Summer 8-14-2009

Optimization of Multidimensional Nuclear Magnetic Resonance Spectroscopy, for Resolution and Sensitivity, Through Application of Radial Sampling

John M. Gledhill
University of Pennsylvania, gledhill@mail.med.upenn.edu

Optimization of Multidimensional Nuclear Magnetic Resonance Spectroscopy, for Resolution and Sensitivity, Through Application of Radial Sampling

Abstract

The high probability of degenerate frequencies in NMR spectra of complex biopolymers such as proteins presented a great barrier to detailed analysis. The combination of multidimensional NMR spectroscopy and high magnetic field strengths has overcome the resulting resonance assignment problem for proteins less than 50 kDa. However, as protein size increases the sampling and sensitivity limited regimes become apparent. As a consequence, the orthogonal linear sampling requirements of conventional multidimensional NMR spectroscopy, combined with increased signal averaging require a longer acquisition time than is feasible. To overcome these limitations, radial sampling of the indirect dimensions of multidimensional experiments is utilized. It is demonstrated here, that through optimization of radial sampling acquisition parameters, it is possible to escape the linear sequential sampling requirements of Cartesian sampling, which allows for the collection of a high resolution spectrum in reduced acquisition time. Further, by exploiting a fundamental statistical advantage of radial sampling, it is possible to obtain a signal-to-noise advantage, over the traditional methodology. The approach is generalized by developing an all inclusive NMR data processing package and associated programs to optimize radial sampling acquisition parameters. An example, which utilizes the resolution and sensitivity advantages, to collect a novel application of a high resolution four-dimensional 13C, 15N edited NOESY is presented in support.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Biochemistry & Molecular Biophysics

First Advisor

A. Joshua Wand

Keywords

NMR, Radial Sampling, Data Processing

Subject Categories

Other Biochemistry, Biophysics, and Structural Biology

Optimization of Multidimensional Nuclear Magnetic Resonance Spectroscopy, for Resolution and Sensitivity, through Application of Radial Sampling

John M. Gledhill, Jr.

A DISSERTATION

in

Biochemistry and Molecular Biophysics

Presented to the Faculties of the University of Pennsylvania

in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2009

Supervisor of Dissertation

KM Ferguson

Graduate Group Chairperson

Dedication

For Leen

Acknowledgements

Foremost, I would like to thank my wife, Charleen, for her loving support and encouragement. She has truly played an unseen, but pivotal, role in all of this work. She is amazing in the way she always knows exactly what to say to keep me going. But realistically, most of the time, she doesn't need to say anything because her smile says it all.

I am also very grateful of my thesis advisor, Josh Wand, who has made all of this work possible. His time, resources and capabilities have been essential. He has given me the freedom to choose my path, but always been available to help define the details.

Thanks are also due to the members of the Wand Lab, both former and current, especially, Kathy Valentine, Ron Peterson and Mike Marlow for helping me get started in the lab and always putting up with my incessant questions. Jakob Dogan, has provided great insight on pulse sequence development. Sabrina Bedard, Sarah Chung, Vignesh Kasinath, Joe Kielec, Vonni Moorman, Nathaniel Nucci and Shoshanna Pokras have all influenced this work.

Finally, I would like to thank my family and friends, all of whom, have played an essential supporting role.

ABSTRACT

Optimization of Multidimensional Nuclear Magnetic Resonance Spectroscopy, for Resolution and Sensitivity, through Application of Radial Sampling

John M. Gledhill, Jr.

Dr. A. Joshua Wand

The high probability of degenerate frequencies in NMR spectra of complex biopolymers such as proteins presented a great barrier to detailed analysis. The combination of multidimensional NMR spectroscopy and high magnetic field strengths has overcome the resulting resonance assignment problem for proteins less than 50 kDa. However, as protein size increases the sampling and sensitivity limited regimes become apparent. As a consequence, the orthogonal linear sampling requirements of conventional multidimensional NMR spectroscopy, combined with increased signal averaging require a longer acquisition time than is feasible. To overcome these limitations, radial sampling of the indirect dimensions of multidimensional experiments is utilized. It is demonstrated here, that through optimization of radial sampling acquisition parameters, it is possible to escape the linear sequential sampling requirements of Cartesian sampling, which allows for the collection of a high resolution spectrum in reduced acquisition time. Further, by exploiting a fundamental statistical advantage of radial sampling, it is possible to obtain a signal-to-noise advantage, over the traditional methodology. The approach is generalized by developing an all inclusive NMR data processing package and associated programs to optimize radial sampling acquisition parameters. An example, which utilizes the resolution and sensitivity advantages, to collect a novel application of a high resolution four-dimensional ^{13}C, ^{15}N edited NOESY is presented in support.

Table Of Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction and Objectives

1.1 Introduction

In general terms, a protein's function is completely determined by its structure. Understanding protein structure, in many cases, can elucidate functional understanding of the protein at a mechanistic level. Stucture-function analysis has proven particularly important in such topics as catalysis, ligand binding, molecular transport and signaling cascades[1]. Protein structure has also played a pivotal role in understanding protein-drug interaction and substantial effort has been applied to rational drug design[2, 3].

Crystallography and nuclear magnetic resonance (NMR) are the primary techniques used to determine protein atomic structure. While crystallography has outpaced NMR in the number of protein structure determined, the additional functionality of NMR makes it appealing in many cases[4]. Namely, NMR allows for biophysical characterization of proteins at site resolved resolution while the protein is in solution. Though NMR has many appealing properties, until recently, the size range of proteins amenable to analysis by NMR is not as broad as the size of proteins that are desirable to study.

The disparity between proteins amenable to NMR analysis and those desired to be studied arises from physical properties of the molecule and means by which NMR signal is acquired. As protein size increases the molecule reorientation time increases and accordingly the spin-spin relaxation rate, T_2, decreases. An increase in T_2 relaxation results in broadened lineshapes and decreased scaler coupling transfer efficiency. Unfortunately, these results decrease the signal to noise of the spectrum and limit the application of some pulse sequences from the reduced coupling efficiency. Combined, these effects have limited protein analysis beyond 35kDa.

Multiple methods have been developed in order to reduce the problematic effects of slow molecular tumbling. The most effective have been extensive deuteration[5-7], TROSY[8] pulse sequence optimization and reverse micelle technology[9]. Extensive deuteration of the protein reduces the dipolar field surrounding the remaining protons and in turn, eliminates many of the spin-spin relaxation modes. With application of deuterium decoupling, this technique has allowed for application of multidimensional NMR experiments to proteins in the 20kDa range[5-7]. TROSY (transverse relaxation optimized spectroscopy) pulse sequences increase the functional protein size by selecting for constructive interference between dipole-dipole relaxation, which arises from slow molecular tumbling, and intrinsic chemical shift anisotropy. Cancellation of the relaxation components allows for selection of a narrow lineshape component. This technique has successfully been applied to proteins beyond 40kDa[10, 11]. The final method to increase the amenable protein size range is application of reverse micelle technology. Reverse micelles are created by encapsulating a protein, which is dissolved in a small pool of water, inside a surfactant micelle that is dissolved in a non-polar, low viscosity, solvent. By using a low viscosity solvent the molecular tumbling time of large proteins is reduced. In turn, all traditional NMR methodology is applicable. This method has been successfully applied to proteins great than 50 kDa[12](unpublished data).

Although the technology is available to study large proteins with NMR, the size range, of protein structure determined by NMR, is not comparable to the size range of proteins of interest. This disparity is apparent if the size of protein structures determined by NMR[13] is compared to the size of protein drug targets[14], Figure 1.1.

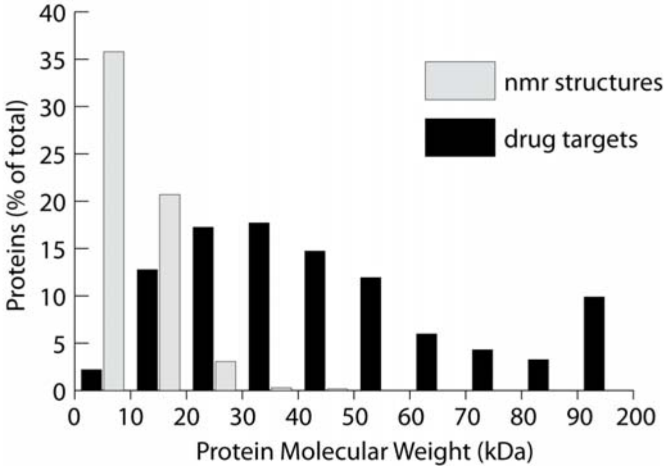


Figure 1.1 A comparison of the molecular weight of protein structures determined by NMR with the molecular weight of protein drug targets, is shown here. The relative frequency versus protein molecular weight demonstrates the disparity, in protein size, between proteins currently being studied and the size of proteins with desirable properties to study. NMR structure information was obtained from the PDB[13] and edited for redundancy. Drug target information was obtained from DrugBank[14].

The lag in protein structure size can be rationalized with the following two reasons: First, the NMR methods available for large proteins function by decreasing the linewidth of the resulting spectra which increases the signal to noise (S/N). These techniques, however, do not account for the added complexity that arises from the increased the number of signals in large proteins. Second, these methods often function at limited sensitivity compared to traditional techniques.

Spectral complexity increases with increasing protein size. In general, NMR spectra contain at least one peak per amino acid residue of the protein. Although in many experiments, such as a NOESY, multiple peaks per residue are present. The dispersion of chemical shifts does not increase coincidently with increasing protein size. Therefore, an increase in the number of peaks directly increases the number of peaks per spectral volume and results in decreased spectral resolution. Spectral complexity is decreased by increasing the dimensionality of the spectrum. Additional dimensions are added by correlating additional atoms in the magnetization transfer pathway. This serves to reduce the degeneracy of the spectrum by increasing the spectral area while retaining a constant number of peaks. This concept is illustrated in Figure 1.2. Here, the number of peaks remains constant but the dimensionality of the spectrum increases. Ubiquitin is used to show the effect of increasing the dimension of the experiment. When one dimension is evolved, amide protons in this case, very few of the peaks are resolved. Evolving two dimensions, Figure 1.2b, resolves a large fraction of the peaks but degeneracy is still present in the spectrum. When three dimensions are evolved all of the degeneracy is resolved because each peak has a unique set of chemical shifts.

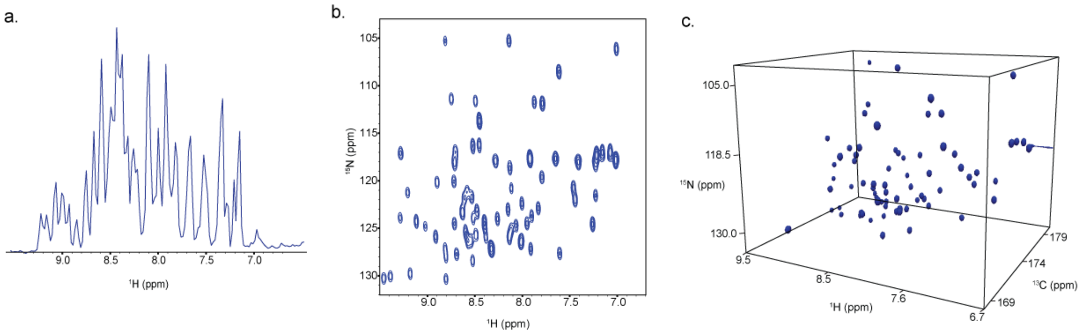


Figure 1.2 An example of increasing the dimension of a NMR experiment to increase the resolution of the experiment is shown here. Ubiquitin is used in all three panels. A one-dimensional amdie 1H spectrum , panel a, a two-dimensional 15N-HSQC[15], panel b and a three-dimensional HNCO[16] resolve an increasing number of peaks as the dimensionality increases.

Increasing the dimensionality of an experiment comes at the expense of acquisition time. The total acquisition time of an experiment is estimated from the product of the number of data points collected in each dimension, the number of transient scans averaged per FID, the interscan, magnetization recovery, delay and the acquisition time of each scan.

$$n_{fid} = \prod_{j=1}^{N-1} n_j n_s d_1 t_a$$

Here n_j is the total number of points in dimension j of N total dimensions, n_s is the number of transients, and d_1 and t_a are the recyle delay and acquisition times respectively. The length of the pulse sequence is comparatively small and ignored. Assuming minimal acquisition time per increment, one transient and 16 increments per dimension; the total acquisition time quickly increases beyond a reasonable range as the dimensionality of an experiment is increased. Using the above parameters a 3D experiment would require .5 hour, a 4D 9 hours, 5D 12 days and a 6D 1.1 years[17]. Even using a minimal acquisition scheme the total acquisition time expands beyond a feasible range beyond four dimensions. Collectively this regime is known as the sampling limit[18]. In this regime acceptable resolution determines the total acquisition time.

Sensitivity is the second limiting parameter to large protein structure determination. Limiting sensitivity arises from dilute samples and/or complex sample preparation protocols. In the case of large proteins, the problems are further compounded by the technologies employed to circumvent tumbling limitations. Extensive, or factional, deuteration decrease the dipolar field but limit the concentration of observable signal by randomly exchanging the observable proton with a non-observed deuterium[6]. TROSY techniques achieve a narrow peak by splitting the signal into four components and selecting for one of the four components that has ideal relaxation parameters, in turn, decreasing the observed signal by 75 percent[8]. Reverse micelles decrease the molecular reorientation time by dissolving the protein in a low viscosity solvent. To minimize the reorientation time, short chained hydrocarbons are used as a solvent. These solvents require pressuring the sample in a special apparatus which limits the sample volume. Some high-pressure NMR tubes limit volume by 67.5%[19]. Further, the concentration of protein filled reverse micelles is limited by the amount of surfactant that can be added without deleteriously increasing the viscosity of the solvent[20].

Typically, when large protein techniques are used, sensitivity is increased by averaging additional transients at the expense of additional measurement time. The variance sum law dictates that doubling the number of transients averaged, which doubles the acquisition time, will increase the signal to noise by the square root of 2. This regime is known as the sensitivity limit[18]. Here the minimal acquisition time is determined by the sensitivity of the experiment. Resolution is also typically limited in this regime because time is spent collecting a large number of scans rather than an increasing number of increments.

In light of the sampling and sensitivity limits, approaches are necessary to collect data with increased resolution, without an exponential increase in acquisition time while achieving a concomitant increase in sensitivity. Recently, various methods have been introduced to speed acquisition. However, no new approaches are available to increase the sensitivity of multidimensional NMR experiments.

All of the recent methods to speed acquisition rely on sparse sampling. In general, sparse sampling speeds acquisition by reducing the number of data points collected in the indirect dimensions. Typically, an order of magnitude time savings is possible using sparse sampling.

1.2 Objectives

The primary objective of this thesis is to alleviate the resolution and sensitivity limitations, imposed by large proteins on NMR, through application of sparse sampling.

The efforts are threefold: First the general application of sparse sampling is improved by developing and extending current methodology. This includes a multidimensional NMR processing program, designed to efficiently process sparse sampled data, chapter 3; novel means to phase correct sparse sampled data, chapter 4; and an optimized sampling angle selection routine for radial sampling, chapter 5. Second, means to obtain an increase in sensitivity are developed. Termed, Sensitivity Enhanced n-Dimensional NMR (SEnD), the approach is presented in chapter 6. Finally, focusing on the resolution and speed of data acquisition, when radial sampling is employed, a new methods is developed. A novel method to collect a 4D ^{13}C, ^{15}N edited NOESY spectrum is presented in chapter . The results presented here are general and will facilitate development of additional novel applications that exploit the acquisition speed, resolution and sensitivity advantages achieved through application of sparse sampling.

CHAPTER 2

SPECTRAL ESTIMATION AND SPARSE SAMPLING

2.1 Introduction

In chapter 1 the sampling and sensitivity limited regimes[21] were presented. In order to overcome these limits, new methods that increase the resolution and sensitivity without a concomitant increase in acquisition time are needed. Of late, substantial work has been performed to alleviate the sampling limits imposed by the strict linear sequential sampling requirements of the standard fast Fourier transform. The majority of the new techniques are base on sparse sampling. Sparse sampling decreases acquisition time by selectively skipping acquisition of points in the indirect dimension. A substantial time savings can be achieved by skipping acquisition points, but this comes at the expense of spectral artifacts. Various spectral estimation methods have been developed to account for or eliminate these artifacts. The various sampling schemes and processing techniques will be reviewed here to determine which is most suitable for our applications.

Prior to reviewing the sparse sampling and data process techniques a review of spectral estimation is presented. This review will serve to further clarify the fundamental limitations imposed by traditional technology.

2.2 Spectral Estimation Review

NMR signal arises from the evolution of transverse magnetization[4]. As the magnetization evolves a time-varying current is generated. This current is measured as a time series of exponentially decaying sinusoid by the spectrometer. The time series data can be represented as:

$$\mathbf{d} = d_0, d_1, \; \dots \;, d_{M-1} \tag{2.1}$$

Where M is the total number of points sampled. In most cases a uniform increment is used between data points to make the data amenable to processing techniques that will be presented below. In all but the simplest cases, determining the underlying frequency components is impossible from direct inspection. Therefore, the time data is converted to the frequency domain using one of the various spectral estimation techniques. Estimation of the frequency domain data from the time domain is represented as:

$$\mathbf{d} \leftrightarrow \mathbf{f} \tag{2.2}$$

Converting the data to the frequency domain allows for a measure of the frequency components to be read directly from the resulting spectrum. The frequency series can be represented as:

$$\mathbf{f} = f_0, f_1, \; \dots \;, f_{N-1} \tag{2.3}$$

Where N is the total number of frequency components determined from the data. An example of converting the time series data to the frequency domain is shown in Figure 2.1.

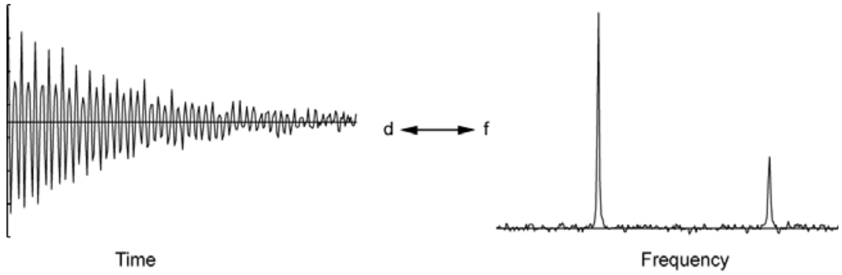


Figure 2.1 An example of spectral estimation is shown for generated data containing two peaks of varying intensity. The time series of an exponentially decaying sinusoid is shown on the left. Only the cosine modulated component is shown for clarity. Application of the Fourier transform to estimate the frequency spectrum produces the spectrum shown on the right.

Here an exponentially decaying sinusoid, of generated data containing two frequency components of different amplitudes is shown. The frequency components are not easily determined from visual inspection of the time data. Estimation of a frequency spectrum allows for direct inspection of the frequency components. Additionally, the frequency spectrum, allows for the relative intensity of the frequency components to be directly assessed. This feature is particularly important when there is a large noise component in the data.

The Fourier transform (FT)[22] is the most common method for spectral estimation. The FT is appealing because it is a linear transform, which allows for the data quality to be directly assessed from the noise level of the spectrum. It is also fast and has no adjustable parameters, making application easy.

Efficient application of the FT requires that the data is sampled at a constant interval. Utilizing a constant interval the data series is written formally as a summation of sinusoids encompassing all of the detectable frequencies multiplied by an exponential decay parameter, T_{2k}.

$$\mathbf{d} = \sum_{k=0}^{N-1} A_k \cos(2\pi\omega_k m\Delta\tau) e^{m\Delta\tau / T_{2k}} \tag{2.4}$$

Where N is the total number of frequency components that can be determined, A_k is the amplitude term of a given component k, with frequency ω_k. m is the series point of M total points and $\Delta\tau$ is the sampling increment. The sampling increment determines the detectable frequency range as dictated by the Nyquist theorem[23]; which states that the range of detectable frequency is ½ the inverse of the time increment. Therefore, if the data is centered at zero frequency the detectable band of frequencies is: $sw = \frac{1}{\Delta\tau}$. In order to determine if a frequency component is positive or negative with respect to the carrier quadrature detection is employed. Quadrature detection is accomplished by collecting two data components, one modulated by cosine and the other sine[4]. The real and imaginary pair is stored as a series of complex numbers. Using the Eulers identity and assuming the summation, the data series is written as:

$$\mathbf{d} = Ae^{-i\omega n\Delta\tau_1 + n\Delta\tau_1/T_2} \tag{2.5}$$

Having defined the possible frequency range of the data series, it is possible to determine all frequencies are present in a given spectrum. The FT solves for the amplitude of each frequency component by first generating a model sinusoid at the given frequency using the same time points as the data. Then the amplitude is determined by summing the product of the data and the sinusoid. The FT of the frequency series is written as:

$$\mathbf{f_n} = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \mathbf{d}_k e^{2\pi kn/M} \tag{2.6}$$

The various terms have the same meaning as above. This representation utilizes the fact that the time and frequency components do not need to be explicitly defined. If the Nyquist sampling theorem is applied then the frequency and time points are both a function of the sweep width and the two terms are reduced to $\frac{nk}{M}$. This generalization assumes that the same number of frequency terms are determined as there are number of data points and the points are equally distributed in the sweep width range. If 16 points are collected, show in black, then the spectrum with broad lines is generated, black as well. If an additional 48 points are collected, resulting in 64 total points then the resulting spectrum in grey is generated with a much narrower line shape. When this concept is expanded to multiple dimensions the limitations of Cartesian sampling are immediately apparent.

When a multiple dimension experiment is collected, each dimension is sampled independently and sequentially using a Cartesian basis. The resulting data is a product of all of the time domains that are evolved.

$$\mathbf{d} = Ae^{-i\omega n\Delta\tau_1 + n\Delta\tau_1 / T_2} e^{-i\omega n\Delta\tau_2 + n\Delta\tau_2 / T_2} \tag{2.7}$$

Sampling each dimension independently allows each dimension to be processed independently. For example, in the case of a 2D experiment the data is collected with respect to two incremented times $d(t_1, t_2)$. This data is first Fourier transformed with respect to t2, resulting in a mixed time-frequency spectrum $d(t_1) f(\omega_2)$. The matrix is then Fourier transformed with respect to t1, resulting in the frequency domain spectrum $f(\omega_1, \omega_2)$. The independence of each dimension requires that a sufficient number of data points be collected to achieve suitable resolution, as discussed above. In turn, increasing the dimensionality of the experiment exponentially increases the required acquisition time.

Collecting a larger number of data points increases the digital resolution of the frequency spectrum. As a secondary effect, the linewidth of peaks in the frequency spectrum are also decreased. Typically, the time data is apodized prior to FT. Apodization is the process of multiplying the data series by a decaying time function to bring the last points of the data to zero[24]. Apodization has the advantage of reducing truncation artifacts and provides a more satisfactory line shape. When more data points are present the apodization function brings the data to zero slower, which, when Fourier transformed, results in a narrowed line. This concept is illustrated in Figure 2.2. To circumvent an exponential increase in acquisition time alternate sampling and processing schemes have been presented.

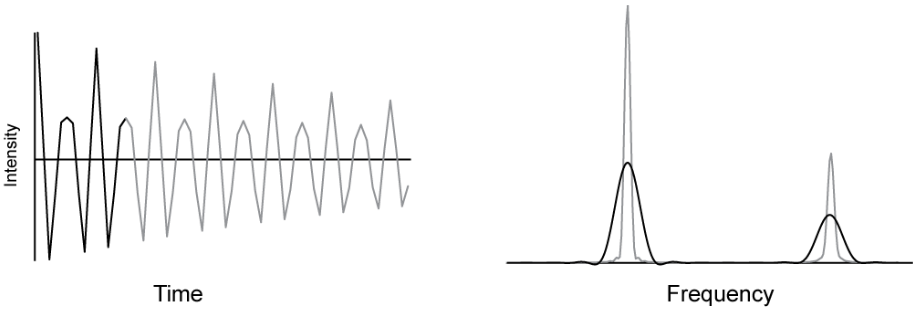


Figure 2.2 Increasing the number of time domain points acquired (left) directly increases the resolution of the frequency domain spectrum (right). The linewidth in the frequency spectrum is substantial decreased by increasing the number of points from 16 to 48.

2.3 Sparse Sampling

Sparse sampling decreases the acquisition time of multidimensional NMR experiments, by reducing the number of points collected in the indirect dimensions. The design of the spectrometer allows for the directly acquired dimension to be collected in real time so only points in the indirect dimensions are sparsely sampled. Various sparse sampling acquisition schemes are available, all of which are designed to provide a suitable level of information while not reducing the spectral resolution. As a result of incompletely sampling each dimension, the sparsely sampled dimensions contain artifacts. The artifacts are directly dependent on the sampling scheme used, as well as the method used to process the spectrum. A review of the various sample schemes and the resulting artifacts are presented here. The sampling schemes are presented first, followed by a discussion of the processing methods.

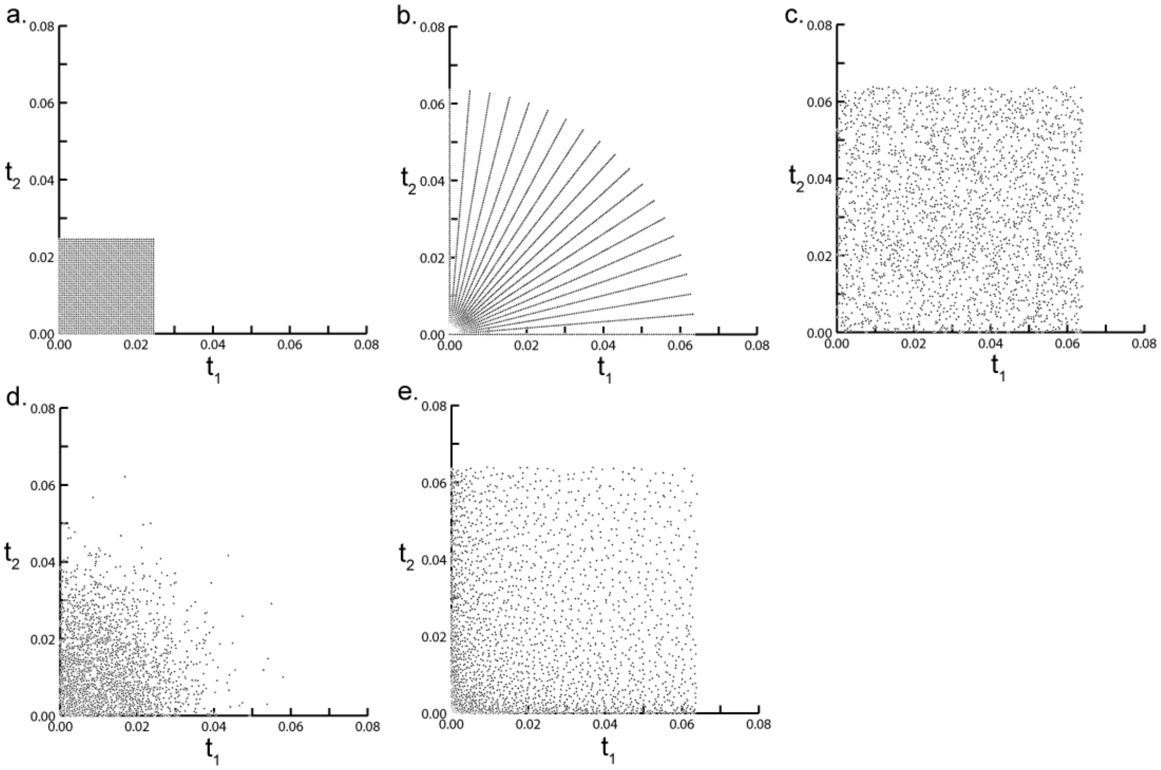


Figure 2.3 Sampling scheme comparison. The indirect dimension of four sparse sampling approaches are shown here compared to Cartesian sampling, a. In all cases 2500 sampling points are used. One of the primary advantages of sparse sampling is that it is capable of sampling much longer evolution times compared to Cartesian sampling. In the sparse sampling schemes 2500 points were plotted over the area of 16000 Cartesian sampled points. Radial sampling is demonstrated in, b, where 20 sampling angles are used and 128 points per angle. Uniform random sampling is shown in c. Gaussian distributed random sampling

in d and optimized random sampling is shown in e. In all schemes a 2000 Hz sweep width was used in both dimensions.

A collection of the various sampling methods are shown in Figure 2.3. The plots only show the sampling pattern for the indirect dimensions of a 3D experiment. As above, the directly acquired dimension is sampled traditionally. All plots in the Figure use the same number of sampling points, 50x50 over the two indirect dimensions. However, all of the sparse sampling schemes are collected over the time domain space equivalent to 128x128. This corresponds to a 6-fold time advantage for sparse sampling. The sampling schemes can be divided into two main categories, radial sampling and random sampling. Radial sampling, Figure 2.3b, is achieved by linking two or more of the indirect dimensions and linearly sampling a vector at an angle (α) with respect to the two orthogonal time domains. In the case of a three dimensional experiment, this is achieved by collecting the directly detected time domain signal normally and linking the indirect dimensions by defining $t_1 = \tau \cos(\alpha)$ and $t_2 = \tau \sin(\alpha)$ and linearly sampling the time period τ[25]. In this case, only one time vector is sampled per two time dimensions. Multiple angles are collected to resolve the degeneracy of collecting data in a lower dimensional space.

Random sampling is the second class of sparse sampling schemes[26]. This method is achieved by using a pseudo-random number generator to choose the acquisition time points in the indirect dimensions. Typically the maximum evolution of each domain is selected depending on the relaxation properties of the atom evolved in a given time domain dimension. The first application of random sampling used a uniform distribution of sampled points in the indirect time domains[26, 27], Figure 2.3c. Initial applications of uniform random sampling demonstrated that a significant resolution advantage can be achieved with this sampling protocol because only a gentle apodization function is required to remove truncation artifacts. Additionally, in many cases the data points are sampled at increments less than the Nyquist frequency which allows for more accurate detection of chemical shifts. Although a higher resolution spectrum was realized with uniform radial sampling artifacts were immediately apparent in the spectrum. Details regarding the artifacts are discussed below. To reduce the detrimental effects of the artifacts more sophisticated schemes have been proposed. These include Gaussian weighted random sampling[28], Figure 2.3d. Here, a probability bias is applied to the pseudo-random number generator. Weighting the distribution of sampled points reduces the effects of the artifacts. Optimally, a Gaussian distribution would be used that matches the decay properties of the nuclei evolved in the indirect dimensions.

Random sampling is further optimized by distributing the data points closer to a Cartesian basis, while still retaining a level of random sampling. Optimized random sampling[29], Figure 2.3e approaches a Cartesian approximation by placing additional restraints on the time points sampled. When Optimized random sampling is used, a grid is generated over the two indirect evolution time dimensions. The area of each cell in the grid increases with a Gaussian weight as the evolution times increase. This sampling scheme allows for collection of a higher density of points at shorter evolution times while still sampling enough points to avoid truncation artifacts. One data point is selected per grid cell. Again this sampling method improved artifacts.

Further normalization of the sampling pattern led to the creation of concentric shell sampling[30]. This sampling scheme produces an artifact free spectrum if specific criteria are met. The sampling scheme functions by collecting data points that are equally spaced on rings with expanding radii. The spacing of the points and rings are dependent on the required sweep width of interest. This sampling scheme requires an equivalent number of points as Cartesian sampling and therefore is not analyzed further here. However, the number of sampling points can be reduced systematically to produce a randomized scheme that is comparable to the optimized random sampling approach.

Regardless of the sampling scheme utilized, quadrature detection is still required to determine the sign of a peak relative to the carrier frequency. This is achieved by acquiring both a real, or cosine modulated component, and an imaginary, or sine modulated component, per dimension. In the case of a 3D experiment, four quadrature components are collected for the two indirect dimensions at each sampling time point[25]. The four data components, represent all combinations of the even and odd functions, which are Cos-Cos modulated, Cos-Sin, Sin-Cos and Sin-Sin. All four of the components are used in the processing techniques that will be discussed.

Traditional sequential 1D Fourier transform data processing methods are no longer applicable when data is sampled outside of the Cartesian basis. In turn, there are two main classes of processing technology to deal with sparse sampled data: projection reconstruction[25] and numerical estimation based[16, 17, 31-36]. The objectives of the projection reconstruction techniques are to generate a final spectrum directly from the data. Numerical estimation based techniques are generally designed to generate a list of the spectral features, then either use the information to create a peak list or generate a final spectrum. Projection reconstruction based techniques are only amenable to radial sampling, while numerical estimation methods are amenable to both radial and random sampling.

2.4 Projection Reconstruction

Application of projection reconstruction to NMR originated as an extension of computerized tomography techniques[37]. In computerized tomography techniques, multiple 2D 'tilted plane' projections of a 3-dimensional object are recorded as a function of sampling angle. The 2D projections are then used, through application of the Radon transform[38], to regenerate a representation of the 3D object. In the case of NMR experiments, radial sampling is used to collect tilted planes of time domain data, which is Fourier transformed to generated tilted planes in the frequency domain. However, unlike tomography, NMR spectra contain discrete peaks rather than continuous objects. Discrete peaks require fewer angles planes to regenerate a final spectrum. An example of the projection reconstruction approach is shown in figure 2.4.

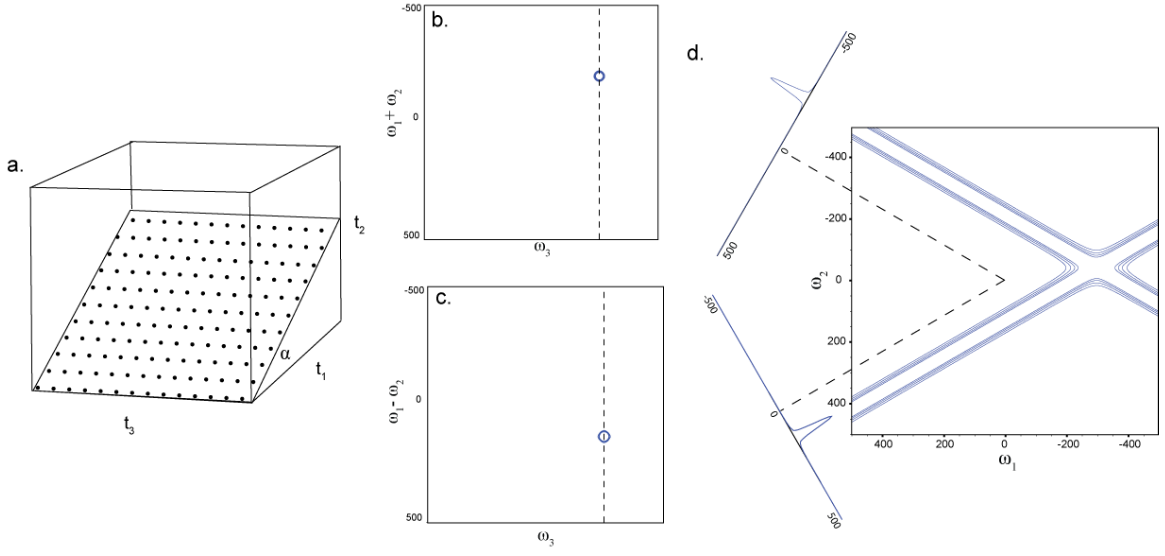


Figure 2.4 Schematic of the projection reconstruction approach to NMR. Radial sampling is use to sample a spectrum, a. To accomplish this, the directly acquired dimension, t$_3$, is sampled using a Cartesian pattern. The two indirect dimensions, t$_1$ and t$_2$, are linked and sampled simultaneously at an angle α (see text for details). The directly detected dimension is processed with standard Fourier transform technology. The indirect dimension cannot be processed directly. From the four quadrature components, sum and differences, employing double angle identities, of the various components are used to generate two complex pairs. The two complex pairs are Fourier transformed, generating two spectrum with the signals modulated by the sum and difference of the frequency components. These spectra are the tilted planes. The sum and differences are shown in panels b and c, respectively. Each indirect vector of the two spectra are projected into the frequency components at 90± α. The two dashed lines in b and c indicate the example vectors used for reconstruction in d. Here, additive back projection is used to resolves the degeneracy of the individual spectra. The vectors are aligned from the carrier frequencies and projected along the perpendicular into the two frequency domains. The intensity from each point on the tilted plane vector is added to the existing intensity in the frequency domains. This generates a ridge of intensity perpendicular to peaks. The peak chemical shift is located at the intersection of the ridge components.

Here, the directly acquired dimension is collected using Cartesian sampling while the two indirect dimensions are sampled using a 30 degree sampling angle, Figure 2.4a. The directly detected dimension is processed using standard Fourier transform methodology. This results in a 2D mixed mode spectrum, where d_1 is frequency and d_2 is an interferogram of time domain data (not shown). The 2D plane is tilted between the two indirect time dimensions. There are no means to distinguish a positive sampling angle from a negative sampling angle. Therefore, double angle identity linear combinations of the quadrature component spectra are calculated to separate the sum and difference of the two frequency domain components[39]. The positive and negative frequency component spectra are then used as a Fourier transform quadrature pair to generate the positive tilt angle spectra, Figure 2.4b, and the other two used to generate the negative tilt angle spectra, Figure 2.4c.

Information is not available to determine the peak location orthogonal to the tilted spectrum plane. Two methods are commonly used to resolve the degeneracy: additive back-projection and lower magnitude comparison[25]. Additive back-projection (ABP), the equivalent of the radon transform, sums all of the component spectra with the intensity, from the tilted plane spectrum projected along a vector orthogonal to the point in the tilted plane[40]. This method produces a ridge of intensity wherever there are peaks in the tilted plane spectrum, Figure 2.4d. When both the sum and difference components are projected into the same spectrum, the ridge intensity constructively sums at the location of ridge intersections. When there is only one peak, in the indirect plane, the intersection of the two ridges corresponds to the chemical shift of the peak. When there are more than one peak in the indirect plane, the ridges intersect at both of the peak chemical shifts, as well as an artifact peak location. Ridges always intersect at the chemical shift of a peak, independent of the sampling angle. Multiple sampling angles are added into the spectrum to determine authentic peaks from artifact peaks. Adding more sampling angles to a spectrum will always reinforce peak intensity, while the artifact peak will remain at a baseline level. This method is capable of producing a readable spectrum, but suffers from severe baseline artifacts. The baseline artifacts can be removed using lower value (LV) method[25]. This method generates a back-projected spectrum for each of the component angles. The individual angle components are compared on an element basis retaining the minimum magnitude intensity value at each point. This removes all of the ridge intensity other than that from the authentic peaks, because only the peaks will have a non-baseline value as the sampling angle is varied. An example of the LV comparison is shown in Figure 2.5.

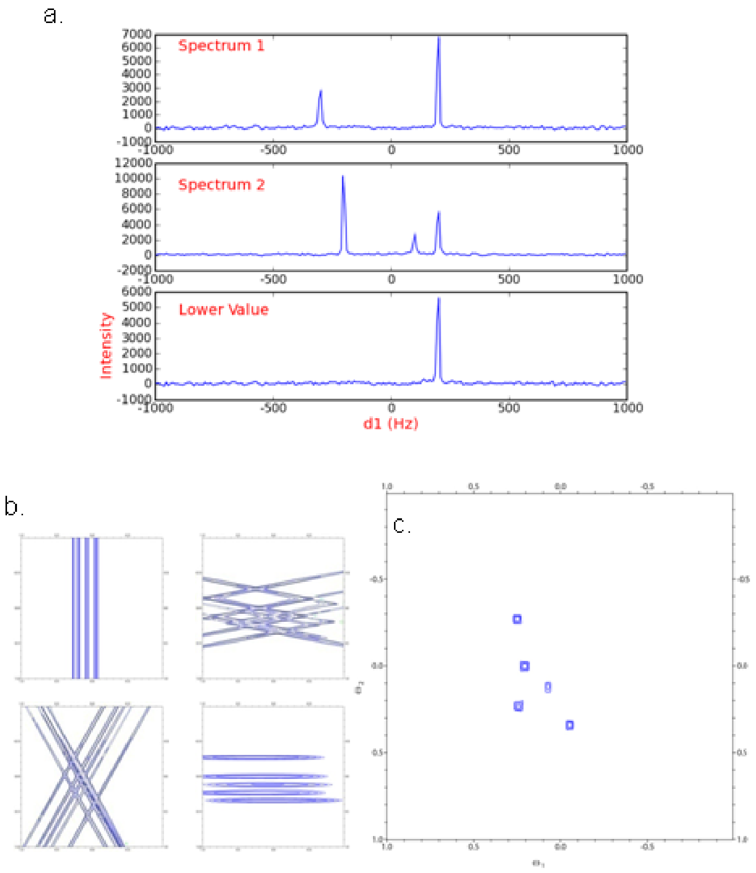


Figure 2.5 Demonstration of the lower value comparison. Panel shows a lower value comparison in 1D. The two spectra, 1 and 2, are compared point-to-point, and the smallest magnitude value is retained and stored in a third spectrum, labeled lower value. Note by comparison artifact peaks are removed. A 2D example, using a 3D radial sampled HNCO of Ubiquitin, is shown in b and c. (b) Four angle spectra are generated using ABP for each sampling angle; 0, 30, 45 and 90. (c) All four of the angle spectra are compared using the lower value to generate a final spectrum, which resolves all of the peaks.

The LV method efficiently removes artifacts from the spectrum, but has the potential to remove authentic peaks from the spectra if the intensity of a peak falls into the noise for a single sampling angle.

Two additional methods are also available to reconstruct a spectrum from the component angle spectra: hybrid reconstruction[41] and distribution reconstruction[42]. Both of these methods were developed to avoid some of the pitfalls of ABP and LV. When the S/N of the component spectra is limiting, peaks can potential be removed during LV comparison. Hybrid reconstruction uses a combination of ABP and LV. Starting with a set of component spectra sampled at various angles, the hybrid method generates ABP spectra from a subset of the component spectra. The sub-group ABP spectra are then used as input for a LV comparison to generate a final spectrum. Generating sub-group ABP spectra prior to LV, the peak intensity is increased prior to the LV, which decreases the likelihood that a peak will be inadvertently removed. The distribution method functions by creating a histogram of intensity values from each of the component spectra at the equivalent positions. A Gaussian is fit to the histogram and the intensity value at the max value of the Gaussian is selected for the final spectrum. This method proposes to avoid some of the flaws that are inherent to the other methods, but it is much more computationally intensive.

Projection reconstruction techniques are digitally limited when projecting the tilted planes into the final spectrum. Often, the data points of the tilted plane do not align with the data points in the final spectrum. Points on the tilted plane must be interpolated in order to determine the intensity values at the points in the final spectrum. This problem is illustrated in Figure 2.6.

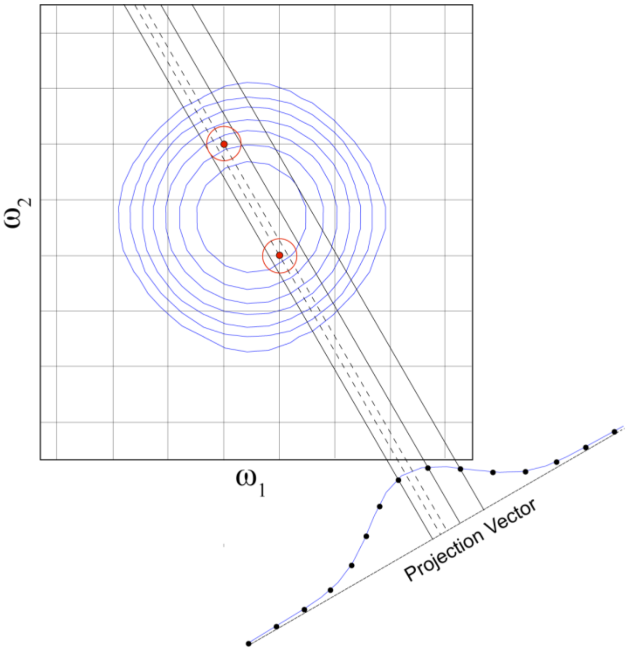


Figure 2.6 Demonstration of the fundamental limitation of projection reconstruction. To demonstrate the problem, the projected vectors are overlaid on a Cartesian sampled spectrum to indicate the chemical shift. To generate a frequency spectrum, data points from the projection vector are extended into the frequency plane. Often the points of the tilted projection vector and the frequency plane do not coincide, because both the projection vector and frequency plane are discretely sampled. The red circles indicate two points that do not fall on the projected intensity. To determine the intensity value at these points, the intensity at the intersection of the dashed line and the projection vector need to be interpolated. The interpolation process is inaccurate and time consuming.

Two studies, APSY[43] and HIFI[44], have proposed to avoid reconstruction by only using the peaks of a tilted plane spectrum. Under appropriate conditions these methods have been successfully applied. However, by not generating a final spectrum all of the existing analysis methodology is dismissed. Therefore, new means to directly solve for the intensity values in the final spectrum without having to interpolate data points on the tilted plane, have been presented to circumvent this limitation. This method realized that the summation used in ABP is essentially a direct multidimensional Fourier transform[26, 45, 46].

2.5 Direct Two-Dimensional Fourier Transform

The direct multidimensional Fourier transform (2D-FT) functions by simultaneously transforming multiple indirect dimensions as opposed to transforming the dimensions sequentially. The discrete 2D –FT can be described as [45-47]:

$$S(\omega_1,\omega_2)=\sum_{t1=0}^{t1\max}\sum_{t2=0}^{t2\max}\exp(-i\omega_1 t_1)\exp(-j\omega_2 t_2)f(t_1,t_2)g(t_1,t_2)w(t_1,t_2) \tag{2.8}$$

Where i and j are quarternion numbers; t_1, t_2 are the incremented times, ω_1 and ω_2 comprise the frequency pair being determined, $f(t_1,t_2)=\exp(-i\Omega_1 t_1)\exp(-j\Omega_2 t_2)$ is the data being transformed, Ω_1 and Ω_2 are the chemical shifts for time domain t_1 and t_2 respectively, $w(t_1,t_2)$ is a weighting factor to account for the unequally spaced sampling of the time domain and is typically applied as a two dimensional apodization function, and $g(t_1,t_2)$ describes the lifetime of the signal, which we will subsequently ignore. In the case of radial sampling $t_1=\tau\cos\alpha$ and $t_2=\tau\sin\alpha$ where τ is the incremented time and α is the sampling angles.

An example of using the direct 2D-FT on generated data is shown in Figure 2.7. The peak position for the data sets was set at (-300 Hz, 75 Hz) and the sampling angle set to 45 degrees. The linewidth was adjusted to 10 Hz by multiplying the data sets by an exponential decay. Further details using this same example are revisited in chapter 4.

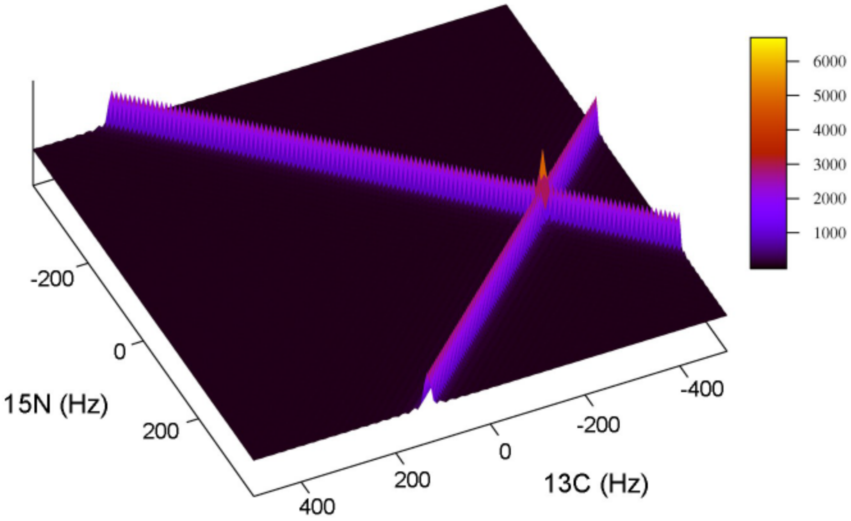


Figure 2.7 An example of the resulting spectrum after a single step two-dimensional Fourier transform. The data was generated with spectral parameters similar to that found in a radial sampled HNCO experiment. The sweep widths were set to 2000 and 1500 Hz for the t_1 (carbon) and t_2 (nitrogen) dimensions respectively. One peak was simulated at -300, 75 hertz with a linewidth of 10 Hz. Radial sampling was realized by incrementing the time in the first dimension as $t_1 = (n/sw_1)\cos\alpha$ and the second dimension as $t_2 = (n/sw_2)\sin\alpha$.

Here, all of the points in the frequency domain were solved for rather than projected from tilted planes. This avoids problems associated with interpolation of data points. However, similar to the projection reconstruction approach, ridges still extend from the peak chemical shifts. All of the methods to remove the ridges presented for projection reconstruction are applicable here.

2.6 Comparison of Sparse Sampling Schemes

The direct multidimensional FT has an additional advantage over projection reconstruction, namely it is amenable to any sampling scheme, not just radial sampling. This occurs because the time points are explicitly defined in the 2D-FT, whereas, the PR techniques use the FFT which assumes equally spaced time points. This allows for a direct comparison of the various sampling schemes. Figure 2.8. shows the resulting spectrum when the various sampling schemes are applied to a generated data set.

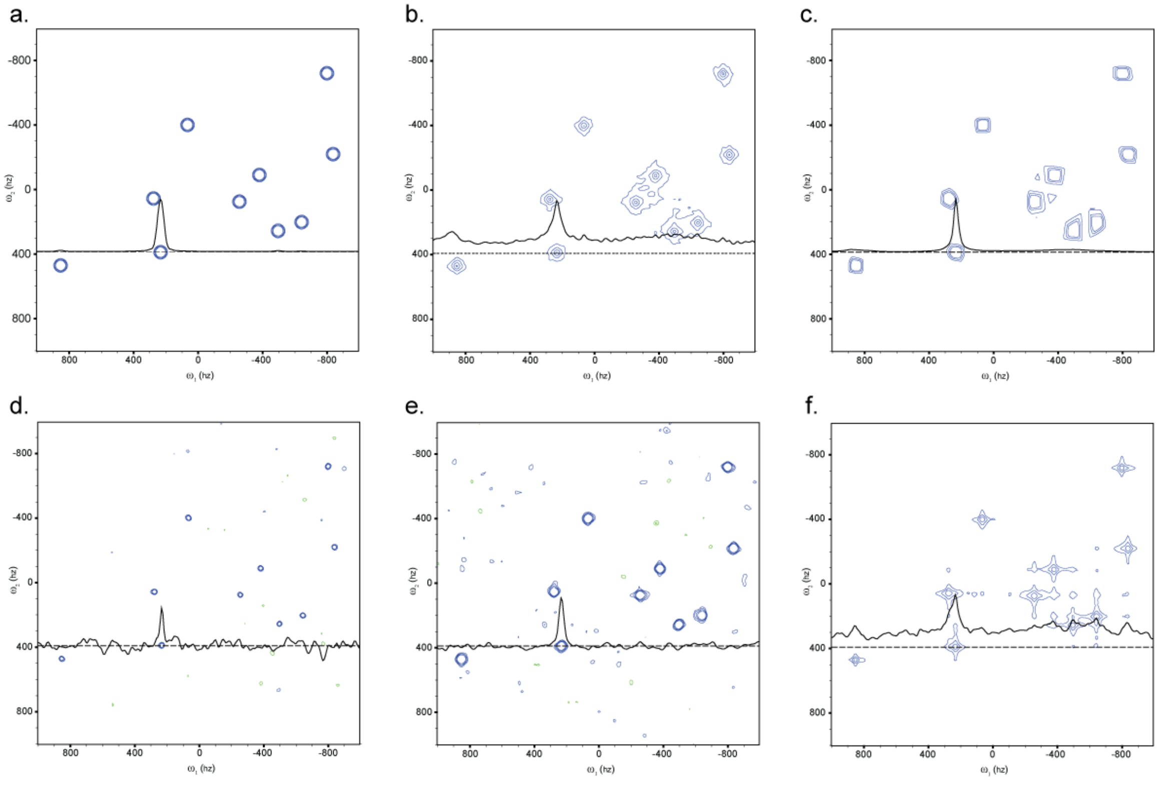


Figure 2.8 Processed spectrum sampling scheme comparison. The reference Cartesian sampled spectrum is shown in a. The radial sampled spectrum processed with ABP and LV are shown in b. and c., respectively.

The uniform random, Gauss weighted random and optimized random samped spectra are shown in d., e. and f., respectively. All spectra were generated from 2500 points using the sampling time points shown in Figure 2.3. The data points for each spectrum were generated using Equation 2.7 using a summation of the ten frequency components seen in panel a. A 0.02 second T_2 was used when generating the data for both dimensions. All spectra were processed using a 2D-FT after applying apodization with a cosine squared function to remove truncation artifacts.

The Cartesian sampled spectrum is shown in Figure 2.8a for comparison. In all of the experiments, the equivalent number of points were generated in order to keep the potential signal volume constant. No noise was added to the generated data, this allows for any baseline artifacts to be directly visible. A 1D slice is shown in all of the spectrum to reference the baseline artifacts. Inspection of the spectrum for all of the sampling schemes demonstrates that all of the peaks are accurately represented, while the baseline artifacts vary for each method. All of the random sampling schemes produce baseline artifacts that appear as noise. Randomized concentric ring sampling also produces baseline artifacts (results not shown). Radial sampling also has baseline artifacts from the ridges extending from all of the peaks, as a function of sampling angle, when no ridge removal technique is applied. When LV comparison is applied the baseline artifacts are removed. To determine the effect of baseline artifacts on the spectrum the signal to noise for the various sampling methods spectra are plotted as a function of data points sampled, Figure 2.9. This figure illustrates the advantage of radial sampling over all of the other sampling schemes, especially when the LV is employed.

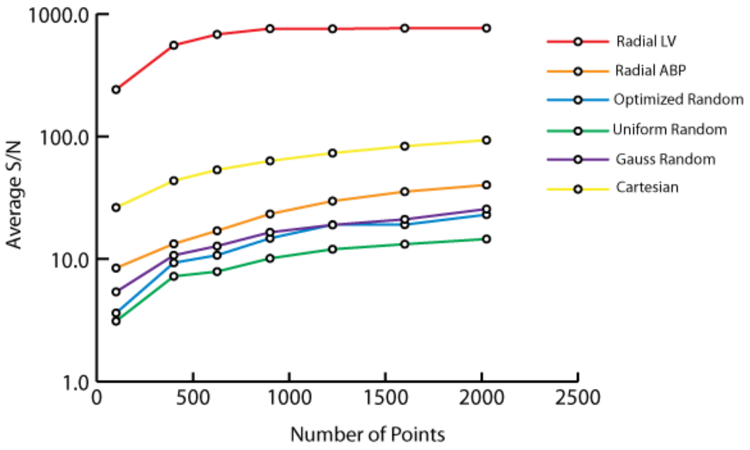


Figure 2.9 Sensitivity Comparison of the various sampling schemes as a function of number of points acquired. The S/N of each spectrum shown in Figure 2.8 is analyzed here. Each point is the average S/N of all 10 peaks in the spectra averaged. When Cartesian sampling was measured the number of points was always increased equally in both dimensions, retaining a square grid. When analyzing radial sampling the number of angles was increased, using approximately 100 points per angle. For the other sampling schemes the points were distributed according to the probability distribution of each sampling type.

2.7 Numerical Methods of Spectral Estimation

Numerical methods are also available to perform the spectra estimation of a frequency domain spectrum on sparse sampled data[16, 17, 31-36]. The general objective of numerical methods is to solve for the spectral parameters, such as the peak chemical shifts, then use the information to generate a final spectrum. In many cases, it is very difficult to generate an accurate representation of the spectrum because the data is corrupted with noise; therefore a deterministic solution is nontrivial. Furthermore, the generated spectrum is not a transform and therefore nonlinear, so the reliability of the chemical shifts is no longer assessable from the noise level in the final spectrum. With this said, methods based on a least-squares fit of estimation parameters to the data have had limited success when the noise level increases; although multiple applications have proven useful when the noise of the spectrum is limited. The successful applications include filter diagonalization[16, 33], maximum likelihood[25, 31], MDD[34] and GFT[17]. One additional method, Maximum Entropy[24], attempts to account for the problems associated with least squares fit, and has had slightly broader success.

Although many successful applications of the various numerical methods have been presented, the application of each technique is dependent upon the data quality and the type of experiment being collected. In order for the work here to be generally applicable the direct multidimensional FT is used here. In most cases numerical methods can also be substituted when applicable.

2.8 Conclusion

Traditional Fourier transform technology requires a large number of data points to achieve a high resolution spectrum. When multiple dimensions are required to resolve spectral degeneracy, the time required to satisfy the linear sampling requirement increases beyond the stability of the spectrometer. Application of sparse sampling allows circumvention of the time limitation of high dimensional NMR experiments. The 2D-FT is used here to process sparse sampled data because of the flexibility to process data collected with any sampling scheme and its ability to directly access the data quality through the noise level of the spectrum.

Comparison of the sampling schemes spectra demonstrate that radial sampling is preferred because of the predictability and ease of removal of the artifacts. Also, the smooth baseline outside of the artifact ridges have superior spectral characteristics compared to the artifacts from random sampling that appear as baseline noise. Finally, comparing the S/N of processed spectra from the various sampling schemes demonstrate that there is a possible sensitivity advantage when using radial sampling combined with artifact removal procedures.

Chapter 3

Al NMR: A Multidimensional NMR Data Processing Package for Cartesian and Arbitrarily Sampled Data

3.1 Introduction

From the previous chapters it should be apparent that the time and resolution advantages of sparse sampling make its general application very appealing. This is especially true in the case of large proteins. As protein molecular weight increases, quite often spectral degeneracy increases concomitantly. Sparse sampling offers an increase in acquisition time which enables higher dimensional experiments to be collected in the same time as the lower dimensional analog.

Although methodology has been developed that utilizes the gains of sparse sampling processed with a direct multidimensional Fourier transform (2D-FT)[27, 48-50]. There is no program available generally available to handle all aspects of sparse sampled data processing. Typically a combination of current processing programs and an external 'in-house' program is utilized for processing. Subsequently, convential programs are available to display and analyze the data, such as, Sparky[51] or Felix (Felix NMR, San Diego, CA).

Two programs are used to process the data because the directly detected dimension is processed with traditional fast Fourier transform methodology, while the indirect dimensions, that utilized a sparse sampling pattern, are processed with a direct multidimensional Fourier transform. In order to process all aspect of sparsely sampled data, including the direct and indirect dimensions, a new data processing package is presented here.

Al NMR incorporates all traditional NMR data processing methodology and new multidimensional Fourier transformed based methodology. The processing program is based on the python scripting language which is becoming one of the standard languages in scientific data analysis. Further, multiple programs, XPLOR-NIH[52] and Sparky[51], familiar to most NMR spectroscopists, utilized the python language.

3.2 Sparse sampling data processing with direct 2D-FT

To demonstrate the processing procedure, a simple example, employing radial sampling[25] for generated (3,2) data set is presented. Here radial sampling is accomplished, in the case of a 3d experiment, by simultaneously evolving both dimensions while collected the directly detected dimension normally. The simultaneously evolved dimensions are set such that the incremented times are $t_1=\tau\cos(a)$ and $t_2=\tau\sin(a)$. Where t is a common, linearly incremented time and a is the radial angle between the two orthogonal time domains. An example of the time points sampled by radial sampling, for a single angle is shown in Figure 3.1. Here, the data was generated using a 45° sampling angle between the two indirect dimensions. For each sampling point in the time domain in Figure 3.1 there are 8 corresponding quadrature components, a real and imaginary for each dimension.

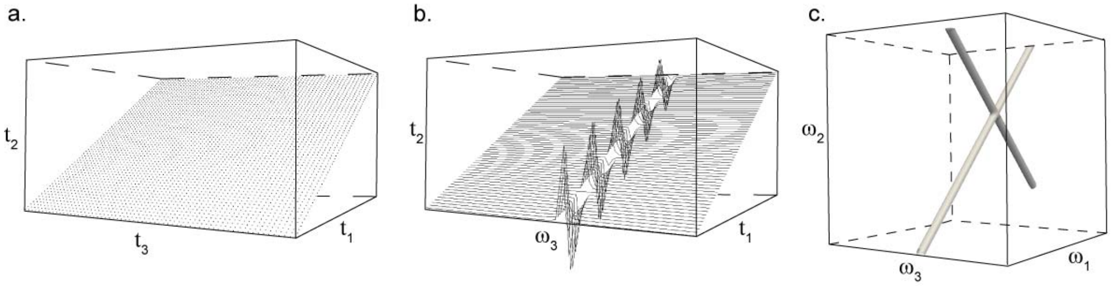


Figure 3.1 Radial sampling data processing example for a 3D spectrum of generated data, with a single peak. a. The time points are collected using a Cartesian basis with respect to the directly acquired dimension. Radial sampling is used in the indirect dimensions by sampling t_1=τcos(a) and t_2=τsin(a). A 45° sampling angle is used. b. The directly acquired dimension is processed using a FFT which results in a mixed mode, frequency, ω_3, time, t_α spectrum. c. The direct 2D-FT is used to process the two indirect dimensions which generates the final frequency domain spectrum.

To process the data set, the directly detected dimension, t_3, which was collected traditionally, is processed using traditional Fourier transform methodology[22]. Each vector along, t_3, is processed separately, as is typically done, including convolution, apodization, zerofilling and Fourier transformation. Processing of t_3 produces a mixed mode spectrum, Figure 3.1b. The spectrum is in the frequency domain along w_3 and the indirect dimensions are still in the time domain. The intensity of the peaks along w_3 is modulated according to the frequency of the peak in the indirect dimensions. This produces an interferogram along the radial sampling angle. For clarity, only one of the four quadrature components is shown in the Figure.

After the directly detected dimension is processed, the two indirect dimensions are processed simultaneously using the direct 2D-FT[45-47]. The 2D-FT generates a 2D frequency matrix for each vector perpendicular to w_3. Typically, each vector is apodized, using a 2D apodization function, or weighed[27] before processing with the direct 2D-FT. The resulting 3D spectrum, after all processing, is shown in Figure 3.1c. As a result of the indirect time domains being underdetermined artifact ridges extend from the authentic peak chemical shifts at 90 +/- the sampling angle. The artifact removal methods for projection reconstruction can be applied here to generate a final, artifact free, spectrum.

The direct 2D-FT is discrete, which requires, that all of the time points and frequency pair values are supplied to the function. By supplying all of the necessary parameters allows the 2D-FT to process data regardless of the sampling scheme employed.

Randomly sampled data is processed using the same flow of operations as presented for radialy sampled data. The random sampled data is collected traditionally in the directly detected dimension and processed with the FFT. The indirect dimensions are sampled simultaneously by randomly selecting coordinate times in the evolution domains of the two indirect dimensions. The time point schedule is recorded and utilized by the

2D-FT to generate a final frequency domain spectrum. Prior to application of the direct 2D-FT the data can be weighted or apodized to increase the resulting spectrum quality. The random time point selections can be modified by weighting the selection criteria to reduce artifacts that are intrinsic to the sampling scheme.

3.3 Al NMR Program Architecture

Al NMR is designed to be a standalone processing package. It does not depend on any of the currently available processing packages for functionality. By designing the program in a self contained manner, allows for increased flexibility to process data collected with arbitrary sampling schemes and dimensions. Additionally, the program is designed to be extendable, by the end user, to develop new methodology. To achieve this flexibility Al NMR uses a Python interface[53]. Python is an established, efficient scripting language. The language is easily learned. Currently, multiple programs, designed for NMR utilize a Python interface. By designing the program as a standalone module data processing is streamlined. Raw data is read directly from the spectrometer and a processed output spectrum is generated. Utilizing Python no scripts needs to be compiled.

The program architecture is shown in Figure 3.2. All aspects of the program are controlled by the user, either through the command line or by a user supplied script. The script or command is passed to the python interrupter which commands subsequent functionality of the program. Typically a script will, at a minimum, load the Al NMR module, read the data from the spectrometer file, process the time domain data into the frequency domain and generate a processed matrix file. All of these commands, listed in the script, are parsed by the python interrupter and passed to the python engine. The first step when a script is executed is to load the Al NMR module. The module extends pythons functionality to include reading, writing and processing NMR data. All of the standard python functionality is retained. With direction from the script, the program reads the NMR data file and creates a data object. The data object includes all of the relevant acquisition parameters and access to the FIDs. Currently, the program supports either Varian or Bruker data files. Further direction of the script controls access of FIDs from the data object and process of the FIDs. In many cases each FID is processed identically. This step can be performed in a parallel using built-in python functions. Finally, the script controls creation of a matrix file object which interfaces with a final matrix file. Currently, either a Sparky or Felix matrix file can be written.

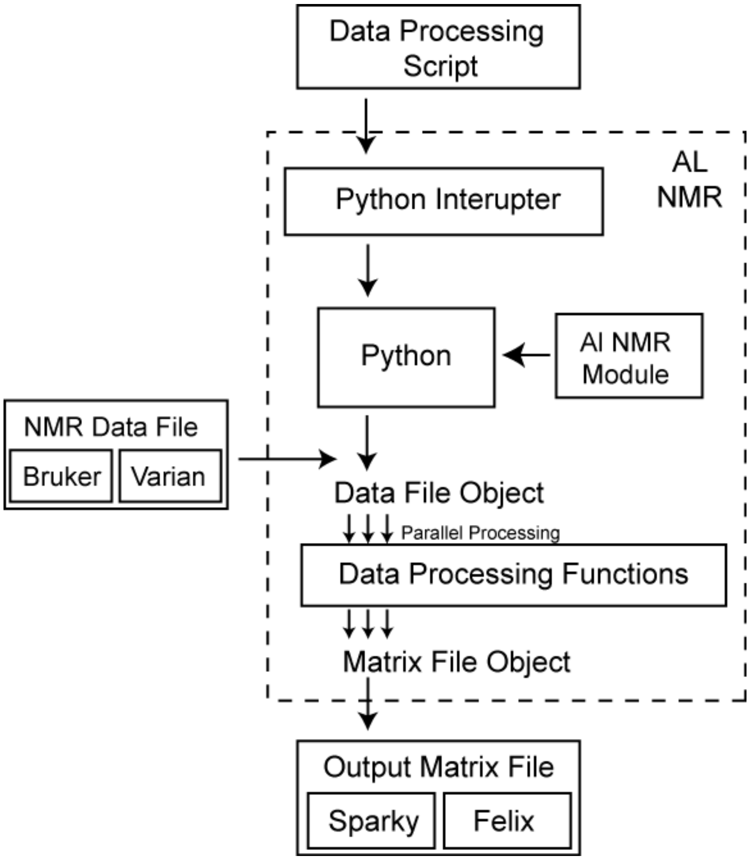


Figure 3.2 Al NMR program architecture.

Python makes up the core of the processing program. Python is one of the most popular, cross platform scripting languages available. Because it is a scripting language the end user doesn't need compile scripts and scripts are easily shared with a colleague using a different operating system. Further, because of the popularity of python there is a large body of tutorial and books available to learn the language. Additionally, when debugging a script the large support environment is advantageous as compared to a scripting language developed specifically for data processing.

Technically, python is desirable because it takes care of all of the overhead in developing a scripting package. It includes means for command interpretation on the command line and through scripts. All methods of looping and computational details such as memory allocation are available. Additionally, utilizing python, scripts can developed easily that include technically complicated concepts such as queuing and multithreading. Finally, utilizing python the core functionality can be expanded with a library of user developed codes. This allows for the package to be extended to meet any users needs.

3.4 Python introduction

An introduction to the python scripting language is presented here to assist the user with developing scripts. Only salient features of the scripting language are presented. The user is referred to many good online tutorials for a complete introduction (www.python.org). The core python functionality contains all of the typical data types, such as int, float, etc. All variables in python are dynamically defined, that is, no memory needs to be reserved prior to using a variable. For example a variable `x` is defined by `x = 5`. When `x` is defined, python parses the data type, integer in this case, and reserves the appropriate amount of memory. Along with the standard data types, python also includes a set of container types, such as lists. Lists contain an ordered group of elements. The elements in a list can be any standard or user defined type. Brackets are used to define a list. For example `angles=[10,45,90]`. Here `angles` is the list which contains three elements. An element of the list is accessed by using the list name and element number: `angles[1]` returns 45. All lists start at the zeroth element.

Lists also contain a built in methods to loop through all of the elements. Which is accomplished by using a for loop. All loops in python are defined by formatting blocks. After a `for` statement the block of subsequent steps in the loop are defined using indentation. For example the elements of the list angles are iterated over, 5 added, and then printed to standard output by the following.

```
for x in angles:
     x=x+5
     print x
```

Loops can also be defined using a `while` statement as shown here:

```
while a < 3:
     x=angles[a]+5
     print x
```

This example also uses a built in logic operator to define the loop. Logic code blocks are defined with the same indentation procedure.

Python is amenable to user extension of the language by importing modules. Modules expand python by declaring new data types, objects and functions. A module is imported my declaring the statement:

```
import alnmr
```

Here the alnmr module is imported to provide all of the NMR data processing functionality to python. In order to use the functions included in the module the interpreter needs to be directed that the function is located inside of the module. The dot operator is used to specify this to the interpreter, this is demonstrated in the example:

```
alnmr.fft(data)
```

This command instructs the interrupter that `fft` is a function found in the `alnmr` module. Multiple modules can be loaded during execution of a single script. Using the `import` command, a user created library of functions can be imported

3.5 Al NMR Data Processing Module

As stated above, when a module is imported, new data types, objects and functions are added to the standard functionality of python. Upon importing Al NMR two new data objects are available and all of the data processing functions. The data objects handle all aspects of reading data files and writing matrix files. All of the data processing functions are shown in Appendix 1. It is important to note that NumPy was utilized in developing this module[54].

There are two types of data objects, one for NMR data files and the other for writing matrix files. Data objects are created to access NMR data files. Currently, both

Bruker and Varian data files are supported. To create a data object, one of the following commands is issued for Bruker or Varian data respectively:

```
datobj  = alnmr.readbruker('data directory')
```

```
datobj = alnmr.readvarian('fid directory')
```

For both functions `datobj` is the name of data object that is created to access the data files. This name is arbitrary. The data and fid directories need to contain complete the path to the specified directories. If the directory or expected files inside of the directory do not exist, an error is returned. There is no limit to the number of data objects that can be created simultaneously to access multiple data files. Working with multiple data objects is particularly import when processing multiple radial sampled angle spectra.

The data object performs all aspects of reading fids from the file. When a data file object is created two event occur: First a file stream is created to access the data matrix. Second, all of the relevant parameters from either the acqu or procpar file, depending on the type of spectrometer used to collect the data, are read and stored. A complete list of the parameters recorded is listed in Appendix 1. The parameters are accessed from the data object in the same manner that functions are accessed from the module, using the dot operator. For example after a data object is created the total number of data points for each dimension is accessed by the following command:

```
tdlist=datobj.td
```

This command returns a list of the total data points for each dimension. The length of the list is equal to the dimension of the experiment.

FIDs are accessed from the data object using the `alnmr.readfid()` command. This command reads one FID from the file and returns the data as a list. When the data object is created, the data file position is set to the first FID. The file position is advanced to sequential FIDs each time the `readfid` command is issued. The read FID command can be directed to read non default fid numbers if the ‘`fidnum`’ option is used as an argument when the command is issued. The fid number can range from 1 to the the total number of fids in the file. Two other options are available for the `readfid` command, `byteswap` and `resize`. The `byteswap` option corrects for a difference in the endianess of the data collected and the system architecture used to run Al NMR. `Resize` removes any trailing zeros that might be included in the fid. Trailing zeros arise from Bruker digital to analog data format conversion.

Matrix output file objects are created in a similar manner to input data file objects. Currently, two output matrix formats, Sparky and Felix, are supported. The commands to create the matrix objects for the two formats are:

```
matrix_obj=alnmr.sparkymat(‘filename’,[d1,d2,d3,…,dn])
```

```
matrix_obj=alnmr.felixmat(‘filename’,[d1,d2,d3,…,dn])
```

When the command is issued a matrix file, as specified by the `filename` option, is either opened or created. The program decides to either open the file or create it depending if the optional matrix dimensions, [d1, d2, d3], arguments are supplied. If dimensions are supplied than a matrix file is created. Else the program attempts to open the matrix file. If no file exists, then an error is returned. Like the data file objects, there is no limit to the number of matrix file objects that can be used simultaneously. Again, this is particularly appealing when dealing with radial sampled data. A unique matrix can be created and accessed, simultaneously, for each sampling angle collected.

When the matrix object is created all of the necessary parameters are either read from the preexisting matrix or generated. A complete list of all the objects parameters is listed in Appendix 2. These parameters are accessed in the same manner as the data object, using the dot operation. For example the size of each matrix dimension is accessed with the following command:

```
dimlist=matrix_obj.matdim
```

Here the returned value is a list of the size of each matrix dimension. The length of the list is equal to the dimension of the matrix.

After the matrix object is established the matrix_obj.read(c1,c2,…,cn) and matrix_obj.write(data, c1,c2,…,cn) commands are used, respectively, to read and write information to and from the file. In the case of the read command the arguments are the coordinates of a point or vector to read. The coordinates are supplied as sequential integers with one value for each dimension, (c1,c2,…,cn). Each dimensions numbering starts at 1, which corresponds to the first point. If one of the dimension points is specified as zero then a vector of data, spanning the dimension, is returned. Only one dimension can be set to zero for each read command. The write command address the coordinates of points and vectors the same way as the read command. When this command is issued, an additional variable, data, is supplied. The data variable is either a point or vector of data. A data point or vector are supplied as a real integer or float.

The program optimizes access to the matrix file using the same block format present in Sparky and Felix. All interaction with the blocks is preformed automatically. Reading and writing to the files has been optimized using a memory buffer system. All of the parameters, regarding the blocks in each dimension, are accessible using the dot operator as before. Again, the parameters are listed in Appendix 2. It is important to note that because a buffer system is used, changes to the buffers have to be committed to the file with the `matrix_obj.update()` command.

All of the data processing functions are listed in Appendix 3. Functions are available for all standard data processing steps, such as: FID manipulation, (i.e. adding fids and zerofilling ; convolution; apodization; linear prediction; Fourier transform and phase correction). Most functions take a fid as input and return the modified data. For example, the following command takes a fid, generated from issuing the `readfid()`, command to act on the data object, the Fourier transform of the fid is preformed and a new data list is returned.

```
ftdata=alnmr.fft(fid1)
```

All of the traditional processing functions, which act on one dimensional data, accept either real or complex data as input. When quadrature detection is employed the data is stored in a complex list. The real component of the list is the cosine modulated data and the imaginary component is the sine modulated data. The functions that act on either random or radial sampled data use a sequential format because there are four quadrature components. There are typically two quadrature components per dimension. Each of the quadrature components are list sequentially as real floats in the data list. When two dimensions are co-evolved the expected data order is cos-cos, sin-cos, cos-sin, sin-sin.

The analogous, processing functions are available for coevolved 2D data. The primary difference of these functions is that they are discrete, unlike 1D data, there are no assumptions that the data is equally spaced. Therefore, all sampling time points, frequency components and sampling angle, if applicable, are specified. For example the arbitrary sampled 2D sinebell squared apodization function function is called with the following command:

```
2d_data=alnmr.ss2dgen(data, sampling_time_points, t1max,
          t2max, shift1, shift2)
```

Here, `data` is a list of data with four quadrature components per sampling increment. The `sampling_time_points` is a list containing the t_1 and t_2 sampling times selected for each increment. The list contains two elements, listed sequentially, for each incremented point sampled. The t_1 time point is listed first, followed by the t_2 time point.

T1max and t2max are the maximum evolution time used for the two time domains. These times are included to specify the times where the apodization function is set to zero. Finally, shift1 and shift2 are the analogs to the 1D sinebell squared shifts with respect to the two dimensions.

3.6 Direct 2D-FT

Defining all of the direct two dimensional functions discretely adds significant complexity to application of these functions compared to the 1D functions. However, the added complexity enables a substantial increase in flexibility of the functions. This is particularly true in the case of the direct two-dimensional Fourier transform (2D-FT). The 2D-FT is called using the following command:

```
2dmat=alnmr.ft2d(data, sampling_time_points, freq1, freq2,
          [ph0a, ph1a, ph0b, ph1b])
```

As before, `data` and `sampling_time_points` are one dimensional lists that contain time domain data and sampling points respectively. `freq1` and `freq2` are the frequency ranges that the data is Fourier transformed into. Typically, the two frequency domains are defined as a list spanning the sweep width used for sampling centered upon zero. Practically this accomplished by setting the first point of the list to $-\frac{sw}{2}$ and the last point to $\frac{sw}{2}$. The number of points in the list is defined by the user by is typically at least two times the number of increments sampled to achieve the same effect as zerofilling 1D data. Therefore the increment of the list is $\frac{sw}{2ni}$. The frequency lists can be generated automatically, by supplying the number of sampling points and sweep width to the following command:

```
freqlist=alnmr.ftfreq(np,sw)
```

The final arguments supplied to the 2D-FT are the zero and first order phase corrections for both the t_1 and t_2 sampling domains. The zero order phase corrections are supplied as degrees and the first order phase corrections are supplied as time values. By default the values, if not supplied, are set to zero. Details of how to determine phase correction for two dimensional data is presented in Chapter 4.

Defining the frequency ranges values to the 2D-FT, the programs allows any region of a spectrum to be processed. As demonstrated in Figure 3.3, an entire sweep width range for both dimension can be utilized or just a sub region of each frequency range. Additionally, single pairs of frequencies can be supplied to generate the Fourier transform at a single point. In turn, generating single points allows for a vector of points to be specified that span across a spectrum, Figure 3.3c. Generating the Fourier transform of a vector is particularly appealing when radial sampled data is analyzed or integrated.

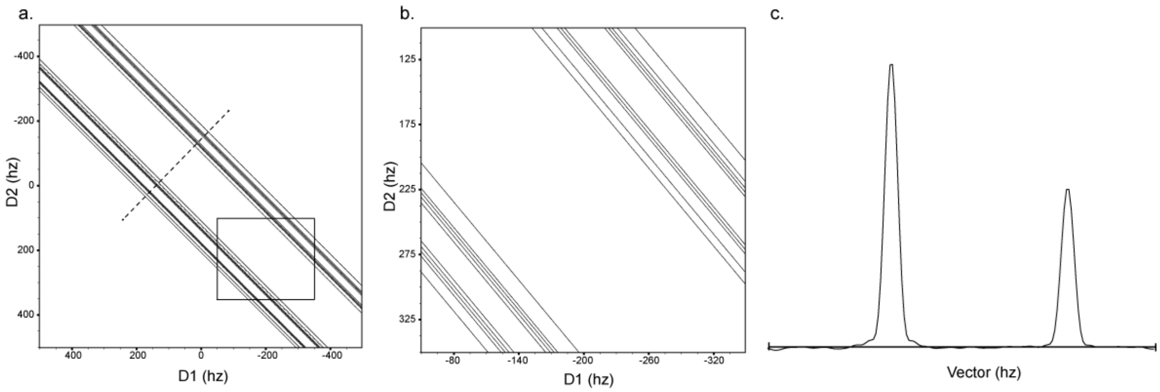


Figure 3.3 A demonstration of the intrinsic flexibility of the 2D-FT to process any region of a spectrum is shown here. The negative 45° sloped ridge component spectra, contain two peaks of generated data, is used for this example. 1000hz sweep width were used for both dimensions. a. The full sweep width was used during processing. b. only a sub-2D window of frequency components were supplied, defined the inset box of a. c. A 1D vector of frequency components are transformed. The frequency components used are shown in a as the dashed line.

3.7. Example Data Processing Scripts

In this section examples of using Al NMR to process both traditionally Cartesian sampled and sparse sampled NMR data are presented. The script codes are supplied in the Appendices 4-6. Three examples were chosen to demonstrate the flexibility of the program. The first example demonstrates how to process a ^{15}N HSQC. An example demonstrating how to phase correct the spectrum is then presented, which utilizes the interactive phase correction interface. Finally, an example demonstrating how to process a 3D radially sampled HNCO is presented.

3.7.1 ^{15}N HSQC Processing Example

A flow chart of the processing script is shown in Figure 3.4. The source code for this script is supplied in Appendix 4. This example was selected because it incorporates all of the essential concepts of processing Cartesian sampled data.

The first step in all Al NMR processing scripts is to import the alnmr package. Additional packages can also be imported, here the python `os` module is imported to allow for easy file path modification across multiple operating systems. Using the `os` module to define the paths simplifies migration across multiple operating systems. The nested directory names, for the input data file and output matrix file are supplied as sequential elements of a list. The elements of the list are used subsequently to define the path using the `os.path.join` command. Prior to generating the paths final output name of the matrix and the matrix dimensions are defined. In this case, only directory names are modified and not the forward or backward slashes used when creating the directory. Next, all of the referencing values that the experiment was collected with are supplied. The phase corrections are also defined. Means to determine the appropriate phase corrections are presented in the following example. The script then generates the paths for both the input data file and the output matrix. After the paths are generated a data object, bdat, and matrix object, smat, are created. As above, the dimensions are supplied when the matrix file object is created which generates a new matrix file. When the matrix file is created, all of the points in the matrix are initialized to zero.

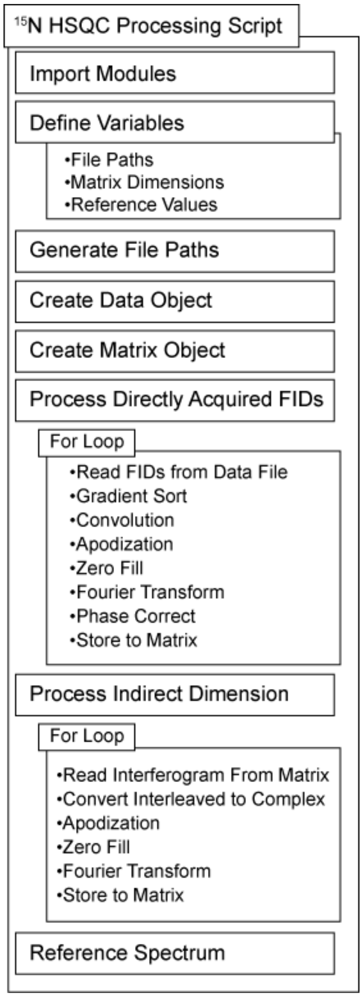


Figure 3.4 ^{15}N HSQC processing script flow chart.

Having established objects for reading and writing the data, the script now processes the data. Each fid is processed by using a `for` loop to iterate over all of the fid numbers. The list of fid numbers is generated using the built-in `range` function. Al NMR relies solely on instruction from the processing script. There are no built-in functions for choosing the appropriate processing functions for a given quadrature mode. Although, it is possible to generate scripts that utilizes the acquisition parameter read in the data object to automatically process the data. This processing script is designed for a sensitivity enhanced, gradient selected experiment[15]. In order to process such data, two FIDs are read then added and subtracted, to select the appropriate components. Because two fids are read at a time, the number of steps in the loop is half of the total number of increments. There is a 90° phase difference between the two components as a result of the sensitivity enhanced data acquisition[55]. This is corrected by exchanging the real and imaginary components using `alnmr.exchange()` and taking then negating the imaginary component by taking the complex conjugate of the data with `alnmr.conjugate()`.

After the two sensitivity enhanced FIDs are generated, the two fids are processed, independently, using traditional methodologies[24]. The processing steps include subtracting a polynomial, using `alnmr.polysub`, for water signal removal, apodization by multiplying a shifted sinebell squared function using, `alnmr.ss1d()`, zerofilling with `alnmr.zerofill()` and Fourier transforming the data. All of the processing steps, including the Fourier transform, accept a complex fid and return a modified complex FID. In the case of the Fourier transform a complex data list is returned with the FT absorptive spectrum stored in the real values and the dispersive spectrum stored in the imaginary values. When both the absorptive and dispersive components are available they are used for phase correction of the data using the `alnmr.phase()` command.

The final steps of processing the directly acquired dimension of this experiment is to discard the downfield half of the spectrum, which, if the carrier frequency is set on water, does not contain any amide proton chemical shifts. This data is deleted using the `alnmr.delete()` command. The dispersive component spectrum is discarded using the `alnmr.reducecomplex()` command. Finally, the two fids are written to sequential positions in the matrix file, as temporary storage, prior to processing the indirect dimension. All of the elements of the fid are wrote at once by supplying 0 for d_1 point at indirect dimension points command. The FIDs are committed to the matrix with `matobj.update()` command.

The indirect dimension is processed by iterating over each vector, which spans the indirect dimension, for each point in the d1 dimension. As before, a `for` loop is used by iterating through the elements of a list generated with the range function. Each vector is read from the matrix by supplying zero as an argument for the read function in the d_2 dimension. In this case there are more points in the matrix than there are data points. The points that do not contain any information are deleted with the `alnmr.delete` command. Unlike the directly acquired data, which is always read from the file as complex, when the indirect vector is read the data is returned as a real values with the real and imaginary components interleaved. The interleaved data is converted to complex with the `alnmr.complexdata` command. The final step before the fid is processed is to multiply the first complex point of each interferogram by .5 to correct for baseline offset associated with using a zero delay for the first point. This data set was collected using gradient selection for quadrature determination[56]. If the data was collected using States-TPPI[57] then a correction, by multiplying every other complex point by negative one should be performed at this step. The command `indvec[1::2]*=-1` performs this operation.

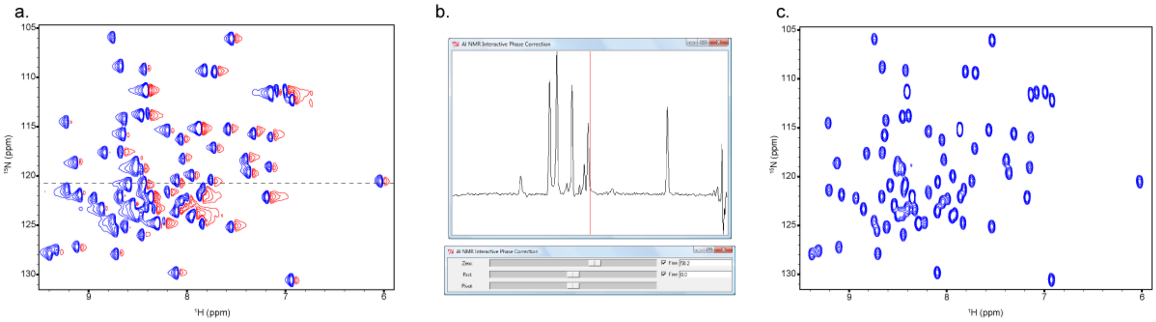
After the complex fid has been read and corrected, the data is processed in a similar manner to the directly acquired dimension using apodization, zero filing and then Fourier transforming. This script does not include a linear prediction step, but the package does include linear prediction using either the `alnmr.lpinv` or `alnmr.lpsvd` commands. The two commands use matrix inversion and singular value decomposition, respectively, to determine the linear prediction coefficients[24]. The final step of processing the indirect dimension is to write the vector back to the same location in the matrix. After all vectors have been processed the matrix access is closed using the `mat_obj.close()` command. When the matrix object close command is issued all changes are committed to the file prior to it actually closing. Therefore the `matobj.update` command does not need to be issued. Access to the data file is also closed at this point. The final step in processing the matrix is to update the referencing information for both of the dimensions using the `alnmr.refsparky` command.

This script only processes two dimensions, additional dimensions would be processed using a similar routine for each of the indirect dimension. In the case of a three dimensional experiment, an additional level of nested loops are needed to cycle through all of the vectors. The vectors would be accessed using three coordinates rather than two.

The general processing routine of reading, processing and writing back to the matrix will remain the same.

3.7.2 Interactive Phase Correction Example

In the ^{15}N HSQC processing script, a phase correction was applied to the data after Fourier transforming. Here a script to determine the appropriate phase correction is presented. The script is shown in Appendix 5. Prior to using this script a spectrum is processed without application of any phase correction. The spectrum is inspected with appropriate NMR analysis software and a vector to use for phase correction is noted. An example spectrum, without any phase correction, is shown in Figure 3.5. The dispersive lineshape along the proton dimension indicates that phase correction is necessary. The dashed line indicates a selected vector for phase correction.



Al NMR Interactive Phase Correction

Zero ☑ Fine 58.2

First ☑ Fine 0.0

Pivot

Figure 3.5 An example of the effects of using the Al NMR interactive phase correction interface. a. A 15N HSQC spectrum of ubiquitin with no phase correction along the proton dimension. A vector for phase correction is selected, indicated by the dashed line. The vector is supplied to the phase correction interface,

b. The sliders are adjusted to determine the necessary phase corrections. c. The phase corrections are used to reprocess the spectrum.

Once a vector has been selected, the coordinates of the vector are supplied to the phase correction script. When the script is run, it first imports the necessary modules and defines the path where the spectrum is stored. A matrix object, `smat`, is created to access the file. The selected vector is read from the matrix using the `smat.read()` command. As described in the HSQC processing macro, when the data is stored the imaginary component is discarded prior to storage. Therefore, when the vector is read, it only contains the real component. The imaginary component is generated by a Hilbert transform, using the command `alnmr.hilbert()`. The resulting complex vector contains both the real and imaginary components which are 90 degrees out of phase. Both complex components are needed for the alnmr interactive phase correction interface. The interface is started with the command, `alnmr.interactivephase()`. When the interactive phase command is issued two windows are opened, as seen in Figure 3.5b. The top, display, window contains the vector selected for phase correction and the bottom, control, window contains the real-time phase correction interface. Changes in the slider positions are represented in the display window. Zero and first order phase corrections are adjustable with the sliders or by direct input into the text boxes. The first order pivot is adjusted by the third slider. Changing the pivot value moves the red indicator line in the display window.

After the phase corrections have been determined, the values can be substituted into the processing macro to reprocess the data. The resulting spectrum is shown in Figure 3.5c. Inspection of the lineshape indicates that no additional phase correction is necessary for this spectrum. If needed, this script can also be used for the indirect dimension by modifying the vector coordinate values.

To avoid reprocessing all dimensions of a spectrum, a script could be generated to read every vector of a selected dimension, perform a Hilbert transform, phase correct using the `alnmr.phase()` command and write the phase corrected vector back to the matrix. This approach is especially appealing for experiments with three dimensions and greater.

3.7.3 3,2 Radial Sampled HNCO Processing Example

The two previous scripts have described means to process and phase correct Cartesian sampled spectra. Here, a script to process radial sampled data, that exploits the versatility of Al NMR, is described. The general processing scheme is depicted in the flow chart of Figure 3.6. The complete processing script is included in Appendix 6.

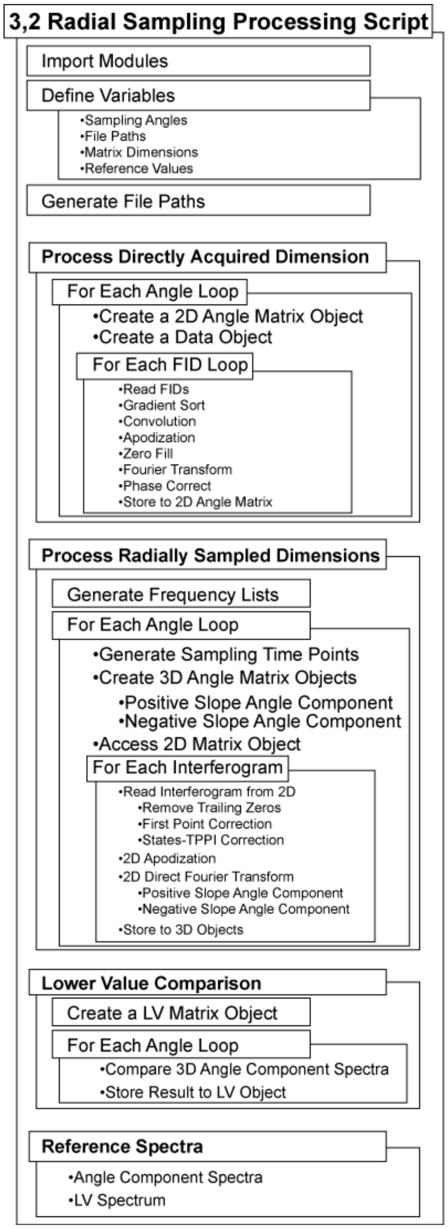


Figure 3.6 Flow diagram of the processing operations for a 3D radial sampled experiment.

The script is designed to process radial sampled data where multiple sampling angles have been collected independently, using separate experiments. There are four primary sections of the script. The first section initializes all of the appropriate variables and defines the necessary paths. Section two processes the directly acquired dimension for each angle data set. Subsequently, section three processes the two radial sampled dimensions for each of the angle ridge components. The final section performs a lower value comparison using all of the angle matrices processed in section three. Spectra are generated for each of the angle ridge components, as well as for the lower value.

Similar to the HSQC example, the first step of the script is to load the `alnmr` and `os` modules. Paths and variables are defined, including a list of the sampling angles and corresponding experiment numbers. These lists are stored using the `angles` and `expnum` variable names. After the appropriate definitions are made the script processes the directly acquired dimension.

In the case of a 3D radial sampled experiment two of the dimensions are coevolved at a supplied sampling angle. In practice this results in the collection of a 2D matrix of data points, where one dimension is directly acquired and the second is the coevolved dimensions. To process the directly acquired data, each FID is read from the data object is processed and stored to an intermediate 2D matrix. To accomplish this, the script builds a 2D matrix for each sampling angle. Then a loop is used to process each fid storing the results in the 2D matrix. As described above, each radial sampled increment contains four quadrature components. Each set of quadrature components are processed simultaneously to allow for the appropriate gradient sorting to be performed. Subsequent steps in processing are equivalent to the directly detected dimension of the HSQC, including, convolution, apodization, zero filling and Fourier transformation[24]. The four processed quadrature component FIDs are stored sequentially in the 2D matrix.

After processing the directly detected dimension, the two indirect dimensions are processed simultaneously. Each radial sampled dimension can be separated in a positive and negative ridge component using a combination of matching and non-matching Fourier transforms, as described in chapter 4. To isolate the angle component spectra each angle is processed independently using a for loop. Two matrix objects are created for each angle, as well as, an object for the appropriate 2D matrix that contains the processed directly detected dimension.

Each vector of the directly detected dimension is processed independently using a for loop. Similar to the HSQC example an entire vector of data is read from the 2D file and then trailing zeros are removed. Unlike the HSQC example the data is not converted to complex, rather each of the four quadrature components are left sequential in the vector. The first incremented time point is scaled by multiplying each of the first four elements by .5. Accordingly, the States-TPPI[57] correction acts on every other four components. The data is then apodized using the `alnmr.ss2d` function. The component angle spectra planes are then generated using the `alnmr.ft2dplus` function. Finally the data is stored to the appropriate 3D spectra.

To generate a final spectrum, absent of any ridge artifacts, the lower value (LV) comparison is employed. Al NMR has an optimized lower value comparison that acts on the entire matrix simultaneously, as opposed to comparing individual vectors sequentially. The results of the lower value comparison are stored in a new matrix. First, the two component angle spectra are compared and stored to the LV matrix. Subsequently, all of the component angle spectra are compared to the LV spectrum, and the new results are used to replace the old values in the LV spectrum. After LV comparison is completed, all of the spectra, including the LV spectrum, are appropriately referenced using the supplied valued.

3.8 Conclusion

In conclusion, a complete data processing package has been presented. This package enables a user to directly process both Cartesian and sparse sampled data. Core to the package are the necessary features to read FIDs directly from the spectrometer format, process the data and write a spectrum directly to an analysis program file format. Currently, the program is capable of reading both Varian and Bruker data formats. All of the traditional data processing functions are available, as well as, the equivalent functions for sparse sampled data. The package is capable of creating spectra files that amenable to direct analysis with either Sparky[51] or Felix. Finally, this program will serve to make sparse sampling generally applicable and facilitate development of a broad range of applications that utilize sparse sampling to decrease acquisition time.

Chapter 4

Phasing Sparse Sampled Multidimensional NMR Data

4.1 Introduction

In the previous chapter, a new data processing program was presented. The program utilizes the direct 2D-FT to process sparsely sampled data. When radial sampling is employed, the transformation results in a fundamental artifact manifested as a ridge of intensity extending through the peak positions perpendicular to +/- the radial sampling angle. The package includes a number of methods to remove the fundamental ridges artifacts, but as we will emphasize below, successful removal of the ridge artifacts requires absorptive line shapes.

Unfortunately, no general procedure for phasing radially sampled NMR data has been presented. Indeed, the emphasis thus far has been on the collection of time domain data that is free of phase distortion or error. Obviously a procedure for retrospective phase correction of radially sampled data is a distinct advantage. Here we present two methods capable of phase correcting arbitrarily sampled NMR data as a solution to this problem.

4.2 Theory

As above, the discrete 2D-FT can be described as[45-47]:

$$S(\omega_1,\omega_2)=\sum_{t1=0}^{t1\max}\sum_{t2=0}^{t2\max}\exp(-i\omega_1 t_1)\exp(-j\omega_2 t_2)f(t_1,t_2)g(t_1,t_2)w(t_1,t_2) \qquad (4.1)$$

Where i and j are quarternion numbers; t_1, t_2 are the incremented times, α_1 and α_2 comprise the frequency pair being determined, $f(t_1,t_2)=\exp(-i\Omega_1 t_1)\exp(-j\Omega_2 t_2)$ is the data being transformed, Ω_1 and Ω_2 are the chemical shifts for time domain t_1 and t_2 respectively, $w(t_1,t_2)$ is a weighting factor to account for the unequally spaced sampling of the time domain and is typically applied as a two dimensional apodization function, and $g(t_1,t_2)$ describes the lifetime of the signal, which we will subsequently ignore. In the case of radial sampling $t_1=\tau\cos\alpha$ and $t_2=\tau\sin\alpha$ where τ is the incremented time and α is the sampling angles.

In accordance with standard Fourier transform quadrature theory, if the carrier frequency is set in the middle of the spectral ranges, eight pieces of data must be collected in order to determine the sign of the frequency components for a three-dimensional spectrum. Typically the proton dimension is processed separately; therefore we will only deal with the indirect evolution terms here. This simplification leaves four terms that are modulated by a mixture of cosine and sine as presented below.

$$f_{CC}(t_1,t_2) = \cos(t_1\Omega_1)\cos(t_2\Omega_2) \tag{4.2a}$$

$$f_{CS}(t_1,t_2) = \cos(t_1\Omega_1)\sin(t_2\Omega_2) \tag{4.2b}$$

$$f_{SC}(t_1,t_2) = \sin(t_1\Omega_1)\cos(t_2\Omega_2) \tag{4.2c}$$

$$f_{SS}(t_1,t_2) = \sin(t_1\Omega_1)\sin(t_2\Omega_2) \tag{4.2d}$$

Four real Fourier transformations can be used to process the four data sets, which we term the cos-cos Fourier transform (CC-FT), the cos-sin Fourier transform (CS-FT), the sin-cos Fourier transform (SC-FT) and the sin-sin Fourier transform (SS-FT). The CC-FT is used to transform the cos-cos modulated data set (Equation 4.2a), the CS-FT to transform the cos-sin modulated data set (Equation 4.2b), and so on. For example, the CC-FT becomes:

$$S_{CC}(\omega_1,\omega_2) = \sum_{t1=0}^{t1\max}\sum_{t2=0}^{t2\max}\cos(t_1\omega_1)\cos(t_2\omega_2)f(t_1,t_2)w(t_1,t_2) \tag{4.3}$$

The three remaining transformations are similarly defined[45, 47].

In order to select the appropriate quadrature image the four resulting spectra, $S_{cc}(\omega_1,\omega_2)$, $S_{cs}(\omega_1,\omega_2)$, $S_{sc}(\omega_1,\omega_2)$ and $S_{ss}(\omega_1,\omega_2)$ are summed, canceling the quadrature images and artifact peaks.

$$S_{RR}(\omega_1,\omega_2)=S_{CC}(\omega_1,\omega_2)+S_{CS}(\omega_1,\omega_2)+S_{SC}(\omega_1,\omega_2)+S_{SS}(\omega_1,\omega_2) \qquad (4.4)$$

To demonstrate the four Fourier transforms and summing procedure, we use four computer generated radially sampled time domain data sets modulated by a mixture of cos and sin as dictated by Equations 4.2a-d with one peak. The peak position for the data sets was set at (-300 Hz, 75 Hz) and the sampling angle set to 45 degrees. The linewidth was adjusted to 10 Hz by multiplying the data sets by an exponential decay. The data sets were Fourier transformed with their respective transform as outlined by Equation 4.3. Sixteen peaks are visible in the $S_{cc}(\omega_1,\omega_2)$ spectrum. Four peaks are the quadrature images at ± 300, ±75 Hz and the twelve arise from intersection of the ridge artifact appearing at ±400, 0; ± 200, 0; 0, ±300; 0, ±150, ±100, ±225. The other three spectra $S_{cs}(\omega_1,\omega_2)$, $S_{sc}(\omega_1,\omega_2)$ and $S_{ss}(\omega_1,\omega_2)$ have the same four quadrature image peaks with varying signs. The variation in signs causes the artifact patterns to change. In the case where two negative ridges intersect a negative artifact peak is present, when two ridges of varying sign intersect no peak is present. When all four spectra are summed the variations in sign of the quadrature and artifact peaks cause them to cancel resulting in a spectrum with just the authentic peaks remaining (Figure 4.1).

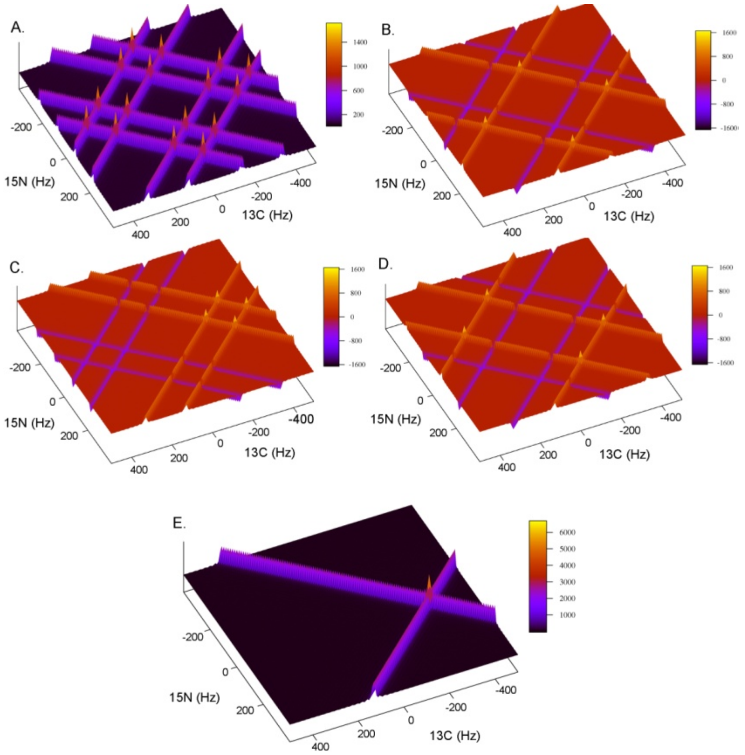


Figure 4.1 An example of how quadrature images are resolved for computer generated radial sampled data processed with a single step two-dimensional Fourier transform. The data was generated with spectral parameters similar to that found in a radial sampled HNCO experiment. Four data sets (A-D) were generated according to Equations 4.7a,b and 4.8. The sweep widths were set to 2000 and 1500 Hz for the t_1 (carbon) and t_2 (nitrogen) dimensions respectively. One peak was simulated at -300, 75 hertz with a linewidth of 10 Hz. Radial sampling was realized by incrementing the time in the first dimension as $t_1 = (n/sw_1)\cos\alpha$ and the second dimension as $t_2 = (n/sw_2)\sin\alpha$. The four data sets were processed with their matching Fourier transform, for example the cos-cos modulated data set was processed with the CC-FT, A. sin-cos with the SC-FT, B. cos-sin with the CS-FT, C. and sin-sin with the SS-FT. Inset E shows the sum of A-D, resolving the appropriate quadrature image.

In addition to the authentic peaks, ridges also extend from the peak at the sampling angle in the $S_{RR}(\omega_1, \omega_2)$ spectrum. Most often one wishes to remove the ridges and in the case where signal to noise is not limiting the lower value (LV) algorithm is preferred[25]. Here multiple data sets are collected at various sampling angles and the data is Fourier transformed independently. Subsequent to the transforms the intensities of multiple $S_{RR}(\omega_1, \omega_2)$ spectra are compared point-wise and the smallest magnitude value at each point is kept in a separate spectrum. If a sufficient number of angle data sets are collected a final spectrum free of ridges is generated.

Providing the data is free of phase error, the above Fourier transform method combined with the lower value algorithm works quickly and accurately to generate a ridge free frequency spectrum. However, this approach is severely limited by its inability to deal with phase distorted data. If a phase error is present the lowest value algorithm will delete authentic peaks. This occurs because the lineshape of dispersive peaks causes the intensity to be zero inside the linewidth of the peak. The zero values are different for each sampling angle, therefore when multiple angles are compared by the LV algorithm the peak will be eliminated.

In order to circumvent this shortcoming, the current strategy is to optimize data collection to reduce phase distortions. Nevertheless, non-ideal spectrometer performance and inherent limitations of pulse sequences often preclude the collection of time domain data free of phase error. Because of the effective convolution of phase error across the various incremented time domains traditional approaches to phase correction are not applicable for radially collected data. As we will illustrate below, the presence of phase distortions severely degrade the quality of the resulting spectrum.

To solve this problem we have developed two novel phase correction methods. The first method presents a correction that is applied in the frequency domain by generating absorptive and dispersive components with the 2D-FT. The second method applies corrections by adding constants to the 2D-FT, essentially applying a correction in the time domain.

The phase corrections in the frequency domain are applied by utilizing properties of the discrete Fourier transform to generate absorptive and dispersive components. Namely an absorptive spectrum is generated by applying a real cos Fourier transform to cos modulated data and a dispersive spectrum is generated if a real sin Fourier transform is applied to the same cos modulated data.

In the case of the 2D Fourier transform we can generate four spectra: real-real, absorptive with respect to both the ω_1 and ω_2, frequency domains, real-imaginary, absorptive with respect to the ω_1 and dispersive with respect to ω_2, and so on. The process for generating these components is summarized in Table 1. For example, the pure absorption spectrum, $S_{RR}(\omega_1,\omega_2)$, is generated by transforming the four data components with the matching Fourier transforms. That is, the cos-cos modulated data is transformed with the CC-FT, the sin-cos modulated data set is transformed with the SC-FT, the cos-sin with the CS-FT and the sin-sin with the SS-FT. The four resulting spectra are summed producing the $S_{RR}(\omega_1,\omega_2)$ spectrum. Similar procedures lead to the remaining three necessary spectra: $S_{RI}(\omega_1,\omega_2)$, $S_{IR}(\omega_1,\omega_2)$ and $S_{II}(\omega_1,\omega_2)$.

Table 4.1 Procedure for Generating Absorptive and Dispersive Spectra

	RR	RI	IR	II
$f_{CC}(t_1,t_2)$	$\xrightarrow{CCFT} S_{CC}^{RR}(\omega_1,\omega_2)$	$\xrightarrow{CSFT} S_{CC}^{RI}(\omega_1,\omega_2)*-1$	$\xrightarrow{SCFT} S_{CC}^{IR}(\omega_1,\omega_2)$	$\xrightarrow{SSFT} S_{CC}^{II}(\omega_1,\omega_2)*-1$
$f_{CS}(t_1,t_2)$	$\xrightarrow{CSFT} S_{CS}^{RR}(\omega_1,\omega_2)$	$\xrightarrow{CCFT} S_{CS}^{RI}(\omega_1,\omega_2)*-1$	$\xrightarrow{SSFT} S_{CS}^{IR}(\omega_1,\omega_2)*-1$	$\xrightarrow{SCFT} S_{CS}^{II}(\omega_1,\omega_2)$
$f_{SC}(t_1,t_2)$	$\xrightarrow{SCFT} S_{SC}^{RR}(\omega_1,\omega_2)$	$\xrightarrow{SSFT} S_{SC}^{RI}(\omega_1,\omega_2)$	$\xrightarrow{CCFT} S_{SC}^{IR}(\omega_1,\omega_2)$	$\xrightarrow{CSFT} S_{SC}^{II}(\omega_1,\omega_2)$
$f_{SS}(t_1,t_2)$	$\xrightarrow{SSFT} S_{SS}^{RR}(\omega_1,\omega_2)$	$\xrightarrow{SCFT} S_{SS}^{RI}(\omega_1,\omega_2)$	$\xrightarrow{CSFT} S_{SS}^{IR}(\omega_1,\omega_2)*-1$	$\xrightarrow{CCFT} S_{SS}^{II}(\omega_1,\omega_2)*-1$
Σ	$S_{RR}(\omega_1,\omega_2))$	$S_{RI}(\omega_1,\omega_2))$	$S_{IR}(\omega_1,\omega_2))$	$S_{II}(\omega_1,\omega_2))$

The four resulting spectra are shown in Figure 4.2 for the one peak generated data set with a sample angle of 45°. From initial inspection it might appear that these 4 spectra are sufficient to allow for phasing the two dimensional spectrum. This is not the case. The signs of the phase correction relative to the +α and –α ridges are opposite and requires that the +α and –α components be isolated and phased separately.

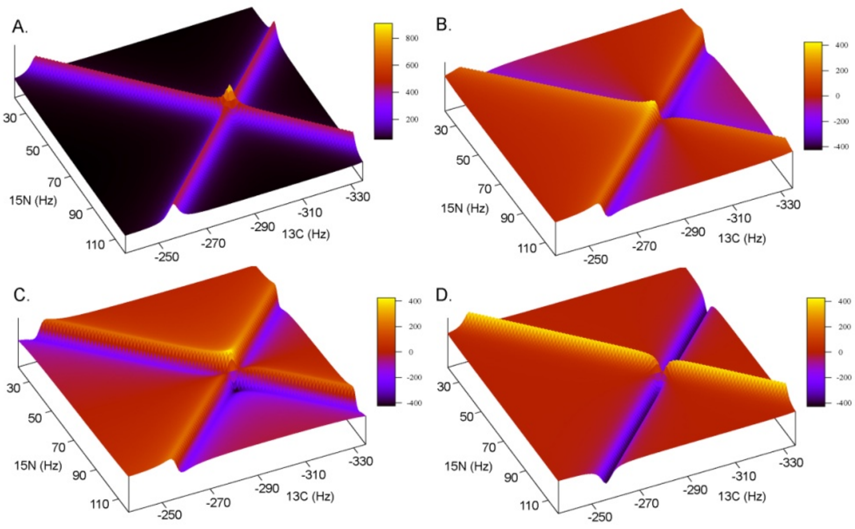


Figure 4.2 An example of how the 2D-FT can be used to generate absorptive and dispersive spectra with respect to t_1 (ω_1) and t_2 (ω_2). Inset A shows the real-real spectra, $S_{RR}(\omega_1,\omega_2))$, generated when the

matching Fourier transform is used, i.e. CC-FT for cos-cos modulated data. Inset B shows the imaginary-real spectra, $S_{IR}(\omega_1,\omega_2))$, generated by not matching the FT with respect t_1 (ω_1) while matching it with respect to t_2 (ω_2), i.e. SC-FT for cos-cos modulated data. Insets C and D show the other two spectra that can be generated $S_{RI}(\omega_1,\omega_2)$ and $S_{II}(\omega_1,\omega_2)$, respectively. Table 1 outlines the complete procedure. The data was generated in the same way as in Figure 4.1 and plotted to view only the area centered on the peak at -300,75 Hz.

The +α and –α real and imaginary components are generated by taking combinations of $S_{RR}(\omega_1,\omega_2)$, $S_{RI}(\omega_1,\omega_2)$, $S_{IR}(\omega_1,\omega_2)$ and $S_{II}(\omega_1,\omega_2)$ as shown below.

$$R^{-\alpha} = S_{RR}(\omega_1,\omega_2) + S_{II}(\omega_1,\omega_2) \tag{4.5a}$$

$$I^{-\alpha} = S_{IR}(\omega_1,\omega_2) - S_{RI}(\omega_1,\omega_2) \tag{4.5b}$$

$$R^{+\alpha} = S_{RR}(\omega_1,\omega_2) - S_{II}(\omega_1,\omega_2) \tag{4.6a}$$

$$I^{+\alpha} = S_{IR}(\omega_1,\omega_2) + S_{RI}(\omega_1,\omega_2) \tag{4.6b}$$

The ±α real and imaginary components are illustrated in Figure 4.3.

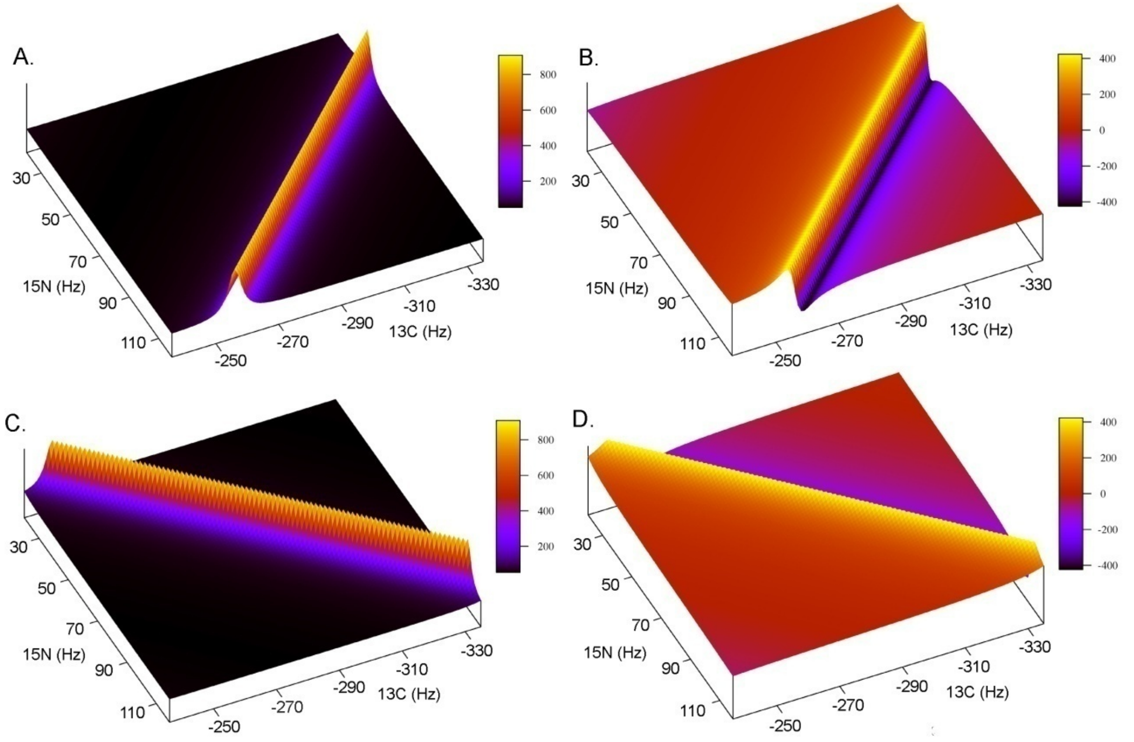


Figure 4.3 An example of the pure ±α real and imaginary component spectra used for phase correction in the frequency domain. Insets A and B show the real and imaginary + α spectra, while insets C and D show the real and imaginary – α spectra. Combinations of the + α and – α components are generated independently and subsequently summed to produce a phased spectrum.

With the real and imaginary components of +α and – α separated the two phased spectra can finally be generated and subsequently summed to produce a phase corrected spectrum, i.e.

$$
\begin{aligned}
S^{+\alpha}(\omega_1,\omega_2) &= R^{+\alpha}\cos(\phi_0^{t_1}+\phi_0^{t_2}+\phi_1^{t_1}\frac{2(\omega_1-\omega_1^{pivot})}{sw_1}-\phi_1^{t_2}\frac{2(\omega_2-\omega_2^{pivot})}{sw_2}) \\
&+ I^{+\alpha}\sin(\phi_0^{t_1}+\phi_0^{t_2}+\phi_1^{t_1}\frac{2(\omega_1-\omega_1^{pivot})}{sw_1}-\phi_1^{t_2}\frac{2(\omega_2-\omega_2^{pivot})}{sw_2})
\end{aligned}
\tag{4.7a}
$$

$$
\begin{aligned}
S^{-\alpha}(\omega_1,\omega_2) &= R^{-\alpha}\cos(\phi_0^{t_2}-\phi_0^{t_1}+\phi_1^{t_1}\frac{2(\omega_1-\omega_1^{pivot})}{sw_1}+\phi_1^{t_2}\frac{2(\omega_2-\omega_2^{pivot})}{sw_2}) \\
&+ I^{-\alpha}\sin(\phi_0^{t_2}-\phi_0^{t_1}+\phi_1^{t_1}\frac{2(\omega_1-\omega_1^{pivot})}{sw_1}+\phi_1^{t_2}\frac{2(\omega_2-\omega_2^{pivot})}{sw_2})
\end{aligned}
\tag{4.7b}
$$

Where $\phi_0^{t_1}$, $\phi_0^{t_2}$, $\phi_1^{t_1}$ and $\phi_1^{t_2}$ are the t_1 and t_2 zero and first order corrections. Also note that a factor of two was included in the first order terms to make setting the first order correction independent of the zero order terms. For example, if the pivot is set in the middle of the spectrum and one adds a half dwell to the incremented time period ±90° phase corrections are needed at the edges of the spectrum. Traditionally the zero and first order phase corrections are set to 90°,-180°. By including the factor of two the phases can be set to 0°,-90°. Therefore only one parameter needs to be adjusted if only a first order phase correction needs to be applied.

To generate a final spectrum, the $S^{+\alpha}$ and $S^{-\alpha}$ are summed (Equation 8).

$$
S(\omega_1,\omega_2) = S^{+\alpha}(\omega_1,\omega_2)+S^{-\alpha}(\omega_1,\omega_2)) \tag{4.8}
$$

Alternatively, the $S^{+\alpha}$ and $S^{-\alpha}$ can be used separately in the lower value-back projection algorithm[41]. This would give an advantage over summing the spectra because more combinations would be available for comparison.

Practically, the zero and first order phase corrections are empirically determined from either an indirectly detected plane of the 3d spectrum with a single peak, so the analysis isn't confused by artifacts, or from the 0° and 90° tilt angle spectra. For the single peak case, four spectra are first generated, as outlined in Table 1. Next, the sum and difference spectra are generated as in Equations 4.5 and 4.6. Finally, the four phase corrections are applied as outlined in Equation 4.7. Here the phases are searched for by varying each phase term until an absorptive spectrum is produced. Else the phase corrections can be determined independently from the 0° and 90° sample angle planes. The 0° and 90° sample angles allow the phase corrections to be isolated for either t_1 or t_2 respectively. In these spectra only one indirect time domain is evolved causing the data to be sampled in Cartesian space. Therefore traditional phasing techniques are applicable so the phase corrections can be determined by employing a Hilbert transform to generate the dispersive components[58]. After the phase corrections are determined from the 0° and 90° sample angle planes, they are applied to all angles using Equation 4.7.

When only zero order corrections are needed it is equally feasible to determine them from a one peak plane or from the 0° and 90° tilt angle spectra. However when any first order correction needs to be applied, it is much easier to determine the phase corrections from the 0° and 90° sample angle spectra. The isolation of the t_1 and t_2 phase correction components by the 0° and 90° sample angle spectra significantly simplify the problem. It is also important to note when first order corrections are applied the ridges do not phase with the peaks. This occurs because of the frequency dependence of the first order correction. Therefore the ridges will in phase proximal to the peak but dispersive as they move further away. Although this might sound problematic, robust schemes are available to remove the ridges if the peaks are properly phased.

Phase corrections can also be applied in the in the time domain by adding constants to the 2D-FT (Equation 4.9).

$$S(\omega_1,\omega_2)=\sum_{t1=0}^{t_1^{\max}}\sum_{t2=0}^{t_2^{\max}}\exp(-i\omega_1(t_1+\phi_1^{t_1})+\phi_0^{t_1})\exp(-j\omega_2(t_2+\phi_1^{t_2})+\phi_0^{t_2})f(t_1,t_2)w(t_1,t_2) \qquad (4.9)$$

Here, $\phi_0^{t_1}$, $\phi_1^{t_1}$, $\phi_0^{t_2}$ and $\phi_1^{t_2}$ are the zero and first order phase corrections for the incremented time domains t_1 and t_2, respectively. This method directly extends from the definition of phase error in time domain data,

$$f_{CC}(t_1,t_2)=\cos((t_1+\phi_1^{t_1})\Omega_1+\phi_0^{t_1})\cos((t_2+\phi_1^{t_2})\Omega_2+\phi_0^{t_2}) \qquad (4.10)$$

and properties of the discrete Fourier transform. Namely a nonzero Fourier series coefficient is determined if the function generated by the Fourier transform matches the data function. We have simply extended this concept to include phase corrections so the function generated by the Fourier transform better matches the experimental data. In turn, an absorptive lineshape is generated upon transformation.

As above the phase corrections are determined empirically from either a plane with one peak or from the 0° and 90° sample angle spectra. Once the phase corrections are determined the data is retransformed with the appropriate corrections applied to Equation 4.9.

It is important to point out that phasing in the time domain has not previously been presented because of inherent limitations of the fast Fourier transform (FFT) algorithm[59]. This can most easily be explained by first inspecting the discrete one dimensional Fourier transform (Equation 4.11).

$$S(\omega) = \sum_{t=0}^{t^{\max}} \exp(-i\omega t) f(t) \tag{4.11}$$

From inspection it is clear that N^2 operations are required to compute $S(\omega)$. This is obviously undesirable if a large number of data points are collected. However, if an extension of the Yates algorithm is applied, the N^2 operations can be reduced to $N \log_2 N$ operations[59]. This is accomplished by iteratively dividing the data to smaller and smaller groups until N groups of size 1 are present. At this point the 1 data point groups can be Fourier transformed and combined in the manner presented by Cooley and Tukey[59].

Properties of the one point Fourier transform are essential to this algorithm. In particular, the Fourier transform of one data point is itself independent of frequency. This is true because t=0 and therefore $\exp(-i\omega t) = 1$ However, if phase corrections are incorporated, t is no longer equal to zero and the Fourier transform is no longer frequency independent and the FFT algorithm is no longer applicable.

4.3 Results

This procedure is illustrated in Figure 4.4 using a standard HNCO[60] modified for radial sampling, such that $t_1 = t_1 \cos(\alpha)$ and $t_2 = t_1 \left(sw_1 / sw_2 \right) \sin(\alpha)$, on a 1mM 1:1 complex between calcium-saturated calmodulin and a peptide corresponding to the calmodulin binding domain of phosphodiesterase 1A. In order to demonstrate the ability to apply a first order phase correction the experiment was setup with a half dwell added to both the t_1 and t_2 increments. Accordingly, the spectra required first order corrections of -90º for both the t_1 (ω_1) and t_2 (ω_2) dimensions. Additionally the spectrum required a zero order correction of 36° in the t_1 dimension.

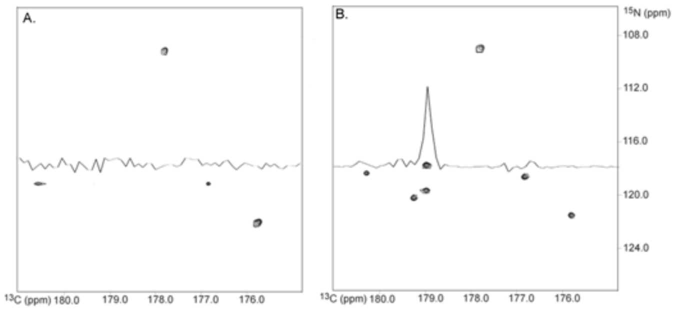


Figure 4.4 Comparison of the same indirect dimension plane for a spectrum processed with no phase correction (Panel A) and with phase correction (Panel B). The phase corrected spectrum shows all peaks at the correct frequencies with the appropriate intensities. The spectrum with no phase correction is missing numerous peaks, as emphasized by the overlaid 1D spectra. Ten sample angle data sets were collected on a 1:1 complex between calcium-saturated calmodulin and a peptide corresponding to the calmodulin binding domain of phosphodiesterase using a HNCO modified for radial sampling. The individual sampling angles were processed separately and compared using the Al NMR implementation of the lower-value algorithm.

4.4 Discussion & Conclusions

The need to have properly phased multidimensional frequency space data is essential to the general application of radial sampling and the 2D-FT. As demonstrated by the HNCO spectrum with no phase correction shown in Figure 4.4a, when a lower magnitude comparison is performed for a data set without phase correction authentic peaks will incorrectly be removed. In many cases, phase error cannot be avoided and explicit phase correction will be required. We have devised two approaches to the phasing of such data, either by manipulation in the frequency domain or in the time domain. The choice of method, time or frequency phase correction, is dependent upon the application at hand. For example, phasing in the frequency domain will be important upon the advent of a fast 2D-FT algorithm. Whereas, phasing in the time domain will easily be implemented in higher dimensional Fourier transforms.

4.5 Methods

NMR data was collected on a ~ 1mM 1:1 complex between calcium-saturated calmodulin and a peptide corresponding to the calmodulin binding domain of the phosphodiesterase at 35º C on a Varian INOVA 600 MHz spectrometer, equipped with a triple-resonance cryogenic probe. The CaM-PDE complex was prepared in 10 mM imidazole pH 6.5, 6mM $CaCl_2$, 100mM KCl and 0.04% azide. Ten sample angles were collected from 0° to 90° degrees in 10° degree increments. Each spectrum was derived from data sets composed of 384 FIDs, four quadrature components at 96 increments. Each FID contained 1024 complex points and was the average of eight scans. The spectral width was set to 14 ppm in the proton dimension. The spectral widths for the indirect dimensions were chosen to assure no peaks were folded and set to 40 and 12 ppm in the nitrogen and carbon dimensions, respectively. The ten angle spectra were processed independently and compared using a lower magnitude algorithm to remove the ridge artifacts. All processing and comparisons were done using Al NMR.

Chapter 5

Optimized Angle Selection for Radial Sampled NMR Experiments

5.1 Introduction

As outlined in Chapter 2, radial sampling of the indirect evolution domain is shown to be appealing under the appropriate conditions because of the smooth baseline compared to the aliasing artifacts of random sampling and the potential for a sensitivity advantage. The ridge artifacts, that are inherent to radial sampling, are easily removed if an efficient set of sampling angles are collected. Further, if quantitative information is to be extracted from the radial sampled spectra it is essential that the ridge intensity only have a contribution from a single peak; as compared to, two peaks lying on the same ridge vector. Therefore, the utility of radial sampling is dependent upon the radial sampling angles chosen during data collection.

In order to increase the utility of the radial sampling approach we present methods to optimize the set of sampling angles employed. The approaches can be classified into two general situations. The first is when the peak resonance frequencies are known and need to be resolved from artifact, and the second is when the peak resonance frequencies are not known and need to resolved and assigned. The former case corresponds to a need to measure variation in intensity such as in a hydrogen exchange or classical relaxation experiment. For this, two algorithms have been developed. One determines the minimum set of angles necessary to distinguish authentic peak intensity from artifactual intensity introduced by the Fourier analysis of radially sampled data (i.e. the ridges). The second algorithm determines the fewest angles needed to produce an artifact free spectrum when a lower value[25] comparison is preformed. Alternatively, if the peak resonance frequencies are not known, an algorithm is developed to provide for iterative post-acquisition determination of the optimal sampling angles to collect and to provide a definitive conclusion regarding the separation of authentic peak intensity from ridge artifacts. This type of algorithm is essential for the optimized application of radial sampling of data to be employed for *de novo* resonance assignment. Both algorithms are tested in the context of a radial sampled HNCO processed with the direct multidimensional Fourier transform[45-47] combined with lower value comparison, but are applicable, with minor modifications to the selection criteria, to more sophisticated artifact removal schemes.

5.2 Theory

When radial sampled data is transformed with the 2D-FT the resulting spectrum is effectively underdetermined and produces ridges that extend through the spectrum where Equation 5.1 is satisfied.

$$\frac{\omega_1 - \Omega_1}{\omega_2 - \Omega_2} = \tan(\alpha) \qquad (5.1)$$

This relationship is true when α is either positive or negative, leading to two ridges extending from the each peak in the spectrum, one with a positive slope and the other with a negative slope.

We define an ordered triple with the directly detected dimension, ω_3, in the first position and the two linked indirect dimensions, ω_1 and ω_2, in the second and third positions respectively. The following linear equation describes the ridge extending from a peak located at point P_1 in a (3,2) radially sampled experiment where we employ the nomenclature of Szyperski[21].

$$P = P_1 + n(0, \cos(\pm(90-\alpha)), \sin(\pm(90-\alpha))) \tag{5.2}$$

P represents a point on the ridge, α is the sampling angle and n is a scalar. As before, the +/- sign is included because two ridges extend, one with a positive slope and another with a negative slope. In the case of a (4,2) radially sampled experiment four ridges would extend from each peak. In this case, Equation 5.2 is expanded to account for two sampling angles, α and β, as described by Equation 5.3.

$$\begin{aligned} P = P_1 + n(0, &\cos(\pm(90-\alpha))\cos(\pm(90-\beta)), \\ &\sin(\pm(90-\alpha))\cos(\pm(90-\beta)), \sin(\pm(90-\beta))) \end{aligned} \tag{5.3}$$

These basic descriptions allow the determination of whether two peaks are resolved at a given sampling angle and where all of the potential artifact positions are located. Further, this description allows all peaks to be analyzed simultaneously, regardless whether they are resolved in the directly detected dimension.

5.2.1 Peak – Peak resolution

Two peaks in a radially sampled experiment are not resolved if the ridge from one of the peaks intersects the second. To determine if two peaks are resolved the distance from one of the peaks to the closest points on the positive and negative ridge components of the other peak is determined. If both distances are greater than a specified cutoff (chosen to reflect a finite line width), the peak is considered resolved. The distance measurement is illustrated in Figure 5.1A, where the peaks are represented by points P_1 and P_2.

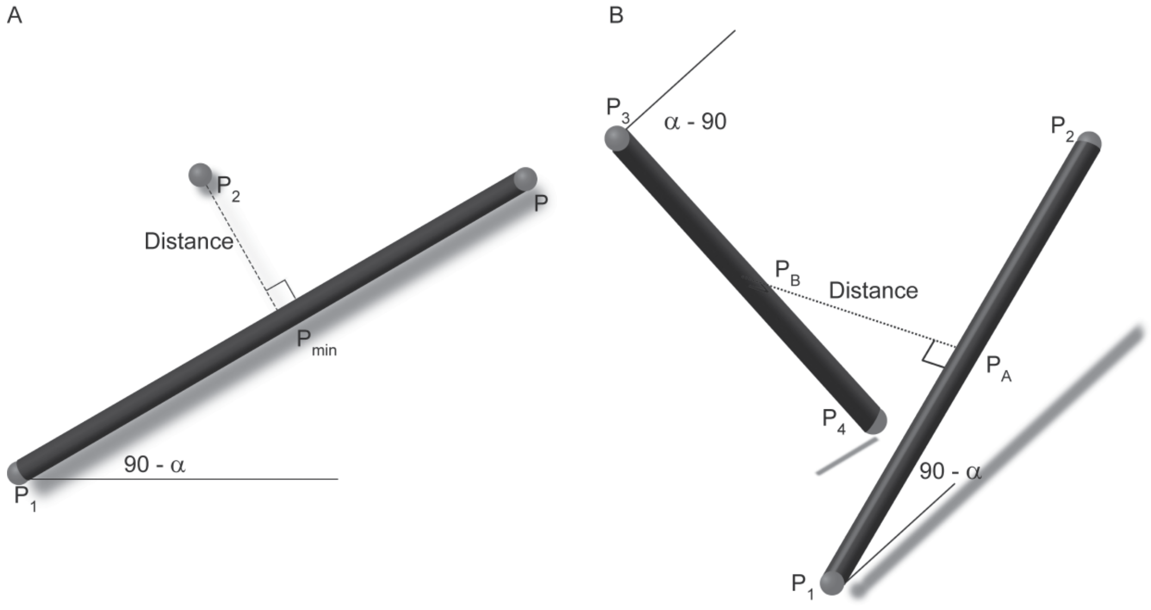


Figure 5.1 Illustration of the peak to ridge distance in 2D space (A). Given two peaks, located at P1 and P3, the shortest distance is calculated between the peak located at position P3 and the nearest point on the ridge, extending from the peak at position P1, located at point P. See text for details regarding this distance calculation. If the distance is greater than a specified cutoff the two peaks are resolved at the given sampling angle α. Illustration of the ridge to ridge distance (B). The ridge to ridge distance is used to determine if an artifact is generated from the intersection of ridges extending from peaks at locations P1 and P3. Additionally, if the distance between the two ridges is less than a specified cutoff, the artifact position is determined as the average of the closest points PA and PB. See text for details regarding this calculation.

For clarity only one of the ridge components is shown in the figure. The distance between P_2 and the ridge from P_1 is determined by applying the point to line distance algorithm commonly encountered in computer graphics[61]. Here we generalize this approach. The first step is to define an equation in order to solve for point P_{min}, the closest point on the ridge to the peak located at P_2.

$$P_{\min} = P_1 + u(P - P_1) \tag{5.4}$$

To determine P an arbitrary non-zero scalar n is used. When the distance between point P_2 and P_{min} is minimized the vector from P_{min} to P_2 is perpendicular to the ridge. Therefore the dot product of the two vectors is zero.

$$(P_2 - P_{\min}) \bullet (P - P_1) = 0 \tag{5.5}$$

In order to solve for the point P_{min}, Equation 5.4 is substituted into the dot product relationship and the scalar u is determined.

$$u = \frac{P_2 \bullet (P - P_1) - P_1 \bullet (P - P_1)}{(P - P_1) \bullet (P - P_1)} \tag{5.6}$$

Finally the expression for u is used to determine the point P_{min}.

$$P_{\min} = P_1 + \frac{P_2 \bullet (P - P_1) - P_1 \bullet (P - P_1)}{(P - P_1) \bullet (P - P_1)} (P - P_1) \tag{5.7}$$

The distance between P_{min} and P_2 is then

$$\text{Distance} = \sqrt{(P_{\min} - P_2) \bullet (P - P_2)} \tag{5.8}$$

The distance defined above corresponds to an infinitely narrow line. To accommodate consideration of a finite linewidth the effective width along the line connecting the two points of interest must be determined. This is accomplished by setting the origin of the Cartesian basis at point P then defining two angles between points P_{min} and P_2 that would be used to describe the latter's position with respect to the former in a polar basis. These two angles are defined as:

$$\alpha_1 = \arctan\left|\frac{P_{\min}(\omega_1) - P_2(\omega_1)}{P_{\min}(\omega_3) - P_2(\omega_3)}\right| \tag{5.9a}$$

$$\alpha_2 = \arctan\frac{\left|P_{\min}(\omega_2) - P_2(\omega_2)\right|}{\sqrt{(P_{\min}(\omega_1) - P_2(\omega_1))^2 + (P_{\min}(\omega_3) - P_2(\omega_3))^2}} \tag{5.9b}$$

Here the subscript defines the Cartesian chemical shift components of vector P_{min} and P_2. Note that the angle is set to 90° if the denominator is zero. The linewidths along the specified distance line can be determined using the above defined angles as follows:

$$linewidth_{\omega_1} = (cartesian\ linewidth_{\omega_1})(\sin\alpha_1)(\cos\alpha_2) \tag{5.10a}$$

$$linewidth_{\omega_2} = (cartesian\ linewidth_{\omega_2})(\sin\alpha_2) \tag{5.10b}$$

$$linewidth_{\omega_3} = (cartesian\ linewidth_{\omega_3})(\cos\alpha_1)(\cos\alpha_2) \tag{5.10c}$$

The effective linewidth for a peak along a given vector is then the Euclidean distance of the scaled components.

$$linewidth = \left(\sum_{q=\omega_1,\omega_2,\omega_3} linewidth_q^2 \right)^{1/2} \tag{5.11}$$

The same scaling components can be used for both the peak at P_2 and the ridge at P_{min} because of the mirror symmetry between them. Note that the linewidth at point P_{min} is the same as the linewidth at peak P_1. Finally, a measure of resolution is obtained by subtracting the two linewidths from the distance measured above.

$$resolution\ distance = distance - linewidth_{P_1} - linewidth_{P_2} \tag{5.12}$$

The correction for finite line width in a higher dimensional experiment expands accordingly while the minimum distance algorithm remains unchanged. In the case of a (4,2) experiment four linewidths will be scaled using three angles defined in the same manner as Eq. [5.9a] and [5.9b]. The four scaling components are: $(\sin\alpha_1)(\cos\alpha_2)(\cos\alpha_3)$, $(\sin\alpha_2)(\cos\alpha_3)$, $(\sin\alpha_3)$ and $(\cos\alpha_1)(\cos\alpha_2)(\cos\alpha_3)$ for the ω_1, ω_2, ω_3 and ω_4 respectively. Where ω_1, ω_2 and ω_3 are the indirectly detected dimensions and ω_4 is the directly detected dimension. Additionally, this treatment also provides a mechanism for filtering a peak list to allow for authentic peaks that will never be resolved to be treated as one peak. For example, one peak that encompasses two non-resolved peaks can be set to have a peak position equal to the average of the two peaks and a broader linewidth to account for both peaks.

5.2.2 Potential artifact positions

The lower value algorithm efficiently removes ridge artifacts if an appropriate combination of angle spectra are compared[25]. In instances where an inappropriate (insufficient) number of angle spectra are employed, ridge intensity may be present at positions in the spectrum not corresponding to authentic peak intensity and represents an artifact in the spectrum. The artifacts occur at locations when multiple ridges intersect. Therefore, determining all possible ridge intersection points can lead to the identification of artifact peaks. If the ridge linewidths were infinitely narrow the potential artifact locations would be the solution to the set of linear equations describing the ridges. In order to accommodate finite linewidths a ridge to ridge distance algorithm is used. The algorithm is an application of the general line to line distance algorithm also often used in computer graphics[61]. If the distance between two ridges is less than a specified cutoff, the average of the two closest points on each ridge is marked as a potential artifact. This procedure is illustrated in Figure 5.1b. Here two ridges extend from points P_1 and P_2 and points P_A and P_B represent the closest points between the two ridges. This figure illustrates only one of the ridge components from each peak. The total number of ridges is defined by the sampling scheme as discussed above.

The two ridges of Figure 5.1b are represented by the linear Equations 5.13a and b:

$$P_A = P_1 + a(P_2 - P_1) \tag{5.13a}$$

$$P_B = P_3 + b(P_4 - P_3) \tag{5.13b}$$

Here, P_1 and P_3 are the positions of authentic peaks. Points P_2 and P_4 are defined as a function of the sampling angle, as presented in Equations 5.2 and 5.3 for a non-zero scalar n. a and b are the scalars used to define points the closest points P_A and P_B. A vector W can be defined between the two closest points as:

$$W = P_A - P_B \tag{5.14}$$

As before, in order to solve for P_A and P_B the scalars a and b must be determined. W becomes

$$W = P_1 + a(P_2 - P_1) - P_3 - b(P_4 - P_3) \tag{5.15}$$

For clarity we can define a vector from the peak at P_1 to the peak at P_3.

$$W_0 = P_1 - P_3 \tag{5.16}$$

The simplified expression for W is presented by substituting 5.16 into 5.15.

$$W = W_0 + a(P_2 - P_1) - P_3 - b(P_4 - P_3) \tag{5.17}$$

From the definition of two skew lines we know the vector describing the line between the closest points is the only line uniquely perpendicular to both the lines describing the ridges. Therefore the dot product and unit vectors $(P_2 - P_1)$ and $(P_4 - P_3)$ that describe the two ridges is zero.

$$W \bullet (P_2 - P_1) = 0 \tag{5.18a}$$

$$W \bullet (P_4 - P_3) = 0 \tag{5.18b}$$

Substituting the expression for W into the dot product definitions puts the equations in terms of the scalars a and b:

$$W_0 \bullet (P_2 - P_1) = b(P_2 - P_1) \bullet (P_4 - P_3) - a(P_2 - P_1) \bullet (P_2 - P_1) \tag{5.19a}$$

$$W_0 \bullet (P_4 - P_3) = b(P_4 - P_3) \bullet (P_4 - P_3) - a(P_4 - P_3) \bullet (P_2 - P_1) \tag{5.19b}$$

For clarity the following scalars are defined: $h = W_0 \bullet (P_2 - P_1)$, $i = (P_2 - P_1) \bullet (P_2 - P_1)$, $j = (P_2 - P_1) \bullet (P_2 - P_1)$, $k = W_0 \bullet (P_2 - P_1)$ and $l = (P_4 - P_3) \bullet (P_4 - P_3)$. Substituting into Equations 5.19a and 5.19b gives:

$$a = \frac{ki - hl}{jl - i^2} \tag{5.20a}$$

$$b = \frac{jk - hi}{jl - i^2} \tag{5.20b}$$

Upon substitution of Equations 5.20a and 5.20b into Equations 5.13a and 5.13b, the points P_A and P_B become:

$$P_A = P_1 + \frac{ki - hl}{jl - i^2}(P_2 - P_1) \tag{5.21a}$$

$$P_B = P_3 + \frac{jk - hl}{jl - i^2}(P_4 - P_3) \tag{5.21b}$$

The Euclidean distance between points P_A and P_B is defined as:

$$distance = \sqrt{(P_A - P_B)\bullet(P_A - P_B)} \tag{5.22}$$

In order to determine if an artifact is present the distance is scaled for the line widths in the same manner as in the peak to ridge distance. If the scaled distance is less than a specified cutoff a potential artifact is located at the average of points P_A and P_B. Again, this algorithm is also applicable to higher dimensional experiments.

We now have the tools necessary to optimize the collection of radially sampled data.

5.2.3 Minimum angles to resolve peak intensities

The first case that we consider is the situation where the positions of authentic peaks are known. This would be encountered during the collection of three-dimensionally resolved relaxation or hydrogen exchange data, for example. Here the goal is to collect data as efficiently as possible such that all authentic peaks are free from contaminating artifact intensity. Importantly, in this situation, artifact intensity that is resolved from authentic peak intensity is of no consequence.

Relaxation rates vary as a function of angle for radial sampled experiments because the data is a product of two relaxation components, one from each of the two indirectly evolved dimensions (spins). The variation in relaxation can be eliminated by using a single sampling angle for a series of experiments. In turn, treating the angles independently, allows for the ridge intensity to be left in the spectrum, skipping any lower value comparison. To increase the number of peaks resolved at a sampling angle the positive and negative ridge components are isolated and analyzed separately. The ridge components are isolated in the same manner as Chapter 4.

If all authentic peaks are not resolved from ridge intensity with a single sampling angle, multiple sampling angles can be used but the resulting data should be treated independently. Treating the angle spectra independently, allows for a simple algorithm to determine the optimal sampling angles. The first step in the algorithm is to edit the peak list grouping authentic peaks that are not resolved from each other as opposed to resolved from artifactual peak intensity. Unresolved authentic peaks will not be resolved by any sampling angle and are therefore treated as a single peak with a linewidth that spans the group of peaks. After the peak list has been edited every combination of peaks is tested for resolution from artifact intensity using the peak to ridge distance algorithm for a selected series of angles. The peak to ridge distance accounts for a difference of the chemical shifts in the directly detected dimension avoiding the need to sort the peaks to assure they are in the same indirect plane of the spectrum. This step generates two lists of peaks for the sampling angle tested, one for the peaks resolved in the positive slope component spectrum, and another for the peaks resolved in the negative slope component spectrum.

The results of a series of sampling angles can be sorted to determine the minimum number and identity of the sampling angles needed to resolve the intensity of all of the authentic peaks in a spectrum. The two lists of resolved peaks at each angle are combined and any redundancy removed; some peaks will be resolved in both of the component spectra. The angle that resolves the most peaks is then found. If the selected angle fails to resolve all of the peaks additional angles are selected on the basis of resolving the most peaks.

This procedure was tested with a simulated data set consisting of 10 peaks, all located in the same plane. The peaks were randomly distributed in two dimensions, with the criteria that they would not be resolved by only one of the two dimensions. The results are shown in Figure 5.2. For comparison the same peak frequencies and linewidths were used to generate a Cartesian sampled data set resulting in the spectrum shown in Figure 2a. Analysis of the peak list concluded that an 85° sampling angle would resolve all of the peaks. The positive slope ridge component of the 85° spectrum is shown in Figure 5.2b. For clarity the Cartesian sampled spectrum is overlaid in gray.

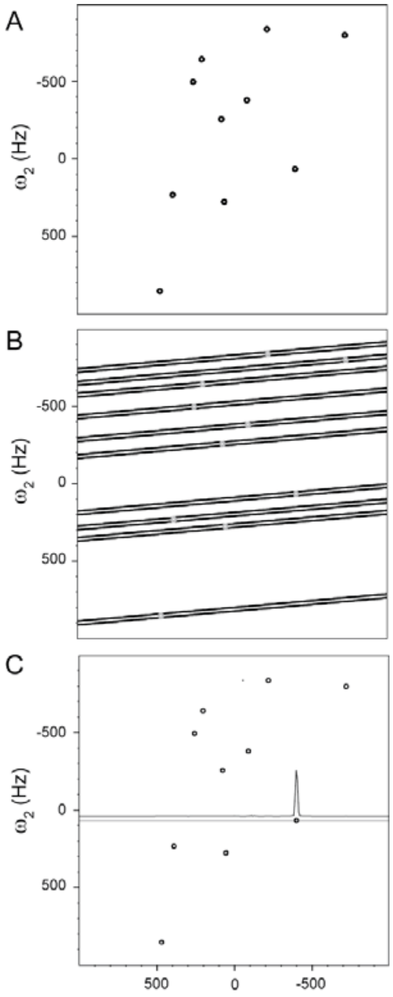


Figure 5.2 Comparison of Cartesian and radial sampled spectra illustrating how appropriate angle selection can speed data collection. Spectrum A shows the comparison Cartesian sampling spectrum. Spectrum B demonstrates the selection of the minimum angles needed to resolve all of the peak intensities. For this set of peaks the positive slope component of 85° sampling angle resolves all of the peaks. The data was processed with the matching and non-matching direct two-dimensional Fourier transform to isolate the positive ridge component. For clarity the Cartesian sampled spectra was overlaid in gray. Spectrum C demonstrates selecting the minimum angles to produce a spectrum with no artifact peaks. Here data was generated with two radial sampling angles, 6° and 85°. The data was processed with the matching and non-matching direct two-dimensional Fourier transforms to isolate the positive and negative ridge components of the two angles generating four spectra. The four spectra were compared with the lower value algorithm

to produce an artifact free spectrum. A slice take at 60 Hz is overlaid, with slight offset for clarity, demonstrates an artifact free baseline.

5.2.4 Minimum sampling angles to generate an artifact free spectrum

In the second scenario that is likely to be encountered, an artifact free spectrum is desired. To produce such a spectrum the lower value algorithm is used to remove the artifacts, the success of which is dependent upon the collection of an appropriate set of sampling angles. If suboptimal sampling angles are used intensity can remain at non-authentic peak locations. In a manner similar to above, we apply the ridge to ridge distance algorithm to determine all of the potential artifact positions, and subsequently apply the peak to ridge distance algorithm to determine if potential artifact positions are resolved in at least one of the selected sampling angles and will consequently be removed by the lower value algorithm.

As before, the first step is to edit the peak list to combine truly unresolved peaks. Sets of unresolved peaks are replaced by a single peak with an adjusted linewidth to account for their unresolved components. Unlike for the previous case described above, the sampling angles are no longer independent, which requires sets of angles to be selected. The number of angles and which angles selected can be definitively determined. If some angles must necessarily be collected, such as the 0° and 90° used to determine phase corrections(Chapter 4), they can be included in every set of angles tested.

Typically, initial tests use a small number of angles and increase the number if all of the artifacts are not removed after a given number of trials.

For a given set of test angles, all of the artifact positions are determined through application of the ridge to ridge distance algorithm. This is accomplished by iterating over each set of angles for each peak against every other ridge. For example, in the case where there are two peaks, P_1 and P_2, and two sampling angles, α_1 and α_2, the ridge extending from peak P_1 at $+\alpha_1$ would be tested against the $-\alpha_1$ ridge of P_2 and both the + and $-\alpha_2$ ridge of P_2. The other three ridge components of P_1 are tested in the same manner. To speed the analysis, only those peaks that are not resolved in the directly detected dimension are tested. The list of all potential artifacts is then edited to remove peaks that are in the same location as the authentic peaks. If the primary concern is to determine the chemical shifts, removing the overlapping artifacts from the list will not affect the final spectrum, it can only affect the peak intensities. If the intensities are a concern the minimum angle to resolve peak intensity algorithm can be run first to select the angles that resolve intensities. The intensities can then be extracted from the appropriate angle spectra.

The final step is to determine if the potential artifacts will be removed during lower value comparison. This is accomplished by applying the peak to ridge distance algorithm. The ridges will only extend from the authentic peaks, and not the potential artifact positions. Accordingly, the potential artifact positions are tested against all of the ridges extending from the authentic peaks. If an artifact position is resolved in one of the angles, the potential artifact will be removed during the lower value comparison. A list of unresolved artifacts is thereby compiled. If the number or location of the remaining artifacts is unsatisfactory a new set of angles is then tested.

This algorithm was test against same 10 peak generated data test case used in the previous algorithm. Analysis of the peak list concluded that two sampling angles, 6° and 85° are needed to remove all of the artifacts. The results are shown in Figure 5.2c. Comparison with the Cartesian sampled spectrum indicates that only authentic peak intensity remains after the lower value comparison.

5.2.5 Spectrum analysis and iterative data collection

In the two previous scenarios, the peak positions are known and the appropriate angles to either resolve peak intensities or resolve all of the artifacts can be determined unambiguously. In situations where the position of the authentic peaks are not known rather than testing for sampling angles that resolve the potential artifacts, the remaining peak intensity in a lower value comparison spectrum needs to be tested. Without knowing the location of the authentic peaks, all intensity in a lower value spectrum must be treated as a potential peak until it is determined to be authentic. If the intensity in a lower value spectrum is resolved in at least one angle, the peak must be authentic. If two peaks are not resolved they are marked as potential artifacts until additional data resolves them.

The first step in this analysis is to collect an initial data set, process the angles separately, compare them with the lower value algorithm and generate a peak list. All such peaks are considered potentially authentic at this point. The peak to ridge distance algorithm is applied to test if a potential peak is resolved from the ridges from all of the other potential peaks. The ridges are generated at each of the sampling angles used. If a peak is not resolved at any of the sampling angles, it is marked as a potential artifact. After a list of potential artifacts is generated, the set of minimum angles to resolve all of the potential peak intensities is determined as described above. Additional data is then collected at the suggested angles. The data is processed and compared, using the lower value algorithm, to the previous spectrum. A new peak list is created and analyzed and the process is repeated until all of the intensity is resolved, or the remaining potential artifacts don't complicate further analysis.

Figure 5.3 demonstrates this method of iterative analysis and data collection. Here the same 10 peak test case was used as before. For the first round of data collection, data was generated at 0° and 90°, processed independently and compared with the lower value algorithm. The resulting lower value spectrum is shown in Figure 5.3a. As anticipated, it is impossible to determine the 10 authentic peaks from the 100 potentially authentic peaks. Analysis of the peak list generated from the spectrum in Figure 5.3a led to the conclusion that a sampling angle of 35° was optimal. An additional data set was generated at 35°, processed and compared using the lower value algorithm to the 0° and 90° lower value spectrum. As shown in Figure 5.3b, inclusion of the additional sampling angle removed a large number of the additional of the potential artifact peaks, leaving 11 peaks in the spectrum. The one artifact in the spectrum is circled to highlight that without the analysis described here it is impossible to distinguish it from an authentic peak. A subsequent iteration determined that a sampling angle of 73° would resolve all of the remaining peaks, removing any potential artifacts. Generating the additional data set, processing and comparison to the 0, 90 and 35 lower value spectrum produces the spectrum shown in Figure 5.3c. Analysis of this spectrums peak list determines that all of the peaks must be authentic because they are all resolved from ridge artifacts.

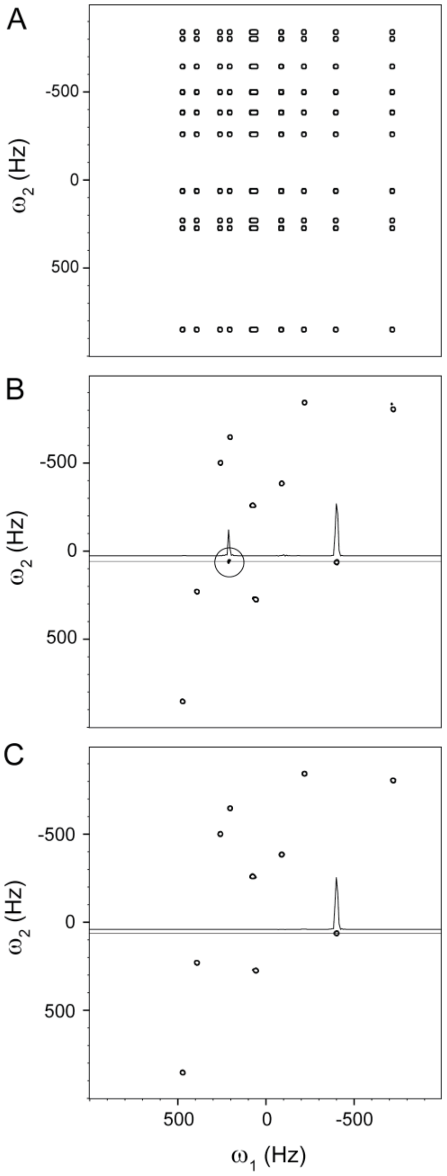


Figure 5.3 Demonstration of iterative angle selection and spectrum analysis. Spectrum A. shows the lower value comparison of the face, 0 and 90 sampling angle, spectra. The peak list of the spectrum A was analyzed and a 35° sampling angle was determined to remove the most artifacts. Spectrum B shows the lower value comparison of the 0°, 35° and 90° sample angle spectra. The circled peak indicates a peak in the spectrum that was determined to be a possible artifact. The overlaid slice demonstrates the relative intensity of the possible artifact peaks. Analysis of the peak list from this spectrum determined that a sampling angle of 73° would remove any remaining artifacts. Spectrum C shows the lower value

comparison of the resulting spectra from sampling angles at 0°, 35°, 73° and 90°. Analysis of the peak list from this spectrum determines that all peaks are resolved at least one of the sampling angles and therefore must be authentic peaks and not artifacts. The overlaid slice demonstrates the removal of the artifact peak. Both the slices in B and C are slightly offset for clarity.

3. Results

We have essentially described three algorithms: finding a minimum set of sampling angles to resolve authentic intensities from ridge artifact intensity; finding a minimum set of sampling angles to remove all ridge artifacts from the spectrum; and an iterative analysis and data collection procedure for obtaining an artifact free spectrum when the positions of authentic peaks are not known *a priori*. Each procedure was tested in the context of the HNCO spectrum of recombinant human ubiquitin. The results are illustrated in Figure 5.4-5.6. To establish the minimum set of sampling angles to resolve authentic intensity, an initial peak list was derived from the conventional Cartesian sampled HNCO spectrum (Figure 5.4a) though such a ^{15}N,^{13}C list could be taken from any reliable source. This spectrum also served as a comparison with equivalent resolution to the radial sampled spectrum. High resolution was achieved by collecting 64 increments in both indirect dimensions. Accordingly, the data collection time was approximately 36 hours. Analysis of the peak list suggested sampling angles of 36° and 90° would resolve all authentic peak intensity from artifactual intensity. Indirect dimension slices of the single step two dimensional FT of the positive slope component of 36° and 90° sampling angle data sets are shown in Figure 5.4b and 5.4c. Total data collection time for the two angle planes was 51 minutes corresponding to a 43 fold time advantage over Cartesian sampling. These slices are all taken at 8.15 ppm in the ^{1}H acquisition dimension (ω_3). Importantly, when measuring peak intensities it is clearly advantageous to separate the spectrum into the individual ridge components. This aids in determining the intensity because distracting artifact peaks are not present. Additionally, by separating the spectrum into its ridge components fewer sampling angles need to be collected. The spectra are not symmetrical so each ridge component contains unique data. Separating the ridges also aids in removing artifact peaks if the lower value algorithm is applied. In summary, these spectra demonstrate the successful resolution of authentic peak intensity from artifactual ridge intensity in a deterministic manner. Analysis of the entire 3D spectrum derived from the two sampling angles confirmed that all of the peak intensities are resolved (data not shown).

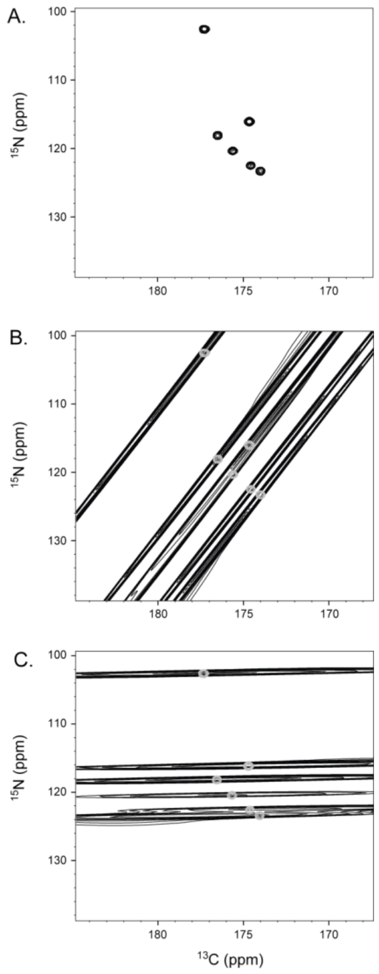


Figure 5.4 Demonstration of the minimum angles needed to determine the peak intensities for ubiquitin using a HNCO. Peak list analysis determined that sampling angles of 36° and 90° would resolve all of the peak intensities. Shown here are the ^{13}C- ^{15}N indirect planes of three HNCO spectra at ^{1}H shift of 8.15 ppm. Spectrum A shows Cartesian sampled experiment as a reference. Spectra B shows the 90° sampling angle spectra while spectrum C shows the positive ridge component of the 36° sampling angle. Note that the peaks that are not resolved at 90° are resolved at 36°.

The two angles selected to resolve peak intensities are not a unique solution. However, the combination of 36° and 90° was selected for multiple reasons. First, it is advantageous to include the 0° or 90° projections or "faces" as they are needed to determine the phase corrections (Chapter 4). Additionally, the 0° and 90° faces can be collected with two quadrature components as compared to four needed for other angles. Finally, the faces are only affected by relaxation arising from spins associated with only one incremented time domain.

In cases where the spectrum is especially complex, it might not be possible to choose a set of angles that resolve all peak intensities. In this circumstance either a subset of peaks must be focused on or an alternate experiment must be chosen. When a subset of the peaks are focused on, the sorting routine can be modified to include a weighting term to favor the angles that resolve the peaks of interest. While the time savings occurred by radial sampling make it appealing, the algorithm described here allows a definitive mechanism for deciding whether radial sampling is applicable.

The same approach was used to test the procedure for defining the minimum set of angles necessary to remove all artifacts from the spectrum. Analysis of the peak list concluded that three sampling angles (0°, 35° and 90°) would suffice. The total measurement time for the three angles is 68 minutes corresponding to 32 fold time advantage over equivalent resolution Cartesian sampling. A representative slice of the HNCO spectrum illustrates that only the desired authentic intensity is present (Figure 5.5). Again, the three angles selected by the algorithm to remove all of the artifacts are not unique; other combinations of angles would produce the equivalent results. In this case the 0° and 90° sampling angles were required to be in the angle set in order to determine the necessary phase corrections(Chapter 4). Other additional angles could be included in the angle set. Time is the only disadvantage to including more angles if the minimum angles are known. Including additional angles will not produce ridge artifacts.

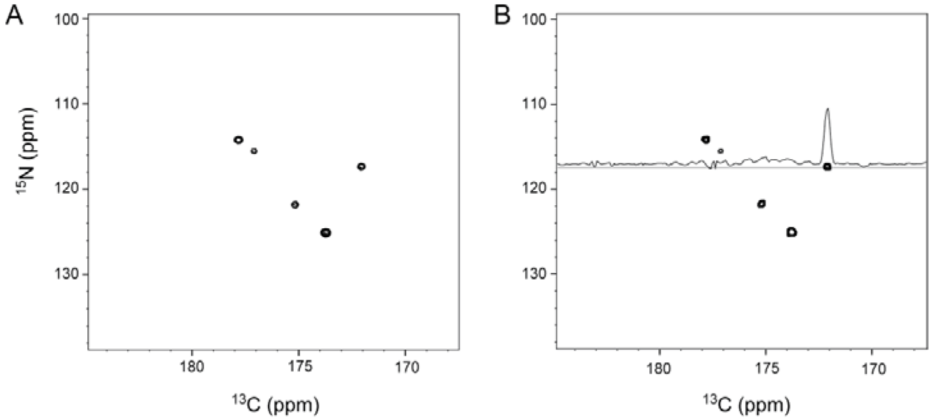


Figure 5.5 An example of calculating the fewest angles needed to generate an artifact free HNCO spectrum of ubiquitin is shown here. Spectrum A shows the comparison Cartesian sampled spectrum at 1H 8.71ppm and Spectrum B shows the same indirect slice of the radial sampled experiment using the calculated sampling angle of 0°,36° and 90°. The overlaid slice demonstrates the typical baseline quality for the entire 3D spectrum.

The number of angles needed to remove all of the artifacts can be decreased by reducing the linewidth of the peaks. The algorithm is based on a distance measurement; therefore the effective distance between the peaks is optimized by reducing the linewidths. Standard methods can be used to decrease the linewidths. Increasing the number of increments if relaxation isn't limiting or using constant time approaches where the line widths is adjusted by the convolution and apodization functions are two obvious options. Effective use of linear prediction adapted to radial sampled experiments would also decrease the linewidths with a concomitant reduction in the minimum number of sampling angles required.

The final example illustrates the iterative data analysis and collection procedure used to faithfully reveal authentic peaks while suppressing artifactual intensity without prior knowledge of the peak positions. Figure 5.6 shows a representative indirect plane of the HNCO generated from lower value comparison of three sampling angles (0°, 45° and 90°). This was used as a starting point. Analysis of the peak list determined that a sampling angle of 67° would resolve the most additional peaks in the spectrum. The 67° sampling angle spectrum was collected and processed and compared to the (0°, 45° and 90°) lower value comparison spectrum. A representative slice is shown in Figure 5.6. Analysis of the resulting peak list concluded that all of the peaks were resolved and therefore authentic.

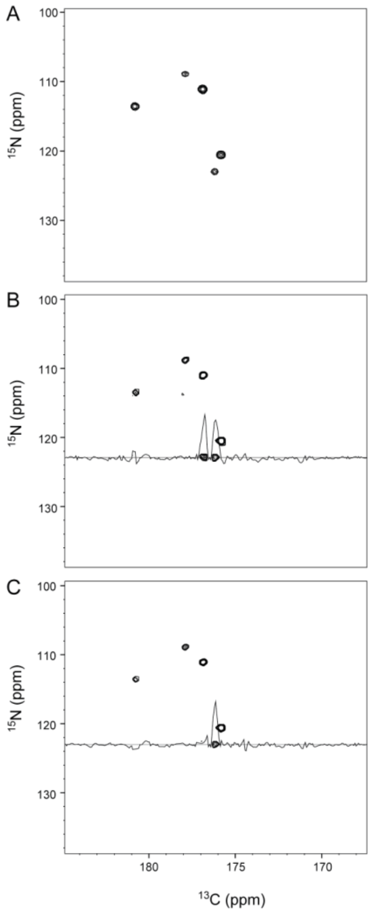


Figure 5.6 Demonstration of the use of iterative angle selection to generate an artifact free HNCO spectrum of ubquitin with radial sampling. For comparison spectrum A shows an indirect slice of the Cartesian sampled HNCO at 1H 8.49 ppm. Spectrum B shows the same indirect plane as A for the radial sampled data generated from the lower value comparison of 0°, 45° and 90° sampling angle spectra. Analysis of the peak list for the entire 3D experiment concludes that a sampling angle of 64° will remove the most remaining artifacts, if any. Spectrum B shows the lower value comparison of 0°, 45° and 90° with a newly collected 64° spectrum. Notice the removal of one artifact peak, as demonstrated by the overlaid slices. Analysis of this peak list concludes that all peaks in the spectra are resolved and therefore authentic.

5.4. Discussion

The three algorithms described here provide a means to confidently collect radially sampled multidimensional NMR data that such that the integrity of peak intensity is maintained (algorithms 1 and 2) or the spectrum entirely free of artifacts arising from ridge intensity inadvertently surviving the lower value data reduction (algorithm 3). The retrospective spectrum analysis described here removes all uncertainty as to whether a peak is authentic or artifact through a quantitative measure of resolution. Furthermore, the approach optimizes the data collection by reducing, if not eliminating, the collection of unnecessary data and identifying when sufficient data has been collected to produce a suitable spectrum. From a practical point of view, any inefficiency that is introduced by the analysis during data collection can be overcome by collecting angles of other experiments while it is being performed. Typically assignment experiments are run as pairs, so a second experiment is collected concurrently. Regardless, the analysis is rapid and not computationally intensive. Additionally, once the first peak list from the initial data set is generated, additional rounds or analysis are much faster. Automation could be applied to this step very easily. Though only a (3,2) radially sampled HNCO spectrum was used to illustrate the potential of the three algorithms described here, all of the methods presented are directly amenable to higher dimensional experiments. Iterative data analysis and collection is particularly appealing in high order nD experiments where sensitivity and resolution are generally limiting. While radial sampling affords an easy method to increase the resolution of such experiments, optimal data collection allows for less angles to be collected and more time to be used for signal averaging with the attendant gain in signal-to-noise. Future work will assess optimized radial sampling in sensitivity limiting cases and in the case of 4-dimensional spectra.

5.5. Methods

All simulated data was created using a set of ten peaks distributed in two dimensions to simulated the two linked indirect dimensions of a (3,2) sampled experiment. The randomly assigned resonance frequencies of the ten peaks are: (248.9, -503.4); (-97.7, -387.2); -(226.5, -844.5); (67.6, -263.5); (462.7, 845.5); (-407.8, 58.3); (194.1, -649.1); (380.9, 224.9); (47.9, 269.3) and (-727.8, -806.1) Hz. The linewidths of all of the peaks was set to 50 Hz in both dimensions. The spectral width was set to 2000 Hz in both dimensions. Each sampling angle used was the result of four quadrature data components collected with 128 increments.

NMR data was collected on a 900 μM ^{13}C, ^{15}N uniformly labeled sample of human ubiquitin at 25° C on a Varian INOVA 500 MHz spectrometer. The sample conditions consisted of 50mM phosphate buffer pH 5.5, 50mM NaCl and 0.04% Azide. Recombinant ubiquitin was prepared as described[62]. NMR data was collected using a standard HNCO[60] or a modified version for radial sampling, such that $t_1 = t_1 \cos(\alpha)$ and $t_2 = t_1 \left(sw_1 / sw_2 \right) \sin(\alpha)$ with the following experimental conditions. For radial sampled data each sampling angle, other than 0 and 90, was collected with four quadrature components at 64 increments composing 256 FIDs. The 0 and 90 sampling angles were collected with two quadrature components at 64 increments composing 128 FIDs. Cartesian sampled data was collected with equivalent resolution using 4 quadrature components at 64 increments in both indirect dimensions composing 16384 FIDs. In both sampling schemes each FID contained 512 complex points and was the average of eight scans, the minimum number of phase cycling steps stated in the original reference. Using a 1.0 second interscan delay the measurement times for 0 and 90 sampling angles was 17 minutes. The measurement time for all other angles was 34 minutes. The measurement time for the Cartesian sampled spectrum was 36.4 hours. The spectral width was set to 12 ppm in the proton dimension. The spectral widths for the indirect dimensions were chosen to assure no peaks were folded and set to 17.5 and 40 ppm in the carbon and nitrogen dimensions respectively. With the corresponding carrier frequencies set at 176 and 119 ppm.

The angle spectra were processed independently using a direct 2D Fourier transform. Prior to Fourier transforming the data was apodized and zero filled. A cosine squared apodization function was applied to remove truncation artifacts and to approximate the correction for unequal spaced data. Subsequently the angle spectra were compared using the lower value (magnitude) algorithm to remove the ridge artifacts. The Cartesian sampled data was processed with corresponding techniques in one dimension. All processing was done using Al NMR and visualized using Sparky[51].

Chapter 6

SEnD NMR: Sensitivity Enhanced n-Dimensional NMR

6.1 Introduction

As presented in Chapter 1, application of NMR to large proteins is inhibited by both resolution and sensitivity. Through application of sparse sampling, the resolution limitation is alleviated. Here the alternative situation is considered, where the goal is to increase the sensitivity of a given spectrum. Briefly, to accomplish this we will use a combination of radial sampling[25] and a previously unexploited statistical property of the data. Chapter 5 demonstrated that the defined pattern of artifacts arising from the multidimensional FT of radial sampled data allows the definition of a set of algorithms to optimize angle selection. As we demonstrate here, optimized angle set collection for radial sampling provides significant freedom for the further optimization of multidimensional NMR spectra with respect to signal-to-noise (S/N). We will first present the theory and resulting criteria that suggests how data optimized for S/N should be collected. We then illustrate how an inherent feature of radial sampling provides the subsequent opportunity to utilize non-linear statistical methods to exponentially reduce the noise without introduction of artifact. Providing the criteria underlying the basic approach are met, a substantial sensitivity advantage can be achieved.

6.2 Theory

Here we are initially interested in the strength of the signal obtained by radial sampling versus that obtained by normal orthogonal (independent) Cartesian (uniform linear) sampling of the time domains t_1 and t_2. Consider heteronuclear spins that are J-coupled. The maximum signal, S_{max}, subsequent to Fourier transform[63] of the Cartesian sampled data for the incremented time domains where J-coupling is suppressed during evolution is:

$$S_{\max} = \sum_{t_2}^{t_2^{\max}} \sum_{t_1}^{t_1^{\max}} \cos^2\left(2\pi\omega_1 t_1\right) \cos^2\left(\pi\omega_1 t_1 / 2t_1^{\max}\right) \cos^2\left(2\pi\omega_2 t_2\right) \cos^2\left(\pi\omega_2 t_2 / 2t_2^{\max}\right) \tag{6.1}$$

where the various terms have their usual meanings. Apodization is included as a cosine squared function. This is compared to the maximum signal intensity of radial sampled data by substituting the two incremented time variables for one variable τ, such that $t_1 = \tau\cos(\alpha)$ and $t_2 = \tau\sin(\alpha)$, in both the signal and apodization terms. Replacement of the two incremented time variables with one term results in only a single summation:

$$\begin{aligned} S_{\max} = &\sum_{\tau}^{\tau^{\max}} \cos^2\left(2\pi\omega_1 \tau \cos\left[\alpha\right]\right) \cos^2\left(\pi\omega_1 \tau \cos\left[\alpha\right] / 2\tau^{\max} \cos\left[\alpha\right]\right) \\ &\times \cos^2\left(2\pi\omega_2 \tau \sin\left[\alpha\right]\right) \cos^2\left(\pi\omega_2 \tau \sin\left[\alpha\right] / 2\tau^{\max} \sin\left[\alpha\right]\right) \end{aligned} \tag{6.2}$$

In both cases, the effects of relaxation are not included for clarity.

Figure 6.1 demonstrates how the maximum signal intensity ratio changes as a function of the number of incremented points assuming the case that the line shape is dictated by the apodization function. A 45 degree sampling angle was used in this example. The captured volume ratio shows a non-linear dependence because the effects of apodization do not scale the signal intensity in radial sampling as substantially as when Cartesian sampling is used. This nonlinear effect is accentuated further if relaxation is included. Additionally, if multiple angles are collected a higher density of points are collected at short evolution times, while the density decreases at long evolution times. The difference in density, between Cartesian sampling and radial sampling produces nearly a 70% gain in total signal intensity if the same numbers of points are evaluated using equally distributed angles.

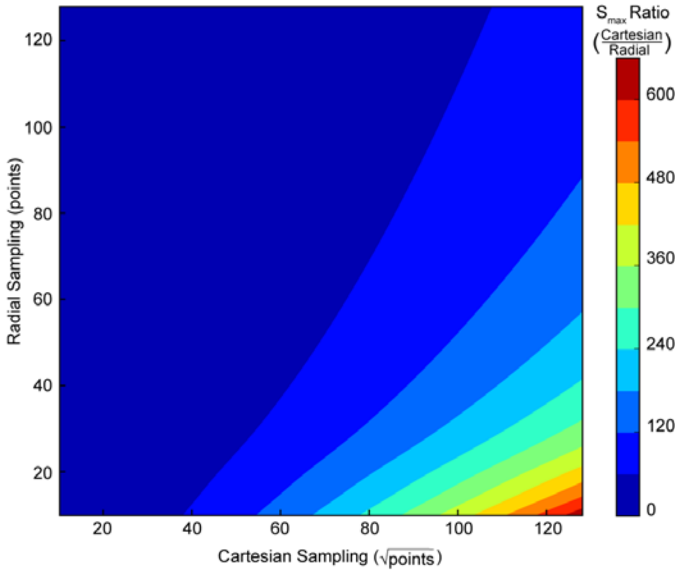


Figure 6.1 The ratio of maximum signal intensity of Cartesian sampling to radial sampling is demonstrated here as a function of number of increments. This plot was generated by solving for the quotient of

equations 1 and 2. The Cartesian data was generated as a NxN grid of points, with the square root of the total number of points indicated. This is compared to the radial sampled data generated at a 45 degree sampling angle. Both data sets were multiplied by a cosine squared apodization function and the real Fourier transformed to calculate the maximum signal intensity. The plot indicates the nonlinear change in volume as the number of sampling points is varied.

The number of transients per free induction decay (FID) and the total number of increments of the time domains influence the signal-to-noise for both Cartesian and radial sampling, the latter in less obvious ways than the former. Consider well-behaved noise described by a Gaussian distribution with a standard deviation, σ, with a probability distribution function described by:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} \tag{6.3}$$

Where σ is the standard deviation of uncorrelated noise distributed about zero. The breadth of the noise changes as the square-root of the number of data points acquired:

$$\sigma_{i+j} = \sigma_i \sqrt{n_j} \tag{6.4}$$

Here σ_i is the starting standard deviation of the noise and σ_{i+j} is the standard deviation of the noise after adding *j* points to the original *i* points. Obviously using radial sampling fewer increments are necessary, per angle, and less noise is introduced into the spectrum. This equation does not account for the change in noise as a function of apodization. The dependence of the noise on the square root of the number of increments is a result of the Fourier transform rescaling the noise of each point as a function of the summation over all of the data points. Each data point measured has a distribution of noise centered on the data point. The distribution is scaled depending on the incremented time value that it is being multiplied by the Fourier transform coefficients. The mean value of the Fourier transform coefficients is $\cos(\pi/4)$. Using the sum of the variance law, the standard deviation of the noise after the Fourier integral is $\sqrt{n}\cos(\pi/4)$ and $\sqrt{n}\sin(\pi/4)$ for the real and imaginary components, respectively. Summing the two Fourier transform components results in the general form shown in equation (6.2).

The noise term as a function of the number of transients is commonly thought of in terms of the variance sum law; which concludes the noise decreases by the square root of 2 as the number of transients doubles. Written as a continuous function it takes the form:

$$\sigma_{i+j} = \sigma_i \left(\sqrt{2}\right)^{-\log_2\left(\frac{n_i+n_j}{n_i}\right)} \tag{6.5}$$

As before, σ_i is the standard deviation of the noise after n_i transients and σ_{i+j} is the standard deviation of the noise after n_{i+j} transients.

The lower magnitude algorithm was originally introduced for the processing of two-dimensional ^{1}H-^{1}H spectra where the asymmetrical features of prominent artifacts such as "t_1 noise" could be used to distinguish them from authentic peaks[64]. The same algorithm has been employed in projection reconstruction spectra with a similar intent[25]. For both projection reconstruction and its direct multidimensional Fourier transform counterpart, the lower magnitude comparison is used to remove ridge artifacts from the spectrum. A complete description of the ridge artifacts is available in Chapter 5. Briefly, the lower magnitude algorithm compares each equivalent data point of spectra obtained with different radial sampling angles and selecting the lowest magnitude value. Because the ridge artifacts are dependent upon the sampling angle, this comparison efficiently removes them. Importantly, because the ridge artifacts are limited to the vectors extending from authentic peaks, the baseline noise is reduced through the comparison. This is the key to a significant S/N advantage offered by the appropriate use of radial sampling and the lower value algorithm.

A formal analysis of the statistical properties of the lower magnitude algorithm has not been described. Typically, it is effectively assumed that the average deviation decreases linearly as a function of the number of angles employed[65]. This description is insufficient to provide means to analyze the change in the standard deviation of the noise. The change in the noise standard deviation can be analyzed by inspection of the probability density function. To determine the probability density, the probability distribution is first derived for the lower magnitude comparison. The lower magnitude comparison compares the absolute magnitude of the values and selects for the minimum magnitude value after n comparisons of different spectra. Assuming a normal Gaussian distribution of noise, the probability distribution for a single angle may be written as:

$$P^{\alpha}(x \leq X) = \frac{1}{2}\left(1 + erf\left[\frac{x}{\sigma\sqrt{2}}\right]\right) \tag{6.6}$$

To account for the magnitude comparison of the lower value (LV) algorithm, we define $y = |x|$. The LV algorithm compares values from *n* trials and selects the single lowest magnitude value. To determine the probability for a single value of y ≤ Y, after n trials, we employ the fact that the probability that a single value that satisfies y ≤ Y is the complement probability that the same value satisfies y ≥ Y.

$$P^{\alpha}(y \geq Y) = 1 - P^{\alpha}(y \leq Y) \tag{6.7}$$

where P^{α} is the probability for a given angle spectrum. Multiple independent sampling angles are compared, making the probability of each event exclusive and therefore the probability that all trials fulfill $y \geq Y$ is expressed as the joint probability of each event. Thus the probability that all values are greater than or equal to Y across the *n* radial angles is:

$$P^{LV}(y \geq Y) = (1 - P^{\alpha}(y \leq Y))^{n} \tag{6.8}$$

The complement of this expression, the probability that a value approaches zero, is then:

$$P^{LV}(y \leq Y) = 1 - \left[1 - P^{\alpha}(y \leq Y)\right]^{n} \tag{6.9}$$

The symmetric property of $P^{LV}(y)$ allows the probability distribution to be converted back into a continuous function in terms of x. The probability of P(x) is written as:

$$P^{LV}(x) = \frac{1}{2}\left(1 - erf\left[\frac{x}{\sigma\sqrt{2}}\right]\right)^{n} \qquad x \geq 0 \tag{6.10a}$$

$$P^{LV}(x) = 1 - \frac{1}{2}\left(1 - erf\left[\frac{x}{\sigma\sqrt{2}}\right]\right)^{n} \qquad x < 0 \tag{6.10b}$$

Having determined the probability distribution for the lower magnitude comparison of n angles, the probability density and subsequently the standard deviation change are can now be written. The derivative of equations 6.10a and b is the probability density function[66].

$$p(x) = \frac{dP^{\alpha}(x \leq X)}{dx} \tag{6.11}$$

The probability density function can then be applied to determine how the noise changes as a function of the number of angle spectra compared[66].

$$\sigma = \sqrt{\int x^{2} p(x) dx} \tag{6.12}$$

The probability distribution function and probability density functions are plotted as a function of angle comparisons in Figures 6.2a and b. The resulting change in the standard deviation of the noise is shown in Figure 6.2c. The exponential decrease in the noise demonstrates that it can be efficiently reduced with a relatively small number of angle comparisons. For example, collecting five angle spectra and processing them into the positive and negative component spectra (Chapter 4) for each angle and then comparing the resulting ten component spectra would reduce the standard deviation of the noise by 84%, as demonstrated in Figure 6.2c.

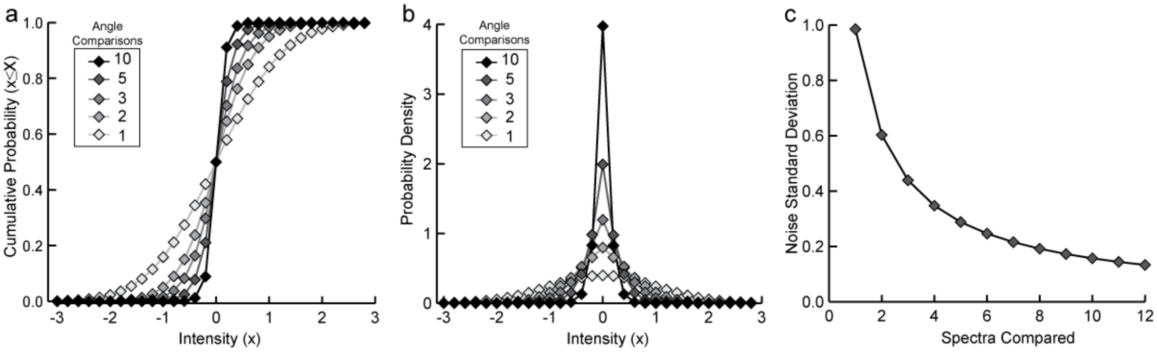


Figure 6.2 The effect of the lower magnitude comparison on the noise is demonstrated as a function of number of angle spectra compared. The probability distribution function is shown in Panel A. The cumulative probability is plotted against the noise intensity for 1 (no comparison), 2, 3, 5 and 10 angle spectra comparisons. The corresponding probability density plots are shown in Panel B for the same numbers of angle comparisons. The change in the standard deviation of the noise is plotted against the number of angle spectra compared in Panel C.

This initial result might suggest that it is advantageous to collect a large number of angles, with minimal transients, in order to take advantage of the exponential decay of the noise standard deviation. However, because the lower magnitude comparison is nonlinear, using a large number of angles collected with fewer transients and attendant lower S/N can potentially significantly degrade the quality of the final spectrum since the distributions of noise and authentic peak intensity may overlap. In order to avoid this situation a requisite signal-to-noise of the angle spectra needs to be defined. For example, a S/N value of 6 will assure that a peak will essentially never be eliminated during lower magnitude comparison as 99.7% of a Gaussian distribution is contained within 3 standard deviations of the mean. The concept is illustrated in Figure 6.3 which depicts the probability density of signal intensity and noise density of a single angle spectrum and the final lower magnitude spectrum from the comparison of 10 angle spectra. This is the essence of the Sensitivity Enhanced n-Dimensional NMR (SEnD) strategy.

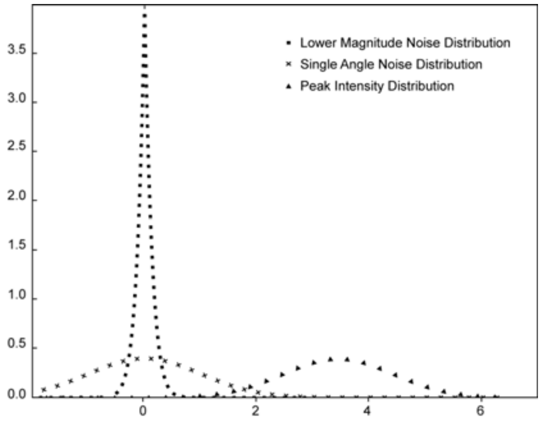


Figure 6.3 A graphical representation of the probability density analysis to retain a peak is shown here. The distributions for peak intensity, and noise intensity before and after lower magnitude comparison are also indicated. The lower magnitude probability density was generated assuming the comparison of 10 angle spectra.

6.3 Results

The SEnD NMR strategy was tested with the HNCO experiment using a 20 μM ^{13}C,^{15}N-ubiquitin sample[62]. Three HxCO projections or faces of the HNCO[60], corresponding to a radial sampling angle of zero, were collected with 4, 8 or 16 transients (Figures 6.4a-c). The corresponding full radial sampled three-dimensional experiments with ten angles equally distributed between 0 and 90 were collected to demonstrate the effects of varying

S/N on the final lower magnitude spectrum. Representative two-dimensional slices of these spectra are shown in Figures 6.4d-f. For this particular sample, use of four transients per FID resulted in an average cross peak S/N of 3 . This is well below the necessary S/N required by the SEnD approach and authentic peaks were indeed removed during lower value comparison (Figure 6.4d). In the case of the spectrum obtained with eight transients the average S/N of peaks was 5.5. Since this is slightly below the SEnD criterion of 6 particular attention would need to be paid to the weakest peak. This is demonstrated in the Figure 6.4e where the lower magnitude processed spectrum contains all of the peaks but the intensities and lineshapes are not uniformly accurate. When 16 transients are used all of the peaks have a signal-to-noise greater than the SEnD minimum of 6 and all are accurately represented accurately represented in the lower value processed three dimensional spectrum (Figure 6.4f).

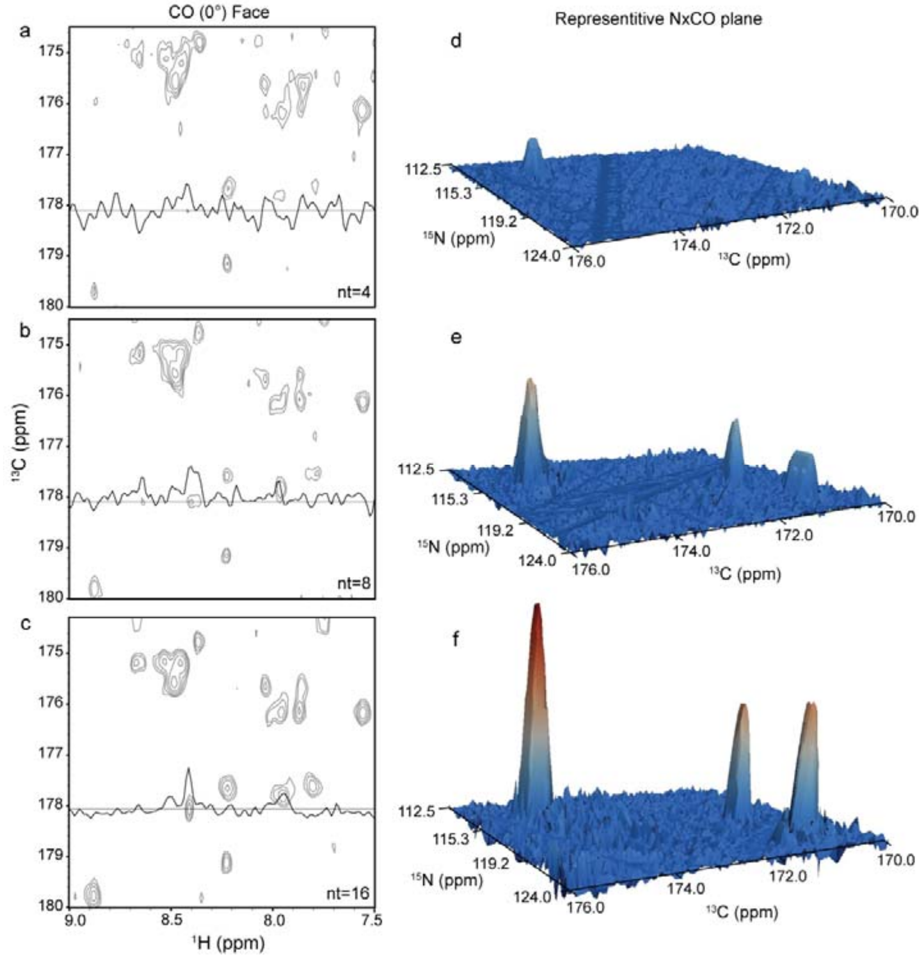


Figure 6.4 The influence of minimum signal-to-noise to retain a peak is demonstrated here using 20 μM ubiquitin. HxCO faces of the HNCO are used to assess the signal to noise of the radial sampled angle planes. The S/N was varied by changing the number of transients. 4, 8 and 16 transient spectra are shown in spectra A, B and C respectively. One-dimensional slices are overlaid to illustrate the quality of the data. The average S/N of the three planes was 3 for 4 transients, 5.5 for 8 transients and 8.2 for 16 transients. Stacked plots of representative indirect planes of the lower magnitude spectra when 10 angles were compared are shown for each of the three settings of transients employed. The n=4 transient/FID spectrum is shown in Panel D, 8 transient spectrum in Panel E and 16 transient spectrum in Panel F.

Statistical theory predicts that a significant sensitivity advantage over Cartesian sampling can be achieved for a fixed unit of acquisition time by applying the SEnD criteria to radial data acquisition. This was tested by varying the number of transients while concomitantly changing the number of angles and keeping the total experiment time constant. The results are shown in Figure 6.5. Here four radial angle experiments were collected on a 1mM ^{13}C,^{15}N-ubiquitin sample , each requiring 7 hours of data collection. A corresponding traditional Cartesian sampling spectrum was also obtained. The radial sampled experiments were collected with equivalent resolution to the Cartesian experiment but varied the number of transients and radial angles as follows: 32 transients with 5 angles, 16 transients with 9 angles, 8 transients with 18 angles, and 4 transients with 36 angles. Each radial sampled data set equally distributed the angles used between 0 and 90. A substantial S/N advantage is achieved over Cartesian sampling when a large number of angles are used. This advantage is achieved because of the reduction in noise from the lower value comparison. When a smaller number of angles are used the S/N is comparable to Cartesian sampling. This indicates that when only a small number of angles are available Cartesian sampling might be desirable.

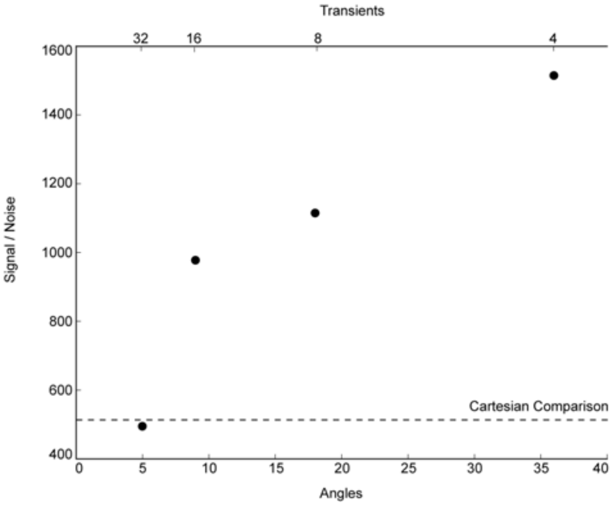


Figure 6.5 The advantage of optimizing data collection parameters is shown here. Five experiments were collected on 1 mM ubiquitin, all requiring 7 hours of measurement time. One experiment employed Cartesian sampling while the other 4 utilized radial sampling. The resolution was held constant by collecting 32 complex in both of the indirect dimensions of the Cartesian experiment, and 32 quatrion points for each angle in the radial sampled experiments. The four radial sampled experiments concomitantly varied the number of transients and angles to keep a constant experiment time. The four combinations used were 32 transients and 5 angles; 16 transients and 9 angles; 8 transients and 18 angles and 4 transients and 32 angles. The average S/N of all the peaks in the resulting lower magnitude spectra are plotted with the average S/N of the Cartesian experiment shown for reference.

To further illustrate the advantage of SEnD optimization equivalent resolution HNCO spectra were collected on a 20 μM ubiquitin sample. When Cartesian sampling was used the total measurement time was 7 hours and employed 4 transients and 36 complex increments in each of the indirect dimensions. For SEnD optimization experiment 5 angles, 32 transients and 36 quatrion data points were used for each angle. The requisite sensitivity of each angle spectrum was determined by collecting the HxCO face as a function of transients. The minimum number of transients required to satisfy the SEnD S/N criterion of 6 was determined to be 16, as demonstated by figure 6.4c.Thirty-two transients were used to ensure that the SEnD criteria was met for all peaks in order to account for any variation in peak intensity as a function of sampling angle. This defines the total number of angles for the fixed total acquisition time to be 6, including 0 and 90 which require half of the quadrature components and therefore require half of the measure time as compared to all other angles. The 6 angles were equally distributed between 0 and 90. Subsequent to generation of the final spectrum the spectrum was analyzed using the algorithms we have previously presented in Chapter 5 and all peaks were determined to be resolved. Comparison of the conventional Cartesian spectrum and the SEnD optimized spectrum clearly indicates the advantage of SEnD optimization (Figure 6.6). Analysis of the SEnD optimized spectrum allowed all expected peaks to be identified and had S/N distributed between 13 and 25. The equivalent peaks in the Cartesian spectrum has a S/N distribution between 4 and 10, further demonstrating the advantage of SEnD optimization.

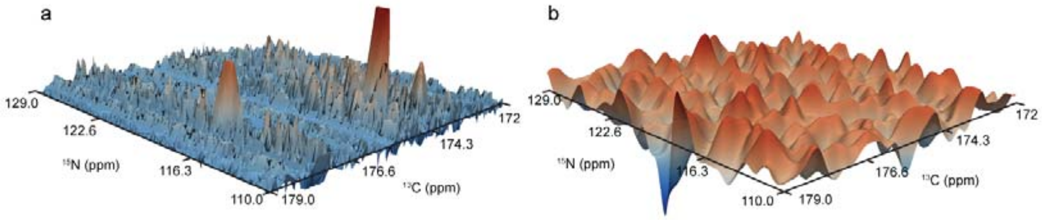


Figure 6.6 Comparison of SEnD optimized radial sampling (Panel A) and Cartesian sampling (Panel B) of an HNCO spectrum obtained on a 20 μM ubiquitin. Both spectra required 7 hours of acquisition time and were collected with equivalent resolution parameters. In the example shown, the SEnD spectrum has identifiable peaks while the corresponding peaks in the Cartesian spectrum are obscured by noise.

6.4 Discussion

Various parameters are associated with the final sensitivity of a multidimensional NMR spectrum. Here we have demonstrated a method, employing radial sampling, to optimize the sensitivity of a multidimensional NMR experiment. This method exploits the redundancy of the data collection, providing that a minimum S/N is achieved in each component radial spectrum. This provides assurance that authentic peaks will survive application of the lower value algorithm. Generally, a minimum signal-to-noise of 6, for each angle spectra, is sufficient. Effectively time allocated to increasing S/N in conventional experiments is redistributed to the collection of additional angle spectra that can be used to exponentially decrease the noise of the spectrum. Clearly the availability of cryogenically cooled probes and preamplifiers allows for the minimum S/N of individual angle spectra to more easily reached and emphasizes the synergy between high sensitivity probes and the SEnD methodology developed here. From a practical point of view, it is important to emphasize that it is possible to test for the satisfaction of the SEnD S/N criterion prior to acquisition of an entire data set. This is most easily accomplished by collecting a two-dimensional face of a three-dimensional experiment or the three-dimensional equivalent of a four-dimensional experiment. The projections allow one to conclude at the outset whether SEnD radial sampling is preferable to conventional Cartesian sampling with respect to final signal-to-noise.

The SEnD approach is generally applicable to all NH-based backbone resonance assignment experiments. Additionally, as we will report elsewhere, the sensitivity gain offered by the SEnD approach provides the opportunity for higher sensitivity and better digital resolution four-dimensional NOESY spectra to be obtained. The application of the SEnD approach in the context of quantitative or even semi-quantitative analysis of NOE peak intensities will require special consideration. This is because the largest negative deviation from a peaks mean is selected for during the lower magnitude comparison of all angle spectra. However, after application of the SEnD method to identify peaks, analyzing the individual angle spectra and treating them with the usual statistics for redundant measurement can recover accurate intensities. This will be described in more detail elsewhere.

Finally, our objective here has not been to carry out a comparison of all of the sparse sampling and processing methods available. Rather we have focused on exploiting the redundancy of the data, unique to radial sampling, in a manner to substantially reduce the spectrum noise and aid in peak identification. Nevertheless, the SEnD criteria be employed in conjunction with other methods capable of processing radial sampled data[34, 36, 43, 44].

6.5 Methods

NMR data was collected on either a 20µM or 1mM ^{13}C, ^{15}N uniformly labeled sample of human ubiquitin at 25°C on a Bruker Avance III 500 MHz NMR spectrometer equiped with a 5 mm triple resonance TCI cryogenic probe. The sample was prepared in 50 mM potassium phosphate buffer pH 5.5 with 50 mM NaCl and 0.04% sodium azide. Recombinant ubiquitin was prepared as described[62]. NMR data was collected using a standard HNCO[60] or a modified version for radial sampling, such that $t_1 = t_1 \cos(\alpha)$ and $t_2 = t_1 \left(sw_1/sw_2 \right) \sin(\alpha)$. The Cartesian experiment was collected using 36 complex points in both of the indirect dimensions for a total of 5184 FIDs. Each fid was the average of 4 transients and contained 512 complex points requiring approximately 7 hours of measurement time. The spectral width was set to 12 ppm, 30 ppm and 12 ppm for proton, nitrogen and carbon respectively. The carriers for each dimension were set to 4.682 ppm, 114.93 ppm and 174 ppm for proton, nitrogen and carbon respectively. The maximum acquisition times for the nitrogen and carbon dimensions were 0.0237 and 0.0239 seconds, respectively. In the case of radial sampling all experimental parameters were set to equivalent values as the Cartesian experiment unless otherwise noted in the main text. All of the radial sampled experiments utilized 36 quatrion data points, requiring 4 quadrature components per data point expect for the 0 and 90 spectra which only require 2 quadrature components. In the case where 4 transients were used, each sampling angle plane required 12 minutes of measurement time. The angle spectra were processed independently using a direct 2D Fourier transform. Prior to Fourier transforming the data was apodized with cosine squared function to remove truncation artifacts and to approximate the correction for unequal spaced data[46]. The data was zero filled to at least twice the number of incremented points. Following processing, individual angle spectra were compared using the lower value (magnitude) algorithm to remove the ridge artifacts[25]. The Cartesian sampled data was processed with corresponding apodization and zero filling. The fast Fourier transform was used in place of the direct 2D Fourier transform. All processing was done using an in-house program and visualized using Sparky[51].

Chapter 7

A Novel Approach to Radially Sampling the 4D ^{15}N, ^{13}C edited NOESY

7.1 Introduction

As presented in Chapter 1, as the molecular weight of a protein increases there is often a concomitant increase in spectral complexity. The complexity is particularly apparent in NOESY experiments[67], where spectral degeneracy of aliphatic protons makes obtaining, well resolved, structural restraints difficult. High resolution four dimensional NOE spectra often resolve the degeneracy and are therefore essential to structural analysis of larger proteins. The ^{13}C, ^{15}N edited 4D NOESY[68] is particularly appealing because it correlates one amide nitrogen atom to multiple aliphatic protons which are then resolved through evolution of the attached Carbon. A new application of radial sampling, which expands a ^{15}N edited NOESY[69] to a ^{15}N, ^{13}C edited NOESY is presented here. The approach relies on the technology developed in the previous Chapters. Particularly, the angle selection algorithm of Chapter 5 and the capabilities of Al NMR, Chapter 3, are utilized.

This application uses a peak list from an existing 3D experiment peak list to optimally select sampling angles for a 4,3 radial sampled ^{15}N, ^{13}C edited experiment. The 4,3 experiment resolves degeneracy present in the existing 3D experiment. Starting from existing information is advantageous because radial sampling angle selection can be optimized using the algorithm presented in Chapter 5. In turn, by optimizing acquired data points, this experimental scheme allows for collection of a high resolution 4D experiment.

To efficiently use the selected sampling angles a modified version of the 4D ^{15}N, ^{13}C edited experiment is presented. This experiment reverses the typical acquisition order of the experiment evolving the Nitrogen dimension first and the Carbon dimension after the NOESY mixing period. Means to accurately integrate the radial sampled experiment are also discussed.

7.2 Methods

7.2.1 Angle Selection

One of the substantial challenges of applying radial sampling to any Cartesian sampled experiment is selecting appropriate sampling angles. Without efficient angle selection time is potentially wasted by collecting angles that do not uniquely resolve all of the peaks in the spectrum. A deterministic method for efficient angle selection based on a starting peak list is presented in Chapter 5. This technology is applied to the 4,3 ^{15}N, ^{13}C edited NOESY experiment here. Starting with a peak list from a 3D ^{15}N edited NOESY[69] peak list a minimum set of angles is selected. The objective of angle selection is to select a set of angles that uniquely resolves all of the peaks in at least one angle. Appropriate angle selection ensures that accurate peak intensities can be determined from the angle spectra. If inappropriate angles are selected analysis will be obscured. If two peaks fall on the same ridge than the ridge intensity is the summation of both of the individual peak intensities.

To select appropriate sampling angles the peaklist from a 3d ^{15}N filtered NOESY is used as a starting set. Typically, the two indirect dimensions are used for radials sampling angle selection. In the case of a 3D ^{15}N filtered NOESY experiment the indirect NOE proton dimension is not suitable for radial sampling. The difficulty arises because of the number and distribution of peaks. An example ^{1}H NOE plane is shown in figure 7.1.

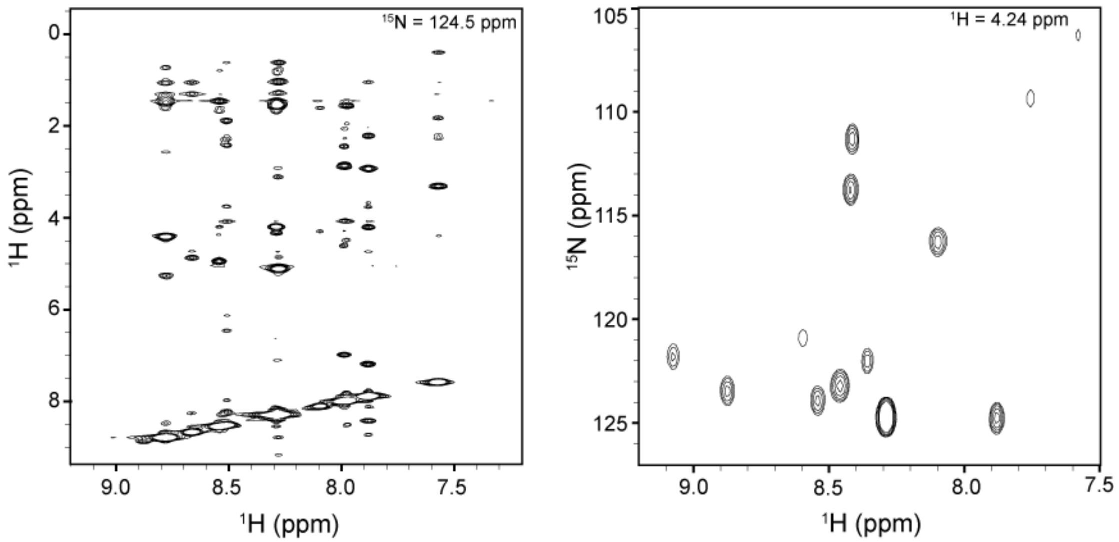


Figure 7.1 An example of the difficulty of using the indirect proton dimensions of the 3D ^{15}N filtered NOESY experiment for angle selection is shown here. An example plane, which demonstrates the complex pattern and number of peaks, of the indirect, aliphatic, proton dimension is shown on the left. This is compared to the directly acquired proton - nitrogen plane of the same experiment (right). The resolution and decreased number of peaks makes the amide proton - nitrogen plane appealing for radial sampling angle selection.

In this experiment the two indirect dimensions are the NOESY ^{1}H and ^{15}N. From inspection of planes between these two dimensions it is clear that selecting appropriate angles would be very difficult because of the large number of peaks per plane and additionally because all of the peaks are distributed along a given NH chemical shift. It is not possible to choose a unique set of angles that would resolve all of the peaks. Since this experiment is not used for radial sampling, the two indirect dimension do not need to be used for radial sampling. Other planes can be inspected. There are relatively few peaks in the amide ^{1}H - ^{15}N planes. The peaks are distributed and therefore, angle selection would be much more efficient. A typical amide ^{1}H - ^{15}N plane is shown in figure 7.1b. Few angles are needed to resolve all of the peak intensities in this plane. These two dimensions should be used for radial sampling.

7.2.2 Pulse Sequence

The tradition ^{13}C, ^{15}N edited NOESY[68] pulse sequence is not directly amenable to radial sampling if the amide ^{1}H and ^{15}N dimensions are to be used for radial sampling. In the traditional pulse sequence, the amide ^{1}H dimension is directly detected and it is not possible to co-evolve this dimension with the nitrogen dimension. In order to radial sample these two dimensions together, the pulse sequence is modified, reversing the acquisition order of the various nuclei. The modified pulse sequence is shown in figure 7.2.

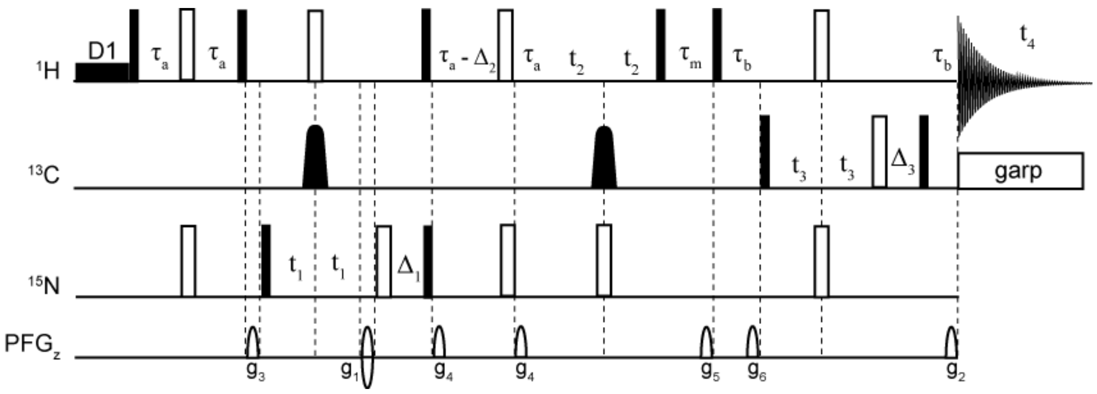


Figure 7.2 The modified ^{13}C, ^{15}N edited NOESY pulse sequence is shown here. Filled and unfilled rectangles indicate 90° and 180° pulses, respectively. The filled shaped pulses on the Carbon channel are 180° adiabatic Chirp pulses. This pulse sequence evolves ^{15}N during the first incremented delay, t_1, and amide ^{1}H during the second incremented delay t_2. To achieve radial sampling t_1 and t_2 are evolved as a function of the cos(α) and sin(α), respectively. ^{13}C is evolved in the third incremented delay t_3. Phase error is eliminated during all 3 incremented delays by refocusing any inadvertent evolution using the Δ delays. This assures that the first time point is set to zero. Decoupling during evolution of the indirect dimensions is achieved using 180° pulses. Hard pulses are used on both the ^{1}H and ^{15}N channels, a Chirp pulse is used on the ^{13}C channel. An INEPT transfer is used to initially transfer magnetization from ^{1}H to ^{15}N with τ_a set to $1/(4J_{NH})$. The same delay is used for the reverse INEPT. The NOESY mixing time, τ_m, was set to 120 ms. An HMQC transfer was used to transfer magnetization of ^{1}H to ^{13}C. The HMQC transfer delay, τ_b, was set to $1/(2J_{CH})$. Quadrature detection was achieved using echo/ anti-echo gradient selection in the ^{15}N dimension and States-TPPI for the amide ^{1}H and ^{13}C dimensions. E/A selection was achieved by modulating the sign of g_1, for the ^{15}N dimension. States-TPPI was achieved by modulating the phase of the pulses prior to the t_2 and t_3 evolution delays. Artifact suppression was achieved through a combination of phase cycling and gradient pulses. The relative gradient strengths used were; g_1:80, g_2:47.1, g_3:50, g_4:5, g_5:9, g_6:39. Water suppression was achieved using presaturation during the interscan delay, D1 and z-filter gradient during the NOESY mixing delay.

The new magnetization pathway is as follows:

$$^{1}H \xrightarrow{^{1}J_{NH}} {}^{15}N(t_1) \xrightarrow{^{1}J_{NH}} {}^{1}H_{NH}(t_2) \xrightarrow{NOE} {}^{1}H \xrightarrow{^{1}J_{CH}} {}^{13}C(t_3) \xrightarrow{^{1}J_{CH}} {}^{1}H_{CH}(t_4)$$

Unlike the traditional pulse sequence, the N and NH dimensions are evolved prior to carbon evolution. This allows for radial sampling of the NH plane, using the sampling angles that were determined from the 3D experiment. The ^{13}C dimension is then sampled using Cartesian sampling scheme. Radial sampling of the amide ^{1}H and ^{15}N allows for a large number of increments to be collected in these dimensions. To best utilize the potential for high resolution these two dimensions are evolved using a HSQC[15]. Alternatively, a HMQC[70] is used in the carbon dimension. This dimension is Cartesian sampled, and the line width of the peaks in this dimension will most likely be dictated by the apodization function. Further, the beneficial relaxation parameters and ease of calibration make the HMQC appealing. Quadrature selection is achieved in the ^{15}N dimension using a echo/anti-echo gradient selection[56] scheme and using States-TPPI[57] in the other two indirect dimensions. Presataturation[4] water suppression is utilized during the inter-scan delay and durint the NOESY mixing time. The selectivity of presaturation minimizes the saturation of Ca attached protons. Additional water suppression is performed through application of Z filter[4] during the NOESY mixing delay.

Compared to the traditional pulse sequence this acquisition scheme has two main advantages: First, resolution in the proton dimensions is optimized and secondly, the carbon dimension is amenable to folding of signals. Sampling the aliphatic ^{1}H dimension directly, decreases the line widths of the aliphatic protons and potentially resolves degeneracy. Additionally, a much smaller indirect proton sweep width is needed to be sample the ^{1}HN dimension. To take advantage of the decreased sweep width the proton carrier is shifted to the center of the amide ^{1}H chemical shifts and returned to water for detection of the aliphatic protons.

On spectrometers that reverse the sign of folded peaks, it is not feasible to narrow the sweep width in carbon, folding a large fraction of the peaks, when radial sampling is applied. If the carbon dimension, with opposite sign folded peaks, was evolved prior to the radial sampled dimension, the opposite signs of peaks could cause peaks to cancel if the ridges overlapped. In this pulse sequence the carbon dimension is evolved subsequent to the radial sampled dimension. This allows for optimization of the ^{13}C sweep width.

7.2.3 Spectrum Analysis

Application of radial sampling to 4D experiments has the potential to dramatically increase the resolution of the experiment but this comes at the expense of increasing matrix sizes. For example, if a radial sampled 4D data set required to 1024 points in the directly detected dimension and 256 points in each of the 3 indirect dimensions, to digitally optimized the resolution, the resulting matrix would be 68 gb. Additionally, if the equivalent sized matrix was generated for each sampling angle component another two-fold increase in storage per angle is necessary. Even if data storage isn't limiting, there is a substantial time requirement to process these matrices. To alleviate the time and storage requirement a directed processing scheme is presented here. This processing scheme exploits the fact that while a 4d matrix contains all of the potential correlation space, only a small fraction of the spectrum actually contains useful information. The rest of the spectrum is noise and doesn't need to be processed.

To reduce the spectrum size only regions of interest are processed. This is accomplished by using an amide ^{1}H ^{15}N peak list as a guide and only processing regions of the spectrum where there is information. A unique spectrum is generated for each amide HSQC region. There is no prior information on the ^{13}C or aliphatic ^{1}H dimensions so the entire sweep width ranges are processed using traditional fast Fourier transform methodology. The ^{15}N and ^{1}H N, radial sampled, dimensions are processed using a direct multidimensional Fourier transform. Using the direct multidimensional Fourier transform the frequency range of the two dimensions can be explicitly determined. Note, care should be taken to process the sub-4D spectrum with high enough resolution to account for the inherent resolution of the data[49].

Two sub-4D spectra are generated for each amide residue, one using the additive back-projection (ABP) method and the other using a lower magnitude (LM) spectrum[25]. Both spectra are used simultaneously to analyze the presence of cross peaks. Analysis is simplified by using the 3D ^{15}N edited spectrum as a guide, where the sub-4D matrix is compared to the appropriate plane of the 3D in order to resolve the degeneracy in the 3d experiment. In many cases it is best to use the ABP spectra as the direct comparison and then use the LM spectrum in order to test for authenticity of a peak. Angles are selected to assure unique resolution of a peak, therefore, the LM spectrum will not contain any false positive peaks. However, if the sensitivity is limiting care should be included to avoid false negatives. False negatives arise when a peak falls into the noise in one angle spectra and is removed by the LM comparison. The peak chemical shift can be cross referenced with the assignment peak list to further assess the authenticity.

The ABP and LM spectra are useful for analyzing the spectrum in terms of peak chemical shift, but they are not suitable for integration. In the case of the ABP spectrum the integral of a peak is only relevant if a peak is resolve in all of the component angle spectra. If the peak is resolved in all of the angle spectra then it should be filtered prior to integration (zhou ref). Even if a peak is resolved in all of the component angle spectra, the lm spectra are not suitable for integration. The lower magnitude algorithm functions by comparing all of the component spectra, on a point to point basis, and selecting for lowest intensity at each point. When the comparison is performed on a peak, the intensity values selected are those with the largest deviation from the mean intensity. Selecting for intensities with the largest deviation decreases the accuracy of the integral value.

The component angle spectra, that resolve a peak, should be used to accurately integrate the peaks. The angle spectra that peaks are resolved in is determined during angle selection. Often multiple angles resolve a given peak. This allows for multiple, redundant, volume measurements to be extracted from the data. To determine the volume Gaussians are fit to the two Cartesian sampled dimension of the peak. Subsequently, a Gaussian is fit to a vector perpendicular to the ridge, in the radial sampled plane, intersecting the peaks chemical shift. To avoid interpolating points the perpendicular vector is generated using the direct 2D-FT[45-47]. The vectors coordinates are generated using the same equations that describe the ridge. The Gaussian fit, to the radial sampled dimensions, is scaled to avoid error when the volume of the peak is determined. This accounts for the different, angle dependent, sweep widths of the vector. Once the individual fit parameters are determined for each of the three domains the standard equation for the area of a Gaussian is utilized to determine the volume.

$$V = A\prod b_n\sqrt{2\pi}$$

Where A is the amplitude of the peak and b is the line width of dimension n. The redundant volumes, for each peak, can be treated with typical statistical analysis.

Determining the peak volumes using the angle spectra is easily automated. After the chemical shifts are determined from the ABP and LV spectra, the peak list and the resolution information is used as input for the fitting routine.

7.3 Results

To demonstrate the application presented here the methodology was applied to 1mM ^{13}C, ^{15}N labeled ubiquitin. The sample was prepared in the same manner as Chapter 6. First a traditional 3D ^{15}N edited NOESY was collected. The spectrum was processed and analyzed to generate a peak list for angle selection. Using our angle selection routine we concluded that 5 angles would resolve 98% of the peaks in the spectrum in at least one angle. These five angles were applied to collect five (4,3) radial sampling experiments using the pulse sequence above. When setting the parameters for the pulse sequence an attempt was made to balance sensitivity and resolution. Sufficient sensitivity is required for application of the lower magnitude comparison during data processing. Without suitable sensitivity authentic peaks can be removed during the magnitude comparison. To this end, 8 transients were averaged per FID. 9216 fids were collected, 48 and 192 total points in ^{13}C and the radial sampled dimension, respectively. Data acquisition required approximately 5 days. Acquisition time could be decreased further by biasing angle selection, collecting only data with objectives of resolving degenerate aliphatic protons, as determined by the ^{15}N edited NOESY.

Post acquisition, each angle spectra was processed separately. The Cartesian sampled dimensions, aliphatic ^{1}H and ^{13}C, were processed traditionally using the fast Fourier transform. A previously collected ^{15}N HSQC was used as a guide to determine the sub spectrum regions to process the amide ^{1}H, ^{15}N planes. Sub 4D spectra were generated for each amide group separating the positive and negative ridge components into separate spectra. In this application, the sub 4D spectra used 16 points per dimension of the ^{1}H-^{15}N planes with frequency range of .25 ppm and 1 ppm for ^{1}H and ^{15}N, respectively. The sweep width ranges were centered at the amide groups chemical shifts. The sweep width and number of points were selected to account for the intrinsic resolution of the spectrum. Each group of sub 4D angle spectra was then used to generate a lower magnitude spectrum and a summation spectrum.

An example of a sub 4D spectra compared to the equivalent plane of a ^{15}N edited NOESY spectra is shown in figure 7.3.

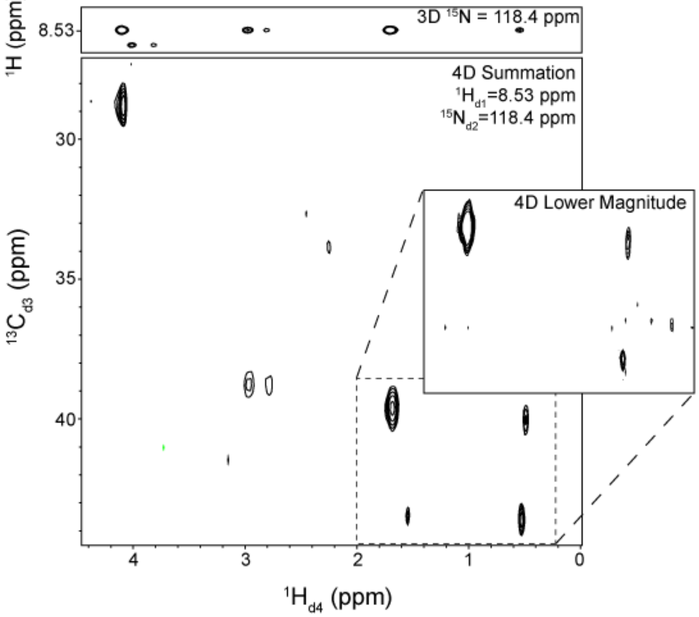


Figure 7.3 An example of using the 4D radial sampled ^{13}C, ^{15}N edited NOESY pulse sequence to resolve the degeneracy present in a 3D ^{15}N edited NOESY spectrum. A reference 1H-1H region of 3D 15N NOESY experiment is shown on the top with the Nitrogen shift fixed. The equivalent plane of the 4D ABP spectrum is shown below. The amide 1H and 15N are fixed at the same plane, the comparison regions of the aliphatic 1H and 13C region are shown. All expected peaks are shown. Potential artifact peaks, defined inside of the dashed line box, are resolved by comparison with the LV spectrum. Three of the peaks are present in the LV spectrum indicating that the fourth peak is a potential artifact.

This example demonstrates how the 4D spectrum is easily analyzed by comparing it to the 3D experiment. Here a ^{1}H- ^{1}H plane, of the 3D spectrum, is shown in panel a. There are 5 peaks resolved in the aliphatic proton dimension. By comparison of the 3D with the

ABP 4D sub spectrum the carbon chemical shifts are directly read from the ABP spectrum for the three downfield peaks. The two upfield peaks, selected inside the box, require comparison to the lower magnitude spectrum to determine which of the peaks are authentic. There are two peaks resolved in carbon for each of the proton shifts. Comparison with the lower magnitude, inset spectrum b., shows that three of the four peaks are authentic. Furhter, to assure that the proposed artifact peak is not a false negative, in the LM spectrum, the peaks chemical shifts are compare to the assignment list. Comparison with aliphatic proton assignments further confirms that three of the four peaks are authentic.

After resolving all of the chemical shifts, each peak was integrated. A Gaussian was first fit to the peak in the aliphatic ^{1}H and the ^{13}C dimensions. Then using the chemical shifts a vector of data was extracted from the amide ^{1}H, ^{15}N dimension for fitting of a Gaussian to the ridge. An example of the extracted vector is shown in figure 7.4a, for the negative sloped ridge component spectra. Here the dashed line extends perpendicular to the authentic peak chemical shifts. A vector is generated for each component angle spectra that resolves the peak. Typical fits of a Gaussian to the angle vector are shown in Figure 7.4b. The fit parameters for each peak in the component angle spectra are then used to determine the volume components for each peak.

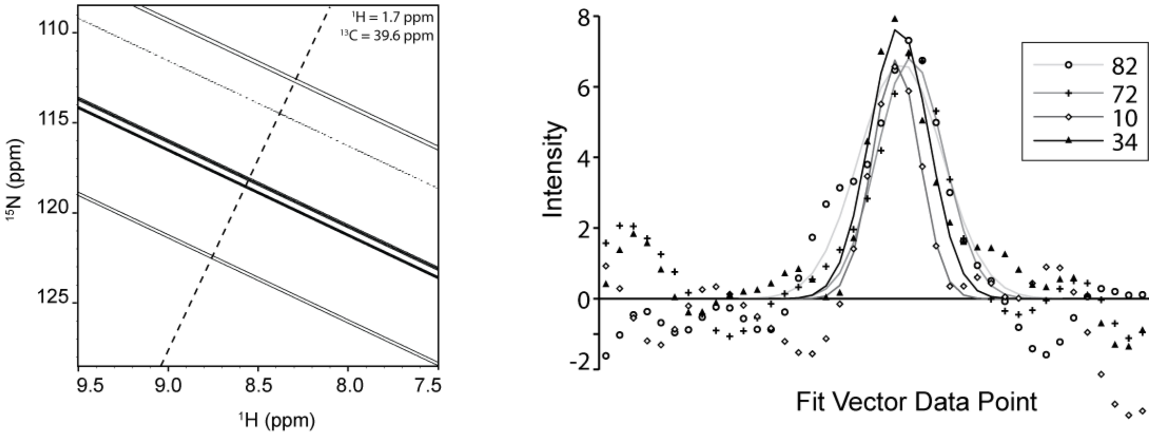


Figure 7.4 An example of extracting vectors from the individual component angle planes is shown here. The entire sweep width range of amide 1H and 15N dimensions are shown on the left. The peak of interested is located at the intersection of the dashed line and the most intense ridge component. The vector proximal to the peak chemical shift is generated and plotted for each of negative sloped ridge component on the right. A Gaussian is fit to the individual components.

Fitting the peaks separately assures that only resolved components contribute to the integral value. Additionally, this method leads to multiple redundant measurements which are amenable to statistical analysis. A simple average of the component volumes is compared to the intensity values from the traditional 3d spectrum in Figure 7.5.

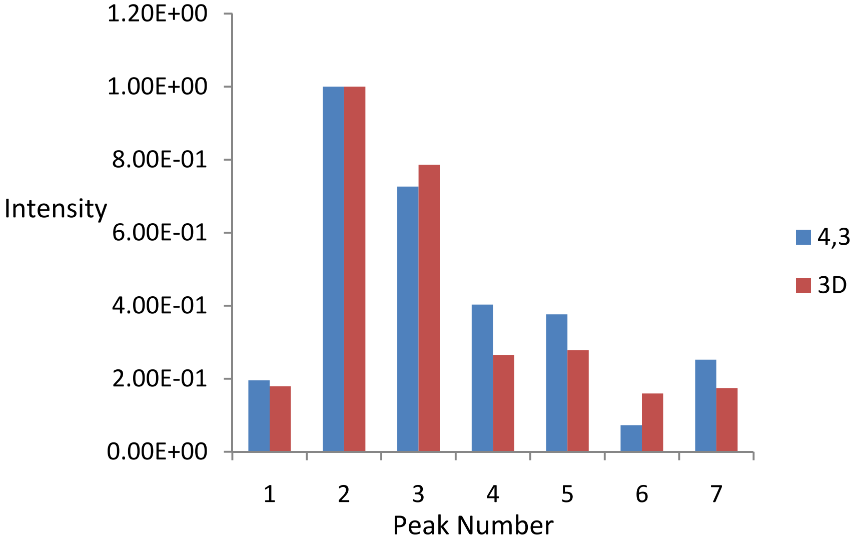


Figure 7.5 A comparison of normalized peak intensities from both the traditional 3D experiment and the average fit intensity from the 4D experiment for residue E64. The volumes from the 4D experiment are shown in blue, those of the 3D are in red. The peak volumes for both experiments were normalized to the most intense peak.

7.4 Conclusion

Four dimensional experiments are becoming commonplace as analysis of large proteins with solution state NMR increases. Expanding traditional 3D experiments, especially in the case of NOESY, resolves much of the degeneracy present in the lower dimensional spectrum. Ideally, it is desirable to increase the dimensionality of the experiment without concomitantly increasing the acquisition time. The methodology presented here expands existing information from a 3D experiment to 4D while minimizing additional acquisition time.

In order to develop this scheme the traditional 4D ^{15}N, ^{13}C pulse sequence was modified in order to optimize angle selection. Modifying the pulse sequence has the advantage of optimizing the resolution of the experiment by collecting the aliphatic proton dimension directly as compared to collecting the amide proton dimension with direct acquisition. Although the resolution is optimized, the sensitivity of the experiment is decreased, in comparison with the traditional pulse sequence. This arises from eliminating the PEP[55] to transfer the evolution from 15N back to ^{1}H for detection. In many cases the disparity in sensitivity is offset by the increasing sensitivity of cryogenically cooled preamplifiers and probes. Additionally, if the appropriate criteria are met, the sensitivity disparity can be more than accounted for using the SEnD methodology presented in Chapter 6. A trade between resolution and sensitivity is necessary. Application of optimized radial sampling affords an increase in resolution, and in many cases, a higher resolution spectrum can be generated, compared to the Cartesian sampled analog, while still collecting enough transients to offset the sensitivity decrease.

Further, the reversed acquisition order presented here could also be applied to the 3D ^{15}N or 3D ^{13}C NOESY[69] experiments. Reversing the acquisition order a higher resolution experiment, optimized for radial sampling can be collected. By radial sampling the equivalent of the hsqc dimension of these experiments iterative angle selection could be employed and generating the final spectrum would not be complicated by the dynamic range of the auto peaks and NOE cross peaks.

Chapter 8

Conclusion

The primary of objective of this thesis was to extend the capabilities of NMR to analyze large proteins. Though untested for all aspects of biomolecular NMR, this goal was largely accomplished. To accomplish the objective, technology was developed, a sensitivity gain achieved and a novel application presented.

A direct two-dimensional Fourier transformed based processing program, Al NMR, was presented in Chapter 3 to allow for general application of sparse sampling. This program includes all of the necessary functionality to process both Cartesian and sparse sampled NMR data, including reading and writing the appropriate file types and, most importantly, all of the necessary processing functions. Further, the program contains a graphical interface to real-time phase correct spectra. This program will be distributed and should serve as an integral tool to the field. Currently no other NMR data processing package contains the equivalent functionality.

The second step in generalizing sparse sampling was to develop means to phase correct the spectra. The limitations of sparse sampling was immediately apparent without ability to phase correct spectra. This is especially true when radial sampling is processed using a lower value comparison. As a result of phase error, the lower value comparison often removes authentic peaks. To circumvent this, the relevant theory was analyzed and two novel phase correct routines were presented. The first demonstrates means to correct the spectrum retrospective of processing by isolating absorptive and dispersive components and solving for linear combinations of the two components with absorptive properties. The second method exploits the flexibility of the 2D-FT, and applies a phase correction in the time domain. Both of these methods are available in Al NMR.

With capabilities to process and phase correct radial sampled data in hand, my efforts turned to optimizing data acquisition. This was accomplished by developing the methodology to minimize the number of sampling angles acquired. The necessary criteria to define a minimum number of sampling angles was defined for two cases; first, when all of the chemical shifts are known and second, if the resonance frequencies are not known. Both cases were successfully tested both computationally and experimentally. This methodology will also be available to the public via the Al NMR distribution. As an aside, the angle set selection was further optimized as a collaboration between myself and another graduate student. This work will be presented in a forthcoming issue of Journal of Biomolecular NMR by Gledhill, Walters and Wand.

Optimization of the radial sampling angle set has led to further optimization of sensitivity parameters associated with collection of radial sampled data. In Chapter 6, Sensitivity Enhanced n-Dimensional NMR (SEnD) is presented. Here, all of the relevant theory and criteria is presented to achieve upwards of a three-fold S/N advantage over the equivalent Cartesian sampled experiment. As demonstrated, this methodology will prove particularly import under sensitivity limiting conditions.

Finally, with all of the technology in place, an example is presented to demonstrate the advantages of radial sampling. The example, Chapter 7, a novel method to collect a 4D ^{13}C, ^{15}N edited NOESY is presented. This study demonstrates the distinct advantage of radial sampling to speed acquisition allowing for collection of a high resolution, high dimensional spectrum. Preliminary evidence is shown to demonstrate the functionality of this method using ubiquitin as a test case.

The sum of the technology presented here provides the necessary foundation for general application of sparse sampling, especially to large proteins. Optimization of resolution and sensitivity parameters is now available. Therefore, the limitations of the sensitivity and sampling limited regimes are reduced. Further, the example presented here is general and should serve as a foundation for a broad array of technology yet to be uncovered through application of sparse sampling.

Finally, with regard to resolution and sensitivity, the methodology presented here can be used to achieve a substantial advantage over Cartesian sampling. The combined effect of increased resolution and sensitivity will potentially allow for more accurate: structure calculation, by resolving more restraints, and measurement of biophysical properties of large proteins. The advantages of optimized radial sampling are dependent upon a large number of parameters, such as sample concentration, protein size and spectrum resolution. Future work will serve to better define the advantages demonstrated here.

Appendices

Appendix 1.

Al NMR Data File Object Properties

Data File			
	Data Object Initialization		
		`data_object=alnmr.readbruker(‘data_directory’)`	
		`data_object=alnmr.readfelix(‘fid_directory’)`	
			These commands initialize a NMR data directory for reading the fids. In the case of readbruker, the experiment number directory is passed and for readfelix the experiment .fid directory is passed. A data object is created from the data file by reading the associated parameter files within the passed data directory.
	Data Object Parameters		
		`data_object.currentfid`	
			Current fid number location in file. This is the next fid that will be read. When the file is initialized this parameter is set to 1.
		`data_object.dname`	
			Directory path of the ser/fid file that is currently being read
		`data_object.td[]`	
			List containing the total number of data points in each acquisition dimension. The list is ordered such that td[0], td[1],... are the first and second dimension points respectively.
		`data_object.fnmode[]`	
			List containing the quadrature mode for each dimension. The default Bruker numbering is used; 0-6: undefined, QF, QSEQ, TPPI, States, States-TPPI, Echo-Antiecho. This parameter is only available for Bruker Data.
		`data_object.angle`	
			Sampling angle from a radial sampled experiment. Assumes that the default Bruker parameter constant 51 is used to set the sampling angle.

		`data_object.numfids`	
			The total number of fids collected in the experiment.
		`data_object.filefidsize`	
			The fid size in the file, defined in number of bytes. In some cases the fid in the file contains trailing zeros which make the expected fid size larger.
	Data Object Commands		
		`fid=data_object.readfid([fidnum=n, byteswap=True/False, resize=True/False])`	
			Read one fid from the file. If no options are supplied, defaults are substituted. fidnum=n; where n is the desired fid to read. The default fidnum is the next fid in the file. byteswap=True/False; default = False. Converts endianess of the fid when reading. resize=True/False; default=True. Resizes the fid when reading to remove any trailing zeros.
		`data_object.movetofid(fid number)`	
			Moves the file pointer to the beginning of the specified fid number. A subsequent readfid issue will read the designated fid.
		`data_object.close()`	
			Closes the fid or ser file and removes the data_object

Appendix 2.

Al NMR Matrix File Object Properties

Matrix File			
	Matrix Object Initialization		
		matrix_object=Al.SparkyMat(‘filename’,[d1,d2,d3,...,dn])	
		matrix_object=Al.FelixMat(‘filename’, [d1,d2,d3,...,dn])	
			Initialize the matrix object while creating or opening the matrix file. If no options are supplied then it is assumed that the matrix file exists. The file is opened and all parameters are intialized from the file. If optional dimensions, d1,d2...dn, are supplied then a new matrix file is generated using the supplied matrix dimensions.
	Matrix Object Parameters		
		matrix_object.blockdim[n]	
			List containing the number of points in one block of the matrix. The list is ordered corresponding to the dimensions where the zero element is the first dimension.
		matrix_object.matdim	
			List containg the matrix dimensions. The list order is the same as blockdim.
		matrix_object.bsize	
			Total number of points in one block of the matrix.
		matrix_object.filename	
			The filename of the corresponding matrix file.
		matrix_object.dim	
			Dimension of the matrix.
		matrix_object.nblocks[n]	
			List containing the total number of blocks in each dimension. The list order is the same as blockdim.
		matrix_object.bstride[n]	
			List containing the number of bytes that need to be skipped in the file to move one block for each of the dimensions.
		matrix_object.pstride[n]	

			List containing the number of bytes that need to be skipped in the file to move one point for each of the dimensions.
		`matrix_object.curblocks[]`	
			Transient list that is used to define a list of blocks that are read per data vector.
		`matrix_object.blockinmem[]`	
			List of the current data blocks being stored in memory.
		`matrix_object.newread`	
			Bool flag that specifies if a new set of blocks need to be read for a different data vector or point.
		`matrix_object.datablocks[n]`	
			Storage list for all of the blocks being held in memory. This is where the actual data is transiently stored. The order of this list corresponds to the blockinmem list.
	Matrix Object Functions		
		`matrix_object.write(data, c1, c2, ..., cn)`	
			Write data to the matrix file. data is either a data point or vector. If a data point is supplied, the matrix coordinate points c1,c2,... are supplied as integers. Where the first point coordinate is 1. If a data vector is supplied, then the coordinate that the vector spans across is set to 0. data is always supplied as a real, not complex, floats. Complex data should be converted to interleaved real data prior to writing. Errors will occurs if more than one dimension is set to 0. Or if a data vector greater than the matrix size is supplied.
		`data=matrix_object.read(c1, c2, ... , cn)`	
			Read a point or vector from a matrix file. Coordinate points, c1,c2,...,cn are define in the same manner as write. If all coordinates points are nonzero then a data point is return. If one dimension of the coordinate points is set to zero then a real data vector that spans the matrix size is returned.
		`matrix_object.update()`	

			Commit changes to the matrix file from the blocks that are stored in memory.
		`matrix_object.close()`	
			Close the matrix file and remove the matrix_object.
	Matrix Operation Functions		
		`alnmr.lv(input matrix file 1, input matrix file 2, output matrix file)`	
			A lower value (magnitude) comparsion is performed between each set of corresponding elements in the two input matrix files. The results are written in the third file name passed. All three files must exist prior to usage.
		`alnmr.matadd(input matrix file 1, input matrix file 2, output matrix file)`	
			Each corresponding set of elements in the two input matrix files are summed. The results are written to the output matrix file.
		`alnmr.refsparky(matrix file name, reference dimension, Nucleus frequency (Mhz), sweep width (hz), carrier (ppm), title)`	
		`alnmr.reffelix(matrix file name, reference dimension, Nucleus frequency (Mhz), sweep width (hz), center point chemical shift (ppm), title)`	
			These commands reference either a sparky or felix matrix file passed to the command. The reference dimension parameter is the matrix dimension that will be acted upon, 1 is the first dimension. The nucleus frequency of the dimension is passed to the command in Mhz and the sweep width is passed in hz. The spectrum is referenced to the center point of the spectrum. This chemical shift is passed in ppm. title is the name of the nucleus: '1H', '13C', '15N', etc. This variable is a string. The command is repeated seperately for each dimension that is to be referenced.

Appendix 3.

Al NMR Matrix File Object Properties

Data Processing Functions			
	Data Operation Functions		
		`data=alnmr.add(fid1, fid2)`	
			Add two fids on an element basis. The function excepts either complex or real data. The corresponding type is returned. The two fids must contain the same number of elements.
		`data=alnmr.sub(fid1, fid2)`	
			Subtract fid2 from fid1 on an element basis. As in the add function, the type returned is the same as the type of fid1 and fid2. The fids must contain the same number of elements.
		`data=alnmr.interleave(fid1, fid2)`	
			Interleave the elements of fid2 between the elements of fid1. It is assumed that each fids data type is real (not complex). The two fids must contain the same number of elements. This function is typically used to combine real and imaginary components that are separated. Following interleaving the data, the data is converted to complex using the complexdata function.
		`data=alnmr.complexdata(fid1)`	
			Convert interleaved, real and imaginary, data to complex data. The first, and every other element, are the real components while the second, and others, are the imaginary components of the data vector returned.
		`data=alnmr.reduce (fid1)`	
			Reduce a complex fid to just a real fid. The imaginary component is discarded.
		`data=alnmr.conjugate (fid1)`	
			Take the complex conjugate of each element in the fid. This negates the imaginary component of the complex data.
		`data=alnmr.exchange(fid1)`	

			Exchange the real and imaginary component of a complex fid.
		`data=alnmr.reverse(data)`	
			Reverse the order to the data vector that is passed to the function. The data that is passed can be either real or complex.
		`data=alnmr.delete(fid1, first point, last point)`	
			Delete a selection of points from the fid data that is passed to the function. First point is inclusive the last point is exclusive. Fid element numbering starts at 0. For example if one desired to delete the first four points of a fid the first point is set to 0 and the last point is set to 4 (which is actually the fifth element).
		`data=alnmr.lowervalue(fid1,fid2)`	
			The lower value (magnitude) comparison is performed between the corresponding elements of the two fids that are passed to the function. It is assumed that the two fids are real, not complex. If a complex fid is passed only the real component is used in the magnitude comparison. The number of elements in the two fids passed are assumed to be equal.
		`data=alnmr.zerofill(fid1, output fid size)`	
			Append the appropriate number of zeros to the fid to make the resulting fid the specified size. The returned data type is equivalent to the type passed.
	Apodization Functions		
		`data=alnmr.ss1d(data,shift)`	
			Apply a shifted sinebell (sine-squared) apodization to the 1d data supplied. The shift is supplied in degrees.
		`data=alnmr.hann1d(data)`	
			Hann window function applied to 1d data.
		`data=alnmr.hamming1d(data)`	
			Hamming window function applied to 1d data.
		`data=alnmr.gauss1d(data,width)`	
			Gaussian window function. Width is relative to the number of elements in the 1d data that is passed. The value of width should be set less than .5.
		`data=alnmr.em(data,line broadening (hz))`	

			Multiply the 1d data by an exponential decay to achieve line broadening by the specified hz. value.
		data=alnmr.ss2d(data,angle, shift1, shift2)	
			Shifted sinebell apodization on radial sampled data. The data is assumed to contain four quadrature components per increment which are listed sequentially in the data vector passed to the function. Angle is the sampling angle the data was sampled at. Shift1 and shift2 are the sinebell shifts for each of the indirect dimensions.
		data=alnmr.ss2dgen(data,time points, ni1max, ni2max, sw1, sw2, shift1[=90], shift2[=90])	
			Shifted sinebell apodization for random sampled data. The function assumes that the data contains four quadrature components per increment, which are listed sequentially in the data vector passed to the function. The function generates a 2d decaying apodization function using the shifts supplied from 0 to the maximum incremented delays. The time points of the sampling scheme are passed to the function in order to determine where on the apodization function surface the point is located.
	Fourier Transform Functions		
		data=alnmr.fft(data)	
			Fast Fourier transform. The function assumes that the passed data is complex. The data vector returned is in decreasing frequency order (sw/2 to -sw/2).
		data=alnmr.ft1d(data, time point list, frequency list, zero order phase correction,first order phase correction)	
			Discrete Fourier transform. The data passed to the function is complex. The time points are defined as a list corresponding to the points which the data was collected at. The frequency list are the frequency the Fourier transform intensities are determined at. A zero and first order phase correct can be applied during the Fourier transform. The zero order phase correction is supplied in degrees. The first order phase correction is supplied as a time. Typically the time correction is a fraction of one increment.
		data=alnmr.ft2d(data, time, freq1, freq2, [ph0a, ph1a,	

		`ph0b, ph1b])`	
			Discrete direct two dimensional Fourier transform. The data passed to the function is supplied as a real list with four components, one for each of the four quadrature components, per sampling point. The time points supplied correspond to the sampling times for the two indirect dimensions. The two times are list sequentially in the time point list. The two frequency lists, one for each of the indirect dimensions, are the values the Fourier transform intensities are solved at. The range of the list should be determined with respect to the sweep widths used, but can be any range of interest. The zero and first order phase corrections are supplied separately for the two dimensions. The zero order phase corrections are supplied in degrees. The first order phase corrections are supplied as time values. A two dimensional matrix is returned, where the first and second dimensions are the freq1 and freq2 values respectively.
		`data=alnmr.ft2dsep(component, data, time, freq1, freq2, [ph0a, ph1a, ph0b, ph1b])`	
			Discrete direct two dimensional Fourier transform that separates the positive and negative ridge component spectra when radial sampling is used. When the component variable is set to 1 the positive ridge component 2d matrix is returned. When set to 2 the negative ridge component spectrum is returned. When set to 3 both components are returned. In this case two variables are returned (data1, data2=Al.ft2dsep(3...))
		`data=alnmr.hilbert(data)`	
			Hilbert transform. This function generates an imaginary component from a real data vector. A complex vector is returned.
	Frequency List generators		
		`data=alnmr.ftfreq(np,sw)`	
			This function generates a list of frequencies given the number of sampling points collected and the sw. The order of the list is in decreasing order (-sw/2 to sw/2). The list are the reference values for the fft function.
	Phase Correction		
		`data=alnmr.phase(data, zero order correction, first order`	

		`correction [, pivot])`	
			Phase correct a complex supplied complex data vector. The zero and first order corrections are both supplied in degrees. The pivot value is optional, by default it is set at the center point of the spectrum.
		`alnmr.interactivephase(data)`	
			This function starts the interactive phase correction interface. The data passed is complex.
		`data=alnmr.phase2d(data, d1 zero order, d1 first order, d2 zero order, d2 first order)`	
			This function phase corrects radial sampled spectra.
	Sampling generator		
		`timepoints=alnmr.maketime (sw,ni, [angle1, sw2, angle2, sw3])`	
			This function generates a time point list for either the discrete 1d Fourier transform or the discrete direct two dimenional Fourier transform, when radial sampled data is used. When a frequency list is desired for a 1d data set the sweep with and number of complex increments are supplied. This returns a list with one time point per data point. When used for radial sampling angle1 and sw2 are also supplied. This generates a time point list with two elements per number of points. The first element is the sampling point for the first indirect dimension and the second element is the time point for the second indirect dimension. The elements are calculated from t1=(n/sw1)cos(a) and t2=(n/sw2)sin(a).
	Digital Water Suppression Functions		
		`data=alnmr.conv(data,window size, convolution filter function)`	
			Convolution water suppression. window size is the number of data points surrounding the current point that are averaged and subtracted from the current point. convolution filter is the weighting of the convolution data points
		`data=alnmr.polysub(data, polynomial order, sweep width)`	

			This function fits a polynomial to every data point in a FID and subtracts the resulting polynomial from the fid to suppress water signal.
	Linear Prediction Functions		
		`data=alnmr.lpinv(data, coefficients, number of predicted points)`	
			Extend the current data through linear prediction. The number of coefficients is supplied, simple matrix inversion is used to solve for the coefficients. The coefficients are used to generate the number of predicted points.
		`data=alnmr.lpsvd (data, coefficients, reduced order parameter, number of predicted points)`	
			This function also performs linear prediction, but uses singular value decomposition to solve for the coefficients.
		`data=alnmr.lpsvdrad (data, coefficients, reduced order parameter, number of predicted points)`	
			This functions performs linear prediction on radial sampled data using singular value decomposition to determine the coefficients of each angle.
		`data=alnmr.averagelp (data, coefficients, reduced order parameter, number of predicted points)`	
			This functions performs linear prediction on radial sampled data. The details will be presented in the forthcoming Gledhill, Kasinath and Wand to be submitted to Journal of Magnetic Resonance.

Appendix 4.

2D Gradient Selected, Sensitivity Enhanced, ^{15}N HSQC Processing Script

```
#import necessary packages
import alnmr
import os

#define directory paths
datapath=['C:\\', 'alnmr_process', 'data']
datadir=['example_hsqc_dir']

matrixpath=['C:\\', 'alnmr_process', 'matrix']
matrixdir=['example_hsqc_dir']
expnum=[102]

#define output matrix name and dimensions
matrixname='example_hsqc.ucsf'

d1=2048
d2=512


#reference information
frequency=[749.613, 75.9662]
sw=[14.0,24.0]
carrier=[4.538, 117.0]
nucleus=['1H','15N']

#phase corrections
d1phase=[50.2, 0, 0]

#generate data path using os.path
datdir=''
for p in datapath:
    datdir=os.path.join(datdir,p)
for p in datadir:
    datdir=os.path.join(datdir,p)
datname=os.path.join(datdir,str(expnum[0]))

#generate matrix path and define name
outdir=''
for p in matrixpath:
    outdir=os.path.join(outdir,p)
for p in matrixdir:
    outdir=os.path.join(outdir,p)
name=os.path.join(outdir,matrixname)
```

```
#initialize nmr data object
bdat=alnmr.readbruker(datname)

#initialize matrix object and build matrix
smat=alnmr.SparkyFile(name,d1,d2)


#process the directly detected dimension
print 'process d1'

for a in range(bdat.td2/2):
    r1=bdat.readfid()
    r2=bdat.readfid()

    se1=alnmr.add(r1,r2)
    se2=alnmr.sub(r2,r1)

    se2=alnmr.exchange(se2)
    se2=alnmr.conjugate(se2)

    se1=alnmr.polysub(se1,5,sw[0]*frequency[0])
    se2=alnmr.polysub(se2,5,sw[0]*frequency[0])

    se1=alnmr.ss1d(se1,90)
    se2=alnmr.ss1d(se2,90)

    se1=alnmr.zerofill(se1,d1*2)
    se2=alnmr.zerofill(se2,d1*2)

    se1=alnmr.fft(se1)
    se2=alnmr.fft(se2)

    se1=alnmr.phase(se1,phase[0], phase[1],phase[2])
    se2=alnmr.phase(se2,phase[0], phase[1],phase[2])

    se1=alnmr.delete(se1,d1,d1*2)
    se2=alnmr.delete(se2,d1,d1*2)

    se1=alnmr.reducecomplex(se1)
    se2=alnmr.reducecomplex(se2)

    smat.write(se1,0,(a*2+1))
    smat.write(se2,0,(a*2+2))
```

```
#commit changes to the matrix file
smat.update()


#process indirect dimension
print 'process d2'

for a in range(d1):
    a+=1
    indvec=smat.read(a,0)
    indvec=alnmr.delete(indvec,td2,len(indvec))
    indvec=alnmr.complexdata(indvec)


    #multiply every other point for states-tppi
    #indvec[1::2]*=-1

    indvec[0]=indvec[0]*.5

    indvec=alnmr.ss1d(indvec,85.0)
    indvec=alnmr.zerofill(indvec,d2)
    ft=alnmr.fft(indvec)
    smat.write(ft,a,0)

smat.close()
bdat.close()

alnmr.refsparky(name,1,frequency[0],(sw[0]*frequency[0])/2,
carrier[0]+(sw[0]/4),nucleus[0])
alnmr.refsparky(name,2,f2,(sw[1]*frequency[1]),
carrier[1],nucleus[1])
```

Appendix 5.
Phase Correction Macro

```
#import necessary packages
import alnmr
import os

vecpt=[0,52]

matrixpath=['C:\\', 'alnmr_process', 'matrix']
matrixdir=['example_hsqc_dir']

#define output matrix name and dimensions
matrixname='example_hsqc.ucsf'

#generate matrix path and define name
outdir=''
for p in matrixpath:
    outdir=os.path.join(outdir,p)
for p in matrixdir:
    outdir=os.path.join(outdir,p)
name=os.path.join(outdir,matrixname)

#initialize matrix object and build matrix
smat=alnmr.SparkyFile(name)

#read vector
vect=smat.read(vecpt[0],vecpt[1])

#hilbert transform
vect=alnmr.hilbert(vect)

#phase correction interface
alnmr.interactivephase(vect)

smat.close()
```

Appendix 6.

3,2 Radial Sampled Processing Script

This script is designed to process data that has been collected using gradient selection, and sensitivity enhancement in the Nitrogen dimension and States-TPPI selection in the Carbon Dimension

```
import alnmr
import os

datapath=['C:\\','alnmr_process', 'data']
matrixpath=['C:\\','alnmr_process','matrix']

datadir=['example_radial_3d']
matrixdir=['example_radial_3d']

prefix=['example_3drad_']

#experiment data directories and corresponding angles
expnum=[1004, 1005, 1006]
angles=[5, 10, 15]

d1phase=[49.2,0]

#output matrix size
d1=1024
d2=128
d3=128

#total number of radial sampled points
td3=128

#spectrum referencing information
sw1=11.9903
sw2=27.0
sw3=35.0

f1=498.81
f2=50.55
```

```
f3=125.43

#proton carrier (discard the downfield component)
c1=4.702+sw1/4
carrier=[str(c1),'118.0','54.0']
nucleus=['1H','15N','13C']

#set the paths for the data and matrix directories
datdir=''
for p in datapath:
    datdir=os.path.join(datdir,p)
for p in datadir:
    datdir=os.path.join(datdir,p)

matdir=''
for p in matrixpath:
    matdir=os.path.join(matdir,p)
for p in matrixdir:
    matdir=os.path.join(matdir,p)


#process the directly acquired dimension
for x in range(len(angles)):
    print angles[x]

    #build a temporary matrix to hold d1 processed data
    name=prefix[0]+str(expnum[x])+'_n'+str(angles[x])+'c.ucsf'
    name=os.path.join(datdir,name)
    smat=alnmr.SparkyFile(name,d1,d2)

    #create the data object for the angle data set
    datdirname=os.path.join(datdir,str(expnum[x]))
    bdat=alnmr.readbruker(datdirname)

    #process all of the FIDs
    for a in range(bdat.td3/4):
        r1=bdat.readfid()
        r2=bdat.readfid()
        r3=bdat.readfid()
        r4=bdat.readfid()

        #p/n selection
        se1=alnmr.add(r2,r1)
        se2=alnmr.sub(r2,r1)
        se3=alnmr.add(r3,r4)
        se4=alnmr.sub(r4,r3)
```

```
#conjugate and exchange
se2=alnmr.exchange(se2)
se2=alnmr.conjugate(se2)
se4=alnmr.exchange(se4)
se4=alnmr.conjugate(se4)


#convolution - polynomial subtraction
se1=alnmr.polysub(se1,5,sw1*f1)
se2=alnmr.polysub(se2,5,sw1*f1)
se3=alnmr.polysub(se3,5,sw1*f1)
se4=alnmr.polysub(se4,5,sw1*f1)

#apodization
se1=alnmr.ss1d(se1,90)
se2=alnmr.ss1d(se2,90)
se3=alnmr.ss1d(se3,90)
se4=alnmr.ss1d(se4,90)

#zerofill
se1=alnmr.zerofill(se1,d1*2)
se2=alnmr.zerofill(se2,d1*2)
se3=alnmr.zerofill(se3,d1*2)
se4=alnmr.zerofill(se4,d1*2)

#fourier transform
se1=alnmr.fft(se1)
se2=alnmr.fft(se2)
se3=alnmr.fft(se3)
se4=alnmr.fft(se4)

#phase
se1=alnmr.phase(se1,phased1[0],phased1[1])
se2=alnmr.phase(se2,phased1[0],phased1[1])
se3=alnmr.phase(se3,phased1[0],phased1[1])
se4=alnmr.phase(se4,phased1[0],phased1[1])

#delete
se1=alnmr.delete(se1,d1,d1*2)
se2=alnmr.delete(se2,d1,d1*2)
se3=alnmr.delete(se3,d1,d1*2)
se4=alnmr.delete(se4,d1,d1*2)

#reduce complex data
se1=alnmr.reducecomplex(se1)
se2=alnmr.reducecomplex(se2)
```

```
        se3=alnmr.reducecomplex(se3)
        se4=alnmr.reducecomplex(se4)

        #write to data matrix
        smat.write(se1,0,(a*4+1))
        smat.write(se2,0,(a*4+2))
        smat.write(se3,0,(a*4+3))
        smat.write(se4,0,(a*4+4))

    smat.close()
    bdat.close()


#process the radial sampled dimensions

#freq list for 2dft
freq1=alnmr.ftfreq(d2,sw2*f2) #nitrogen
freq2=alnmr.ftfreq(d3,sw3*f3) #carbon

#loop over all angles
for x in range(len(angles)):
    #sampling points list
    timepoints=alnmr.maketime(ni,sw2*f2,angle,sw3*f3)

    #build matrix objects for the angle components
    prname=os.path.join(outdir,prefix[0]+str(expnum[x])+
                    '_n'+str(angles[x])+'c_pr.ucsf')
    outmatpr=alnmr.SparkyFile(prname,d1,d2,d3)

    mrname=os.path.join(outdir,prefix[0]+str(expnum[x])+
                    '_n'+str(angles[x])+'c_mr.ucsf')
    outmatmr=alnmr.SparkyFile(mrname,d1,d2,d3)

    #data matrix directory
    inname=os.path.join(datdir,prefix[0]+str(expnum[x])+
                    '_n'+str(angles[x])+'c.ucsf')
    inmat=alnmr.SparkyFile(inname)

    #loop over all of the d1 points
    for d1p in range(d1):
        print d1p

        #read vector
        data=inmat.read(d1p+1,0)
        data=alnmr.delete(data,td3,len(data)) #delete zeros

        #multiply first point by 1/2
        data[0]=.5*data[0]
        data[1]=.5*data[1]
        data[2]=.5*data[2]
        data[3]=.5*data[3]

        #states tppi on bruker correction
        data[4::8]*=-1
        data[5::8]*=-1
        data[6::8]*=-1
        data[7::8]*=-1

        #apodize
        data=alnmr.ss2d(data,angle,90,90)

        ftp,ftm=alnmr.ft2dplus(3,data,timepoints,
                               freq1,freq2,0.0,0.0,0.0,0.0)

        for y in range(d2):
            outmatpr.write(ftp[y],d1p+1,y+1,0)
            outmatmr.write(ftm[y],d1p+1,y+1,0)


    #outmatmr.close()
    outmatpr.close()
    outmatmr.close()
    inmat.close()

#lower value comparison

#build lv matrix
print 'build'
lvname=os.path.join(outdir,prefix[0]+'lv_3d.ucsf')
lvmat=alnmr.SparkyFile(lvname,d1,d2,d3)
lvmat.close()

#compare components of the first angle
name1=os.path.join(outdir,prefix[0]+str(expnum[0])+
                   '_n'+str(angles[0])+'c_pr.ucsf')
name2=os.path.join(outdir,prefix[0]+str(expnum[0])+
                   '_n'+str(angles[0])+'c_mr.ucsf')

#lower value
alnmr.lv(name1,name2,lvname)

#compare all of the remaining angles
for x in range(len(angles)-1):
    name=os.path.join(outdir,prefix[0]+
                      str(expnum[x+1])+'_n'+
                      str(angles[x+1])+'c_pr.ucsf')
    alnmr.lv(lvname,name,lvname)
    name=os.path.join(outdir,prefix[0]+
                      str(expnum[x+1])+'_n'+
                      str(angles[x+1])+'c_mr.ucsf')
    alnmr.lv(lvname,name,lvname)


#reference all of the spectra
for x in range(len(angles)):
    name=os.path.join(outdir,prefix[0]+str(expnum[x])+
                      '_n'+str(angles[x])+'c_pr.ucsf')
    alnmr.refsparky(name,1,f1,(sw1*f1),float(carrier[0]),
                      nucleus[0])
    alnmr.refsparky(name,2,f2,(sw2*f2),float(carrier[1]),
                      nucleus[1])
    alnmr.refsparky(name,3,f3,(sw3*f3),float(carrier[2]),
                      nucleus[2])

    name=os.path.join(outdir,prefix[0]+str(expnum[x])+
                      '_n'+str(angles[x])+'c_mr.ucsf')
    alnmr.refsparky(name,1,f1,(sw1*f1),float(carrier[0]),
                      nucleus[0])
    alnmr.refsparky(name,2,f2,(sw2*f2),float(carrier[1]),
                      nucleus[1])
    alnmr.refsparky(name,3,f3,(sw3*f3),float(carrier[2]),
                      nucleus[2])

alnmr.refsparky(lvname,1,f1,(sw1*f1),float(carrier[0]),
                      nucleus[0])
alnmr.refsparky(lvname,2,f2,(sw2*f2),float(carrier[1]),
                      nucleus[1])
alnmr.refsparky(lvname,3,f3,(sw3*f3),float(carrier[2]),
                      nucleus[2])
```

References

[1] G.A. Petsko, D. Ringe, Protein structure and function, New Science Press ; Sinauer Associates ; Blackwell Publishing, London, Sunderland, MA, Oxford, 2004.

[2] M. Rami Reddy, M.D. Erion, Knovel (Firm), Free energy calculations in rational drug design, Kluwer Academic/Plenum Publishers, New York, 2001, pp. xxii, 384 p., [6] p. of plates.

[3] A.L. Parrill, M. Rami Reddy, American Chemical Society. Meeting, Rational drug design : novel methodology and practical applications, American Chemical Society : Distributed by Oxford University Press, Washington, DC, 1999.

[4] J. Cavanagh, Protein NMR spectroscopy : principles and practice, Academic Press, San Diego, 1996.

[5] S. Grzesiek, J. Anglister, H. Ren, A. Bax, C-13 Line Narrowing by H-2 Decoupling in H-2/C-13/N-15-Enriched Proteins - Application to Triple-Resonance 4d J-Connectivity of Sequential Amides. Journal of the American Chemical Society 115 (1993) 4369-4370.

[6] R.A. Venters, W.J. Metzler, L.D. Spicer, L. Mueller, B.T. Farmer, Use of H-1(N)-H-1(N) Noes to Determine Protein Global Folds in Perdeuterated Proteins. Journal of the American Chemical Society 117 (1995) 9592-9593.

[7] T. Yamazaki, W. Lee, M. Revington, D.L. Mattiello, F.W. Dahlquist, C.H. Arrowsmith, L.E. Kay, An Hnca Pulse Scheme for the Backbone Assignment of N-15,C-13,H-2-Labeled Proteins - Application to a 37-Kda Trp Repressor DNA Complex. Journal of the American Chemical Society 116 (1994) 6464-6465.

[8] K. Pervushin, R. Riek, G. Wider, K. Wuthrich, Attenuated T2 relaxation by mutual cancellation of dipole-dipole coupling and chemical shift anisotropy indicates an avenue to NMR structures of very large biological macromolecules in solution. Proc Natl Acad Sci U S A 94 (1997) 12366-71.

[9] A.J. Wand, M.R. Ehrhardt, P.F. Flynn, High-resolution NMR of encapsulated proteins dissolved in low-viscosity fluids. Proc Natl Acad Sci U S A 95 (1998) 15299-302.

[10] C. Fernandez, G. Wider, TROSY in NMR studies of the structure and function of large biological macromolecules. Curr Opin Struct Biol 13 (2003) 570-80.

[11] R. Riek, K. Pervushin, K. Wuthrich, TROSY and CRINEPT: NMR with large molecular and supramolecular structures in solution. Trends Biochem Sci 25 (2000) 462-8.

[12] J.M. Kielec, K.G. Valentine, C.R. Babu, A.J. Wand, Reverse micelles in integral membrane protein structural biology by solution NMR spectroscopy. Structure 17 (2009) 345-51.

[13] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The Protein Data Bank. Nucleic Acids Res 28 (2000) 235-42.

[14] D.S. Wishart, C. Knox, A.C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, M. Hassanali, DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Res 36 (2008) D901-6.

[15] J. Schleucher, M. Schwendinger, M. Sattler, P. Schmidt, O. Schedletzky, S.J. Glaser, O.W. Sorensen, C. Griesinger, A general enhancement scheme in heteronuclear multidimensional NMR employing pulsed field gradients. J Biomol NMR 4 (1994) 301-6.

[16] V.A. Mandelshtam, The multidimensional filter diagonalization method - I. Theory and numerical implementation. Journal of Magnetic Resonance 144 (2000) 343-356.

[17] S. Kim, T. Szyperski, GFT NMR, a new approach to rapidly obtain precise high-dimensional NMR spectral information. Journal of the American Chemical Society 125 (2003) 1385-1393.

[18] T. Szyperski, D.C. Yeh, D.K. Sukumaran, H.N.B. Moseley, G.T. Montelione, Reduced-dimensionality NMR spectroscopy for high-throughput protein resonance assignment. Proceedings of the National Academy of Sciences of the United States of America 99 (2002) 8009-8014.

[19] R.W. Peterson, A.J. Wand, Self-contained high-pressure cell, apparatus, and procedure for the preparation of encapsulated proteins dissolved in low viscosity fluids for nuclear magnetic resonance spectroscopy. Review of Scientific Instruments 76 (2005) -.

[20] C.R. Babu, P.F. Flynn, A.J. Wand, Preparation, characterization, and NMR spectroscopy of encapsulated proteins dissolved in low viscosity fluids. Journal of Biomolecular Nmr 25 (2003) 313-323.

[21] T. Szyperski, H.S. Atreya, Principles and applications of GFT projection NMR spectroscopy. Magn. Reson. Chem. 44 Spec No (2006) S51-60.

[22] R.N. Bracewell, The Fourier transform and its applications, McGraw-Hill, New York, 1986.

[23] H. Nyquist, Certain topics in telegraph transmission theory. Trans. AIEE 47 (1928) 617-644.

[24] J.C. Hoch, A.S. Stern, NMR data processing, Wiley-Liss, New York, 1996.

[25] E. Kupce, R. Freeman, Projection-reconstruction technique for speeding up multidimensional NMR spectroscopy. J Am Chem Soc 126 (2004) 6429-40.

[26] K. Kazimierczuk, A. Zawadzka, W. Kozminski, I. Zhukov, Random sampling of evolution time space and Fourier transform processing. J Biomol NMR 36 (2006) 157-68.

[27] N. Pannetier, K. Houben, L. Blanchard, D. Marion, Optimized 3D-NMR sampling for resonance assignment of partially unfolded proteins. J Magn Reson 186 (2007) 142-9.

[28] K. Kazimierczuk, A. Zawadzka, W. Kozminski, I. Zhukov, Lineshapes and artifacts in Multidimensional Fourier Transform of arbitrary sampled NMR data sets. J Magn Reson 188 (2007) 344-56.

[29] K. Kazimierczuk, A. Zawadzka, W. Kozminski, Optimization of random time domain sampling in multidimensional NMR. J Magn Reson 192 (2008) 123-30.

[30] B.E. Coggins, P. Zhou, Sampling of the NMR time domain along concentric rings. J Magn Reson 184 (2007) 207-21.

[31] R.A. Chylla, J.L. Markley, Theory and Application of the Maximum-Likelihood Principle to Nmr Parameter-Estimation of Multidimensional Nmr Data. Journal of Biomolecular Nmr 5 (1995) 245-258.

[32] J.C. Hoch, A.S. Stern, Maximum entropy reconstruction, spectrum analysis and deconvolution in multidimensional nuclear magnetic resonance. Methods Enzymol 338 (2001) 159-78.

[33] H.T. Hu, A.A. De Angelis, V.A. Mandelshtam, A.J. Shaka, The multidimensional filter diagonalization method - II. Application to 2D projections of 2D, 3D, and 4D NMR experiments. Journal of Magnetic Resonance 144 (2000) 357-366.

[34] V. Jaravine, I. Ibraghimov, V.Y. Orekhov, Removal of a time barrier for high-resolution multidimensional NMR spectroscopy. Nat Methods 3 (2006) 605-7.

[35] N. Trbovic, S. Smirnov, F.L. Zhang, R. Bruschweiler, Covariance NMR spectroscopy by singular value decomposition. Journal of Magnetic Resonance 171 (2004) 277-283.

[36] J.W. Yoon, S. Godsill, E. Kupce, R. Freeman, Deterministic and statistical methods for reconstructing multidimensional NMR spectra. Magn Reson Chem 44 (2006) 197-209.

[37] G.N. Hounsfield, Computerized transverse axial scanning (tomography). 1. Description of system. Br J Radiol 46 (1973) 1016-22.

[38] S. Deans, The Radon Transform and Some of Its Applications, Krieger Publishing Company.

[39] W. Kozminski, I. Zhukov, Multiple quadrature detection in reduced dimensionality experiments. Journal of Biomolecular Nmr 26 (2003) 157-166.

[40] K. Nagayama, P. Bachmann, K. Wuthrich, R.R. Ernst, Use of cross-sections and of projections in 2-dimensional NMR-spectroscopy. Journal of Magnetic Resonance 31 (1978) 133-148.

[41] R.A. Venters, B.E. Coggins, D. Kojetin, J. Cavanagh, P. Zhou, (4,2)D projection-reconstruction experiments for protein backbone assignment: Application to human carbonic anhydrase II and calbindin D-28K. Journal of the American Chemical Society 127 (2005) 8785-8795.

[42] C.D. Ridge, V.A. Mandelshtam, On projection-reconstruction NMR. Journal of Biomolecular Nmr 43 (2009) 151-159.

[43] S. Hiller, G. Wider, K. Wuthrich, APSY-NMR with proteins: practical aspects and backbone assignment. J Biomol NMR 42 (2008) 179-95.

[44] H.R. Eghbalnia, A. Bahrami, M. Tonelli, K. Hallenga, J.L. Markley, High-resolution iterative frequency identification for NMR as a general strategy for multidimensional data collection. J Am Chem Soc 127 (2005) 12528-36.

[45] B.E. Coggins, P. Zhou, Polar Fourier transforms of radially sampled NMR data. J Magn Reson 182 (2006) 84-95.

[46] D. Marion, Processing of ND NMR spectra sampled in polar coordinates: a simple Fourier transform instead of a reconstruction. J Biomol NMR 36 (2006) 45-54.

[47] K. Kazimierczuk, W. Kozminski, I. Zhukov, Two-dimensional Fourier transform of arbitrarily sampled NMR data sets. J Magn Reson 179 (2006) 323-8.

[48] B.E. Coggins, P. Zhou, High resolution 4-D spectroscopy with sparse concentric shell sampling and FFT-CLEAN. J Biomol NMR 42 (2008) 225-39.

[49] K. Kazimierczuk, A. Zawadzka, W. Kozminski, Narrow peaks and high dimensionalities: exploiting the advantages of random sampling. J Magn Reson 197 (2009) 219-28.

[50] K. Kazimierczuk, A. Zawadzka, W. Kozminski, I. Zhukov, Determination of spin-spin couplings from ultrahigh resolution 3D NMR spectra obtained by optimized random

sampling and multidimensional Fourier transformation. J Am Chem Soc 130 (2008) 5404-5.

[51] T.D. Goddard, D.G. Kneller, SPARKY 3, University of California, San Francisco.

[52] A.T. Brünger, X-PLOR, Version 3.1 : a system for X-ray crystallography and NMR, Yale University Press, New Haven, 1992.

[53] G. Van Rossum, Python Programming Language (www.python.org).

[54] T. Oliphant, Numerical Python NumPy (http://numpy.scipy.org).

[55] A.G. Palmer, J. Cavanagh, P.E. Wright, M. Rance, Sensitivity improvement in proton-detected two-dimensional heteronuclear relay spectroscopy. J. Magn. Reson. 91 (1991) 429-436

[56] L.E. Kay, P. Keifer, T. Saarinen, Pure absorption gradient enhanced heteronuclear single quantum correlation spectroscopy with improved sensitivity. 114 (1992) 10663–10665.

[57] D. Marion, M. Ikura, R. Tschudin, A. Bax, Rapid recording of 2D NMR spectra without phase cycling. Application to the study of hydrogen exchange in proteins. J. Magn. Reson. 85 (1989) 393-399.

[58] R.R. Ernst, Numerical Hilbert transform and automatic phase correction in magnetic resonance spectroscopy. Journal of Magnetic Resonance 1 (1969) 7-26.

[59] J.W. Cooley, J.W. Tukey, An algorithm for machine calculation of complex Fourier series. Mathematics of Computation 19 (1965) 297-301.

[60] D.R. Muhandiram, L.E. Kay, Gradient-enhanced triple-resonance 3-Dimensional NMR experiments with improved sensitivity. J. Magn. Reson. Ser. B 103 (1994) 203-216.

[61] D.H. Eberly, 3D game engine design : a practical approach to real-time computer graphics, Morgan Kaufmann, San Francisco 2000.

[62] A.J. Wand, J.L. Urbauer, R.P. McEvoy, R.J. Bieber, Internal dynamics of human ubiquitin revealed by 13C-relaxation studies of randomly fractionally labeled protein. Biochemistry 35 (1996) 6116-6125.

[63] R.R. Ernst, W.A. Anderson, Application of Fourier Transform Spectroscopy to Magnetic Resonance. Review of Scientific Instruments 37 (1966) 93-&.

[64] R. Baumann, G. Wider, R.R. Ernst, K. Wuthrich, Improvement of 2d Noe and 2d Correlated Spectra by Symmetrization. Journal of Magnetic Resonance 44 (1981) 402-406.

[65] B.E. Coggins, R.A. Venters, P. Zhou, Filtered backprojection for the reconstruction of a high-resolution (4,2)D CH3-NH NOESY spectrum on a 29 kDa protein. J Am Chem Soc 127 (2005) 11562-3.

[66] G.P. Wadsworth, J.G. Bryan, Introduction to probability and random variables, McGraw-Hill, New York, 1960.

[67] D. Neuhaus, M.P. Williamson, The nuclear Overhauser effect in structural and conformational analysis, Wiley-VCH, New York ; Chichester [England], 2000.

[68] L.E. Kay, G.M. Clore, A. Bax, A.M. Gronenborn, Four-dimensional heteronuclear triple-resonance NMR spectroscopy of interleukin-1 beta in solution. Science 249 (1990) 411-4.

[69] S.W. Fesik, E.R.P. Zuiderweg, Heteronuclear three-dimensional nmr spectroscopy. A strategy for the simplification of homonuclear two-dimensional NMR spectra J. Magn. Reson. 78 (1988) 588-593.

[70] L. Mueller, Sensitivity enhanced detection of weak nuclei using heteronuclear multiple quantum coherence. J Am Chem Soc 101 (1979) 4481-4484.