly

deal with non-trivial workflows including branching and iteration, while interfacing with a relational database to provide centralized data storage. In addition, it is a valuable learning tool. This is covered in Chapters 3, 4, and 8.

A.3 A bioinformatics sandbox

In order to meet the growing need for scientific software that is simultaneously robust, flexible, and maintainable, this paper [142] applies an alternative approach, Functional Programming, to developing scientific software. This approach offers several potential benefits which are explored using several small software projects. This paper also presents a public repository for programmers and bioinformaticians seeking to apply Functional Programming to biologically relevant problems; the repository serves as a means to learn about Functional Programming and to share interesting and useful code with other bioinformaticians. This is covered in Chapter 8.

A.4 Accessing archived NMR data

This follow-up paper to the sandbox paper applies Functional Programming to create [45] an application capable of reading data directly from standard NMR archives. This problem is difficult to solve, such that most existing solutions are incomplete and difficult to use. The solution presented in this paper is complete and simple, in part due to the Functional Programming concepts applied when implementing the solution. This is covered in Chapters 6, 7, and 8.

A.5 Random phase detection

While non-uniform sampling in time increments had previously been studied, this paper [51] presents the additional concept and characterization of non-uniformly sampling the phases as well. This offers potential reductions for spectrometer usage without sacrificing data quality. This is covered in Chapters 3, 4, and 8.

A.6 Software architecture for effective NMR data processing

This paper [143] explores the architecture and design of CONNJUR software and how it contributes to the creation of a robust and powerful framework for describing and solving NMR problems. This is covered in Chapter 8.

A.7 CONNJUR Spectrum Translator

This paper [49] describes the design and use of a tool for converting between several formats for binary time- and frequency-domain NMR data. ST provides a novel architecture for reducing the amount of work required to translate between formats. This is covered in Chapters 3, 4, and 8.

Appendix B

Library of Deductive Reasons

This appendix presents commonly used deductive reasons during the process of NMR data analysis. It is grouped by the datatype being deduced, and then broken down into the data used to make that deduction; each deduction is followed by an explanation of its meaning and applicability.

Pick new peak

Local extremum of spectral intensity

Peaks can be naively picked by identifying local maxima and minima in spectra. This may be done by a software tool or manually.

Chemical shift matching

One or more peaks assigned to a GSS may be used to pick a new peak, based on matching one or more chemical shifts of the existing peaks as well as spectral features. For instance, a

peak in an NHSQC spectrum may be used to locate peaks in an HNCACB spectrum using the hydrogen and nitrogen chemical shifts.

Initialize new GSSs

NHSQC peak

Use NHSQC signal peaks to initialize GSSs. This is a convenient choice due to the many through-bond experiments which build upon the NHSQC's H-N coupling.

Peak classification as signal, noise, or artifact

Peak intensity + noise level

Signal peaks generally have high intensities and noise peaks have low ones.

Peak position + pulse sequence

True signals are expected in limited spectral regions. Projections of true signal peaks along spectral edges may indicate artifacts.

Peak sign + pulse sequence

In experiments where all true peaks are expected to have the same sign, such as an NHSQC, a peak of the opposite sign may be a signal or an artifact.

Chemical shift matching + peak-GSS + pulse sequence

A specific number of peaks are expected for a GSS in a given spectrum. Extra peaks matching a GSS may be artifacts or noise.

Peak pattern

Phase errors and truncations give rise to characteristic dispersive lineshapes and sinc wiggles, respectively. When using a naive peak picker, these may be picked as a series of smaller peaks radiating out from a more intense, central peak.

Chemical shift matching + pulse sequence + lack of peak

A potential signal peak may be recognized as noise or artifactual based on missing of matching peaks in another or the same spectra that would be expected if it were a true peak.

Assign peak cross section to resonance

Peak cross section lineshape

Peak cross sections from the same nucleus may have similar lineshapes; if they are grouped into the same GSS, the matching cross sections can be assigned to the same resonance.

Chemical shift matching

The assignment of a peak cross section to a resonance can be used to assign another peak cross section from the same GSS to the same resonance, if the chemical shifts match.

Assign peak to GSS

Peak-GSS: disambiguate peak-GSS assignment

When peaks overlap, it is difficult to assign them correctly to GSSs. However, the resonances may be resolvable in additional spectra, which then makes it possible to return to the first spectrum and correctly resolve its overlap.

Chemical shift matching

Based on matching of corresponding peak cross sections, peaks are combined into GSSs. The matching of peaks may be within a single spectrum, or between multiple spectra – as long as the spectral dimensions match (both nucleus and resonance type). The tolerances allowed are important in determining which peaks match.

GSS typing

Peak pattern

In the C(CO)NH-TOCSY and HC(CO)NH-TOCSY, GSSs of specific types consistently exhibit characteristic patterns of peak intensity and sign. Observation of a characteristic pattern can be used to deduce the type of a previously untyped GSS.

BMRB statistics: sidechain Trp, Asn/Gln, Arg

Sidechain tryptophan GSSs exhibit a characteristic peak pattern with characteristic chemical shifts, which may be found in the BMRB. Asparagine and glutamine produce characteristic peak patterns in H-N-based experiments, due to their two sidechain protons. These resonances

consistently appear in the same spectral region, making them easy to identify. Arginine sidechain GSSs resonate at characteristic chemical shifts, and may even be subject to splitting if they appear sufficiently far outside of the decoupling band.

Resonance typing

The typings of the resonances of a GSS constrain the GSS typing. For example, a GSS with a CB(i) resonance cannot be assigned to glycine.

BMRB statistics: backbone Ala, Gly, Ser/Thr

Alanine's CB resonates at a characteristic chemical shift relative to other CB's due to its lack of additional carbon nuclei. Glycine's CA resonates at a characteristic chemical shift relative to other CA's, and also lacks a CB nucleus. In experiments such as an HNCACB or C(CO)NH-TOCSY, glycine strips appear without a CB peak. Serine's and threonine's CB resonates at a characteristic chemical shift relative to other CB's due to the -OH groups.

Assign sequential GSS chain

Chemical shift matching: GSS-GSS and resonance typing

Matching carbon strips from 3-dimensional experiments are used to build sequential GSS assignments and resolve some resonance types simultaneously. For example, given two HNCACB strips, chemical shift matching, and relative intensities, the sequential GSS assignments and CA(i)/CA(i-1) and CB(i)/CB(i-1) resonance typing assignments can be made, such that the following conditions are satisfied (note that the last two are not inviolable):

- i-1 peaks in following ss should be matched by i peaks in preceding ss

- intensity of HNCACB i-1 peaks should usually be less than intensity of i peaks in same ss
- intensity of HNCACB i-1 peaks should usually be less than intensity of matching i peaks in preceding ss

Resonance matching

If the resonances are already typed (perhaps with the help of an auxiliary experiment such as a C(CO)NH-TOCSY or HN(CO)CACB in addition to the HNCACB), sequential GSS assignment can be accomplished by matching resonances with appropriate nucleus type assignments: for example, a GSS with a CA(i) and CB(i) resonance whose chemical shifts match the CA(i-1) and CB(i-1) resonance frequencies can be deduced to form a sequential chain.

Sequence-specific GSS-residue assignment

GSS typing + primary sequence

A GSS chain can be assigned to residues based on the amino acid types of the residues and their match to the GSS typings. It is not necessary that every single GSS is unambiguously typed, merely that each GSS typing is consistent with the primary sequence.

GSS typing + primary sequence + GSS-residue

This is similar to the previous rule, but takes previous sequence-specific assignments into account using the process of elimination: for example, if a GSS is typed as an arginine, and

there is only one unassigned arginine residue remaining, the GSS may correspond to the arginine.

Extend GSS-residue fragment

A GSS chain already assigned to specific residues may be extended at the ends. In addition to the constraints rules given in "chemical shift matching: GSS-GSS and resonance typing", GSS typings must match the residue typing, or the BMRB statistics if untyped.

Resonance typing

Peak sign + pulse sequence

In experiments such as the HNCACB, the peak sign can contain information about the resonance typing in the ^{13}C axis: CA resonances from backbone GSSs have opposite signs from the CB resonances.

Resonance typing

By process of elimination, if one resonance is typed as a specific nucleus, another resonance from the same GSS can not be assigned the same type.

Peak sign + GSS type

The GSS type and peak sign can be used in conjunction with certain pulse sequences to determine resonance types. For example, in an HNCACB spectrum, the positive peak of a glutamine sidechain is a CG and the negative is a CB (or vice versa).

Sequential GSS assignment

Resonance typing may be done during sequential GSS assignments, as covered in section B.

Characteristic splitting pattern

Methylene groups, such as CB/HB2/HB3 in many amino acids, often show splitting in the ^{1}H axis due to the two non-equivalent protons. When such a splitting pattern is observed, assignment to a methyl group or to a group with a single hydrogen attached to a carbon can both be ruled out.

Pulse sequence

The pulse sequence places constraints on the possible resonance typings of the cross sections of each peak: there is a limited list of spin systems which the pulse sequence is capable of capturing; each peak from the spectrum must have resonance typings corresponding to one of the possible spin systems. For example, hbCBcgcdHD experiment targets aromatics; a peak from that spectrum that matches a GSS not typed as an aromatic, can not be assigned to that GSS.

GSS typing

A GSS's assigned type places constraints on the types of its resonances. For example, a resonance in a glycine GSS cannot be assigned to CB(i).

BMRB statistics

Resonances can often be assigned a nucleus type unambiguously using BMRB statistics in conjunction with the spectra in which a resonance appears. Process of elimination may also be employed.

BMRB statistics

The BMRB average chemical shift stats in conjunction with the GSS typing are useful for spectra such as the C(CO)NH-TOCSY, HBHA(CO)NH, and HC(CO)NH-TOCSY, where there are multiple peaks in a strip along a carbon or hydrogen dimension. Most amino acid types feature good dispersion, making it easy to get the correct assignments directly from the statistics.

TOCSY aliphatic sidechain

After assigning HNCACB and HBHA(CO)NH spectra, C(CO)NH-TOCSY, HC(CO)NH-TOCSY and HCCH-TOCSY are used in conjunction: the C(CO)NH-TOCSY peaks are used to find HCCH-TOCSY strips, which yields proton chemical shifts and match the HC(CO)NH-TOCSY peaks. Also, each HCCH-TOCSY strip should have peaks in all the same ^{1}H shifts. BMRB statistics can be used to assign most peaktypes unambiguously. Splitting patterns also help to identify methylene groups.

TOCSY peak pattern

Several aliphatic C/H groups are difficult to distinguish using BMRB statistics, including Leucine's CG, CD1, and CD2, as well as Isoleucine's CG2 and CD1, and its QG2 and QD1.

These can often be resolved by characteristic intensity patterns for each amino acid type due to the TOCSY nature of experiment; also, peaks from methyl gruops are often sharper and more intense.

CYANA: stereospecific resonance typing

Many pairs of nuclei give rise to two peaks which can not be immediately assigned unambiguously, although it is known that each nucleus corresponds to one of the peaks, and the other nucleus to the other peak. Examples include HB2/HB3 of Y, and QD1/QD2 of L. For additional examples of ambiguities see 3.11. CYANA can often resolve these ambiguities during structure calculation.

Appendix C

Assignment: Considerations, Notations, Strategy

This appendix presents several strategies and tools for understanding and applying the concepts of resonances and GSSs during chemical shift assignment. These were all developed during the course of analysis of the Samp3 data set, and were concurrently applied in order to facilitate correct analysis.

C.1 Typing of H-N-rooted GSSs

Type assignment of GSSs is an important intermediate for obtaining sequence-specific GSS assignments. There are several categories of H-N-rooted GSSs. Backbone GSSs are rooted in the H-N of the peptide backbone. Sidechain GSSs are rooted in H-N groups of amino acid sidechains. The backbone ones may span multiple amino acid residues; the typing of the corresponding residues may be known partially or fully for each piece.

A simple system is presented in Figure C.1 for representing partial and full GSS typing. First, the typing of a portion of a GSS may be ambiguous or unambiguous. If it is unambiguous, then a single type is assigned to that portion. If it is ambiguous, then multiple types are assigned to that portion. An ambiguous assignment may be one of several, such as either a serine or a threonine; it may be one of many, such as any backbone type; or it may be any H-N-rooted GSS type. A sequential assignment, which only applies to backbone GSSs as they are able to span multiple amino acids, enables independent typing of each amino acid spanned by the GSS.

C.2 Graph patterns of pulse sequences

Modeling pulse sequences as patterns which act upon graphs is simplistic but useful. The graphs are molecules, the nodes are nuclei, and scalar and dipolar couplings are edges. Wherever a pulse sequence pattern matches a molecular graph, signals are expected in the corresponding spectra.

The model separates nuclei into groups based on atomic number and chemical shift range, yielding protons, nitrogens, aliphatic carbon, aromatic carbon, and carbonyl carbon (see Table C.2). Nuclei are connected by J-couplings or through-space couplings, indicating transfer of magnetization between nuclei. Finally, a differentiation is made between nuclei whose chemical shift is and is not recorded. See Tables C.2 and C.3.

This approach is similar to that applied in [144]; a major difference is that NMR-specific details such as specific J-coupling values, decoupling pulses, shaped pulses, and transfer efficiencies have all been omitted. These were omitted in order to keep the model simple and easy to use, so that it is possible to quickly determine which nuclei and spin systems are

expected to appear in an NMR experiment.

C.3 Partial ambiguities in resonance typing

Resonance typing interacts with GSS typing, as the GSS type(s) determine which nuclei may possibly be present in a GSS. However, it is still possible to obtain partial resonance typings – depending on the pulse sequences used – without unambiguously typing the containing GSS. Two cases where such information is obtainable are an HNCACB, which distinguishes between the "CA" and "CB" nuclei by means of a 180-degree phase shift, and experiment pairs such as the HNCACB and HN(CO)CACB, which allows for distinguishing between CA and CA(i-1), and CB and CB(i-1) based on peaks which do and do not appear at the same chemical shift in both spectra. Figures C.1, C.2, C.3, C.4, C.5 and C.6 present a scheme for capturing these partial resonance typings based on the key given in Table C.4.

C.4 Tables

GSS typing	Shorthand notation
Unknown	?
Single backbone	A,C,D,E,F,G,H,I,K,L,M,N,Q,R,S,T,V,W,Y
Single backbone (i-1)	A,C,D,E,F,G,H,I,K,L,M,N,P,Q,R,S,T,V,W,Y
Ambiguous backbone	b
Single sidechain	sQ, sN, sW, sR
Backbone S/T	S/T
Sidechain Q/N	sQ/sN
Sequential	b-b, I-S, P/R-G

Table C.1: GSS types used in typing of H-N-rooted GSSs. In addition to the basic GSS types from standard H-N groups present in protein backbones and amino acid sidechains, fully and partially ambiguous GSS types are useful. GSSs that span multiple amino acids are given a separate type for each amino acid.

Symbol	Explanation
Hn	Any H
Nn	Any N
Ca	Aliphatic C
Co	Carbonyl C
Cr	Aromatic C
–	J-coupling
<–>	Through-space coupling
[...]	Chemical shift not collected
...*	Zero or more
...+	One or more

Table C.2: Key used to model various common pulse sequences as patterns on graphs. Nuclei form the nodes of the graph, and J-couplings form the edges.

NHSQC	Hn – Nn
HNCO	Hn – Nn – Co
HN(CA)CO	Hn – Nn – [Ca] – Co
HNCA	Hn – Nn – Ca
HNCACB	Hn – Nn – Ca1
HNCACB	Hn – Nn – [Ca1] – Ca2
HN(CO)CACB	Hn – Nn – [Co] – Ca1
HN(CO)CACB	Hn – Nn – [Co] – [Ca1] – Ca2
C(CO)NH-TOCSY	Hn – Nn – [Co] – [Ca]* – Ca
H(CCO)NH-TOCSY	Hn – Nn – [Co] – [Ca]+ – Hn
HBHA(CO)NH	Hn – Nn – [Co] – [Ca] – Hn
HBHA(CO)NH	Hn – Nn – [Co] – [Ca] – [Ca] – Hn
HCCH-TOCSY	Hn – Ca – [Ca]* – Hn
TOCSY-NHSQC	Hn – Nn – [Ca]+ – Hn
hbCBcgcdHD	[Hn] – Ca – [Cr]2 – Hn
hbCBcgcdceHE	[Hn] – Ca – [Cr]2,3 – Hn
CHSQC	Hn – C
NOESY-NHSQC	Hn <–> Hn – Nn
NOESY-CHSQC	Hn <–> Hn – C

Table C.3: Common pulse sequences and their graph patterns using the key given in Table C.2.

Nuclei	Together	Separate
HA2/HA3	HA*	HA*1, HA*2
HB2/HB3	HB*	HB*1, HB*2
HE21/HE22	HE2*	HE2*1, HE2*2
QG1/QG2	QG*	QG*1, QG*2
Ca11/Ca12	Ca1*	Ca1*1, Ca1*2
Ca11/Ca21	Ca*1	Ca*11, Ca*21

Table C.4: Notation for partial resonance typings. In general, each group includes multiple nuclei which may be ambiguous depending on the pulse sequences used and the state of the assignment of the full data set. For stereospecifically ambiguous nuclei, the names are formed by replacing the number with a * to signify all nuclei, and appending a unique index to signify one but not all of the nuclei.

C.5 Figures

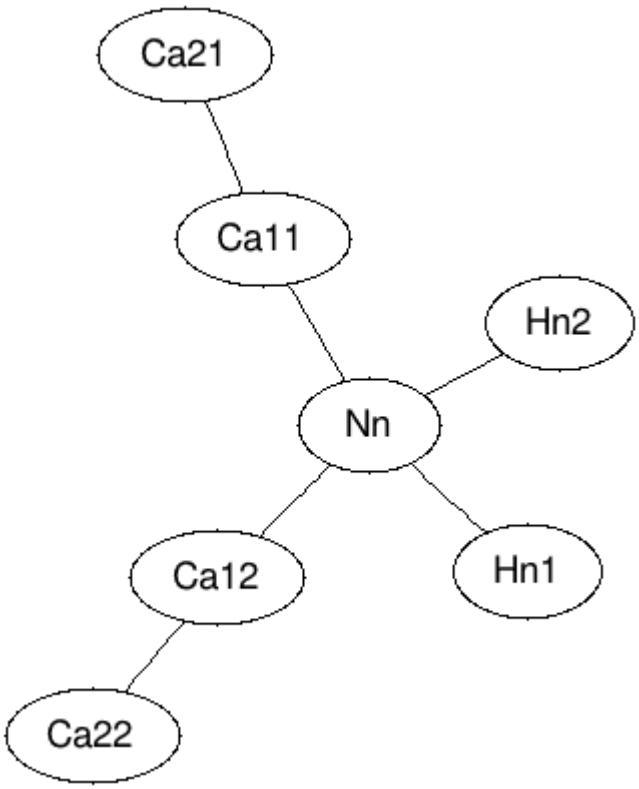

Figure C.1: The portion of a GSS observable from an HNCACB experiment. The names of the nuclei reflect their relationships: Ca11 and Ca21 are distinguishable from Ca12 and Ca22 by a 180-degree phase shift, while Ca11 and Ca12 form a distinct covalent group from Ca21 and Ca22.

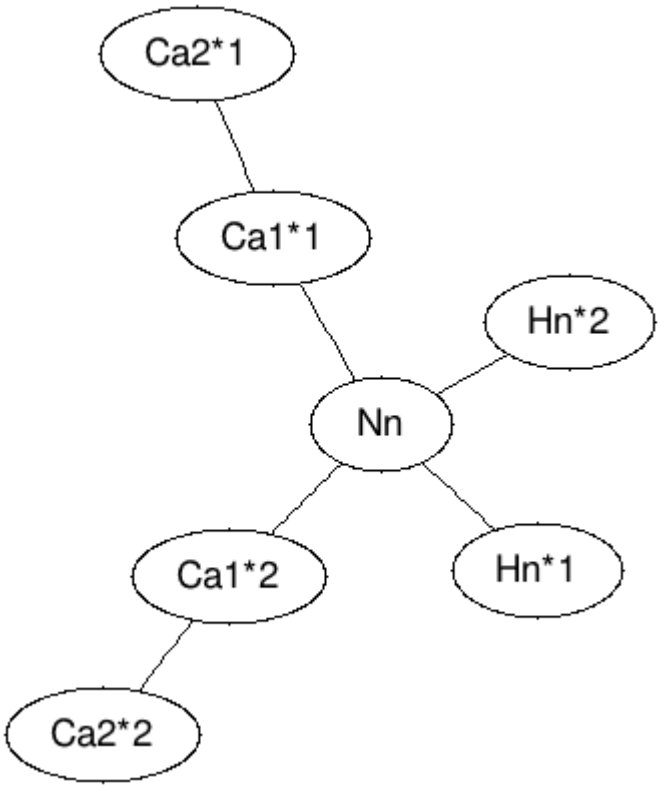

Figure C.2: An alternative naming scheme which accounts for ambiguity for GSSs observed through an HNCACB. Distinct, placeholder names are used to indicate the important relationships between resonances: the number of bonds distance of the aliphatic carbons from the nitrogen.

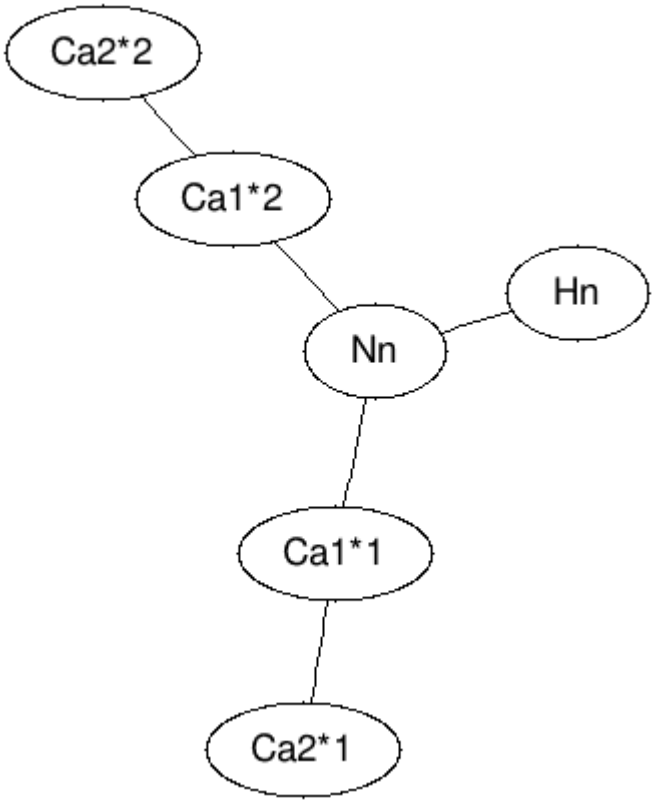

Figure C.3: A backbone GSS in HNCACB. Note that the Ca11/Ca21 and Ca12/Ca22 pairs may be ambiguous.

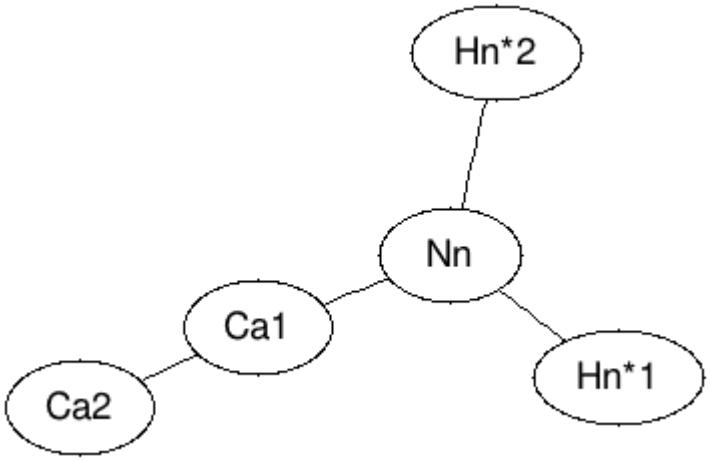

Figure C.4: The extent of a sidechain Q or N GSS in an HNCACB. Note that there are two protons, which may be ambiguously assigned.

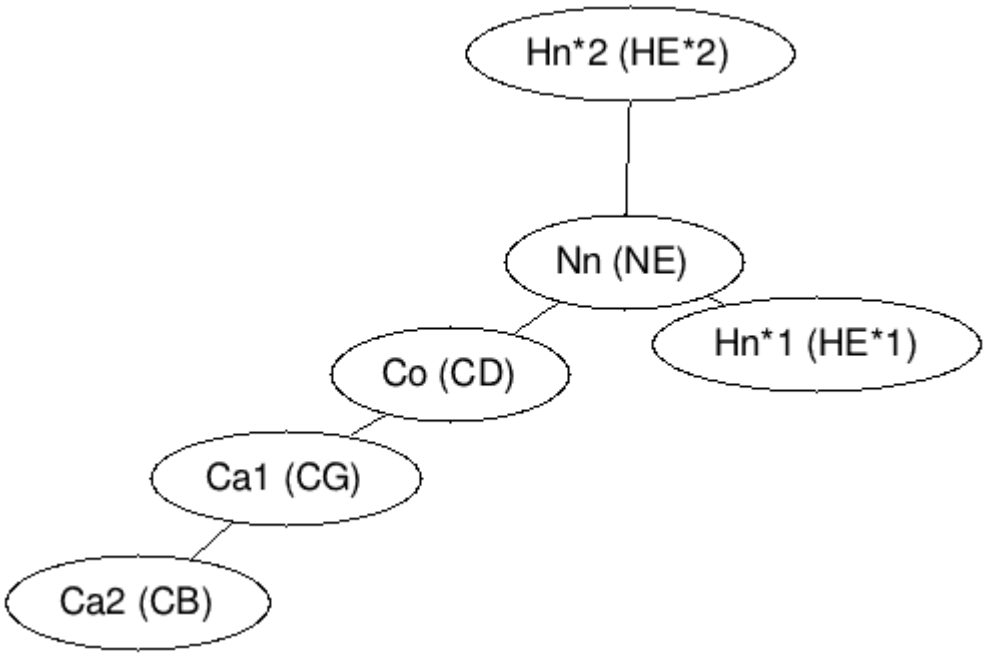

Figure C.5: A GSS in an HNCACB, assigned to a Q sidechain. The nitrogen and carbon resonances have been unambiguously typed, while the proton resonances are ambiguous.

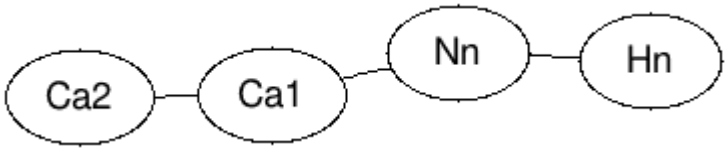

Figure C.6: A sidechain R GSS in an HNCACB. The resonances have not yet been typed.

Bibliography

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[2] E. L. Ulrich, H. Akutsu, J. F. Doreleijers, Y. Harano, Y. E. Ioannidis, J. Lin, M. Livny, S. Mading, D. Maziuk, Z. Miller, *et al.*, "BioMagResBank," *Nucleic acids research*, vol. 36, no. suppl 1, pp. D402–D408, 2008.

[3] B. J. Stockman and C. Dalvit, "NMR screening techniques in drug discovery and drug design," *Progress in Nuclear Magnetic Resonance Spectroscopy*, vol. 41, no. 3, pp. 187–231, 2002.

[4] J. Moore, N. Abdul-Manan, J. Fejzo, M. Jacobs, C. Lepre, J. Peng, and X. Xie, "Leveraging structural approaches: applications of NMR-based screening and X-ray crystallography for inhibitor design," *Journal of synchrotron radiation*, vol. 11, no. 1, pp. 97–100, 2003.

[5] S. Reckel, D. Gottstein, J. Stehle, F. Löhr, M.-K. Verhoefen, M. Takeda, R. Silvers, M. Kainosho, C. Glaubitz, J. Wachtveitl, *et al.*, "Solution NMR structure of prote-

orhodopsin," *Angewandte Chemie International Edition*, vol. 50, no. 50, pp. 11942–11946, 2011.

[6] C. Drummond, "Reproducible Research: a Dissenting Opinion," 2012.

[7] J. F. Russell, "If a job is worth doing, it is worth doing twice," *Nature*, vol. 496, no. 7443, pp. 7–7, 2013.

[8] F. S. Collins and L. A. Tabak, "Policy: NIH plans to enhance reproducibility.," *Nature*, vol. 505, no. 7485, pp. 612–613, 2014.

[9] M. Rubacha, A. K. Rattan, and S. C. Hosselet, "A review of electronic laboratory notebooks available in the market today," *Journal of the Association for Laboratory Automation*, vol. 16, no. 1, pp. 90–98, 2011.

[10] T. Talbott, M. Peterson, J. Schwidder, and J. D. Myers, "Adapting the electronic laboratory notebook for the semantic era," in *Collaborative Technologies and Systems, 2005. Proceedings of the 2005 International Symposium on*, pp. 136–143, IEEE, 2005.

[11] J. D. Myers, E. S. Mendoza, and B. Hoopes, "A Collaborative Electronic Laboratory Notebook.," in *IMSA*, pp. 334–338, 2001.

[12] M. R. Macleod, M. Fisher, V. O'Collins, E. S. Sena, U. Dirnagl, P. M. Bath, A. Buchan, H. B. van der Worp, R. J. Traystman, K. Minematsu, *et al.*, "Reprint: Good laboratory practice: preventing introduction of bias at the bench," *Journal of Cerebral Blood Flow & Metabolism*, vol. 29, no. 2, pp. 221–223, 2008.

[13] C. Unger, H. Skottman, P. Blomberg, M. S. Dilber, and O. Hovatta, "Good manufacturing practice and clinical-grade human embryonic stem cell lines," *Human molecular genetics*, vol. 17, no. R1, pp. R48–R53, 2008.

[14] J. Savović, H. E. Jones, D. G. Altman, R. J. Harris, P. Jüni, J. Pildal, B. Als-Nielsen, E. M. Balk, C. Gluud, L. L. Gluud, *et al.*, "Influence of reported study design characteristics on intervention effect estimates from randomized, controlled trials," *Annals of internal medicine*, vol. 157, no. 6, pp. 429–438, 2012.

[15] J. P. Simmons, L. D. Nelson, and U. Simonsohn, "False-positive psychology undisclosed flexibility in data collection and analysis allows presenting anything as significant," *Psychological science*, vol. 22, no. 11, pp. 1359–1366, 2011.

[16] D. L. Donoho, A. Maleki, I. U. Rahman, M. Shahram, and V. Stodden, "Reproducible research in computational harmonic analysis," *Computing in Science & Engineering*, vol. 11, no. 1, pp. 8–18, 2009.

[17] R. D. Peng, "Reproducible research in computational science," *Science (New York, Ny)*, vol. 334, no. 6060, p. 1226, 2011.

[18] J. B. Buckheit and D. L. Donoho, *Wavelab and reproducible research*. Springer, 1995.

[19] D. C. Ince, L. Hatton, and J. Graham-Cumming, "The case for open computer programs," *Nature*, vol. 482, no. 7386, pp. 485–488, 2012.

[20] A. Nekrutenko and J. Taylor, "Next-generation sequencing data interpretation: enhancing reproducibility and accessibility," *Nature Reviews Genetics*, vol. 13, no. 9, pp. 667–672, 2012.

[21] D. Blankenberg, G. Von Kuster, E. Bouvier, D. Baker, E. Afgan, N. Stoler, J. Taylor, A. Nekrutenko, *et al.*, "Dissemination of scientific software with Galaxy ToolShed," *Genome Biology*, vol. 15, no. 2, p. 403, 2014.

[22] S. C. Landis, S. G. Amara, K. Asadullah, C. P. Austin, R. Blumenstein, E. W. Bradley, R. G. Crystal, R. B. Darnell, R. J. Ferrante, H. Fillit, *et al.*, "A call for transparent reporting to optimize the predictive value of preclinical research," *Nature*, vol. 490, no. 7419, pp. 187–191, 2012.

[23] D. L. Sackett, "Bias in analytic research," *Journal of chronic diseases*, vol. 32, no. 1, pp. 51–63, 1979.

[24] J. P. Ioannidis, "Why most published research findings are false," *PLoS medicine*, vol. 2, no. 8, p. e124, 2005.

[25] R. Nuzzo, "STATISTICAL ERRORS," 2014.

[26] C. G. Begley, "Reproducibility: six red flags for suspect work," *Nature*, vol. 497, no. 7450, pp. 433–434, 2013.

[27] H. Pashler and C. R. Harris, "Is the replicability crisis overblown? Three arguments examined," *Perspectives on Psychological Science*, vol. 7, no. 6, pp. 531–536, 2012.

[28] D. L. Vaux, "Research methods: Know when your numbers are significant," *Nature*, vol. 492, no. 7428, pp. 180–181, 2012.

[29] D. MacArthur, "Methods: Face up to false positives," *Nature*, vol. 487, no. 7408, pp. 427–428, 2012.

[30] E.-J. Wagenmakers, R. Wetzels, D. Borsboom, H. L. van der Maas, and R. A. Kievit, "An agenda for purely confirmatory research," *Perspectives on Psychological Science*, vol. 7, no. 6, pp. 632–638, 2012.

[31] C. L. Borgman, "The conundrum of sharing research data," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 6, pp. 1059–1078, 2012.

[32] J. Rung and A. Brazma, "Reuse of public genome-wide gene expression data," *Nature Reviews Genetics*, vol. 14, no. 2, pp. 89–99, 2013.

[33] A. Mullard, "Reliability of'new drug target'claims called into question," *Nature Reviews Drug Discovery*, vol. 10, no. 9, pp. 643–644, 2011.

[34] F. Prinz, T. Schlange, and K. Asadullah, "Believe it or not: how much can we rely on published data on potential drug targets?," *Nature reviews Drug discovery*, vol. 10, no. 9, pp. 712–712, 2011.

[35] C. G. Begley and L. M. Ellis, "Drug development: Raise standards for preclinical cancer research," *Nature*, vol. 483, no. 7391, pp. 531–533, 2012.

[36] J. F. Doreleijers, W. F. Vranken, C. Schulte, J. L. Markley, E. L. Ulrich, G. Vriend, and G. W. Vuister, "NRG-CING: integrated validation reports of remediated experimental biomolecular NMR data and coordinates in wwPDB," *Nucleic acids research*, vol. 40, no. D1, pp. D519–D524, 2012.

[37] R. A. Laskowski, J. A. C. Rullmann, M. W. MacArthur, R. Kaptein, and J. M. Thornton, "AQUA and PROCHECK-NMR: programs for checking the quality of protein structures solved by NMR," *Journal of biomolecular NMR*, vol. 8, no. 4, pp. 477–486, 1996.

[38] A. Bhattacharya, R. Tejero, and G. T. Montelione, "Evaluating protein structures determined by structural genomics consortia," *Proteins: Structure, Function, and Bioinformatics*, vol. 66, no. 4, pp. 778–795, 2007.

[39] S. Neal, A. M. Nip, H. Zhang, and D. S. Wishart, "Rapid and accurate calculation of protein 1H, 13C and 15N chemical shifts," *Journal of biomolecular NMR*, vol. 26, no. 3, pp. 215–240, 2003.

[40] B. Han, Y. Liu, S. W. Ginzinger, and D. S. Wishart, "SHIFTX2: significantly improved protein chemical shift prediction," *Journal of biomolecular NMR*, vol. 50, no. 1, pp. 43–57, 2011.

[41] J. F. Doreleijers, J. A. Rullmann, and R. Kaptein, "Quality assessment of NMR structures: a statistical survey," *Journal of molecular biology*, vol. 281, no. 1, pp. 149–164, 1998.

[42] A. J. Nederveen, J. F. Doreleijers, W. Vranken, Z. Miller, C. A. Spronk, S. B. Nabuurs, P. Güntert, M. Livny, J. L. Markley, M. Nilges, *et al.*, "RECOORD: a recalculated coordinate database of 500+ proteins from the PDB using restraints from the BioMagResBank," *PROTEINS: Structure, Function, and Bioinformatics*, vol. 59, no. 4, pp. 662–672, 2005.

[43] W. F. Vranken, W. Boucher, T. J. Stevens, R. H. Fogh, A. Pajon, M. Llinas, E. L. Ulrich, J. L. Markley, J. Ionides, and E. D. Laue, "The CCPN data model for NMR spectroscopy: development of a software pipeline," *Proteins: Structure, Function, and Bioinformatics*, vol. 59, no. 4, pp. 687–696, 2005.

[44] T. Goddard and D. Kneller, "SPARKY 3," *University of California, San Francisco*, vol. 14, p. 15, 2004.

[45] M. Fenwick, G. Weatherby, H. J. Ellis, and M. R. Gryk, "Parser Combinators: A Practical Application for Generating Parsers for NMR Data," in *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on*, pp. 241–246, IEEE, 2013.

[46] S. R. Hall, "The STAR file: A new format for electronic data transfer and archiving," *Journal of Chemical Information and Computer Sciences*, vol. 31, no. 2, pp. 326–333, 1991.

[47] S. R. Hall and N. Spadaccini, "The STAR file: Detailed specifications," *Journal of Chemical Information and Computer Sciences*, vol. 34, no. 3, pp. 505–508, 1994.

[48] S. R. Hall and A. P. Cook, "STAR dictionary definition language: initial specification," *Journal of chemical information and computer sciences*, vol. 35, no. 5, pp. 819–825, 1995.

[49] R. J. Nowling, J. Vyas, G. Weatherby, M. W. Fenwick, H. J. Ellis, and M. R. Gryk, "CONNJUR spectrum translator: an open source application for reformatting NMR spectral data," *Journal of biomolecular NMR*, vol. 50, no. 1, pp. 83–89, 2011.

[50] M. Fenwick, G. Weatherby, and M. Gryk, "Connjur Workflow Builder.".

[51] M. W. Maciejewski, M. Fenwick, A. D. Schuyler, A. S. Stern, V. Gorbatyuk, and J. C. Hoch, "Random phase detection in multidimensional NMR," *Proceedings of the National Academy of Sciences*, vol. 108, no. 40, pp. 16640–16644, 2011.

[52] D. Rovnyak, D. P. Frueh, M. Sastry, Z.-Y. J. Sun, A. S. Stern, J. C. Hoch, and G. Wagner, "Accelerated acquisition of high resolution triple-resonance spectra using non-uniform sampling and maximum entropy reconstruction," *Journal of Magnetic Resonance*, vol. 170, no. 1, pp. 15–21, 2004.

[53] A. E. Derome and A. E. Derome, *Modern NMR techniques for chemistry research*, vol. 6. Pergamon press Oxford, 1987.

[54] F. Bloch, "Nuclear induction," *Physical review*, vol. 70, no. 7-8, p. 460, 1946.

[55] D. G. Davis and A. Bax, "Assignment of complex proton NMR spectra via two-dimensional homonuclear Hartmann-Hahn spectroscopy," *Journal of the American Chemical Society*, vol. 107, no. 9, pp. 2820–2821, 1985.

[56] I. Solomon, "Relaxation processes in a system of two spins," *Physical Review*, vol. 99, no. 2, p. 559, 1955.

[57] J. H. Ardenkjær-Larsen, B. Fridlund, A. Gram, G. Hansson, L. Hansson, M. H. Lerche, R. Servin, M. Thaning, and K. Golman, "Increase in signal-to-noise ratio of> 10,000 times in liquid-state NMR," *Proceedings of the National Academy of Sciences*, vol. 100, no. 18, pp. 10158–10163, 2003.

[58] D. S. Wishart, C. G. Bigam, J. Yao, F. Abildgaard, H. J. Dyson, E. Oldfield, J. L. Markley, and B. D. Sykes, "1H, 13C and 15N chemical shift referencing in biomolecular NMR," *Journal of biomolecular NMR*, vol. 6, no. 2, pp. 135–140, 1995.

[59] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.

[60] H. Nyquist, "Certain topics in telegraph transmission theory," *American Institute of Electrical Engineers, Transactions of the*, vol. 47, no. 2, pp. 617–644, 1928.

[61] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[62] J. C. Hoch, "Maximum entropy signal processing of two-dimensional NMR data," *Journal of Magnetic Resonance (1969)*, vol. 64, no. 3, pp. 436–440, 1985.

[63] D. Neuhaus and M. P. Williamson, *The nuclear Overhauser effect in structural and conformational analysis*. VCH New York, 1989.

[64] M. P. Williamson and C. J. Craven, "Automated protein structure calculation from NMR data," *Journal of biomolecular NMR*, vol. 43, no. 3, pp. 131–143, 2009.

[65] M. C. Baran, Y. J. Huang, H. N. Moseley, and G. T. Montelione, "Automated analysis of protein NMR assignments and structures," *Chemical reviews*, vol. 104, no. 8, pp. 3541–3556, 2004.

[66] P. Guerry and T. Herrmann, "Advances in automated NMR protein structure determination," *Quarterly reviews of biophysics*, vol. 44, no. 03, pp. 257–309, 2011.

[67] L. E. Kay, M. Ikura, R. Tschudin, and A. Bax, "Three-dimensional triple-resonance NMR spectroscopy of isotopically enriched proteins," *Journal of Magnetic Resonance (1969)*, vol. 89, no. 3, pp. 496–514, 1990.

[68] S. Grzesiek and A. Bax, "An efficient experiment for sequential backbone assignment of medium-sized isotopically enriched proteins," *Journal of Magnetic Resonance (1969)*, vol. 99, no. 1, pp. 201–207, 1992.

[69] S. Grzesiek and A. Bax, "Correlating backbone amide and side chain resonances in larger proteins by multiple relayed triple resonance NMR," *Journal of the American Chemical Society*, vol. 114, no. 16, pp. 6291–6293, 1992.

[70] S. Grzesiek and A. Bax, "Amino acid type determination in the sequential assignment procedure of uniformly 13C/15N-enriched proteins," *Journal of biomolecular NMR*, vol. 3, no. 2, pp. 185–204, 1993.

[71] S. Grzesiek, J. Anglister, and A. Bax, "Correlation of Backbone Amide and Aliphatic Side-Chain Resonances in $^{13}C/^{15}N$-Enriched Proteins by Isotropic Mixing of ^{13}C Magnetization," *Journal of Magnetic Resonance, Series B*, vol. 101, no. 1, pp. 114–119, 1993.

[72] A. Lemak, C. A. Steren, C. H. Arrowsmith, and M. Llinás, "Sequence specific resonance assignment via Multicanonical Monte Carlo search using an ABACUS approach," *Journal of biomolecular NMR*, vol. 41, no. 1, pp. 29–41, 2008.

[73] P. Güntert, "Automated structure determination from NMR spectra," *European Biophysics Journal*, vol. 38, no. 2, pp. 129–143, 2009.

[74] A. Majumdar and E. Zuiderweg, "Improved ^{13}C-Resolved HSQC-NOESY Spectra in H_2O, Using Pulsed Field Gradients," *Journal of Magnetic Resonance, Series B*, vol. 102, no. 2, pp. 242–244, 1993.

[75] M. Takeda, T. Ikeya, P. Güntert, and M. Kainosho, "Automated structure determination of proteins with the SAIL-FLYA NMR method," *Nature protocols*, vol. 2, no. 11, pp. 2896–2902, 2007.

[76] P. Güntert, "Automated NMR structure calculation with CYANA," in *Protein NMR Techniques*, pp. 353–378, Springer, 2004.

[77] J. P. Linge, M. Habeck, W. Rieping, and M. Nilges, "ARIA: automated NOE assignment and NMR structure calculation," *Bioinformatics*, vol. 19, no. 2, pp. 315–316, 2003.

[78] Y. Shen, F. Delaglio, G. Cornilescu, and A. Bax, "TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts," *Journal of biomolecular NMR*, vol. 44, no. 4, pp. 213–223, 2009.

[79] M. Karplus, "Contact electron-spin coupling of nuclear magnetic moments," *The Journal of chemical physics*, vol. 30, no. 1, pp. 11–15, 1959.

[80] M. Karplus, "Vicinal proton coupling in nuclear magnetic resonance," *Journal of the American Chemical Society*, vol. 85, no. 18, pp. 2870–2871, 1963.

[81] C. D. Schwieters, J. J. Kuszewski, N. Tjandra, and G. Marius Clore, "The Xplor-NIH NMR molecular structure determination package," *Journal of Magnetic Resonance*, vol. 160, no. 1, pp. 65–73, 2003.

[82] H. Sreepad, "First-principles Study of L-Alanine,"

[83] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, "Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms," *Journal of Magnetic Resonance*, vol. 172, no. 2, pp. 296–305, 2005.

[84] R. Kaiser, "Use of the Nuclear Overhauser Effect in the Analysis of High-Resolution Nuclear Magnetic Resonance Spectra," *Journal of Chemical Physics*, vol. 39, pp. 2435–2442, Nov. 1963.

[85] S. Spera and A. Bax, "Empirical correlation between protein backbone conformation and C. alpha. and C. beta. 13C nuclear magnetic resonance chemical shifts," *Journal of the American Chemical Society*, vol. 113, no. 14, pp. 5490–5492, 1991.

[86] A. Cavalli, X. Salvatella, C. M. Dobson, and M. Vendruscolo, "Protein structure determination from NMR chemical shifts," *Proceedings of the National Academy of Sciences*, vol. 104, no. 23, pp. 9615–9620, 2007.

[87] Y. Shen, O. Lange, F. Delaglio, P. Rossi, J. M. Aramini, G. Liu, A. Eletsky, Y. Wu, K. K. Singarapu, A. Lemak, *et al.*, "Consistent blind protein structure generation from NMR chemical shift data," *Proceedings of the National Academy of Sciences*, vol. 105, no. 12, pp. 4685–4690, 2008.

[88] T. Ikeya, M. Takeda, H. Yoshida, T. Terauchi, J.-G. Jee, M. Kainosho, and P. Güntert, "Automated NMR structure determination of stereo-array isotope labeled ubiquitin from minimal sets of spectra using the SAIL-FLYA system," *Journal of biomolecular NMR*, vol. 44, no. 4, pp. 261–272, 2009.

[89] R. R. Ernst and W. Anderson, "Application of Fourier transform spectroscopy to magnetic resonance," *Review of Scientific Instruments*, vol. 37, no. 1, pp. 93–102, 2004.

[90] G. M. Crippen, A. Rousaki, M. Revington, Y. Zhang, and E. R. Zuiderweg, "SAGA: rapid automatic mainchain NMR assignment for large proteins," *Journal of biomolecular NMR*, vol. 46, no. 4, pp. 281–298, 2010.

[91] E. R. Zuiderweg, I. Bagai, P. Rossi, and E. B. Bertelsen, "EZ-ASSIGN, a program for exhaustive NMR chemical shift assignments of large proteins from complete or incomplete triple-resonance data," *Journal of biomolecular NMR*, vol. 57, no. 2, pp. 179–191, 2013.

[92] H. R. Eghbalnia, A. Bahrami, L. Wang, A. Assadi, and J. L. Markley, "Probabilistic identification of spin systems and their assignments including coil–helix inference as output (PISTACHIO)," *Journal of Biomolecular NMR*, vol. 32, no. 3, pp. 219–233, 2005.

[93] D. E. Zimmerman, C. A. Kulikowski, Y. Huang, W. Feng, M. Tashiro, S. Shimotakahara, C.-y. Chien, R. Powers, and G. T. Montelione, "Automated analysis of protein NMR assignments using methods from artificial intelligence," *Journal of molecular biology*, vol. 269, no. 4, pp. 592–610, 1997.

[94] H. N. Moseley, D. Monleon, and G. T. Montelione, "Automatic determination of protein backbone resonance assignments from triple-resonance NMR data," *Methods in enzymology*, vol. 339, pp. 91–108, 2001.

[95] C. Bartels, T.-h. Xia, M. Billeter, P. Güntert, and K. Wüthrich, "The program XEASY for computer-supported NMR spectral analysis of biological macromolecules," *Journal of biomolecular NMR*, vol. 6, no. 1, pp. 1–10, 1995.

[96] J. Cavanagh, W. J. Fairbrother, A. G. Palmer III, and N. J. Skelton, *Protein NMR spectroscopy: principles and practice*. Academic Press, 1995.

[97] F. Delaglio, S. Grzesiek, G. W. Vuister, G. Zhu, J. Pfeifer, and A. Bax, "NMRPipe: a multidimensional spectral processing system based on UNIX pipes," *Journal of biomolecular NMR*, vol. 6, no. 3, pp. 277–293, 1995.

[98] J. Hoch and A. Stern, "The Rowland NMR Toolkit," *Rowland Institute for Science Technical Memorandum*, no. 18t, 1985.

[99] V. Jaravine, I. Ibraghimov, and V. Y. Orekhov, "Removal of a time barrier for high-resolution multidimensional NMR spectroscopy," *Nature methods*, vol. 3, no. 8, pp. 605–607, 2006.

[100] J. C. Hoch and A. S. Stern, *NMR data processing*. Wiley-Liss New York:, 1996.

[101] B. Johnson, "NMRViewJ, version 8.2. 36," *One Moon Scientific Inc., Newark*, 2010.

[102] B. Alipanahi, X. Gao, E. Karakoc, L. Donaldson, and M. Li, "PICKY: a novel SVD-based NMR spectra peak picking method," *Bioinformatics*, vol. 25, no. 12, pp. i268–i275, 2009.

[103] V. Y. Orekhov, I. V. Ibraghimov, and M. Billeter, "MUNIN: a new approach to multi-dimensional NMR spectra interpretation," *Journal of biomolecular NMR*, vol. 20, no. 1, pp. 49–60, 2001.

[104] D. M. Korzhnev, I. V. Ibraghimov, M. Billeter, and V. Y. Orekhov, "MUNIN: Application of three-way decomposition to the analysis of heteronuclear NMR relaxation data**," *Journal of biomolecular NMR*, vol. 21, no. 3, pp. 263–268, 2001.

[105] N. H. Pawley, J. D. Gans, and R. Michalczyk, "APART: automated preprocessing for NMR assignments with reduced tedium," *Bioinformatics*, vol. 21, no. 5, pp. 680–682, 2005.

[106] R. Koradi, M. Billeter, M. Engeli, P. Güntert, and K. Wüthrich, "Automated peak picking and peak integration in macromolecular NMR spectra using AUTOPSY," *Journal of Magnetic Resonance*, vol. 135, no. 2, pp. 288–297, 1998.

[107] A. Bahrami, A. H. Assadi, J. L. Markley, and H. R. Eghbalnia, "Probabilistic interaction network of evidence algorithm and its application to complete labeling of peak lists from protein NMR spectroscopy," *PLoS computational biology*, vol. 5, no. 3, p. e1000307, 2009.

[108] A. S. Altieri and R. A. Byrd, "Automation of NMR structure determination of proteins," *Current opinion in structural biology*, vol. 14, no. 5, pp. 547–553, 2004.

[109] Y.-S. Jung and M. Zweckstetter, "Mars-robust automatic backbone assignment of proteins," *Journal of biomolecular NMR*, vol. 30, no. 1, pp. 11–23, 2004.

[110] G. T. Montelione, B. A. Lyons, S. D. Emerson, and M. Tashiro, "An efficient triple resonance experiment using carbon-13 isotropic mixing for determining sequence-specific resonance assignments of isotopically-enriched proteins," *Journal of the American Chemical Society*, vol. 114, no. 27, pp. 10974–10975, 1992.

[111] A. Bax, G. M. Clore, and A. M. Gronenborn, "1H-1H correlation via isotropic mixing of 13C magnetization, a new three-dimensional approach for assigning 1H and 13C spectra of 13C-enriched proteins," *Journal of Magnetic Resonance*, vol. 88, pp. 425–431, 1990.

[112] M. Mobli, A. S. Stern, W. Bermel, G. F. King, and J. C. Hoch, "A non-uniformly sampled 4D HCC (CO) NH-TOCSY experiment processed using maximum entropy for rapid protein sidechain assignment," *Journal of Magnetic Resonance*, vol. 204, no. 1, pp. 160–164, 2010.

[113] S. Hiller, R. Joss, and G. Wider, "Automated NMR assignment of protein side chain resonances using automated projection spectroscopy (APSY)," *Journal of the American Chemical Society*, vol. 130, no. 36, pp. 12073–12079, 2008.

[114] P. K. Weiner and P. A. Kollman, "AMBER: Assisted model building with energy refinement. A general program for modeling molecules and their interactions," *Journal of Computational Chemistry*, vol. 2, no. 3, pp. 287–303, 1981.

[115] M. C. Baran, H. N. Moseley, J. M. Aramini, M. J. Bayro, D. Monleon, J. Y. Locke, and G. T. Montelione, "SPINS: a laboratory information management system for organizing and archiving intermediate and final results from NMR protein structure determinations," *Proteins: Structure, Function, and Bioinformatics*, vol. 62, no. 4, pp. 843–851, 2006.

[116] Z. Zolnai, P. T. Lee, J. Li, M. R. Chapman, C. S. Newman, G. N. Phillips Jr, I. Rayment, E. L. Ulrich, B. F. Volkman, and J. L. Markley, "Project management system for structural and functional proteomics: Sesame," *Journal of structural and functional genomics*, vol. 4, no. 1, pp. 11–23, 2003.

[117] G. Liu, Y. Shen, H. S. Atreya, D. Parish, Y. Shao, D. K. Sukumaran, R. Xiao, A. Yee, A. Lemak, A. Bhattacharya, *et al.*, "NMR data collection and analysis protocol for high-

throughput protein structure determination," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 30, pp. 10487–10492, 2005.

[118] B. Nuseibeh, S. Easterbrook, and A. Russo, "Leveraging inconsistency in software development," *Computer*, vol. 33, no. 4, pp. 24–29, 2000.

[119] E. F. Codd, "A relational model of data for large shared data banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, 1970.

[120] B. G. Buchanan, E. H. Shortliffe, *et al.*, *Rule-based expert systems*, vol. 3. Addison-Wesley Reading, MA, 1984.

[121] R. Reiter, "A theory of diagnosis from first principles," *Artificial intelligence*, vol. 32, no. 1, pp. 57–95, 1987.

[122] P. Baudiš, "Current concepts in version control systems," Master's thesis, Charles University, Prague, 2008.

[123] K. Hinsen, K. Läufer, and G. K. Thiruvathukal, "Essential tools: Version control systems," *Computing in science & engineering*, vol. 11, no. 6, pp. 84–91, 2009.

[124] J. Loeliger and M. McCullough, *Version Control with Git: Powerful tools and techniques for collaborative software development*. " O'Reilly Media, Inc.", 2012.

[125] T. Morse, "Cvs," *Linux Journal*, vol. 1996, no. 21es, p. 3, 1996.

[126] B. Collins-Sussman, B. Fitzpatrick, and M. Pilato, *Version control with subversion*. O'Reilly Media, Inc., 2004.

[127] M. Fischer, M. Pinzger, and H. Gall, "Populating a release history database from version control and bug tracking systems," in *Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on*, pp. 23–32, IEEE, 2003.

[128] M. P. Robillard and G. C. Murphy, "Representing concerns in source code," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 16, no. 1, p. 3, 2007.

[129] Y. Shen and A. Bax, "SPARTA+: a modest improvement in empirical NMR chemical shift prediction by means of an artificial neural network," *Journal of biomolecular NMR*, vol. 48, no. 1, pp. 13–22, 2010.

[130] D. S. Wishart, M. S. Watson, R. F. Boyko, and B. D. Sykes, "Automated 1H and 13C chemical shift prediction using the BioMagResBank," *Journal of biomolecular NMR*, vol. 10, no. 4, pp. 329–336, 1997.

[131] "Jmol: an open-source Java viewer for chemical structures in 3D." `http://www.jmol.org/`. Accessed: 2014-09-05.

[132] S. G. Hyberts, K. Takeuchi, and G. Wagner, "Poisson-gap sampling and forward maximum entropy reconstruction for enhancing the resolution and sensitivity of protein NMR data," *Journal of the American Chemical Society*, vol. 132, no. 7, pp. 2145–2147, 2010.

[133] M. W. Maciejewski, H. Z. Qui, I. Rujan, M. Mobli, and J. C. Hoch, "Nonuniform sampling and spectral aliasing," *Journal of Magnetic Resonance*, vol. 199, no. 1, pp. 88–93, 2009.

[134] J. C. Hoch, M. W. Maciejewski, and B. Filipovic, "Randomization improves sparse sampling in multidimensional NMR," *Journal of Magnetic Resonance*, vol. 193, no. 2, pp. 317–320, 2008.

[135] H. A. Kautz, B. Selman, and M. Coen, "Bottom-up design of software agents," *Communications of the ACM*, vol. 37, no. 7, pp. 143–146, 1994.

[136] M. Jørgensen, "Top-down and bottom-up expert estimation of software development effort," *Information and Software Technology*, vol. 46, no. 1, pp. 3–16, 2004.

[137] M. P. O'Brien, J. Buckley, and T. M. Shaft, "Expectation-based, inference-based, and bottom-up software comprehension," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 16, no. 6, pp. 427–447, 2004.

[138] C. Rajaraman and M. R. Lyu, "Reliability and maintainability related software coupling metrics in C++ programs," in *Software Reliability Engineering, 1992. Proceedings., Third International Symposium on*, pp. 303–311, IEEE, 1992.

[139] P. Bellstedt, T. Seiboth, S. Häfner, H. Kutscha, R. Ramachandran, and M. Görlach, "Resonance assignment for a particularly challenging protein based on systematic unlabeling of amino acids to complement incomplete NMR data sets," *Journal of biomolecular NMR*, vol. 57, no. 1, pp. 65–72, 2013.

[140] S.-R. Tzeng, M.-T. Pai, and C. G. Kalodimos, "NMR studies of large protein systems," in *Protein NMR Techniques*, pp. 133–140, Springer, 2012.

[141] J. Keeler, *Understanding NMR spectroscopy*. John Wiley & Sons, 2013.

[142] M. Fenwick, C. Sesanker, M. R. Schiller, H. J. Ellis, M. L. Hinman, J. Vyas, and M. R. Gryk, “An Open-Source Sandbox for Increasing the Accessibility of Functional Programming to the Bioinformatics and Scientific Communities,” in *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on*, pp. 89–94, IEEE, 2012.

[143] H. J. Ellis, G. Weatherby, R. J. Nowling, J. Vyas, M. Fenwick, and M. Gryk, “A Pipeline Software Architecture for NMR Spectrum Data Translation,” *Computing in Science & Engineering*, vol. 15, no. 1, pp. 76–83, 2013.

[144] S. Fox-Erlich, T. O. Martyn, H. J. Ellis, and M. R. Gryk, “Delineation and analysis of the conceptual data model implied by the "IUPAC Recommendations for Biochemical Nomenclature",” *Protein Science*, vol. 13, no. 9, pp. 2559–2563, 2004.