

---

# NMR methods for characterising protein structure in solution and solid state

Henry William Orton

---

Research School of Chemistry

February 2020

A thesis submitted for the degree of  
Doctor of Philosophy of the  
Australian National University



Australian  
National  
University

© Copyright by Henry William Orton 2020

All Rights Reserved

# Declaration of Originality

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. The content of my thesis is the result of work I have carried out since the commencement of my research degree candidature (February 2017 to February 2020) and has not been previously submitted for another degree or diploma in any university or tertiary institution.

Henry William Orton

February 2020

# Acknowledgements

In many ways a PhD is a personal undertaking which requires the self-directed development of research skills and understandings. However, I have come to appreciate the professional and social input of colleagues and friends who have collaborated with me to make the past 3 years an enjoyable journey.

To Professor Gottfried Otting, I am always grateful for your humble approach to ambitious projects and broad expert knowledge of the field of NMR. Your approachability and support encouraged me to pursue a challenging field which I have grown to cherish greatly.

Many thanks to Professor Guido Pintacuda and his group in Lyon, France. Your hospitality and generous sharing of knowledge during my 6 month visit was inspiring. I am particularly thankful to Doctor Jan Stanek for your generosity and patience in teaching me all that I know about solid-state NMR.

To my colleagues at the ANU, I am always grateful for your expertise when helping me to troubleshoot the intricacies of synthetic biology. Our everyday social interactions have kept me sane, whether it be popping down to the shop with Damian, or building a lightning detector with Matt, I really appreciated these good times.

Many thanks to my parents. Although I only visited you occasionally in New Zealand, I always appreciated talking on the phone most weeks. Thank you for helping me so much while I had a fractured ankle.

Finally, to the Westpac Foundation, I am grateful for the future leaders scholarship which has allowed me to focus on my studies and opened to me the world of international collaborative research.

# Abstract

Unpaired electrons give rise to spatially dependent paramagnetic phenomena observable by NMR spectroscopy which can be utilised for molecular structure determination. Paramagnetic lanthanide protein-complexes display distance-dependent relaxation enhancements of nuclear spins which can be used to refine molecular structure, an area of great interest to structural biology and drug discovery. Such relaxation enhancements are usually measured using spin labels with slowly relaxing unpaired electrons. The relaxation enhancement driven by paramagnetic lanthanide ions with fast relaxing electrons is theoretically well described by a magnetic susceptibility tensor, however, poor agreement with experiment has been observed previously.

Chapter 1 of this thesis reports the development of a unified theoretical framework for calculation of paramagnetic NMR phenomena and describes its implementation as a software tool for fitting magnetic susceptibility tensors. A coordinate-free matrix representation was derived for calculation of all paramagnetic NMR observables, including pseudocontact shifts, residual dipolar couplings, paramagnetic relaxation enhancements and cross-correlation effects between Curie-spin relaxation, chemical shift anisotropy and dipole-dipole relaxation. The software was then used to expose the shortcomings of established methods for measurement of paramagnetic relaxation enhancements with an experimental investigation of the model protein calbindin D<sub>9k</sub>. It was found that residual dipolar couplings and intermolecular relaxation enhancements contaminate the experimental measurement of relaxation, leading to artificially short distances as derived from theory. Novel NMR experiments were established to eliminate these artefacts, leading to reliable measurements and drastically improved distance predictions in proteins up to 25 Å from the paramagnetic centre.

To study higher molecular weight biological systems including amyloid fibrils, viral assemblies and membrane proteins in liposomes, solid-state NMR achieves distance measurements by methods of dipolar recoupling. Recent developments in ultra-fast magic angle spinning technology have allowed for proton-detected higher-dimensional NMR spectroscopy, and with



it, long-range inter-nuclear distance measurements for structure determination. An essential step towards structure analysis requires assignment of NMR chemical shifts to atoms of the molecule. Chapter 2 of this thesis pushes the limits of spin coherence transfer in NMR with the design and demonstration of a five-dimensional NMR experiment in the solid state for complete protein backbone assignment. Automated projection spectroscopy was used as a method for sparse sampling of the 5D experiment which was acquired for micro-crystalline GB1 protein, micro-crystalline superoxide dismutase and fibrillar B2 microglobulin. Fully automated assignment of chemical shifts was then demonstrated, showing that the new 5D experiment performs robust measurements that require very little human interpretation to complete an otherwise labour-intensive process.

Chapter 3 of this thesis describes efforts in establishing equipment for ultra-fast magic angle spinning at the Australian National University. The chapter includes protocols for sample packing and handling of 1.2 mm and 0.5 mm rotors as well as protocols for magic angle spinning at 60 kHz and 150 kHz respectively.

# List of publications

1. **Orton, H. W.**, Huber, T., and Otting, G.: Paramagpy: Software for fitting magnetic susceptibility tensors using paramagnetic effects measured in NMR spectra (2020) *Magnetic Resonance*, 1:1–12
2. **Orton, H. W.**, Otting, G.: Accurate electron-nucleus distances from paramagnetic relaxation enhancements (2018) *Journal of the American Chemical Society*, 140:7688–7697
3. **Orton, H. W.**, Stanek, J., Schubeis, T., Foucaudeau, D., Ollier, C., Draney, A. W., Le Marchand, T., Cala-De Paepe, D., Felli I. C., Pierattelli, R., Hiller, S., Bermel, W., Pintacuda G., Protein NMR resonance assignment without spectral analysis: 5D SOLid-state Automated Projection SpectroscopY (SO-APSY) (2020), *Angewandte Chemie International Edition*, 59:2380–2384

# Contents

<b>Declaration of Originality</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>List of publications</b>	<b>vi</b>
<b>Glossary</b>	<b>x</b>
<b>General introduction</b>	<b>1</b>
<b>1 Paramagnetic NMR in solution: theory and practice</b>	<b>3</b>
1.1 The magnetic susceptibility tensor . . . . .	3
1.2 Observing magnetic susceptibility in NMR spectra . . . . .	4
1.2.1 Pseudocontact shift (PCS) . . . . .	4
1.2.2 Paramagnetic relaxation enhancement (PRE) . . . . .	5
1.2.3 Cross-correlated relaxation (CCR) . . . . .	7
1.2.4 Residual dipolar coupling (RDC) . . . . .	8
1.3 Limitations of paramagnetic NMR theory . . . . .	8
1.4 A coordinate-free matrix formalism . . . . .	9
1.4.1 Optimising PCS calculation . . . . .	9
1.4.2 Optimising PRE calculation . . . . .	10
1.4.3 Optimising CCR calculation . . . . .	11
1.4.4 Further cross-correlation . . . . .	14
1.5 Python software for paramagnetic NMR . . . . .	17
1.6 Calbindin D <sub>9k</sub> : a model paramagnetic system . . . . .	19
1.7 Measuring distances derived from PREs . . . . .	20
<b>2 Protein NMR assignment in the solid state</b>	<b>22</b>
2.1 Ultra-fast magic angle spinning . . . . .	23
2.2 Sparse sampling with automated projection spectroscopy . . . . .	24

---

<b>3</b>	<b>Solid state NMR at the ANU: protocols established for UFMAS of biosolids</b>	<b>25</b>
3.1	60 kHz MAS broad-band probe . . . . .	26
3.1.1	Biosolid packing of 1.2 mm diameter rotors . . . . .	26
3.1.2	Probe change protocol . . . . .	28
3.2	150 kHz MAS probe . . . . .	29
3.2.1	Biosolid packing of 0.5 mm diameter rotors . . . . .	29
	<b>Conclusion and future directions</b>	<b>31</b>
	<b>Bibliography</b>	<b>39</b>
	<b>Appendix A Paramagpy: Software for Fitting Magnetic Susceptibility Tensors Using Paramagnetic Effects Measured in NMR Spectra</b>	<b>40</b>
	<b>Appendix B Accurate Electron-Nucleus Distances from Paramagnetic Relaxation Enhancements</b>	<b>157</b>
	<b>Appendix C Protein NMR Resonance Assignment without Spectral Analysis: 5D SOLid-State Automated Projection Spectroscopy (SO-APSY)</b>	<b>213</b>

# Glossary

**A** alignment tensor. 9

**APSY** automated projection spectroscopy. 24, 32

**CCR** cross-correlated relaxation. 7, 8, 11, 14, 18, 19

$\chi$  magnetic susceptibility tensor. 3, 4, 5, 6, 8, 9, 10, 11, 17, 18

$\Delta\chi$  anisotropic component of magnetic susceptibility tensor. ix, 4, 5, 10, 14, 18

$\Delta\chi_{\text{ax}}$  axial component of  $\Delta\chi$ . 5

$\Delta\chi_{\text{rh}}$  rhombic component of  $\Delta\chi$ . 5

$\chi_{\text{iso}}$  isotropic component of magnetic susceptibility tensor. 11

**CP** cross polarisation. 24

**CPMAS** cross polarisation with magic angle spinning. 24

**CSA** chemical shift anisotropy. 6, 9, 14, 23

**DD** dipole-dipole. 6, 7, 9, 11, 13, 14, 23, 24

$\Delta$  second invariant of a tensor. 10

$\delta^{\text{CSA}}$  CSA tensor. 14

**FKM** a type of fluoroelastomer. 26, 27

$\gamma_{\text{H}}$   $^1\text{H}$  gyromagnetic ratio. 8

$\gamma_{\text{N}}$   $^{15}\text{N}$  gyromagnetic ratio. 8

$g$  electronic g-factor. 3

**GUI** graphical user interface. 18, 19, 31

$\hbar$  Planck's constant divided by  $2\pi$ . 8

**INEPT** insensitive nuclei enhanced by polarization transfer. 23

$I$  nuclear spin angular momentum. 12

**J** spectral density. 6, 8

$k_{\text{B}}$  Boltzmann constant. 3

**MAS** Magic angle spinning. 22, 23

$\mu$  electron spin magnetic moment. 6

$\mu_0$  permeability of free space. 3

- $\langle \mu \rangle$  population averaged electron spin magnetic moment. 3
- $\mu_B$  Bohr magneton. 3
- NUS** non-uniform sampling. 24
- $\omega$  nuclear Larmor frequency. 6
- $\omega_H$   $^1\text{H}$  Larmor frequency. 8
- PCS** pseudocontact shift. 4, 5, 8, 9, 18, 19, 31
- $\phi$  azimuth angle of spherical polar coordinates. 5, 8
- PRE** paramagnetic relaxation enhancement. 6, 10, 18, 19, 32
- PTFE** polytetrafluoroethylene. 26, 28
- $R_1$  longitudinal nuclear spin relaxation rate. 6, 20, 21
- $R_2$  transverse nuclear spin relaxation rate. 6, 13, 20, 21
- $R_2^{\text{Curie} \times \text{DD}}$  Curie spin dipole-dipole cross-correlated transverse relaxation rate. 8
- RACS** residual anisotropic chemical shift. 14
- $r$  euclidean distance. 5, 8, 10
- RDC** residual dipolar coupling. 8, 9, 18, 19, 32
- $r_{\text{HN}}$   $^1\text{H}$ – $^{15}\text{N}$  internuclear bond length. 8, 12
- $\vec{r}_{\text{HN}}$   $^1\text{H}$ – $^{15}\text{N}$  internuclear bond vector. 13
- RMSE** root mean squared deviation. 32
- $R$   $3 \times 3$  rotation matrix. 4
- $\vec{r}$  Cartesian position vector. 8, 9, 13
- SBM** Solomon Bloembergen Morgan. 6
- $\sigma$  paramagnetic shielding tensor. 4, 5, 6, 9, 11
- $\sigma_{\text{iso}}$  isotropic component of paramagnetic shielding tensor. 5, 9
- $\sigma_{\text{aniso}}$  anisotropic component of paramagnetic shielding tensor. 5, 6, 7, 10
- $\sigma_N$  dipole-dipole coupling tensor from  $^{15}\text{N}$ . 11
- $S$  electron spin angular momentum. 3
- SVD** singular value decomposition. 18
- $T_{1e}$  electronic relaxation time. 6
- $\tau_r$  rotational correlation time. 6
- $T$  temperature in Kelvin. 3
- $\theta$  polar angle of spherical polar coordinates. 5, 8
- $\theta_{\text{HN}}$  angle subtended by  $^1\text{H}$ – $^{15}\text{N}$  internuclear bond vector with electron nuclear vector. 8, 13
- Tr** trace operation (summation of diagonal components of a matrix). 10
- UFMAS** ultra-fast magic angle spinning. 25, 26, 31, 32
- $\otimes$  Kronecker product. 10

# General introduction

NMR spectroscopy can be used to indirectly probe the local environment of nuclei by observing their magnetic spin. A number of experiments have been developed to infer structure of proteins from their NMR spectra, often featuring 3 dimensions or more (Sattler et al., 1999). For diamagnetic proteins dissolved in solution, structural information can be inferred from the nuclear Overhauser effect (NOE) which allows coherence transfer within distances less than 5 Å (Wüthrich, 1986). Additionally, torsion angles can be derived from measurement of chemical shifts and J-coupling constants (Haasnoot et al., 1980; Shen et al., 2009). Such phenomena contain valuable local information to aid calculation of protein secondary structure and domain contacts. However, propagating these many short distance restraints to infer tertiary structure on the nanometre scale can result in errors in the global protein fold which can be disastrous for subsequent predictions of protein function (Güntert, 2003).

Longer distances in the range 10–30 Å are accessible from NMR phenomena arising in paramagnetic protein complexes (Otting, 2010). Lanthanide complexes with fast relaxing unpaired electrons give rise to pseudocontact shifts (PCS), residual dipolar couplings (RDC), paramagnetic relaxation enhancements (PRE) and cross-correlated relaxation (CCR) (Bertini et al., 2002b). All such effects can be predicted from a magnetic susceptibility tensor  $\chi$  which is characterised by a total of 9 parameters which describe position, magnitude, anisotropy and orientation. The PCS contains information regarding distance and orientation with respect to the magnetic susceptibility tensor. RDC and CCR offer constraints for internuclear bond orientations. Distance from the paramagnetic centre can be calculated directly from PREs. Consequently, accurate knowledge of the magnetic susceptibility tensor for a given paramagnetic protein complex permits refinement of the protein structure or calculation of a ligand binding mode from experimentally observed paramagnetic NMR restraints.

Structural studies in solution NMR spectroscopy are limited to soluble proteins with molecular weights of usually less than 50 kDa, and less than 200 kDa when using perdeuterated pro-

teins with TROSY techniques. Solid-state NMR spectroscopy permits the study of larger and insoluble systems that include amyloid fibrils, viral assemblies and membrane proteins in liposomes (Stanek et al., 2016; Schubeis et al., 2018). To achieve acceptable resolution and sensitivity without perdeuteration, proton-detected ultra-fast magic angle spinning (UFMAS) in excess of 100 kHz is highly advantageous for protein samples. This relatively new technology has allowed the application of well-established dipolar recoupling experiments to high gyromagnetic ratio nuclei such as  $^1\text{H}$  and  $^{19}\text{F}$  for distances measurements on the order of 10 Å (Linser et al., 2011; Wi et al., 2004). Solid-state NMR spectroscopy thus allows structural characterisation within a useful distance range to determine the tertiary structures of a broader range of proteins.



# Chapter 1

## Paramagnetic NMR in solution: theory and practice

### 1.1 The magnetic susceptibility tensor

Unpaired electrons with spin  $S$  placed in an external magnetic field have non-degenerate spin energies which arise from Zeeman splitting. If the spin transitions are rapid compared to the rotational diffusion of the molecule (picosecond electronic relaxation) the spin populations follow the Boltzmann law. The resulting time-averaged population difference gives a magnetic moment  $\langle \mu \rangle$  which to first order is proportional to the applied field as dictated by the Curie law (see equation 1.1) (Bleaney and Bleaney, 1976). This proportionality for a given paramagnetic centre is described by the magnetic susceptibility  $\chi$  (see equation 1.2) where  $\mu_0$  is the permeability of free space,  $\mu_B$  is the Bohr magneton,  $g$  is the electronic  $g$  factor,  $k_B$  is the Boltzmann constant and  $T$  is the temperature (Bertini et al., 2002b).

$$\langle \mu \rangle = \frac{\chi}{\mu_0} \cdot B_0 \quad (1.1)$$

$$\chi = \mu_0 \mu_B^2 g^2 \frac{S(S+1)}{3k_B T} \quad (1.2)$$

The electronic  $g$  factor is not necessarily isotropic as spin–orbit coupling leads to orientation dependent energy with respect to the external magnetic field. Many paramagnetic molecular complexes, particularly those with lanthanide ions, have non-degenerate energies which can be described by an anisotropic  $g$  factor with a tensor property. This anisotropy is reflected in the magnetic susceptibility tensor  $\chi$  (Bertini et al., 2002b). In practical terms, anisotropy of the

magnetic susceptibility introduces a dependence of paramagnetic spin interactions on the orientation of the molecular complex with respect to the external magnetic field. Consequently, spatially dependent effects within the molecular reference frame are observable in NMR spectra.

A  $3 \times 3$  symmetric matrix along with a Cartesian position  $[x, y, z]$  gives a complete description of the  $\chi$  tensor in the molecular coordinate frame (see equation 1.3). The  $\chi$  tensor can also be diagonalised by a rotation matrix  $R$  (see equation 1.4) which represents a rotation of the molecular coordinate frame to the  $\chi$  tensor principal axes. Many spin interactions only depend on the anisotropic component of the  $\chi$  tensor which is denoted  $\Delta\chi$ . The  $\Delta\chi$  tensor can be calculated by subtracting the isotropic component of  $\chi$  from the diagonal elements of the matrix.

$$\chi = \begin{bmatrix} \chi_{xx} & \chi_{xy} & \chi_{xz} \\ \chi_{xy} & \chi_{yy} & \chi_{yz} \\ \chi_{xz} & \chi_{yz} & \chi_{zz} \end{bmatrix} \quad (1.3)$$

$$\chi = R \begin{bmatrix} \chi_{xx} & 0 & 0 \\ 0 & \chi_{yy} & 0 \\ 0 & 0 & \chi_{zz} \end{bmatrix} R^{-1} \quad (1.4)$$

## 1.2 Observing magnetic susceptibility in NMR spectra

The magnetic susceptibility is a source for hyperfine interactions and molecular alignment which are observable in NMR spectra as perturbations to chemical shift, relaxation and nuclear spin couplings.

### 1.2.1 Pseudocontact shift (PCS)

The dipolar magnetic field arising from the  $\chi$  tensor can be described at the site of a nuclear spin by a paramagnetic shielding tensor  $\sigma$  which defines the hyperfine coupling. In the case of isotropic magnetic susceptibility, the  $\sigma$  tensor at the position of the nuclear spin is traceless, meaning the average shielding over all molecular orientations is zero and therefore chemical shift is unperturbed. However, the presence of anisotropy in the magnetic susceptibility contributes a spatially dependent isotropic component to the shielding tensor which is observable in the NMR spectrum as the pseudocontact shift (PCS) (Kurland and McGarvey, 1970). This is

represented by the isotropic component  $\sigma_{\text{iso}}$  of the tensor representation in Figure 1.1.

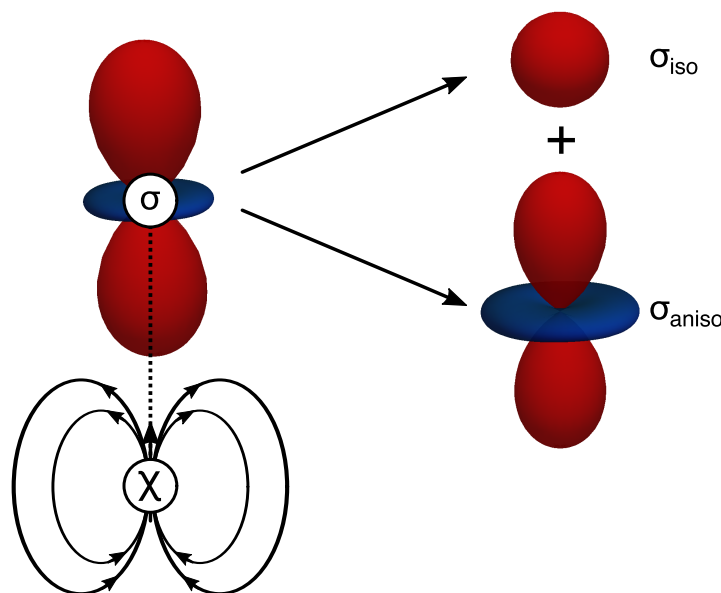


Figure 1.1: Schematic representation of electron–nuclear spin interactions. The magnetic field arising from the  $\chi$  tensor yields a shielding tensor  $\sigma$  which can be decomposed into an isotropic  $\sigma_{\text{iso}}$  and anisotropic  $\sigma_{\text{aniso}}$  component.

The calculation of the PCS commonly uses equation 1.5 where  $r$ ,  $\theta$  and  $\phi$  define the polar coordinates of the electron-nuclear vector within the coordinate frame of the  $\chi$  tensor (Otting, 2010).  $\Delta\chi_{\text{ax}}$  and  $\Delta\chi_{\text{rh}}$  are the axial and rhombic components of the  $\Delta\chi$  tensor respectively as in equation 1.4.

$$\delta^{\text{PCS}} = \frac{1}{12\pi r^3} \left[ \Delta\chi_{\text{ax}} (3 \cos^2(\theta) - 1) + \frac{3}{2} \Delta\chi_{\text{rh}} \sin^2(\theta) \cos(2\phi) \right] \quad (1.5)$$

PCS is measured in the NMR spectra of a molecular complex as the change in chemical shift in the presence and absence of a paramagnetic centre. The measured PCS contains information about the length and orientation of the vector between the nuclear spin and the paramagnetic centre and has been used for structure refinement of proteins and the identification of ligand binding sites (Otting, 2010; Pintacuda et al., 2007; Schmitz et al., 2012; Gaponenko et al., 2004).

### 1.2.2 Paramagnetic relaxation enhancement (PRE)

The relaxation of a nuclear spin is driven by fluctuating magnetic fields stemming from the surrounding molecular environment. Typical sources for such magnetic fields are neighbouring

nuclear spins (dipole-dipole (DD) interactions) or the surrounding electrons interacting with the external magnetic field (chemical shift anisotropy (CSA) relaxation) (Abragam, 1961). For a protein molecule, the magnetic field experienced at the nuclear spin is predominantly modulated by the rotational diffusion of the molecule on the nanosecond time-scale as characterised by the rotational correlation time  $\tau_r$ . This stochastic process typically yields an exponential behaviour for relaxation which can be measured in an NMR experiment either as the rate of recovery of longitudinal magnetisation  $R_1$  or as the rate of decay of transverse magnetisation  $R_2$  (Cavanagh et al., 2007).

The anisotropic component  $\sigma_{\text{aniso}}$  of the  $\sigma$  tensor in Figure 1.1 characterises the changing magnetic field at the site of the nuclear spin due to the  $\chi$  tensor. In the context of a model for stochastic rotational diffusion of the molecule, the paramagnetic relaxation enhancement (PRE) arises from the magnetic field fluctuations over time (Guéron, 1975; Vega and Fiat, 1976). This so called Curie-spin relaxation arises due to the magnetic susceptibility and is the dominant relaxation mechanism for paramagnetic lanthanides with fast-relaxing electrons. The isotropic component of the magnetic susceptibility is the predominant contributor to the  $\sigma_{\text{aniso}}$  tensor from which the longitudinal and transverse PRE can be calculated using equation 1.6 and 1.7 respectively, where  $\omega$  is the nuclear Larmor frequency and  $\mathbf{J}$  is the spectral density as defined in equation 1.8.

$$R_1^{\text{Curie}} = \frac{2}{5} \left( \frac{\mu_0}{4\pi} \right)^2 \frac{\omega^2 \chi_{\text{iso}}^2}{r^6} \mathbf{J}(\omega) \quad (1.6)$$

$$R_2^{\text{Curie}} = \frac{1}{5} \left( \frac{\mu_0}{4\pi} \right)^2 \frac{\omega^2 \chi_{\text{iso}}^2}{r^6} [4\mathbf{J}(0) + 3\mathbf{J}(\omega)] \quad (1.7)$$

$$\mathbf{J}(\omega) = \frac{\tau_r}{1 + \omega^2 \tau_r^2} \quad (1.8)$$

For paramagnetic centres with slowly relaxing electrons the instantaneous magnetic moment  $\mu$  contributes to the PRE as described by Solomon Bloembergen Morgan (SBM) theory (Solomon, 1955). This is the case for  $\text{Gd}^{3+}$ , where the electronic relaxation time  $T_{1e}$  is much longer than the rotational correlation time  $\tau_r$  for most macromolecules in solution. SBM relaxation is negligible for all other paramagnetic lanthanide ions where  $T_{1e} \ll \tau_r$ .

Both theories postulate a dependence of the PRE as  $r^{-6}$  from the paramagnetic centre. PREs have been measured to help characterise protein solvent exposure and for protein structure refinement (Pintacuda and Otting, 2002; Bertini et al., 2001a).

### 1.2.3 Cross-correlated relaxation (CCR)

Nuclear dipole-dipole (DD) relaxation can interfere with Curie-spin relaxation in a process called cross-correlated relaxation (CCR). CCR relaxation is most commonly considered between  $^1\text{H}$  and  $^{15}\text{N}$  spins as are present in the backbone of a protein molecule. The magnetic fields at the  $^1\text{H}$  spin from  $\sigma_{\text{aniso}}$  and the  $^{15}\text{N}$ -dipole moment can constructively interfere to enhance relaxation or destructively interfere to reduce relaxation. This scenario is observable in the  $^1\text{H}$ -NMR spectrum as a difference in the linewidth of the two components of the  $^{15}\text{N}$ -coupled doublet which is depicted in Figure 1.2.

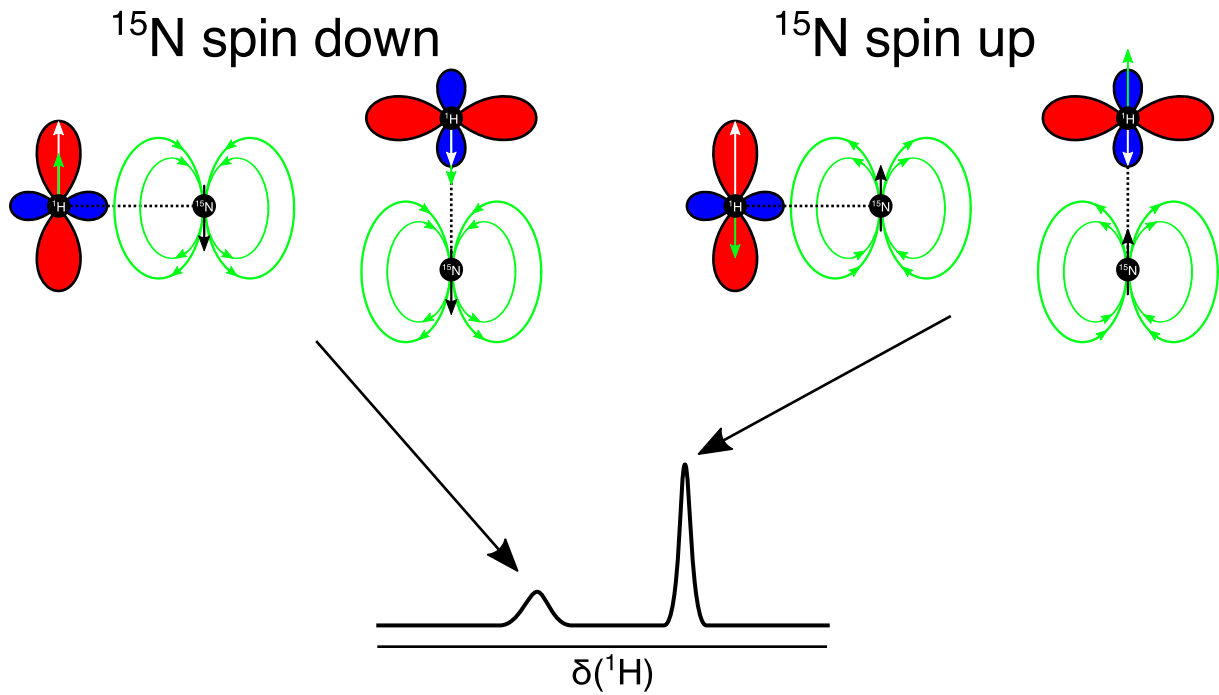


Figure 1.2: Schematic representation of CCR observable as difference in NMR peak line-width of a  $^1\text{H}$ - $^{15}\text{N}$  spin pair. The  $\sigma_{\text{aniso}}$  tensor as found in Figure 1.1 is shown at the  $^1\text{H}$  spin with the induced field indicated by the white arrow. The  $^{15}\text{N}$  dipole field is shown in green with the field at the site of the  $^1\text{H}$  spin shown by a green arrow. Left: the  $^{15}\text{N}$  spin-down state gives rise to constructive interference between the Curie-spin and spin-dipole instantaneous fields and a broad peak in the  $^1\text{H}$  NMR spectrum. Right: the  $^{15}\text{N}$  spin-up state gives rise to destructive interference between the Curie-spin and spin-dipole instantaneous fields and a narrow peak in the  $^1\text{H}$ -NMR spectrum.

$$R_2^{\text{Curie} \times \text{DD}} = \frac{9}{45} \frac{\mu_0}{4\pi} \frac{B_0 \gamma_H^2 \gamma_N \hbar}{r_{\text{HN}}^3} \frac{\chi_{\text{iso}}}{4\pi r^3} (3 \cos^2(\theta_{\text{HN}}) - 1) [4\mathbf{J}(0) + 3\mathbf{J}(\omega_{\text{H}})] \quad (1.9)$$

The CCR difference in transverse relaxation rate  $R_2^{\text{Curie} \times \text{DD}}$  can be calculated using equation 1.9 where  $r_{\text{HN}}$  is the  $^1\text{H}$ - $^{15}\text{N}$  internuclear bond length,  $\gamma_{\text{H}}$  and  $\gamma_{\text{N}}$  are the  $^1\text{H}$  and  $^{15}\text{N}$  gyromagnetic ratios respectively,  $\hbar$  is Planck's constant divided by  $2\pi$ ,  $\theta_{\text{HN}}$  is the angle between the  $^1\text{H}$ - $^{15}\text{N}$  bond vector and the electron-nuclear vector  $\vec{r}$ ,  $\mathbf{J}$  is the spectral density and  $\omega_{\text{H}}$  is the Larmor frequency of  $^1\text{H}$  (Ghose and Prestegard, 1997). CCR measurements have been used to restrain nuclear bond orientations for molecular structure refinement (Bertini et al., 2002a; Pintacuda et al., 2003). A generalisation of this theory considers the anisotropy of the  $\chi$  tensor (Bertini et al., 2001c).

### 1.2.4 Residual dipolar coupling (RDC)

Anisotropy of the magnetic susceptibility means the electronic Zeeman splitting energy varies with the orientation of the paramagnetic centre with respect to the external magnetic field. Consequently, some orientations of the paramagnetic complex have lower energy, and become more populated, following a Boltzmann distribution. This process means that molecular orientations are no longer isotropically distributed for samples dissolved in solution. The net alignment becomes observable in the NMR spectrum by partially introducing direct dipole couplings between nuclear spins in what is called the residual dipolar coupling (RDC) (Saupe and Englert, 1963).

Although the RDC is most often described for diamagnetic molecules in the presence of alignment media, the mathematical formalism has been generalised for paramagnetically induced RDCs in terms of the magnetic susceptibility as shown in equation 1.10 for a  $^1\text{H}$ - $^{15}\text{N}$  spin pair.

$$\nu^{\text{RDC}} = -\frac{1}{4\pi} \frac{B_0^2}{15k_{\text{B}}T} \frac{\gamma_{\text{H}}\gamma_{\text{N}}\hbar}{2\pi r_{\text{HN}}^3} \left[ \Delta\chi_{\text{ax}} (3\cos^2(\theta) - 1) + \frac{3}{2}\Delta\chi_{\text{rh}} \sin^2(\theta) \cos(2\phi) \right] \quad (1.10)$$

In practice, RDC measurements can be used to restrain inter-nuclear bond orientations for protein structure refinement (Tolman et al., 1995; Tjandra and Bax, 1997). Although containing similar information to CCR, the induced alignment affects the entire molecule, meaning RDCs are independent of the distance from the metal centre and can be observed for all spins.

## 1.3 Limitations of paramagnetic NMR theory

In order to calculate PCS, CCR and RDC effects, familiar equations are quoted in terms of spherical polar coordinates  $r$ ,  $\theta$  and  $\phi$  for the electron-nuclear vector with respect to the  $\chi$  tensor

principal axes (for example equations 1.5, 1.9 and 1.10). The equations in this form require a rotation of the molecular coordinates to be aligned with the principal axes of the  $\chi$  tensor.

Equations in this form pose the following limitations:

1. Calculations are only valid for a given molecular orientation and often require heavy trigonometric calculations to rotate atomic coordinates.
2. Two interaction tensors cannot be represented within the same molecular coordinate frame without requiring angle-dependent terms between their principal axes. Cross-correlated effects such as those with relaxation and molecular alignment then often require additional calculations to rotate coordinates and apply the relevant theory.

## 1.4 A coordinate-free matrix formalism

To overcome the limitations posed in section 1.3 it is necessary to move to a representation where paramagnetic phenomena can be calculated independent of the molecular orientation. To aid computational speed, a  $3 \times 3$  matrix representation for tensor interactions is an obvious choice. Although a matrix formalism has been described in the literature for most paramagnetic NMR phenomena (Bertini et al., 2002b), it has not been widely adopted by the community. There are 5 possible spin interaction tensors to consider, and each may have an independent principal axis orientation for a given molecular orientation:

- $\chi$  for magnetic susceptibility of the paramagnetic molecule
- $\sigma$  for the paramagnetic shielding at the nuclear spin
- $\mathbf{A}$  for the alignment tensor for describing RDC (often coincident with the  $\chi$  tensor unless external alignment media are used).
- CSA arising from chemical shift anisotropy
- DD (dipole-dipole) tensor arising from nuclear spin coupling

### 1.4.1 Optimising PCS calculation

As an alternative to equation 1.5, the PCS is equivalent to the value of  $\sigma_{\text{iso}}$  which can be calculated as the trace of the shielding tensor  $\sigma$  in equation 1.11, where  $\vec{r}$  is the electron-nuclear

vector,  $r$  is the electron-nuclear distance,  $\Delta\chi$  is the traceless part of the  $\chi$  tensor,  $\otimes$  is the Kronecker product and  $\text{Tr}$  denotes the trace operation (Bertini et al., 2002b).

$$\delta^{\text{PCS}} = \frac{1}{3} \text{Tr}[\sigma] \quad (1.11)$$

$$\text{where } \sigma = \frac{1}{4\pi r^5} [3\vec{r} \otimes \vec{r}^T - r^2] \cdot \Delta\chi \quad (1.12)$$

Equation 1.11 can be simplified substantially to give equation 1.17. In equation 1.15 we use the fact that  $\Delta\chi$  is traceless. In the final step, we use the fact that the trace only takes diagonal elements, which can be derived from the product shown.

$$\delta^{\text{PCS}} = \frac{1}{12\pi r^5} \text{Tr} \left[ (3\vec{r} \otimes \vec{r}^T - r^2) \cdot \Delta\chi \right] \quad (1.13)$$

$$= \frac{1}{12\pi r^5} \left( 3 \text{Tr} \left[ (\vec{r} \otimes \vec{r}^T) \cdot \Delta\chi \right] - r^2 \text{Tr} [\Delta\chi] \right) \quad (1.14)$$

$$= \frac{1}{4\pi r^5} \text{Tr} \left[ (\vec{r} \otimes \vec{r}^T) \cdot \Delta\chi \right] \quad (1.15)$$

$$= \frac{1}{4\pi r^5} \text{Tr} \left( \begin{bmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{bmatrix} \cdot \begin{bmatrix} \Delta\chi_{xx} & \Delta\chi_{xy} & \Delta\chi_{xz} \\ \Delta\chi_{xy} & \Delta\chi_{yy} & \Delta\chi_{yz} \\ \Delta\chi_{xz} & \Delta\chi_{yz} & \Delta\chi_{zz} \end{bmatrix} \right) \quad (1.16)$$

$$= \frac{1}{4\pi r^5} \vec{r}^T \cdot \Delta\chi \cdot \vec{r} \quad (1.17)$$

Compared to equation 1.5, this elegant form of the PCS equation requires only a product of the electron-nuclear vector with the  $\Delta\chi$  tensor matrix and can be calculated for any molecular orientation, thus addressing problem 1 of section 1.1. Additional matrix implementations of RDC and PRE effects are described in the publication of Appendix A.

### 1.4.2 Optimising PRE calculation

A matrix formalism for Curie-spin relaxation has been established by Suterina et al. (Suterina et al., 2018). The theory describes the second invariant  $\Delta$  with components  $\sigma_{ii}$  of the  $\sigma_{\text{aniso}}$  tensor in equation 1.18. The Curie-spin relaxation rates are then given by equations 1.19 and 1.20. This theory has the added benefit of accounting for anisotropy of the  $\chi$  tensor which gives an orientation dependence to the PRE magnitude.



$$\Delta^2 = \sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx}\sigma_{yy} - \sigma_{xx}\sigma_{zz} - \sigma_{yy}\sigma_{zz} + \frac{3}{4} [((\sigma_{xy} + \sigma_{yx})^2 + (\sigma_{xz} + \sigma_{zx})^2 + (\sigma_{yz} + \sigma_{zy})^2)] \quad (1.18)$$

$$R_1^{\text{Curie}} = \frac{2}{15} \Delta^2 \omega^2 \left[ \frac{\tau_r}{1 + \omega^2 \tau_r^2} \right] \quad (1.19)$$

$$R_2^{\text{Curie}} = \frac{1}{45} \Delta^2 \omega^2 \left[ 4\tau_r + \frac{3\tau_r}{1 + \tau_r^2 \omega^2} \right] \quad (1.20)$$

### 1.4.3 Optimising CCR calculation

CCR can be described with a matrix formalism by considering the interference of the DD tensor and the paramagnetic shielding tensor with the relaxation theory of section 1.4.2. This formalism is proven below to be equivalent to the isotropic case as described by the original implementation in equation 1.9 (the proof was also given in the supplement to Appendix A).

We consider the case of a  $^{15}\text{N}$ - $^1\text{H}$  spin pair, with paramagnetic shielding tensor  $\sigma$  by expanding equation 1.12.

$$\sigma = \frac{1}{4\pi r^5} \begin{bmatrix} (3x^2 - r^2) & 3xy & 3xz \\ 3xy & (3y^2 - r^2) & 3yz \\ 3xz & 3yz & (3z^2 - r^2) \end{bmatrix} \cdot \begin{bmatrix} \chi_{xx} & \chi_{xy} & \chi_{xz} \\ \chi_{xy} & \chi_{yy} & \chi_{yz} \\ \chi_{xz} & \chi_{yz} & \chi_{zz} \end{bmatrix} \quad (1.21)$$

We choose the orientation of the electron- $^1\text{H}$  vector to be aligned with the z-axis such that  $z \rightarrow r$ . In the case of an isotropic magnetic susceptibility,  $x = y = 0$  and the  $\chi$  tensor is represented by a diagonal matrix with three identical elements  $\chi_{\text{iso}}$ , yielding the following simplification

$$\sigma = \frac{1}{4\pi r^5} \begin{bmatrix} -r^2 & 0 & 0 \\ 0 & -r^2 & 0 \\ 0 & 0 & 2r^2 \end{bmatrix} \cdot \begin{bmatrix} \chi_{\text{iso}} & 0 & 0 \\ 0 & \chi_{\text{iso}} & 0 \\ 0 & 0 & \chi_{\text{iso}} \end{bmatrix} \quad (1.22)$$

$$= \zeta_{\text{DSA}} \begin{bmatrix} -r^2 & 0 & 0 \\ 0 & -r^2 & 0 \\ 0 & 0 & 2r^2 \end{bmatrix} \quad (1.23)$$

$$\text{where } \zeta_{\text{DSA}} = \frac{\chi_{\text{iso}}}{4\pi r^5} \quad (1.24)$$

The nuclear DD-tensor  $\sigma_{\text{N}}$  arising from the interaction of the  $^{15}\text{N}$  spin with the  $^1\text{H}$  spin

can be described in the same coordinate frame for an arbitrary orientation of the bond vector  $\vec{r}_{\text{HN}} = [x, y, z]$  with bond length  $r_{\text{HN}}$  by

$$\sigma_N = \frac{1}{B_0} \frac{\mu_0}{4\pi} \gamma_N \hbar I \left[ 3 \frac{\vec{r}_{\text{HN}} \otimes \vec{r}_{\text{HN}}^T}{r_{\text{HN}}^5} - \frac{1}{r_{\text{HN}}^3} \right] \quad (1.25)$$

$$\sigma_N = \zeta_{\text{DD}} \begin{bmatrix} (3x^2 - r_{\text{HN}}^2) & 3xy & 3xz \\ 3xy & (3y^2 - r_{\text{HN}}^2) & 3yz \\ 3xz & 3yz & (3z^2 - r_{\text{HN}}^2) \end{bmatrix} \quad (1.26)$$

$$\text{where } \zeta_{\text{DD}} = \frac{1}{B_0} \frac{\mu_0}{4\pi} \frac{\gamma_N \hbar I}{r_{\text{HN}}^5} \quad (1.27)$$

and  $x, y, z$  denote the coordinates of the  $^{15}\text{N}$  spin relative to the  $^1\text{H}$  spin (Abragam, 1961).  $I$  denotes the nuclear spin angular momentum of  $^{15}\text{N}$ . Note the factor of  $\frac{1}{B_0}$  is necessary to make the tensor dimensionless.

The effective tensor at the site of the  $^1\text{H}$  spin, when the  $^{15}\text{N}$  partner is in the spin-up state, is given by the sum of the two tensors

$$\sigma_{\uparrow} = \sigma + \sigma_N \quad (1.28)$$

$$= \begin{bmatrix} (3x^2 - r_{\text{HN}}^2)\zeta_{\text{DD}} - (r^2)\zeta_{\text{DSA}} & (3xy)\zeta_{\text{DD}} & (3xz)\zeta_{\text{DD}} \\ (3xy)\zeta_{\text{DD}} & (3y^2 - r_{\text{HN}}^2)\zeta_{\text{DD}} - (r^2)\zeta_{\text{DSA}} & (3yz)\zeta_{\text{DD}} \\ (3xz)\zeta_{\text{DD}} & (3yz)\zeta_{\text{DD}} & (3z^2 - r_{\text{HN}}^2)\zeta_{\text{DD}} + (2r^2)\zeta_{\text{DSA}} \end{bmatrix} \quad (1.29)$$

Elements of the effective tensor  $\sigma_{\uparrow}$  are then substituted into equation 1.18. Expanding and simplifying (via symbolic processing in the program Mathematica), yields

$$\begin{aligned}
\Delta(\sigma_{\uparrow})^2 = & ((3x^2 - r_{\text{HN}}^2)\tilde{\zeta}_{\text{DD}} - (r^2)\tilde{\zeta}_{\text{DSA}})^2 \\
& + ((3y^2 - r^2)\tilde{\zeta}_{\text{DD}} - (r^2)\tilde{\zeta}_{\text{DSA}})^2 \\
& + ((3z^2 - r_{\text{HN}}^2)\tilde{\zeta}_{\text{DD}} + (2r^2)\tilde{\zeta}_{\text{DSA}})^2 \\
& - ((3x^2 - r_{\text{HN}}^2)\tilde{\zeta}_{\text{DD}} - (r^2)\tilde{\zeta}_{\text{DSA}})((3y^2 - r^2)\tilde{\zeta}_{\text{DD}} - (r^2)\tilde{\zeta}_{\text{DSA}}) \\
& - ((3x^2 - r_{\text{HN}}^2)\tilde{\zeta}_{\text{DD}} - (r^2)\tilde{\zeta}_{\text{DSA}})((3z^2 - r_{\text{HN}}^2)\tilde{\zeta}_{\text{DD}} + (2r^2)\tilde{\zeta}_{\text{DSA}}) \\
& - ((3y^2 - r^2)\tilde{\zeta}_{\text{DD}} - (r^2)\tilde{\zeta}_{\text{DSA}})((3z^2 - r_{\text{HN}}^2)\tilde{\zeta}_{\text{DD}} + (2r^2)\tilde{\zeta}_{\text{DSA}}) \\
& + \frac{3}{4} [(6xy\tilde{\zeta}_{\text{DD}})^2 + (6xz\tilde{\zeta}_{\text{DD}})^2 + (6yz\tilde{\zeta}_{\text{DD}})^2] \tag{1.30}
\end{aligned}$$

$$= 9\tilde{\zeta}_{\text{DD}}^2 r_{\text{HN}}^4 + 9\tilde{\zeta}_{\text{DSA}}^2 r^4 - 9\tilde{\zeta}_{\text{DD}}(r_{\text{HN}}^2 - 3z^2)\tilde{\zeta}_{\text{DSA}}r^2 \tag{1.31}$$

The angle  $\theta_{\text{HN}}$  between the electron-nuclear vector  $\vec{r}$  and the nuclear bond vector  $\vec{r}_{\text{HN}}$  is captured by the dot product formula

$$\vec{r} \cdot \vec{r}_{\text{HN}} = |\vec{r}| |\vec{r}_{\text{HN}}| \cos \theta_{\text{HN}} \tag{1.32}$$

$$\implies [0, 0, r] \cdot [x, y, z] = r r_{\text{HN}} \cos \theta_{\text{HN}} \tag{1.33}$$

$$\implies z = r_{\text{HN}} \cos \theta_{\text{HN}} \tag{1.34}$$

Substituting equation 1.34 into equation 1.31 gives

$$\Delta(\sigma_{\uparrow})^2 = 9\tilde{\zeta}_{\text{DD}}^2 r_{\text{HN}}^4 + 9\tilde{\zeta}_{\text{DSA}}^2 r^4 - 9\tilde{\zeta}_{\text{DD}} r_{\text{HN}}^2 (1 - 3 \cos^2 \theta_{\text{HN}}) \tilde{\zeta}_{\text{DSA}} r^2 \tag{1.35}$$

where the first two terms account for the DD and Curie-spin auto-relaxation terms respectively, and the last term accounts for their cross-correlation. The  $R_2$  relaxation rate can be calculated by substitution of  $\Delta^2$  into equation 1.20.

$$R_2^{\text{Curie}}(\sigma_{\uparrow}) = \frac{1}{45} \omega_{\text{H}}^2 [4\mathbf{J}(0) + 3\mathbf{J}(\omega_{\text{H}})] \left( 9\tilde{\zeta}_{\text{DD}}^2 r_{\text{HN}}^4 + 9\tilde{\zeta}_{\text{DSA}}^2 r^4 + 9\tilde{\zeta}_{\text{DD}} r_{\text{HN}}^2 (3 \cos^2 \theta_{\text{HN}} - 1) \tilde{\zeta}_{\text{DSA}} r^2 \right) \tag{1.36}$$

The same derivation for  $\sigma_{\downarrow}$  yields the same result except for a sign change in the cross term:

$$R_2^{\text{Curie}}(\sigma_{\downarrow}) = \frac{1}{45} \omega_{\text{H}}^2 [4\mathbf{J}(0) + 3\mathbf{J}(\omega_{\text{H}})] \left( 9\tilde{\zeta}_{\text{DD}}^2 r_{\text{HN}}^4 + 9\tilde{\zeta}_{\text{DSA}}^2 r^4 - 9\tilde{\zeta}_{\text{DD}} r_{\text{HN}}^2 (3 \cos^2 \theta_{\text{HN}} - 1) \tilde{\zeta}_{\text{DSA}} r^2 \right) \tag{1.37}$$

Taking the difference we obtain

$$R_2^{\text{Curie} \times \text{DD}} = R_2^{\text{Curie}}(\sigma_{\uparrow}) - R_2^{\text{Curie}}(\sigma_{\downarrow}) \quad (1.38)$$

$$= \frac{1}{45} B_0^2 \gamma_H^2 (18 \zeta_{\text{DD}} r_{\text{HN}}^2 (3 \cos^2 \theta_{\text{HN}} - 1) \zeta_{\text{DSA}} r^2) [4\mathbf{J}(0) + 3\mathbf{J}(\omega_{\text{H}})] \quad (1.39)$$

$$= \frac{9}{45} \frac{\mu_0}{4\pi} \frac{B_0 \gamma_H^2 \gamma_N \hbar}{r_{\text{HN}}^3} \frac{\chi_{\text{iso}}}{4\pi r^3} (3 \cos^2 \theta_{\text{HN}} - 1) [4\mathbf{J}(0) + 3\mathbf{J}(\omega_{\text{H}})] \quad (1.40)$$

which is equivalent to the original form of CCR theory as in equation 1.9.

#### 1.4.4 Further cross-correlation

The matrix formalism also lends itself to solve problem 2 of section 1.3. Calculating cross-correlation effects can be as simple as adding together two interaction tensors. For example, to calculate the cross-correlation of CSA relaxation with Curie-spin relaxation (which arises from the DD-coupling tensor), the two tensors are added together element-wise and then propagated through the standard equations for relaxation theory (Orton et al., 2016). This is described in detail in Appendix A.

For another example, the original publication of the residual anisotropic chemical shift (RACS) effect reports equation 1.41 (John et al., 2005). This equation describes a cross-correlation between the chemical shift anisotropy tensor  $\delta^{\text{CSA}}$  and the alignment tensor induced by the magnetic susceptibility  $\Delta\chi$ .

$$\Delta\delta^{\text{RACS}} = \frac{B_0^2}{15\mu_0 k_B T} \sum_{ij} \delta_{ii}^{\text{CSA}} \cos^2 \theta_{ij} \Delta\chi_{jj} \quad (1.41)$$

The matrix representation presents a more natural description of the RACS effect for which a derivation is shown below. To begin, we consider the principal axis vectors  $\vec{v}_i$  and  $\vec{w}_i$  of the  $\delta^{\text{CSA}}$  and  $\Delta\chi$  tensors respectively, such that  $i \in [x, y, z]$ . Equation 1.42 uses the dot product formula to relate the principal axis vectors to the cosine of their angle. The summation is then converted to the vector notation of equation 1.43. From here, equation 1.44 uses the trace operation of a  $3 \times 3$  matrix equation. The matrices are then simplified into their component row and column vectors for the principal axes in equation 1.45. The principal axes presented as column vectors in a  $3 \times 3$  matrix form a rotation transformation which has been labelled  $R_{\delta^{\text{CSA}}}$  for the vectors  $\vec{v}_i$  and  $R_{\Delta\chi}$  for the vectors  $\vec{w}_i$  in equation 1.46. Note that the vectors  $\vec{v}_i$  and  $\vec{w}_i$  when written as row vectors simply form the transpose of the transformation. The trace operation is invariant under cyclic permutation of the arguments and so they are reshuffled in

---

equation 1.47. In equation 1.48 we apply equation 1.4 to substitute the native tensor from its diagonal form. In the final step, we exploit a theorem that says the trace of the product of two matrices is equivalent to the sum of their element-wise product to give equation 1.49 (Styan, 1973).

$$\Delta\delta^{\text{RACS}} = \frac{B_0^2}{15\mu_0 k_B T} \sum_{i,j \in [x,y,z]} \delta_{ii}^{\text{CSA}} (\vec{v}_i \cdot \vec{w}_j)^2 \Delta\chi_{jj} \quad (1.42)$$

$$= \frac{B_0^2}{15\mu_0 k_B T} \begin{bmatrix} \delta_{xx}^{\text{CSA}} & \delta_{yy}^{\text{CSA}} & \delta_{zz}^{\text{CSA}} \end{bmatrix} \begin{bmatrix} (\vec{v}_x \cdot \vec{w}_x)^2 & (\vec{v}_x \cdot \vec{w}_y)^2 & (\vec{v}_x \cdot \vec{w}_z)^2 \\ (\vec{v}_y \cdot \vec{w}_x)^2 & (\vec{v}_y \cdot \vec{w}_y)^2 & (\vec{v}_y \cdot \vec{w}_z)^2 \\ (\vec{v}_z \cdot \vec{w}_x)^2 & (\vec{v}_z \cdot \vec{w}_y)^2 & (\vec{v}_z \cdot \vec{w}_z)^2 \end{bmatrix} \begin{bmatrix} \Delta\chi_{xx} \\ \Delta\chi_{yy} \\ \Delta\chi_{zz} \end{bmatrix} \quad (1.43)$$

$$= \frac{B_0^2}{15\mu_0 k_B T} \text{Tr} \left( \begin{bmatrix} \delta_{xx}^{\text{CSA}} & 0 & 0 \\ 0 & \delta_{yy}^{\text{CSA}} & 0 \\ 0 & 0 & \delta_{zz}^{\text{CSA}} \end{bmatrix} \begin{bmatrix} \vec{v}_x \cdot \vec{w}_x & \vec{v}_x \cdot \vec{w}_y & \vec{v}_x \cdot \vec{w}_z \\ \vec{v}_y \cdot \vec{w}_x & \vec{v}_y \cdot \vec{w}_y & \vec{v}_y \cdot \vec{w}_z \\ \vec{v}_z \cdot \vec{w}_x & \vec{v}_z \cdot \vec{w}_y & \vec{v}_z \cdot \vec{w}_z \end{bmatrix} \right) \quad (1.44)$$

$$\begin{bmatrix} \Delta\chi_{xx} & 0 & 0 \\ 0 & \Delta\chi_{yy} & 0 \\ 0 & 0 & \Delta\chi_{zz} \end{bmatrix} \begin{bmatrix} \vec{w}_x \cdot \vec{v}_x & \vec{w}_x \cdot \vec{v}_y & \vec{w}_x \cdot \vec{v}_z \\ \vec{w}_y \cdot \vec{v}_x & \vec{w}_y \cdot \vec{v}_y & \vec{w}_y \cdot \vec{v}_z \\ \vec{w}_z \cdot \vec{v}_x & \vec{w}_z \cdot \vec{v}_y & \vec{w}_z \cdot \vec{v}_z \end{bmatrix} \right) \\ = \frac{B_0^2}{15\mu_0 k_B T} \text{Tr} \left( \begin{bmatrix} \delta_{xx}^{\text{CSA}} & 0 & 0 \\ 0 & \delta_{yy}^{\text{CSA}} & 0 \\ 0 & 0 & \delta_{zz}^{\text{CSA}} \end{bmatrix} \begin{bmatrix} \vec{v}_x \\ \vec{v}_y \\ \vec{v}_z \end{bmatrix} \begin{bmatrix} \vec{w}_x & \vec{w}_y & \vec{w}_z \end{bmatrix} \right) \quad (1.45)$$

$$\begin{bmatrix} \Delta\chi_{xx} & 0 & 0 \\ 0 & \Delta\chi_{yy} & 0 \\ 0 & 0 & \Delta\chi_{zz} \end{bmatrix} \begin{bmatrix} \vec{w}_x \\ \vec{w}_y \\ \vec{w}_z \end{bmatrix} \begin{bmatrix} \vec{v}_x & \vec{v}_y & \vec{v}_z \end{bmatrix} \right) \\ = \frac{B_0^2}{15\mu_0 k_B T} \text{Tr} \left( \begin{bmatrix} \delta_{xx}^{\text{CSA}} & 0 & 0 \\ 0 & \delta_{yy}^{\text{CSA}} & 0 \\ 0 & 0 & \delta_{zz}^{\text{CSA}} \end{bmatrix} R_{\delta^{\text{CSA}}}^T R_{\Delta\chi} \begin{bmatrix} \Delta\chi_{xx} & 0 & 0 \\ 0 & \Delta\chi_{yy} & 0 \\ 0 & 0 & \Delta\chi_{zz} \end{bmatrix} R_{\Delta\chi}^T R_{\delta^{\text{CSA}}} \right) \quad (1.46)$$

$$= \frac{B_0^2}{15\mu_0 k_B T} \text{Tr} \left( R_{\delta^{\text{CSA}}} \begin{bmatrix} \delta_{xx}^{\text{CSA}} & 0 & 0 \\ 0 & \delta_{yy}^{\text{CSA}} & 0 \\ 0 & 0 & \delta_{zz}^{\text{CSA}} \end{bmatrix} R_{\delta^{\text{CSA}}}^T R_{\Delta\chi} \begin{bmatrix} \Delta\chi_{xx} & 0 & 0 \\ 0 & \Delta\chi_{yy} & 0 \\ 0 & 0 & \Delta\chi_{zz} \end{bmatrix} R_{\Delta\chi}^T \right) \quad (1.47)$$

$$= \frac{B_0^2}{15\mu_0 k_B T} \text{Tr} [\delta^{\text{CSA}} \cdot \Delta\chi] \quad (1.48)$$

$$= \frac{B_0^2}{15\mu_0 k_B T} \sum_{i,j \in [x,y,z]} \delta_{ij}^{\text{CSA}} \Delta\chi_{ji} \quad (1.49)$$

Clearly equation 1.49 should be preferred for its numerical simplicity. Optimisation of tensor parameters to fit experimental data is often achieved by non-linear regression methods. Such algorithms require thousands of evaluations of the calculated data and therefore speed

of computation is essential. Appendix A reports similar optimised calculations for additional cross-correlation effects in paramagnetic NMR theory.

## 1.5 Python software for paramagnetic NMR

Python is a popular software for the scientific community because of its independence from hardware level operations. Python is an object-oriented programming language which means that code may be compartmentalised into structures called *objects*. Objects can possess data within variables called *attributes* as well as complete tasks through the use of functions called *methods*. Such a program structure enables code which is more human-readable and less prone to errors.

An object structure is naturally adapted to the world of paramagnetic protein NMR, where the nuclear spin and the paramagnetic centre can be defined as objects. An overview of some reasonable attributes and methods is given in Figure 1.3. A method like Calculate PCS can take arguments, such as the coordinates of the nuclear spin, to evaluate at the given position. Methods can access the object's attributes directly, meaning the user does not have to provide arguments for the  $\chi$  tensor which are available directly from the attribute of the current object.

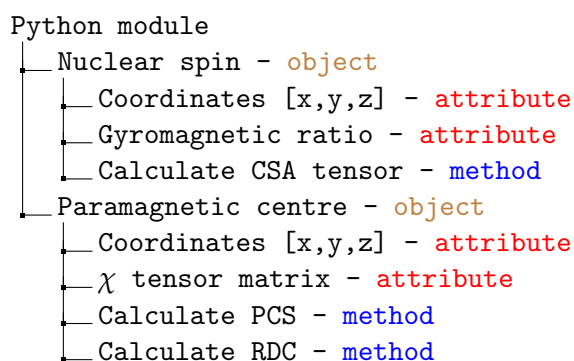


Figure 1.3: Tree diagram of python object structure for a nuclear spin and paramagnetic centre

Given a protein structure in the presence of a paramagnetic centre, it can be useful to calculate the paramagnetic effects to aid interpretation of NMR spectra. It can also be useful to fit the parameters of the paramagnetic centre (such as the  $\chi$  tensor) using experimentally measured paramagnetic values. There exist many programs capable of these calculations whose individual shortcomings are discussed in detail in Appendix A (John et al., 2005; Schmitz et al., 2008; Rinaldelli et al., 2015; Zweckstetter and Bax, 2000; Valafar and Prestegard, 2004; Bieri et al., 2011). The associated publication reports the development of the new software 'Paramagpy',

written in Python, which combines the functionality of previous programs and aims to unify all paramagnetic NMR phenomena. Paramagpy has a Python object to define the paramagnetic centre which is common to all PCS, RDC, PRE and CCR calculations, with associated methods for calculation and fitting. Consequently, there is a seamless work flow between fitting to experimental values and back-calculation of any paramagnetic NMR effects. For example, a  $\Delta\chi$  tensor could be fitted using experimental PCS data and then immediately used for calculation of the expected RDC values for  $^1\text{H}$ - $^{15}\text{N}$  spin pairs.

Python is often described as a computationally slow language, however, it is a simple task to cast low-level repetitive operations (such as the core mathematics of paramagnetic calculations) to a more suitable programming language like C++. Paramagpy uses the numerical library Numpy to achieve this, and has algorithms that incorporate the matrix formalism defined in Section 1.4 for rapid parameter optimisation (van der Walt et al., 2011). Paramagpy principally uses non-linear regression to fit parameters of the  $\chi$  tensor to experimental data, but can also perform singular value decomposition (SVD) for linear systems such as fitting a spatially constrained  $\Delta\chi$  tensor to PCSs and RDCs.

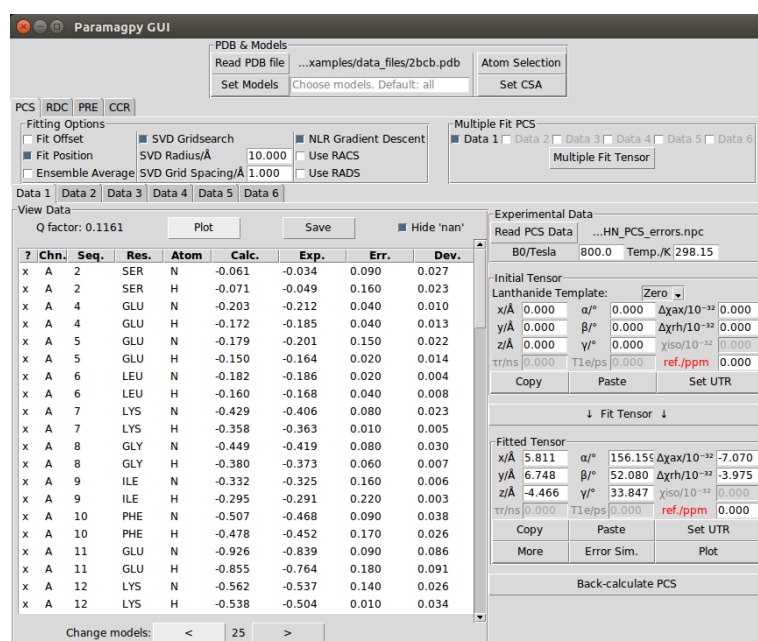


Figure 1.4: Paramagpy GUI running on Ubuntu 16.04 with PCS tab displayed

Although many powerful programs are written for a scripted environment, most rely on a graphical user interface (GUI) to provide intuitive accessibility to a broader audience. The GUI for Paramagpy was written for the Tcl/Tk interface, which is a standard GUI tool kit available on Unix and Windows operating systems. The GUI is shown in Figure 1.4 which displays the



PCS tab of the program. The GUI has a similar layout among each of the PCS/RDC/PRE/CCR tabs which allows the same  $\chi$  tensor parameters to be used for calculating different paramagnetic phenomena via the *copy* and *paste* buttons.

Paramagpy is an open-source project and has extensive documentation to describe its functionality. The documentation was written into the source code using the `__doc__` object attribute and then compiled into web HTML using the python module *Sphinx*. This can be viewed from within the archived project: <https://doi.org/10.5281/zenodo.3594568>, and includes many example calculations with real experimental data.

Often interpretation of paramagnetic NMR spectra requires consideration of multiple PCS, RDC, PRE or CCR effects for a given sample. Even if the spectroscopist wishes only to measure paramagnetically induced RDCs, they will require knowledge of the PCS in order to correctly assign chemical shifts, and knowledge of the PREs to predict which signals will be broadened beyond detection. Paramagpy is a useful tool for such cases by adopting a comprehensive and modern programming approach.

## 1.6 Calbindin D<sub>9K</sub>: a model paramagnetic system

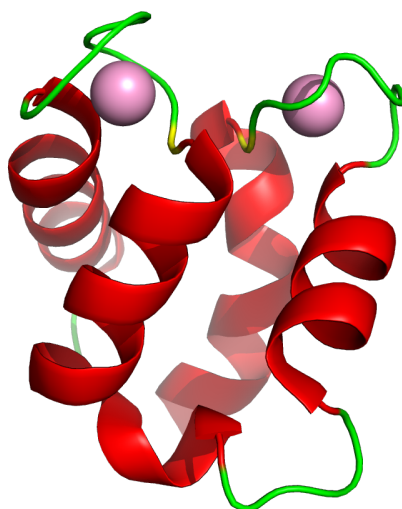


Figure 1.5: Calbindin D<sub>9K</sub> cartoon representation. The N- and C-terminal calcium binding sites are shown on the left and right respectively. Pink spheres represent bound metal ions.

Calbindin D<sub>9K</sub> is a calcium binding protein with two EF-hand domains that can each bind a single Ca<sup>2+</sup> ion (see Figure 1.6) (Chazin et al., 1989). The C-terminal binding site will pref-

erentially bind a lanthanide ion over  $\text{Ca}^{2+}$ . This makes it a model system for paramagnetic investigations because the metal position and orientation are fixed with respect to the molecular structure. There have been numerous investigations using calbindin for paramagnetic lanthanide studies, mostly by the Florence NMR group (Bertini et al., 2001b,a; Banci et al., 2004; Bertini et al., 2002a)

## 1.7 Measuring distances derived from PREs

In its simplest form,  $R_1$  can be measured by the inversion-recovery experiment where the magnetisation is inverted by a  $180^\circ$  pulse, after which its recovery towards equilibrium is described by Equation 1.50 of Figure 1.7. Similarly,  $R_2$  can be measured by a spin-echo sequence where the magnetisation decays to zero with Equation 1.51. By fitting the relevant equation to measured peak heights for given delay values  $\tau$ , the relaxation rate can be extracted.

To measure PREs experimentally it is necessary to subtract the relaxation measurement for a diamagnetic sample from that of a paramagnetic sample. Often a diamagnetic metal ion such as  $\text{Y}^{3+}$  or  $\text{Lu}^{3+}$  is substituted for the reference spectrum (Otting, 2010). Measurement of the relaxation rate by one-dimensional spectra is not generally practical for protein complexes due to overlapped peaks. Therefore, pseudo-3D spectra are often used, which generally encode a temporal dimension determined by the relaxation delay of a 2D HSQC experiment.

In this thesis, pseudo-3D spectra were analysed using the program CCPNMR analysis (Vranken et al., 2005) for which a Python macro was written to extract peak amplitudes in each relaxation plane of the experiment. Fitting of the exponential equations for relaxation was achieved with a custom Python script. Theoretical PREs were calculated using the software Paramagpy (see section 1.5). Such a work flow ensured expedite accuracy in repeated processing of such spectra.

Appendix B reports the development of new methods for the accurate measurement of electron-nuclear distances derived from PREs as demonstrated for the protein calbindin  $\text{D}_{9k}$ . The publication highlights the importance of considering additional sources to relaxation, including RDCs and intermolecular effects. The software Paramagpy was used for many of the calculations described in the publication.

$$I_z(\tau) = I_z^{\text{eq}} - (I_z^{\text{eq}} - I_z^0) e^{-R_1\tau} \quad (1.50)$$

$$I_{\pm}(\tau) = I_{\pm}^0 e^{-R_2\tau} \quad (1.51)$$

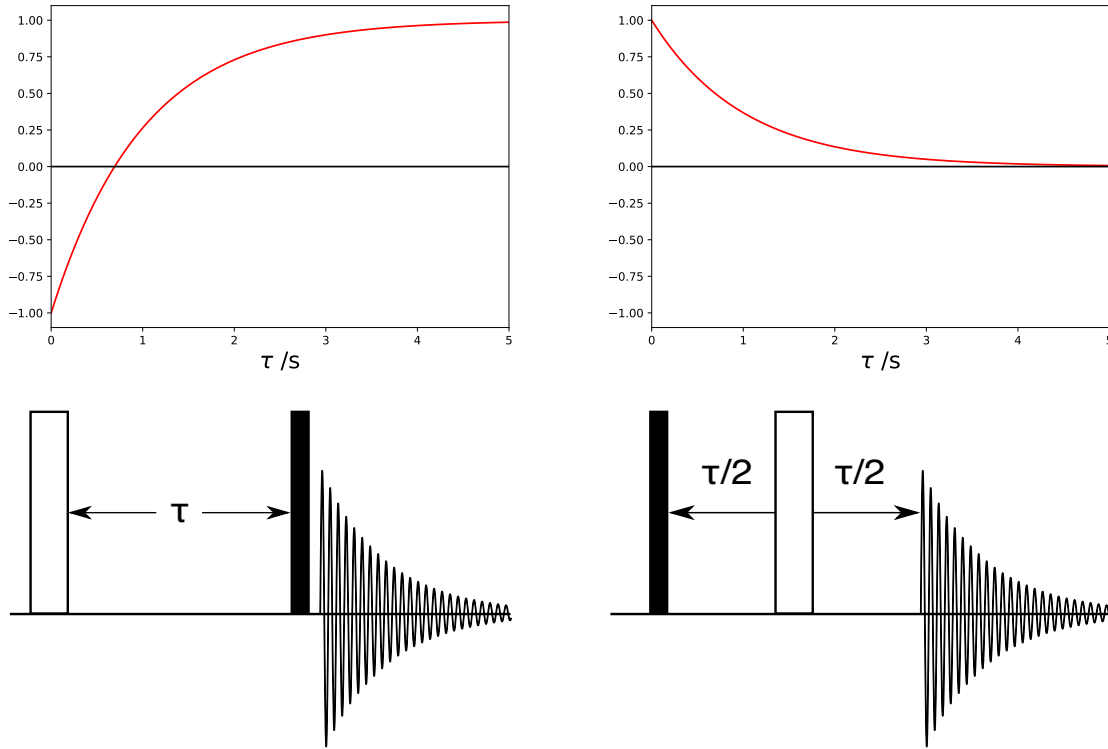


Figure 1.6:  $R_1$  and  $R_2$  relaxation depicted on the left and right respectively. The equation for magnetisation evolution, its graphical representation and the corresponding pulse sequence are depicted from top to bottom. In the pulse sequences, open bars represent 180° pulses, while filled bars represent 90° pulses. In the equations,  $I_z$  is the longitudinal magnetisation,  $I_z^{\text{eq}}$  is the equilibrium magnetisation,  $I_z^0$  is the initial magnetisation at  $\tau = 0$  and  $I_{\pm}$  is the transverse magnetisation.

## Chapter 2

# Protein NMR assignment in the solid state

Paramagnetic NMR offers methods for measuring distances in a useful range for protein structure determination, but the presented theory from Chapter 1 only applies to molecules or molecular complexes in solution. To access a broader range of biological samples, including membrane proteins, viral assemblies and amyloid fibrils, it is necessary to move to the much different realm of solid-state NMR.

Most nuclear spin interactions are anisotropic which means that their magnitude has an angular dependence to the externally applied magnetic field (Levitt, 2008). Although not usually seen in solution NMR spectra, this feature is clearly observed in the spectra of static unordered samples. Spectral phenomena such as chemical shift anisotropy (CSA) give rise to powder patterns and direct dipole-dipole (DD) couplings are characterised by Pake patterns (Duer, 2004). For NMR spectra of macromolecular solids, such broad features complicate interpretation of spectra and prevent resolved signals (Ramamoorthy, 2005).

Magic angle spinning (MAS) describes the process of physically rotating a sample about the angle  $\theta$  to the external magnetic field as defined by equation 2.1 (Andrew et al., 1958).  $\theta$  represents the angle subtended by the diagonal through a cube with its axes and leads to complete averaging of all rank 2 spin interactions. MAS improves sensitivity and resolution in solid-state NMR spectra allowing resonance assignment and structural studies of macromolecular complexes (Ramamoorthy, 2005).

$$\theta = \arccos \left[ \frac{1}{\sqrt{3}} \right] = 54.7356^\circ \quad (2.1)$$

## 2.1 Ultra-fast magic angle spinning

Until recently, solid-state NMR of protein molecules was limited to heteronuclear detection of  $^{13}\text{C}$  and  $^{15}\text{N}$  spins. This was because the technology allowed a maximum MAS rate of 40 kHz. At these spinning rates the CSA and DD-couplings for heteronuclear spins are efficiently removed, allowing the acquisition of resolved spectra (Laws et al., 2002). However, the limited sensitivity of low gyromagnetic ratio heteronuclei makes the acquisition of higher dimensional spectra for protein assignment infeasible.

In contrast, the relatively large gyromagnetic ratio of  $^1\text{H}$  permits the acquisition of higher dimensional NMR spectra and more information-rich NMR experiments. However, with this come relatively large  $^1\text{H}$ - $^1\text{H}$  DD couplings that require MAS rates beyond 100 kHz for resolved spectra (Struppe et al., 2017). It is only with the recent development of sub-millimetre diameter rotors that such spinning speeds have been unlocked, and with them, a suite of new  $^1\text{H}$ -detected NMR experiments (Andreas et al., 2016; Chevelkov et al., 2014).

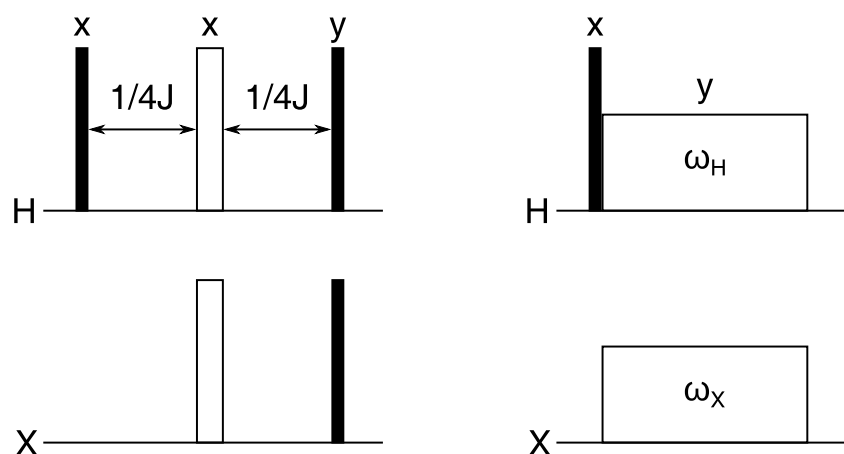


Figure 2.1: Coherence transfer pulse schemes for solution and solid state. Left: INEPT coherence transfer by J-coupling. Right: Cross-polarisation transfer by CPMAS. Example pulse phases are shown above the pulses. Filled bars represent  $90^\circ$  pulses and open bars represent  $180^\circ$  pulses.  $\omega_H$  and  $\omega_X$  denote continuous radio-frequency power levels and must fulfil the CPMAS matching condition.

Coherence transfer is an essential step in many NMR experiments and is used to correlate nuclei via their spin interactions. Standard solution NMR experiments for protein assignment use J-coupled coherence transfer steps like the famous INEPT sequence. Such scalar transfers

often suffer from relatively long periods during which transverse relaxation severely attenuates the NMR signal (Cavanagh et al., 2007). Consequently, NMR experiments for assignment of globular proteins in solution are usually limited in the number of magnetisation transfers before the experiment becomes very insensitive. In solid-state NMR, coherence transfer can be achieved using relatively short mixing times by exploiting DD couplings through the cross polarisation (CP) transfer (Pines et al., 1973). CPMAS stands for CP with MAS. It requires appropriate power levels to fulfil the matching condition (see Figure 2.1). Appendix C explores the limits of sequential CPMAS transfers with the development of a 5-dimensional solid-state NMR experiment for complete protein backbone assignment.

## 2.2 Sparse sampling with automated projection spectroscopy

Acquisition of NMR spectra with more than three dimensions often requires sparse sampling techniques in order to reduce the experiment time while retaining desirable resolution. The most popular method for sparse sampling is non-uniform (NUS) which involves acquiring a subset of the points in the indirect time domain and filling the missing values with a reconstruction algorithm (Maciejewski et al., 2012).

Another method for sparse sampling is projection spectroscopy which convolves several dimensions by simultaneously incrementing all indirect time domains. The choice of projection angle dictates the contribution from each indirect dimension from which information about chemical shifts can be obtained. By analysing the peak locations in several projections with the GAPRO algorithm, unique chemical shifts present in the fully sampled spectrum can be reconstructed in a process called automated projection spectroscopy (APSY) (Hiller et al., 2005). APSY is traditionally a solution NMR technique, however with the improved resolution of  $^1\text{H}$ -detected NMR spectra with ultra-fast MAS techniques, APSY is a viable technique for determining chemical shifts also in the solid state. Appendix C demonstrates APSY methods for sparse sampling of a 5-dimensional experiment for assignment of chemical shifts in three different solid protein samples.

## Chapter 3

# Solid state NMR at the ANU: protocols established for UFMAS of biosolids



Figure 3.1: Photo of solid-state NMR rotors. Upper: 0.5 mm diameter rotor. Middle: 1.2 mm diameter rotor. Lower: Australian 5 cent coin.

The recent advances in ultra-fast magic angle spinning (UFMAS) technology have opened the door to facile  $^1\text{H}$ -detected structural studies of biomolecular solids. This realm of NMR complements solution-state NMR studies which are usually limited to relatively small soluble biomolecules. This chapter describes efforts to implement hardware required for solid-state studies with UFMAS at the ANU and includes protocols for packing biosolid samples. Figure 3.1 depicts a photo of the rotors used for the experimental set-ups considered in this chap-

ter.

### 3.1 60 kHz MAS broad-band probe

The probe described in this section is a 60 kHz MAS probe produced by Phoenix NMR Colorado. The probe is a triple resonance (H/F-X-Y) probe with the high frequency channel tunable to 700 MHz  $^1\text{H}$  NMR frequency or the corresponding  $^{19}\text{F}$  NMR frequency. The X and Y channels are broad-band tunable between  $^{31}\text{P}$  and  $^{15}\text{N}$ . The rotor diameter used by the probe is 1.2 mm.

#### 3.1.1 Biosolid packing of 1.2 mm diameter rotors

A cross-section of the 1.2 mm diameter rotor is given in Figure 3.2. Note the presence of the spacer plugs shown in black. For powder samples, a polytetrafluoroethylene (PTFE) spacer is often used to limit the total sample to the active volume of the rotor. However, for biomolecular solids which are often a suspension of solid material in buffer, the liquid is likely to leak from the ends of the rotor due to the hydrostatic pressure induced by UFMAS. Consequently, it is necessary to insert a fluoroelastomer (known as FKM) plug to seal the sample and prevent drying at maximum spinning speeds.

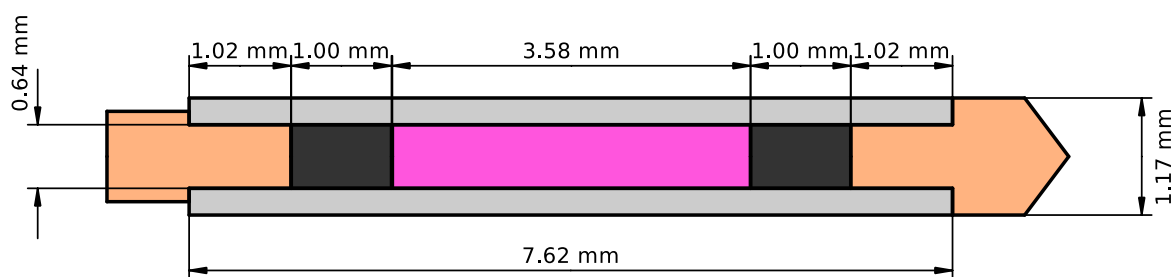


Figure 3.2: 1.2 mm diameter rotor cross-section and dimensions. Grey: Ceramic rotor body. Brown: End-cap and turbine from left to right respectively. Black: Spacer plugs. Pink: Sample volume.

The FKM-rubber plugs can be cut from a 1.0 mm rubber sheet using an appropriate leather hole punch tool. To fabricate such a hole punch, a 0.9 mm hypodermic needle (yellow colour) was modified. Initially, the sharp end was ground flat, then sharpened to give a cutting edge. Finally the bur was removed from the lumen using a 0.6 mm drill bit. A gentle pressure is applied to cut the plug with a twisting motion. Note that if excessive pressure is applied, a



plug with a steep conical shape will be formed which is not desirable. The plug was expelled from the cutting tip by inserting a 0.6 mm drill bit down the lumen of the hole punch.

After insertion of a single plug at one end of an empty rotor the sample can be packed into the rotor. For packing of biosolid samples a device for use with an ultracentrifuge was designed and fabricated by the workshop at the ANU (see Figure 3.3). The apparatus is screwed together with the rotor located within the central collar. The sample suspended in buffer is loaded into the funnel at the top and the entire device is placed into the bucket of the ultra-centrifuge. Micro-crystalline samples are typically centrifuged at 150 000 g for 2 hours.

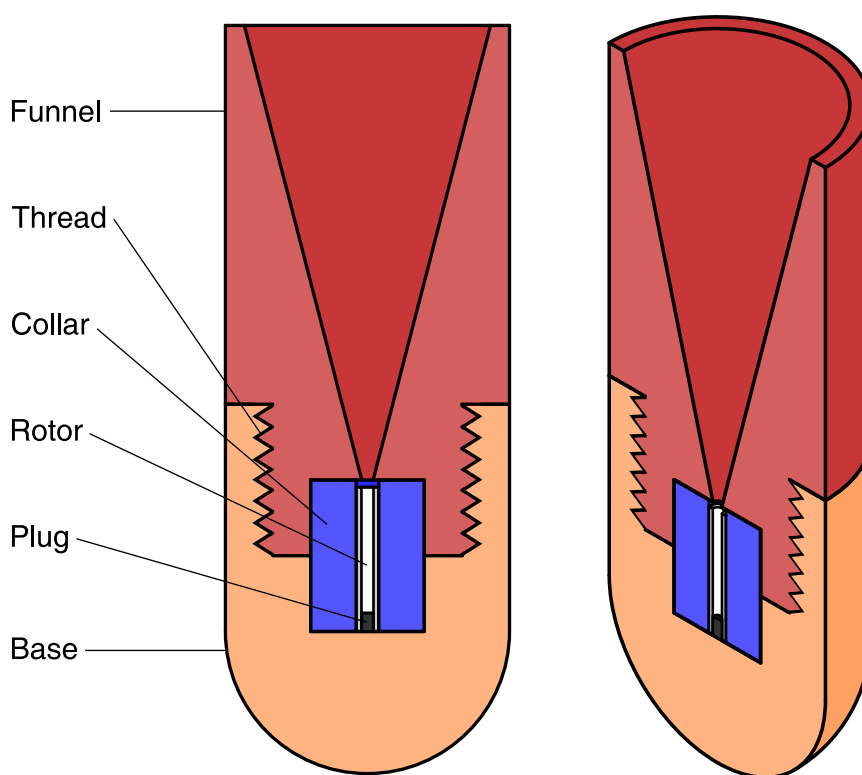


Figure 3.3: Cross-section of biosolids packing apparatus for 1.2 mm diameter rotors. The funnel and base were fabricated from Delrin plastic. The collar is polyether ether ketone (PEEK) plastic. The rotor is depicted with a FKM rubber plug installed.

Once the sample is packed into the rotor, it is necessary to remove excess sample so that enough interior space is available for all components of Figure 3.2. A depth gauge tool was fabricated for this purpose. After inserting the second FKM-plug, it is advisable to rinse the interior end-spaces with water using a fine needle. This removes any protein residue and prevents the cap and turbine from becoming stuck in the rotor. Finally the rotor is closed by

inserting the cap and turbine following standard methods outlined in the Phoenix NMR manual. Note that the sample volume here is slightly larger than the active volume recommended if PTFE spacers are used. This is because the spacer adjacent to the turbine is usually 1.4 mm in length. For opening the rotor, the liquid nitrogen method as described in the Phoenix NMR user manual was most successful.

### 3.1.2 Probe change protocol

The standard operation of the 700 MHz NMR spectrometer is for solution experiments. For changing the set-up to solids experiments, the following protocol should be followed

#### Protocol 1: Solution to solids probe change

1. Ensure no acquisition is running
2. Execute **sx ej** to eject sample
3. Switch off the sample changer
4. Set BCU target power to **off** and VT target temperature to 20.0 °C
5. Wait 20 min for temperature to start rising with 0 % heater power
6. Switch the VTU state to **off**
7. Disconnect all probe connections and remove the probe
8. Cover the magnet bore and place probe in box
9. Insert screws into shim stack first, then engage solids probe with the magnet
10. Adjust probe orientation to match silver markings so that the rotor is aligned with the  $x$ -axis of the shim stack and tighten screws.
11. Insert the solids probe and attach all connections
12. Insert the VT gas exhaust chimney from the top of the magnet
13. Execute **edhead** command and select **phoenix 1.2** to set as current
14. Execute **bsmsdisp** command, turn off the lock and disable lock sweep under 'lock/level' tab
15. Execute **rsh** and select recent shims for solids
16. Execute **edasp** and ensure probe connections are valid

17. On the MAS controller, select **Module**→**Open**→**PNMR 1.2mm Default**, then **Module**→**Send Profile**

The following protocol is used to change the solids probe back to the solution probe.

#### Protocol 2: Solids to solution probe change

1. Ensure no acquisition is running and stop spinning via MAS controller
2. Set BCU target power to **off** and VT target temperature to 20.0 °C
3. Wait 20 min for temperature to start rising with 0 % heater power
4. Switch the VTU state to **off**
5. Remove the VT gas exhaust chimney from the top of the magnet
6. Disconnect all probe connections
7. Lower the probe and then detach from magnet
8. Insert the solution probe and attach all connections
9. Plug in the sample changer
10. Execute **edhead** and ensure solutions probe is recognised
11. Execute **rsh** and select recent shims for solution
12. Switch the VTU state to **on**
13. Set BCU target power to **low** and VT target temperature to 25.0 °C. This may take 20 min to stabilise
14. Execute **sx N** where N is number for the dummy sample

## 3.2 150 kHz MAS probe

The probe described in this section is a 150 kHz MAS probe produced by Ago Samoson. The probe is triple resonance (H–C–N) with the high frequency channel tunable to 800 MHz <sup>1</sup>H NMR frequency. The rotor diameter used by the probe is 0.5 mm.

### 3.2.1 Biosolid packing of 0.5 mm diameter rotors

Packing of the 0.5 mm rotors requires very careful handling methods as it can easily be lost by the user. There are a number of key differences in the packing methods when compared

to those previously outlined in section 3.1.1. Figure 3.4 shows the cross-section of the 0.5 mm rotors where it can be seen that there are no plug seals and the end cap is flush with the ceramic rotor.

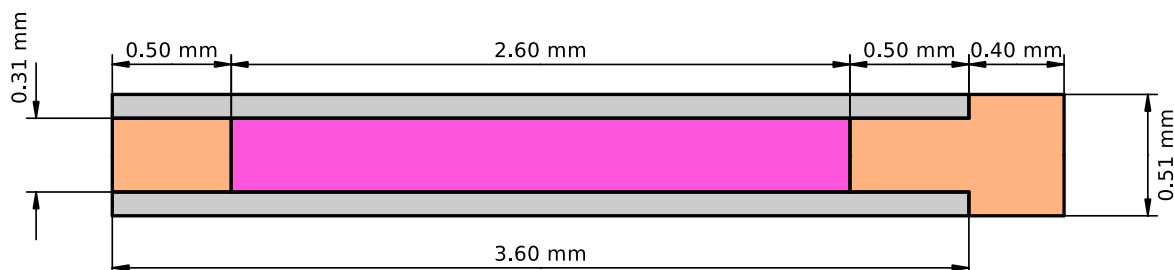


Figure 3.4: 1.2 mm diameter rotor cross-section and dimensions. Grey: Ceramic rotor body. Brown: End-cap and turbine from left to right respectively. Pink: Sample volume.

It is advisable to never remove the turbine from the body of the rotor. Consequently, a similar packing apparatus as shown in Figure 3.3 is used to load the sample. The rotor is placed in the device with the turbine facing down. After packing the sample, an end cap is inserted into the rotor and trimmed to a shorter length by using a guillotine jig. The additional end cap material is then ground flush with the rotor using sand paper.

For unloading the rotor, a small drill with a 0.35 mm diameter drill bit is used to remove the end cap. After breaching the plastic plug, the sample is removed by drilling by hand until the base of the turbine is felt. The rotor can be cleaned with the turbine in place by sonication in an appropriate solvent such as water or ethanol.

# Conclusion and future directions

This thesis has investigated new methods for characterising protein structures in solution and solid state. Paramagnetic NMR theory was refined and implemented in a new software tool. This tool was then used to investigate the practical aspects of deriving distances from paramagnetic relaxation enhancements. To increase the scope of proteins that can be studied by NMR, a new 5-dimensional experiment for complete backbone assignment of proteins was developed. Finally, protocols for UFMAS sample packing were implemented for the specific hardware set up at the ANU.

Paramagnetic NMR phenomena are useful effects for investigation of structure in lanthanide-protein complexes. The first part of Chapter 1 developed a matrix formalism to support rapid calculations of paramagnetic NMR effects. This theoretical framework was then implemented in a software tool written in Python. The software has a GUI and supports a seamless flow for calculations between paramagnetic phenomena. This allows fitting of parameters of the magnetic susceptibility tensor to experimental data such as PCSs. Next, the fitted magnetic susceptibility tensor can immediately be used for calculation of a different NMR phenomenon such as RDC or PREs. Future development of the code could aim at improving the convention for reporting magnetic susceptibility tensors. The popular Euler angles which are used to report the tensor orientation suffer from non-uniqueness and gimbal lock. These artefacts reduce the performance of gradient-descent algorithms used during optimisation and lead to confusion in communicating results. An improved description for orientation could use a quaternion definition which overcomes these issues. The Euler angle formalism represents three sequential rotations whereas a quaternion represents a single rotation about some three-dimensional axis. The quaternion formalism has 3 parameters to characterise the rotation axis and each are always independent, unlike Euler angles. However, the magnetic susceptibility is a second rank tensor and therefore contains inherent symmetries with three mirror planes and three two-fold rotations. These symmetries give rise to non-unique descriptions for orientation when com-

pared to an asymmetric object. Consequently only a subspace of the quaternions is required to describe all tensor orientations. An appropriate formalism should identify the smallest subspace and its transformations to and from the common Euler angle representations.

The second section of Chapter 1 investigated the practical aspects of deriving distances from paramagnetic relaxation enhancements. PREs were measured for the model protein calbindin  $D_{9k}$  where it was found that experimentally measured values were much larger than those predicted by theory. The origins of this discrepancy were found in the presence of RDCs and intermolecular PREs which were efficiently controlled through a new experimental method. The contributions from RDCs were suppressed by using short sampling schemes with the lanthanide ion  $Er^{3+}$ . Intermolecular PREs were efficiently removed by mixing the diamagnetic reference with the paramagnetic species. Consequently, accurate electron-nuclear distances were derived in the range 10–25 Å with an RMSD of 0.9 Å. The molecular complex investigated in the present study contains a lanthanide binding site which is fixed with respect to the molecule. Future studies could investigate the measurement of PREs in molecules with paramagnetic tags. These are often flexible. A changing metal position obviously introduces uncertainty in the measured distances, as well as changing the dynamics of spin relaxation. The accuracy of correlation between experimentally measured and predicted PREs warrants further investigation in these more realistic systems.

Chapter 2 developed a new NMR pulse sequence with 5 dimensions for complete protein backbone assignment in the solid state. APSY was used as a sparse sampling technique where the 5-dimensional peaks were reconstructed from 2-dimensional projections. The experiment was demonstrated for two micro-crystalline proteins and one amyloid fibril sample. Automated peak assignment was employed using the software FLYA and excellent agreement between published assignments was obtained. This new experiment demonstrates how modern techniques combined with  $^1H$ -detected UFMAS NMR allow facile assignment of proteins with very little labour required by the user. Future investigations could add an additional dimension to the experiment to allow  $C^\beta$  evolution during the scalar transfer step. The  $C^\beta$  chemical shift contains valuable information for determining the amino acid type and can be added to the experiment with little cost to the sensitivity. Currently, the APSY method suffers from poor resolution in the 2-dimensional projections. Future experiments could investigate the acquisition of 3-dimensional projections to allow more peaks to be identified with fewer projections.

The final chapter of this thesis recounts the establishment of UFMAS hardware at the ANU with some useful information for the NMR spectroscopist. The fabrication of new tools and

development of protocols for packing of solid state samples into millimetre-scale rotors is described.

# Bibliography

Abragam, A. (1961) *The principles of nuclear magnetism*. 32. Oxford university press.

Andreas, L. B., Jaudzems, K., Stanek, J., Lalli, D., Bertarello, A., Le Marchand, T., Calade Paepe, D., Kotlovica, S., Akopjana, I., Knott, B., Wegner, S., Engelke, F., Lesage, A., Emsley, L., Tars, K., Herrmann, T., and Pintacuda, G. (2016) Structure of fully protonated proteins by proton-detected magic-angle spinning NMR. *PNAS*, 113:9187–9192. doi:10.1073/pnas.1602248113.

Andrew, E. R., Bradbury, A., and Eades, R. G. (1958) Nuclear magnetic resonance spectra from a crystal rotated at high speed. *Nature*, 182:1659–1659. doi:10.1038/1821659a0.

Banci, L., Bertini, I., Cavallaro, G., Giachetti, A., Luchinat, C., and Parigi, G. (2004) Paramagnetism-based restraints for Xplor-NIH. *J. Biomol. NMR*, 28:249–261. ISSN 0925-2738. doi:10.1023/B:JNMR.0000013703.30623.f7.

Bertini, I., Cavallaro, G., Cosenza, M., Kümmerle, R., Piccioli, M., and Poggi, L. (2002a) Cross correlation rates between Curie spin and dipole-dipole relaxation in paramagnetic proteins: The case of Cerium substituted Calbindin D<sub>9k</sub>. *J. Biomol. NMR*, 23:11. doi:10.1023/A:1016341507527.

Bertini, I., Donaire, A., Jiménez, B., Luchinat, C., Parigi, G., Piccioli, M., and Poggi, L. (2001a) Paramagnetism-based versus classical constraints: an analysis of the solution structure of Ca Ln calbindin D<sub>9k</sub>. *J. Biomol. NMR*, 21:85–98.

Bertini, I., Janik, M. B. L., Lee, Y.-M., Luchinat, C., and Rosato, A. (2001b) Magnetic susceptibility tensor anisotropies for a lanthanide ion series in a fixed protein matrix. *J. Am. Chem. Soc.*, 123:4181–4188. doi:10.1021/ja0028626.

Bertini, I., Kowalewski, J., Luchinat, C., and Parigi, G. (2001c) Cross correlation between the



- dipole–dipole interaction and the Curie spin relaxation: The effect of anisotropic magnetic susceptibility. *J. Magn. Reson.*, 152:103–108. ISSN 10907807. doi:10.1006/jmre.2001.2378.
- Bertini, I., Luchinat, C., and Parigi, G. (2002b) Magnetic susceptibility in paramagnetic NMR. *Prog. Nucl. Mag. Res. Sp.*, 40:249–273. doi:10.1016/S0079-6565(02)00002-X.
- Bieri, M., d’Auvergne, E. J., and Gooley, P. R. (2011) relaxGUI: a new software for fast and simple NMR relaxation data analysis and calculation of ps-ns and  $\mu$ s motion of proteins. *J. Biomol. NMR*, 50:147–155. doi:10.1007/s10858-011-9509-1.
- Bleaney, B. I. and Bleaney, B. (1976) *Electricity and Magnetism*. Oxford University Press.
- Cavanagh, J., Fairbrother, W. J., Palmer III, A. G., Rance, M., and Skelton, N. J. (2007) *Protein NMR spectroscopy: principles and practice*. Academic Press, Amsterdam ; Boston, 2 edition. ISBN 978-0-12-164491-8.
- Chazin, W. J., Koerdel, J., Thulin, E., Hofmann, T., Drakenberg, T., and Forsen, S. (1989) Identification of an isoaspartyl linkage formed upon deamidation of bovine calbindin D<sub>9k</sub> and structural characterization by 2D proton NMR. *Biochemistry*, 28:8646–8653. doi:10.1021/bi00447a055.
- Chevelkov, V., Habenstein, B., Loquet, A., Giller, K., Becker, S., and Lange, A. (2014) Proton-detected MAS NMR experiments based on dipolar transfers for backbone assignment of highly deuterated proteins. *J. Magn. Reson.*, 242:180–188. doi:10.1016/j.jmr.2014.02.020.
- Duer, M. J. (2004) *Introduction to solid-state NMR spectroscopy*. Blackwell Oxford, Oxford.
- Gaponenko, V., Sarma, S. P., Altieri, A. S., Horita, D. A., Li, J., and Byrd, R. A. (2004) Improving the accuracy of NMR structures of large proteins using pseudocontact shifts as long-range restraints. *J. Biomol. NMR*, 28:205–212. ISSN 0925-2738. doi:10.1023/B:JNMR.0000013706.09264.36.
- Ghose, R. and Prestegard, J. H. (1997) Electron spin–nuclear spin cross-correlation effects on multiplet splittings in paramagnetic proteins. *J. Magn. Reson.*, 128:138–143. ISSN 10907807. doi:10.1006/jmre.1997.1227.
- Guéron, M. (1975) Nuclear relaxation in macromolecules by paramagnetic ions: a novel mechanism. *J. Magn. Reson.*, 19:58–66. doi:10.1016/0022-2364(75)90029-3.

- Güntert, P. (2003) Automated NMR protein structure calculation. *Prog. Nucl. Mag. Res. Sp.*, 43:105–125. doi:10.1016/S0079-6565(03)00021-9.
- Haasnoot, C., de Leeuw, F., and Altona, C. (1980) The relationship between proton-proton NMR coupling constants and substituent electronegativities—I: an empirical generalization of the karplus equation. *Tetrahedron*, 36:2783–2792. doi:10.1016/0040-4020(80)80155-4.
- Hiller, S., Fiorito, F., Wuthrich, K., and Wider, G. (2005) Automated projection spectroscopy (APSY). *PNAS*, 102:10876–10881. doi:10.1073/pnas.0504818102.
- John, M., Park, A. Y., Pintacuda, G., Dixon, N. E., and Otting, G. (2005) Weak alignment of paramagnetic proteins warrants correction for residual CSA effects in measurements of pseudocontact shifts. *J. Am. Chem. Soc.*, 127:17190–17191. doi:10.1021/ja0564259.
- Kurland, R. J. and McGarvey, B. R. (1970) Isotropic NMR shifts in transition metal complexes: The calculation of the fermi contact and pseudocontact terms. *J. Magn. Reson.*, 2:286–301. doi:10.1016/0022-2364(70)90100-9.
- Laws, D. D., Bitter, H.-M. L., and Jerschow, A. (2002) Solid-state NMR spectroscopic methods in chemistry. *Angew. Chem. Int. Ed.*, 41:3096–3129. doi:10.1002/1521-3773(20020902)41:17<3096::AID-ANIE3096>3.0.CO;2-X.
- Levitt, M. H. (2008) *Spin dynamics: basics of nuclear magnetic resonance*. John Wiley & Sons, Chichester, England ; Hoboken, NJ, 2nd ed edition. ISBN 978-0-470-51118-3 978-0-470-51117-6.
- Linser, R., Bardiaux, B., Higman, V., Fink, U., and Reif, B. (2011) Structure calculation from unambiguous long-range amide and methyl  $^1\text{H}$ - $^1\text{H}$  distance restraints for a microcrystalline protein with MAS solid-state NMR spectroscopy. *J. Am. Chem. Soc.*, 133:5905–5912. doi:10.1021/ja110222h.
- Maciejewski, M. W., Mobli, M., Schuyler, A. D., Stern, A. S., and Hoch, J. C. (2012) Data sampling in multidimensional NMR: fundamentals and strategies. In *Novel sampling approaches in higher dimensional NMR*, pages 49–77. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-27160-1. doi:10.1007/128\_2011\_185.
- Orton, H. W., Kuprov, I., Loh, C.-T., and Otting, G. (2016) Using paramagnetism to slow down nuclear relaxation in protein NMR. *J. Phys. Chem. Lett.*, 7:4815–4818. doi:10.1021/acs.jpclett.6b02417.

- Otting, G. (2010) Protein NMR using paramagnetic ions. *Annu. Rev. Biophys.*, 39:387–405. doi:10.1146/annurev.biophys.093008.131321.
- Pines, A., Gibby, M. G., and Waugh, J. S. (1973) Proton-enhanced NMR of dilute spins in solids. *J. Chem. Phys.*, 59:569–590. doi:10.1063/1.1680061.
- Pintacuda, G., Hohenthanner, K., Otting, G., and Müller, N. (2003) Angular dependence of dipole-dipole-Curie-spin cross-correlation effects in high-spin and low-spin paramagnetic myoglobin. *J. Biomol. NMR*, 27:18.
- Pintacuda, G., John, M., Su, X.-C., and Otting, G. (2007) NMR structure determination of protein-ligand complexes by lanthanide labeling. *Acc. Chem. Res.*, 40:206–212. ISSN 0001-4842, 1520-4898. doi:10.1021/ar050087z.
- Pintacuda, G. and Otting, G. (2002) Identification of protein surfaces by NMR measurements with a paramagnetic Gd(III) chelate. *J. Am. Chem. Soc.*, 124:372–373. ISSN 0002-7863, 1520-5126. doi:10.1021/ja016985h.
- Ramamoorthy, A. (2005) *NMR spectroscopy of biological solids*. CRC Press.
- Rinaldelli, M., Carlon, A., Ravera, E., Parigi, G., and Luchinat, C. (2015) FANTEN: a new web-based interface for the analysis of magnetic anisotropy-induced NMR data. *J. Biomol. NMR*, 61:21–34. doi:10.1007/s10858-014-9877-4.
- Sattler, M., Schleucher, J., and Griesinger, C. (1999) Heteronuclear multidimensional NMR experiments for the structure determination of proteins in solution employing pulsed field gradients. *Prog. Nucl. Mag. Res. Sp.*, 34:93–158. doi:10.1016/S0079-6565(98)00025-9.
- Saupe, A. and Englert, G. (1963) High-resolution nuclear magnetic resonance spectra of orientated molecules. *Phys. Rev. Lett.*, 11:462–464. doi:10.1103/PhysRevLett.11.462.
- Schmitz, C., Stanton-Cook, M. J., Su, X.-C., Otting, G., and Huber, T. (2008) Numbat: an interactive software tool for fitting  $\Delta\chi$ -tensors to molecular coordinates using pseudocontact shifts. *J. Biomol. NMR*, 41:179–189. doi:10.1007/s10858-008-9249-z.
- Schmitz, C., Vernon, R., Otting, G., Baker, D., and Huber, T. (2012) Protein structure determination from pseudocontact shifts using ROSETTA. *J. Mol. Biol.*, 416:668–677. ISSN 00222836. doi:10.1016/j.jmb.2011.12.056.

- Schubeis, T., Le Marchand, T., Andreas, L. B., and Pintacuda, G. (2018)  $^1\text{H}$  magic-angle spinning NMR evolves as a powerful new tool for membrane proteins. *J. Magn. Reson.*, 287:140–152. doi:10.1016/j.jmr.2017.11.014.
- Shen, Y., Delaglio, F., Cornilescu, G., and Bax, A. (2009) TALOS+: a hybrid method for predicting protein backbone torsion angles from NMR chemical shifts. *J. Biomol. NMR*, 44:213–223. doi:10.1007/s10858-009-9333-z.
- Solomon, I. (1955) Relaxation processes in a system of two spins. *Phys. Rev.*, 99:559–565. doi:10.1103/PhysRev.99.559.
- Stanek, J., Andreas, L. B., Jaudzems, K., Cala, D., Lalli, D., Bertarello, A., Schubeis, T., Akopjana, I., Kotelovica, S., Tars, K., Pica, A., Leone, S., Picone, D., Xu, Z.-Q., Dixon, N. E., Martinez, D., Berbon, M., El Mammeri, N., Noubhani, A., Saupe, S., Habenstein, B., Loquet, A., and Pintacuda, G. (2016) NMR spectroscopic assignment of backbone and side-chain protons in fully protonated proteins: microcrystals, sedimented assemblies, and amyloid fibrils. *Angew. Chem. Int. Ed.*, 55:15504–15509. doi:10.1002/anie.201607084.
- Struppe, J., Quinn, C. M., Lu, M., Wang, M., Hou, G., Lu, X., Kraus, J., Andreas, L. B., Stanek, J., Lalli, D., Lesage, A., Pintacuda, G., Maas, W., Gronenborn, A. M., and Polenova, T. (2017) Expanding the horizons for structural analysis of fully protonated protein assemblies by NMR spectroscopy at MAS frequencies above 100 kHz. *Solid State Nucl.*, 87:117–125. doi:10.1016/j.ssnmr.2017.07.001.
- Styan, G. P. (1973) Hadamard products and multivariate statistical analysis. *Linear Algebra and its Applications*, 6:217 – 240. ISSN 0024-3795. doi:https://doi.org/10.1016/0024-3795(73)90023-2.
- Suturina, E. A., Mason, K., Geraldès, C. F. G. C., Chilton, N. F., Parker, D., and Kuprov, I. (2018) Lanthanide-induced relaxation anisotropy. *Phys. Chem. Chem. Phys.*, 20:17676–17686. ISSN 1463-9076, 1463-9084. doi:10.1039/C8CP01332B.
- Tjandra, N. and Bax, A. (1997) Measurement of dipolar contributions to  $1\text{JCH}$  splittings from magnetic-field dependence of J-modulation in two-dimensional NMR spectra. *Journal of Magnetic Resonance*, 124:512 – 515. ISSN 1090-7807. doi:https://doi.org/10.1006/jmre.1996.1088.

- Tolman, J. R., Flanagan, J. M., Kennedy, M. A., and Prestegard, J. H. (1995) Nuclear magnetic dipole interactions in field-oriented proteins: information for structure determination in solution. *Proceedings of the National Academy of Sciences*, 92:9279–9283. ISSN 0027-8424. doi:10.1073/pnas.92.20.9279.
- Valafar, H. and Prestegard, J. H. (2004) REDCAT: a residual dipolar coupling analysis tool. *J. Magn. Reson.*, 167:228–241. doi:10.1016/j.jmr.2003.12.012.
- van der Walt, S., Colbert, S. C., and Varoquaux, G. (2011) The NumPy array: a structure for efficient numerical computation. *Comput. Sci. Eng.*, 13:22–30. doi:10.1109/MCSE.2011.37.
- Vega, A. J. and Fiat, D. (1976) Nuclear relaxation processes of paramagnetic complexes the slow-motion case. *Mol. Phys.*, 31:347–355. doi:10.1080/00268977600100261.
- Vranken, W. F., Boucher, W., Stevens, T. J., Fogh, R. H., Pajon, A., Llinas, M., Ulrich, E. L., Markley, J. L., Ionides, J., and Laue, E. D. (2005) The CCPN data model for NMR spectroscopy: development of a software pipeline. *Proteins*, 59:687–696. ISSN 08873585. doi:10.1002/prot.20449.
- Wi, S., Sinha, N., and Hong, M. (2004) Long-range  $^1\text{H}$ - $^{19}\text{F}$  distance measurement in peptides by solid-state NMR. *J. Am. Chem. Soc.*, 126:12754–12755. doi:10.1021/ja0462732.
- Wüthrich, K. (1986) *NMR of Proteins and Nucleic Acids*. A Wiley-interscience publication. Wiley. ISBN 9780471828938.
- Zweckstetter, M. and Bax, A. (2000) Prediction of sterically induced alignment in a dilute liquid crystalline phase: aid to protein structure determination by NMR. *J. Am. Chem. Soc.*, 122:3791–3792. doi:10.1021/ja0000908.

## **Appendix A**

# **Paramagpy: Software for Fitting Magnetic Susceptibility Tensors Using Paramagnetic Effects Measured in NMR Spectra**



Australian  
National  
University

## Statement of Contribution

This thesis is submitted as a Thesis by Compilation in accordance with:

[https://policies.anu.edu.au/ppl/document/ANUP\\_003405](https://policies.anu.edu.au/ppl/document/ANUP_003405)

I declare that the research presented in this Thesis represents original work that I carried out during my candidature at the Australian National University, except for contributions to multi-author papers incorporated in the Thesis where my contributions are specified in this Statement of Contribution.

**Title:** Paramagpy: Software for fitting magnetic susceptibility tensors using paramagnetic effects measured in NMR spectra

**Authors:** Henry W. Orton, Thomas Huber and Gottfried Otting

**Publication Outlet:** Magnetic Resonance

**Current status of paper:** Published

**Contribution:** I contributed all aspects of this publication including writing the code, documentation and manuscript.

Henry William Orton

CANDIDATE

SIGNATURE

10/02/2020

DATE

Endorsed

Gottfried Otting

SENIOR AUTHOUR

SIGNATURE

10/02/2020

DATE

Gottfried Otting

PRIMARY SUPERVISOR

SIGNATURE

10/02/2020

DATE

Luke Connell

DELEGATED AUTHORITY

SIGNATURE

11/02/2020

DATE



# Paramagpy: software for fitting magnetic susceptibility tensors using paramagnetic effects measured in NMR spectra

Henry William Orton, Thomas Huber, and Gottfried Otting

Research School of Chemistry, Australian National University, Canberra, ACT 2601, Australia

**Correspondence:** Gottfried Otting ([gottfried.otting@anu.edu.au](mailto:gottfried.otting@anu.edu.au))

Received: 4 November 2019 – Discussion started: 13 November 2019

Revised: 29 December 2019 – Accepted: 20 January 2020 – Published: 14 February 2020

**Abstract.** Paramagnetic metal ions with fast-relaxing electrons generate pseudocontact shifts (PCSs), residual dipolar couplings (RDCs), paramagnetic relaxation enhancements (PREs) and cross-correlated relaxation (CCR) in the nuclear magnetic resonance (NMR) spectra of the molecules they bind to. These effects offer long-range structural information in molecules equipped with binding sites for such metal ions. Here we present the new open-source software Paramagpy, which has been written in Python 3 with a graphic user interface. Paramagpy combines the functionalities of different currently available programs to support the fitting of magnetic susceptibility tensors using PCS, RDC, PRE and CCR data and molecular coordinates in Protein Data Bank (PDB) format, including a convenient graphical user interface. Paramagpy uses efficient fitting algorithms to avoid local minima and supports corrections to back-calculated PCS and PRE data arising from cross-correlation effects with chemical shift tensors. The source code is available from <https://doi.org/10.5281/zenodo.3594568> (Orton, 2019).

## 1 Introduction

Paramagnetic metal ions with fast-relaxing electrons produce a number of spatially dependent effects in nuclear magnetic resonance (NMR) spectra of biomolecules which are useful for probing molecular structure and interactions. These effects arise from the magnetic susceptibility of unpaired electrons, which manifests in NMR spectra most notably as pseudocontact shifts (PCSs), paramagnetic relaxation enhancements (PRE) and residual dipolar couplings (RDCs), but also as cross-correlated relaxation (CCR) effects. PCSs and RDCs only arise when the magnetic susceptibility is anisotropic, which is the case for all trivalent paramagnetic lanthanide ions except  $\text{Gd}^{3+}$ .

A number of programs have been developed for fitting the parameters of magnetic susceptibility tensors,  $\chi$ , to atomic coordinates of biomolecules using the paramagnetic effects experimentally observed in NMR spectra. The program Numbat supports calculation and fitting of the magnetic susceptibility anisotropy tensor,  $\Delta\chi$ , from experimental PCS

data with corrections for residual anisotropic chemical shifts (RACSs) (John et al., 2005) in a convenient graphical user interface (GUI) (Schmitz et al., 2008). The Python module PyParaTools offers similar functionality to Numbat but in a scripting environment and adds methods for fitting  $\chi$  tensors and alignment tensors using PREs and RDCs, respectively (Stanton-Cook et al., 2014). The software FANTEN offers a convenient web-based GUI for fitting  $\Delta\chi$  and alignment tensors from PCS and RDC data sets, respectively (Rinaldelli et al., 2015).

RDCs arise not only from paramagnetism, but also in the presence of alignment media such as dilute liquid crystals. The programs PALES (Zweckstetter and Bax, 2000) and REDCAT (Valafar and Prestegard, 2004) fit alignment tensors to atomic coordinates using RDCs. The program Module can use RDCs to fit alignment tensors for molecular structure refinement (Dosset et al., 2001). PCS and RDC restraints have also been implemented in the software packages CYANA (Balayssac et al., 2006), XPLOR-NIH (Banci et al., 2004), Rosetta (Schmitz et al., 2012; Raman et al., 2010) and



HADDOCK (Dominguez et al., 2003; de Vries et al., 2010) for structure determination and refinement.

The coordinates of paramagnetic centres can also be determined from PREs, and suitable fitting programs include the programs RelaxGUI (Bieri et al., 2011) and Spinach (Hogben et al., 2011). CCR effects can occur between Curie-spin and dipole–dipole relaxation (Ghose and Prestegard, 1997) and also between Curie-spin and chemical shift anisotropy (CSA) relaxation (Pintacuda et al., 2004a). The former is observed as a difference in relaxation rates between the multiplet components of scalar coupled resonances (Ghose and Prestegard, 1997; Bertini et al., 2002a). The software FANTACROSS supports calculation of this CCR effect, but does not allow fitting of the  $\chi$  tensor position (Bertini et al., 2001b). The latter CCR effect was experimentally demonstrated only recently (Orton et al., 2016).

NMR spectra of biomolecules labelled with paramagnetic metal ions with fast electronic relaxation rates, as afforded by lanthanide tags, simultaneously display PCS, RDC, PRE and CCR effects in the same spectrum (Pintacuda et al., 2004b). Due to their common origin in the paramagnetism of the metal ion, all these effects are interrelated. For example, the  $\Delta\chi$  tensor determined from PCS measurements can, in principle, be used to predict RDCs, and RDCs arising from paramagnetic alignment allow predictions of some of the  $\Delta\chi$ -tensor parameters. The software PyParaTools offers convenient integration of all of these effects, but it lacks many refinements, such as the computation of RACS effects which may affect PCS measurements (John et al., 2005), explicit routines for calculating PREs based on Solomon–Bloembergen–Morgan (SBM) or Curie-spin relaxation theory including anisotropic effects arising from non-vanishing  $\Delta\chi$  tensors, calculation of cross-correlated Curie-spin–CSA PRE effects or Curie-spin and dipole–dipole CCR involving anisotropic  $\Delta\chi$  tensors, or anisotropic SBM (Suturina et al., 2018) calculations.

Here we present a new Python-based program, Paramagpy, which offers a graphical interface for fitting magnetic susceptibility tensors using PCS, RDC, PRE and CCR data and seamless transition between these calculations. The fitting routine of Paramagpy for determining  $\Delta\chi$  tensors from PCSs employs an efficient grid search algorithm as previously implemented in GPS-Rosetta (Schmitz et al., 2012). The algorithm is adept at overcoming the local minima problem that sometimes compromises the results obtained with Numbat and PyParaTools. Paramagpy uses both Curie-spin (Guéron, 1975) and Solomon–Bloembergen–Morgan (Solomon, 1955) theory to calculate PREs, and it includes cross-correlation effects with anisotropic chemical shift tensors (Pintacuda et al., 2004a), which have not been taken into account by any previous tensor-fitting software. Paramagpy can be installed as a Python module and scripted for efficient calculations, or run via an intuitive GUI.

Calculations using Paramagpy have been verified with data from previous publications. This includes fitting of

$\Delta\chi$  tensors to amide PCS data of lanthanide-loaded calbindin D<sub>9k</sub> and calculating PREs for amide <sup>1</sup>H spins (Orton and Otting, 2018). Paramagpy has also been used successfully to predict cross-correlated CSA–Curie-spin relaxation giving rise to negative PREs for amide <sup>15</sup>N spins (Orton et al., 2016). CCR calculations have been verified with data from high- and low-spin paramagnetic myoglobin (Pintacuda et al., 2003). Paramagpy has been shown to fit alignment tensors consistent with previous results for lanthanide-tagged ubiquitin (Pearce et al., 2017), but may also be applied to data sets arising from alignment media where Paramagpy reports alignment and Saupe tensors alongside  $\Delta\chi$  tensors. Paramagpy can thus be used with RDC data obtained by any means of weak molecular alignment in the magnetic field, substituting software like Module (Dosset et al., 2001).

## 2 Pseudocontact shifts

The magnetic susceptibility tensor  $\chi$  associated with a paramagnetic centre creates a dipolar shielding tensor  $\sigma$  at a given position  $\mathbf{r}$  and distance  $r$  from the paramagnetic centre as shown in Eq. (1), where  $\mathbf{I}_3$  is the  $3 \times 3$  identity matrix,  $\otimes$  denotes the Kronecker product and  $\cdot$  denotes the matrix multiplication.

$$\sigma = \frac{1}{4\pi} \left[ 3 \frac{\mathbf{r} \otimes \mathbf{r}^T}{r^5} - \frac{\mathbf{I}_3}{r^3} \right] \cdot \chi \quad (1)$$

$$= \frac{1}{4\pi r^5} \begin{bmatrix} (3x^2 - r^2) & 3xy & 3xz \\ 3xy & (3y^2 - r^2) & 3yz \\ 3xz & 3yz & (3z^2 - r^2) \end{bmatrix} \cdot \begin{bmatrix} \chi_{xx} & \chi_{xy} & \chi_{xz} \\ \chi_{xy} & \chi_{yy} & \chi_{yz} \\ \chi_{xz} & \chi_{yz} & \chi_{zz} \end{bmatrix} \quad (2)$$

The PCS is given by the trace of the shielding tensor as shown in the PCS Eq. (3). The  $\Delta\chi$  tensor is given by the traceless part of the  $\chi$  tensor. Considering only the  $\Delta\chi$  tensor, a linear form of the PCS equation can be obtained, which characterises the  $\Delta\chi$  tensor by five explicit parameters as shown in Eq. (4). Including the three position parameters represented by the coordinates of the metal centre ( $x, y, z$ ), solving the PCS equation requires determining eight parameters in total.

$$\delta^{\text{PCS}} = \frac{1}{3} \text{Tr}[\sigma] \quad (3)$$

$$= \frac{1}{4\pi r^5} [x^2 - z^2, \quad y^2 - z^2, \quad 2xy, \quad 2xz, \quad 2yz] \cdot \begin{bmatrix} \Delta\chi_{xx} \\ \Delta\chi_{yy} \\ \Delta\chi_{xy} \\ \Delta\chi_{xz} \\ \Delta\chi_{yz} \end{bmatrix} \quad (4)$$

## 2.1 Singular value decomposition (SVD) grid search

Equation (4) can be rewritten in matrix form to give Eq. (5), where  $\mathbf{b}$  is a column vector of length  $n$  of the calculated PCS values,  $\mathbf{x}$  is a column vector of length 5 of the  $\Delta\chi$ -tensor parameters and  $\mathbf{A}$  is a  $n \times 5$  matrix with rows defined by the row vector in Eq. (4) containing coordinate parameters.

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad (5)$$

$$\mathbf{x} = \mathbf{A}^+ \cdot \mathbf{b} \quad (6)$$

$$\mathbf{x} = (\mathbf{W} \cdot \mathbf{A})^+ \cdot (\mathbf{W} \cdot \mathbf{b}) \quad (7)$$

Populating vector  $\mathbf{b}$  with many experimental PCS values and the matrix  $\mathbf{A}$  with atomic coordinates from a molecule of known structure, the system is likely overdetermined and a least-squares solution for the  $\Delta\chi$ -tensor parameters  $\mathbf{x}$  can be obtained analytically by considering the singular values of the matrix  $\mathbf{A}$  and constructing the pseudo-inverse  $\mathbf{A}^+$ . This allows calculation of the best-fitting tensor at a given position by Eq. (6) (Schmitz et al., 2012). A weighted least-squares fit can be obtained using Eq. (7), where the square matrix  $\mathbf{W}$  contains the weights along the diagonal  $W_{ii} = 1/S_{\text{PCS},i}$ , which may be sourced from the experimental standard deviations  $S_{\text{PCS},i}$  of the  $i$ th spin.

Since this calculation is fast, a grid search over many positions of the paramagnetic centre is feasible, providing a robust initial guess prior to iterative refinement of the tensor position by non-linear gradient-descent methods. Paramagpy can evaluate 5000 grid points for 50 PCS values in under 1 s using a 2 GHz Intel i5 2016 processor of a typical laptop computer.

## 2.2 Non-linear gradient descent

When fitting of the position of the paramagnetic centre is required, the PCS equation becomes non-linear. A fit can be found iteratively by minimising the sum of squares of the differences between experimental and back-calculated PCS values. An efficient method for minimisation is by non-linear gradient descent. We chose the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm (Fletcher, 1988) for non-linear least-squares minimisation of the cost function in Eq. (8). Here,  $\text{PCS}_i^{\text{exp}}$  and  $\text{PCS}_i^{\text{cal}}$  are, respectively, the experimental and back-calculated PCSs for spin  $i$ , and  $S_{\text{PCS},i}$  is the experimental uncertainty in the PCS of spin  $i$ .

$$\text{cost} = \sum_i \frac{(\text{PCS}_i^{\text{cal}} - \text{PCS}_i^{\text{exp}})^2}{S_{\text{PCS},i}^2} \quad (8)$$

## 2.3 Multiple PCS data sets

Often there are multiple PCS data sets available for different metal ions bound at the same position, obtained from multiple samples prepared with different metal ions. A simultaneous fit of the common position is possible, independently

fitting the tensor magnitude and orientation for each data set, and can lead to a more accurate overall position of the paramagnetic centre. Paramagpy supports multiple data sets for simultaneous fitting of a common metal position by both the SVD grid-search and non-linear gradient-descent algorithms.

## 2.4 Corrections to PCS calculations

An anisotropic magnetic susceptibility causes alignment of the molecule in the external magnetic field. As molecular orientations are no longer sampled uniformly, shielding tensors may no longer average to their isotropic values. In this situation, the chemical shift actually observed in the paramagnetic sample contains contributions from residual anisotropic chemical shifts (RACSs) arising from non-zero averaging of the chemical shift anisotropy (CSA) tensor. Paramagpy supports PCS calculations that include RACS corrections (John et al., 2005). Paramagpy provides standard CSA tensors for amide  $^1\text{H}$  spins and backbone amide  $^{15}\text{N}$  and carbonyl  $^{13}\text{C}$  spins (Cornilescu and Bax, 2000). Customised CSA tensors may also be set for any of the nuclear spins.

In addition to the CSA tensor, there is also a dipolar shielding tensor  $\sigma$  at the site of a nuclear spin, which arises from the magnetic susceptibility of the paramagnetic centre. In analogy to the RACS effect, this can lead to a residual anisotropic dipolar shift (RADS), which is a small perturbation to the observed PCS in paramagnetic samples arising from molecular alignment (Bertini et al., 2002b). Paramagpy includes RADS as an option in the PCS calculation and  $\Delta\chi$ -tensor fitting routines.

Systematic errors in experimental PCS values can arise due to variations in the carrier frequency or calibration of the recorded NMR spectra of the diamagnetic and paramagnetic species. This offset can be included as a parameter during the fitting of  $\Delta\chi$  tensors, although doing so is meaningful only if a sufficient number of PCS data are available to avoid overfitting.

## 3 Residual dipolar couplings

An anisotropic magnetic susceptibility tensor induces a co-incident alignment tensor  $\mathbf{A}$ , giving rise to RDCs between nuclear spins. The alignment tensor can be found from the  $\Delta\chi$  tensor using Eq. (9), where  $B_0$  is the magnetic field,  $\mu_0$  the vacuum permeability,  $k_B$  the Boltzmann constant and  $T$  the temperature (Bertini et al., 2002b).

$$\mathbf{A} = \frac{B_0^2}{15\mu_0 k_B T} \Delta\chi \quad (9)$$

The RDC values can be calculated using Eq. (10), where  $\mathbf{r}_{AB}$  is the internuclear vector and  $r_{AB}$  the distance between the two nuclei  $A$  and  $B$  (Kramer et al., 2004). This can be expanded into the vector equation Eq. (11), where  $x$ ,  $y$  and  $z$

are the Cartesian coordinates of the internuclear vector  $\mathbf{r}_{AB}$ .

$$\text{RDC} = \frac{3\gamma_A\gamma_B\mu_0\hbar}{8\pi^2r_{AB}^5}\mathbf{r}_{AB}^T\mathbf{A}\cdot\mathbf{r}_{AB} \quad (10)$$

$$= \frac{3\gamma_A\gamma_B\mu_0\hbar}{8\pi^2r_{AB}^5}$$

$$[x^2 - z^2, \quad y^2 - z^2, \quad 2xy, \quad 2xz, \quad 2yz] \cdot \begin{bmatrix} \mathbf{A}_{xx} \\ \mathbf{A}_{yy} \\ \mathbf{A}_{xy} \\ \mathbf{A}_{xz} \\ \mathbf{A}_{yz} \end{bmatrix} \quad (11)$$

Unlike the PCS tensor, the RDC tensor does not require parameters for position and can therefore be described by five parameters for magnitude and orientation. Fitting can therefore be achieved by a linear least-squares fit.

### 3.1 SVD fitting algorithm

Paramagpy uses the SVD algorithm similar to the original implementation in the program REDCAT (Valafar and Prestegard, 2004). It is functionally the same as the algorithm applied to solving the PCS equation in Sect. 2.1. A  $n \times 5$  matrix with rows defined by the row vector in Eq. (11) containing coordinate parameters is constructed. From this, a pseudo-inverse matrix is calculated and applied to the experimental RDC values, yielding the best-fitting alignment tensor.

## 4 Paramagnetic relaxation enhancements

PREs describe the relaxation rates of longitudinal magnetisation,  $R_1 = 1/T_1$ , or transverse magnetisation,  $R_2 = 1/T_2$ , of nuclear spins, where  $T_1$  and  $T_2$  are the longitudinal and transverse relaxation times, respectively. For PREs of paramagnetic molecules in solution, the relaxation rates are governed by dipole–dipole interactions as described by the SBM equations or the shielding tensor anisotropy as described by the Curie-spin equations (Solomon, 1955; Guéron, 1975).

### 4.1 Solomon–Bloembergen–Morgan theory

The SBM equations for  $R_1$  and  $R_2$  are shown in Eqs. (12) and (13), respectively, where  $\gamma$  is the nuclear gyromagnetic ratio,  $r$  the distance of the nucleus from the paramagnetic centre, and  $\omega$  and  $\omega_S$  the nuclear and electronic Larmor frequencies, respectively.  $\tau_c$  is the correlation time calculated as  $1/\tau_c = 1/\tau_r + 1/T_{1e}$ , where  $\tau_r$  is the rotational correlation time of the molecule and  $T_{1e}$  is the electronic relaxation time.  $\mu_{\text{eff}}$  is the effective magnetic moment of the paramagnetic centre, which can be predicted from the Landé  $g$  factor, the Bohr magneton  $\mu_B$  and the total angular momentum quantum number  $J$  (Eq. 14).

$$R_1^{\text{SBM}} = \frac{2}{15} \left( \frac{\mu_0 \gamma \mu_{\text{eff}}}{4\pi r^3} \right)^2 \left[ \frac{3\tau_c}{1 + \tau_c^2 \omega^2} + \frac{7\tau_c}{1 + \tau_c^2 \omega_S^2} \right] \quad (12)$$

$$R_2^{\text{SBM}} = \frac{1}{15} \left( \frac{\mu_0 \gamma \mu_{\text{eff}}}{4\pi r^3} \right)^2 \left[ 4\tau_c + \frac{3\tau_c}{1 + \tau_c^2 \omega^2} + \frac{13\tau_c}{1 + \tau_c^2 \omega_S^2} \right] \quad (13)$$

$$\mu_{\text{eff}} = g\mu_B \sqrt{J(J+1)} \quad (14)$$

An extension to the SBM theory which accounts for anisotropy of the dipolar spectral density is described by Eqs. (15) and (16) where  $\mathbf{G}(\omega)$  describes the spectral power density tensor (Suturina et al., 2018).  $\hat{\mathbf{r}}$  is the unit vector from the paramagnetic centre to the nuclear spin. The spectral power density tensor usually cannot be derived theoretically, but is instead fitted to experimental data.

$$R_1^{\text{SBM-aniso}} = \frac{2}{3} \left( \frac{\mu_0 \gamma}{4\pi r^3} \right)^2 \text{Tr} \left[ (3\hat{\mathbf{r}} \otimes \hat{\mathbf{r}} - \mathbf{I}_3)^2 \cdot \mathbf{G}(\omega) \right] \quad (15)$$

$$R_2^{\text{SBM-aniso}} = \frac{1}{3} \left( \frac{\mu_0 \gamma}{4\pi r^3} \right)^2 \text{Tr} \left[ (3\hat{\mathbf{r}} \otimes \hat{\mathbf{r}} - \mathbf{I}_3)^2 \cdot (\mathbf{G}(0) + \mathbf{G}(\omega)) \right] \quad (16)$$

### 4.2 Curie-spin theory

Curie-spin relaxation is governed by the dipolar shielding tensor  $\sigma$  as calculated in Eq. (1), which must include the isotropic component of the  $\chi$  tensor,  $\chi_{\text{iso}}$ , which can be predicted using Eq. (17). The first invariant  $\Lambda$  and second invariant  $\Delta$  of the shielding tensor are calculated by Eqs. (18) and (19), where  $\sigma_{ij}$  denotes the  $i$ th and  $j$ th components of the shielding tensor  $\sigma$  (Suturina et al., 2018). This allows calculation of the  $R_1$  and  $R_2$  PREs by Eqs. (20) and (21), respectively. These equations account for anisotropy of the magnetic susceptibility, provided Eq. (1) is used to calculate  $\sigma$  (Vega and Fiat, 1976).

$$\chi_{\text{iso}} = \frac{\mu_0 \mu_{\text{eff}}^2}{3k_B T} \quad (17)$$

$$\Lambda^2 = (\sigma_{xy} - \sigma_{yx})^2 + (\sigma_{xz} - \sigma_{zx})^2 + (\sigma_{yz} - \sigma_{zy})^2 \quad (18)$$

$$\Delta^2 = \sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx}\sigma_{yy} - \sigma_{xx}\sigma_{zz} - \sigma_{yy}\sigma_{zz} + \frac{3}{4} [(\sigma_{xy} + \sigma_{yx})^2 + (\sigma_{xz} + \sigma_{zx})^2 + (\sigma_{yz} + \sigma_{zy})^2] \quad (19)$$

$$R_1^{\text{Curie}} = \frac{1}{2} \Lambda^2 \omega^2 \left[ \frac{\tau_r}{1 + 9\tau_r^2 \omega^2} \right] + \frac{2}{15} \Delta^2 \omega^2 \left[ \frac{\tau_r}{1 + \omega^2 \tau_r^2} \right] \quad (20)$$

$$R_2^{\text{Curie}} = \frac{1}{4} \Lambda^2 \omega^2 \left[ \frac{\tau_r}{1 + 9\tau_r^2 \omega^2} \right] + \frac{1}{45} \Delta^2 \omega^2 \left[ 4\tau_r + \frac{3\tau_r}{1 + \tau_r^2 \omega^2} \right] \quad (21)$$

When PREs due to Curie-spin relaxation are cross-correlated with CSA relaxation, the CSA tensor is added to the dipolar shielding tensor to give an effective shielding tensor  $\sigma_{\text{eff}}$ . The PRE including CSA cross-correlation  $R_{\text{Curie} \times \text{CSA}}$  is determined as the difference in relaxation rates in the paramagnetic and diamagnetic state as shown in Eq. (22). This can give rise to negative PREs as shown previously and confirmed by experiment (Pintacuda et al., 2004a; Orton et al., 2016).

$$R_{\text{Curie} \times \text{CSA}} = R^{\text{Curie}}(\sigma_{\text{eff}}) - R^{\text{Curie}}(\sigma_{\text{CSA}}) \quad (22)$$

### 4.3 Fitting algorithm

Paramagpy includes routines to calculate PREs and fit all parameters for each of the above relaxation theories, including cross-correlated relaxation with CSA effects. This is achieved by non-linear gradient descent to minimise the cost function of Eq. (23). Here,  $\text{PRE}_i^{\text{exp}}$  and  $\text{PRE}_i^{\text{cal}}$  are, respectively, the experimental and back-calculated PREs for spin  $i$ , and  $S_{\text{PRE},i}$  is the experimental uncertainty in the PRE of spin  $i$ . The user can choose to fit or constrain different parameters, such as the magnetic susceptibility or power spectral density tensor position, magnitude, correlation time  $\tau_c$ , etc. Parameter templates for lanthanide ions are also provided, based on tensor magnitudes and anisotropies previously reported for lanthanide complexes of calbindin D<sub>9k</sub> (Bertini et al., 2001a). These may be used to give a quick estimate of expected PRE values.

$$\text{cost} = \sum_i \frac{(\text{PRE}_i^{\text{cal}} - \text{PRE}_i^{\text{exp}})^2}{S_{\text{PRE},i}^2} \quad (23)$$

## 5 Curie-spin dipole–dipole cross-correlated relaxation

Interference of the internuclear dipole–dipole (DD) relaxation with Curie-spin relaxation provides a mechanism for differential relaxation rates of multiplet components by cross-correlated relaxation (CCR) (Ghose and Prestegard, 1997). This effect is readily observed and measured as the difference in the relaxation rate  $R_2$  of the two doublet components of an amide  $^1\text{H}$ – $^{15}\text{N}$  spin pair. In this case, the shielding tensor arising at the  $^1\text{H}$  spin due to the  $^{15}\text{N}$  dipole is given by Eq. (24), where  $\mathbf{r}_{\text{HN}}$  is the H–N bond vector,  $r_{\text{HN}}$  is the internuclear distance,  $\gamma_{\text{N}}$  is the gyromagnetic ratio of  $^{15}\text{N}$  and  $I = \frac{1}{2}$  is the spin of  $^{15}\text{N}$ . The factor of  $1/B_0$  is necessary to express the  $^{15}\text{N}$  shielding tensor in units of parts per million to match the units of the Curie-spin shielding tensor. The effective shielding tensor for the  $^1\text{H}$  spin due to both the Curie spin and the  $^{15}\text{N}$  dipole in either the up or down spin state is given by Eqs. (25) and (26), respectively. The relaxation rate  $R_2^{\text{Curie}}$  is then calculated using Eq. (21) for both the up and down effective shielding tensors  $\sigma_{\uparrow}$  and  $\sigma_{\downarrow}$ , and their difference is taken to represent the Curie  $\times$  DD differential line broadening  $R^{\text{Curie} \times \text{DD}}$ . In this way the auto-correlated relaxation mechanisms arising from the separate DD and Curie mechanisms are subtracted out, leaving the pure cross-correlated term. A derivation showing the equivalence of Eqs. (24)–(27) to those reported by Ghose and Prestegard (Ghose and Prestegard, 1997) is given in the Supplement.

$$\sigma_{\text{N}} = \frac{1}{B_0} \frac{\mu_0}{4\pi} \gamma_{\text{N}} \hbar I \left[ 3 \frac{\mathbf{r}_{\text{HN}} \otimes \mathbf{r}_{\text{HN}}^T}{r_{\text{HN}}^5} - \frac{\mathbf{I}_3}{r_{\text{HN}}^3} \right] \quad (24)$$

$$\sigma_{\uparrow} = \sigma + \sigma_{\text{N}} \quad (25)$$

$$\sigma_{\downarrow} = \sigma - \sigma_{\text{N}} \quad (26)$$

$$R^{\text{Curie} \times \text{DD}} = R^{\text{Curie}}(\sigma_{\uparrow}) - R^{\text{Curie}}(\sigma_{\downarrow}) \quad (27)$$

Paramagpy uses the above equations for all DD  $\times$  Curie relaxation calculations. By using Eq. (1) for calculating the Curie-spin shielding tensor  $\sigma$ , these equations also account for anisotropy of the magnetic susceptibility  $\chi$ . CCR values can be calculated between any two atoms in the specified Protein Data Bank (PDB) file. The calculations have been shown to agree with previous experimental CCR data on high- and low-spin myoglobin (Pintacuda et al., 2003).

### 5.1 Fitting algorithm

Paramagpy includes routines to fit all parameters of the  $\chi$  tensor, including position, magnitude and anisotropy, to experimentally measured CCR data. This is achieved by non-linear gradient descent to minimise the cost function of Eq. (28). Here,  $\text{CCR}_i^{\text{exp}}$  and  $\text{CCR}_i^{\text{cal}}$  are, respectively, the experimental and back-calculated CCRs for spin  $i$ , and  $S_{\text{CCR},i}$  is the experimental uncertainty in the CCR of spin  $i$ .

$$\text{cost} = \sum_i \frac{(\text{CCR}_i^{\text{cal}} - \text{CCR}_i^{\text{exp}})^2}{S_{\text{CCR},i}^2} \quad (28)$$

## 6 Uncertainty calculations

To judge the quality of a  $\Delta\chi$  or  $\chi$  tensor fitted using PCS, RDC, PRE or CCR data, Paramagpy offers three methods to test the robustness of the fit: structure-sourced, bootstrap and Monte Carlo. The structure-sourced method assumes that multiple models in a PDB file represent experimental uncertainty in the atomic coordinates as is common for NMR structures (see Sect. 7.1). In this approach, a tensor is fitted to each individual model and uncertainties in the fitted tensor parameters are reported. The alternative bootstrapping method repeats the fit many times, with each iteration randomly sampling a specified proportion of the data, and subsequently reports the standard deviation in the fitted tensor parameters. The Monte Carlo method repeats the fit using all the data, but each time adds noise to the experimental values. The noise is sourced from a uniform distribution that has been scaled by values provided by the user for each atom. These scaling values are ideally calculated from noise in the spectrum to reflect uncertainty in peak positions or amplitudes (Kontaxis et al., 2000). The standard deviations in the fitted tensor parameters are then reported.

## 7 Molecular structures with multiple models

### 7.1 Structures with uncertainties represented by a family of models

Biomolecular structures in the PDB, which have been determined by solution NMR, usually report experimental uncer-



tainty in the atomic coordinates by including multiple models, which individually fulfil the experimental restraints. The default behaviour of Paramagpy is to fit a magnetic susceptibility tensor to each model independently and then report an average of all these tensors. The tensor averaging is achieved by Eq. (29) where the summation runs over the tensors fitted to each of the  $n$  models. This ensures no errors are introduced by averaging prolate/oblate tensors with different principal axis definitions. All other parameters involved in the fit, such as origin of the tensor position, rotational correlation time or electronic relaxation time, are averaged in the conventional way. Note that the final result is sensitive to different relative orientations of the models.

$$\chi_{\text{average}} = \frac{1}{n} \sum_i^n \begin{bmatrix} (\chi_{xx})_i & (\chi_{xy})_i & (\chi_{xz})_i \\ (\chi_{xy})_i & (\chi_{yy})_i & (\chi_{yz})_i \\ (\chi_{xz})_i & (\chi_{yz})_i & (\chi_{zz})_i \end{bmatrix} \quad (29)$$

## 7.2 Structures represented by a conformational ensemble

Some coordinate sets in the PDB have been determined by molecular dynamics, where the ensemble of models deposited fulfils the experimental restraints better than each individual model. For this case, Paramagpy has the option for calculation of ensemble-averaged paramagnetic effects at all stages of calculations and fitting. Ensemble-averaged fitting presents a subtle but important difference compared to the multiple-model method described in Sect. 7.1 above. This is particularly noticeable for RDCs, where the ensemble average can be much smaller than the corresponding RDC of a single model, and therefore several models representing different bond orientations may be simultaneously required to fit an appropriate alignment tensor or  $\Delta\chi$  tensor.

The implementation of ensemble averaging in Paramagpy averages the paramagnetic values calculated for each atom in the different models, identifying the specific atoms by identical atom numbers in the PDB file. Custom ensemble averaging behaviour can be changed by the user in the scripted environment. In the implementations of the SVD algorithm, ensemble averaging involves summation of rows for common atoms of the matrix  $\mathbf{A}$  of Eq. (5) before calculation of the singular values. In the implementations of the non-linear gradient descent algorithm, the values calculated for the common atoms are averaged prior to calculating the sum of squares of differences. This is shown in Eq. (30) where  $a^{\text{cal}}$  and  $a^{\text{exp}}$  are the calculated and experimental PCS, RDC, PRE or CCR values, respectively. The index  $m$  is for atoms that are common between models, and the index  $i$  runs over all atoms in the structure.

$$\text{cost}_{\text{ensemble}} = \sum_i \frac{\left( \sum_m [a_{m,i}^{\text{cal}} - a_i^{\text{exp}}] \right)^2}{\sigma_{a(i)}^2} \quad (30)$$

## 7.3 Fitting tensor parameters to multimers

In the case of symmetric multimers composed of monomers with each containing a paramagnetic metal ion, the ensemble averaging feature of Paramagpy can be exploited to fit the  $\Delta\chi$  tensor associated with a given monomer. This is achieved simply by defining the monomeric units in the PDB file as models of the same structure and applying the ensemble averaging routine to fit the  $\Delta\chi$  tensor using the experimental PCSs, which reflect the average of the PCSs observed in each monomer. Note that, due to the averaging, the final fitted  $\Delta\chi$  tensor must be scaled by the user  $n$ -fold, where  $n$  is the number of monomers. This feature can also be exploited in NMR crystallography (Kervern et al., 2009).

## 8 Quality factors

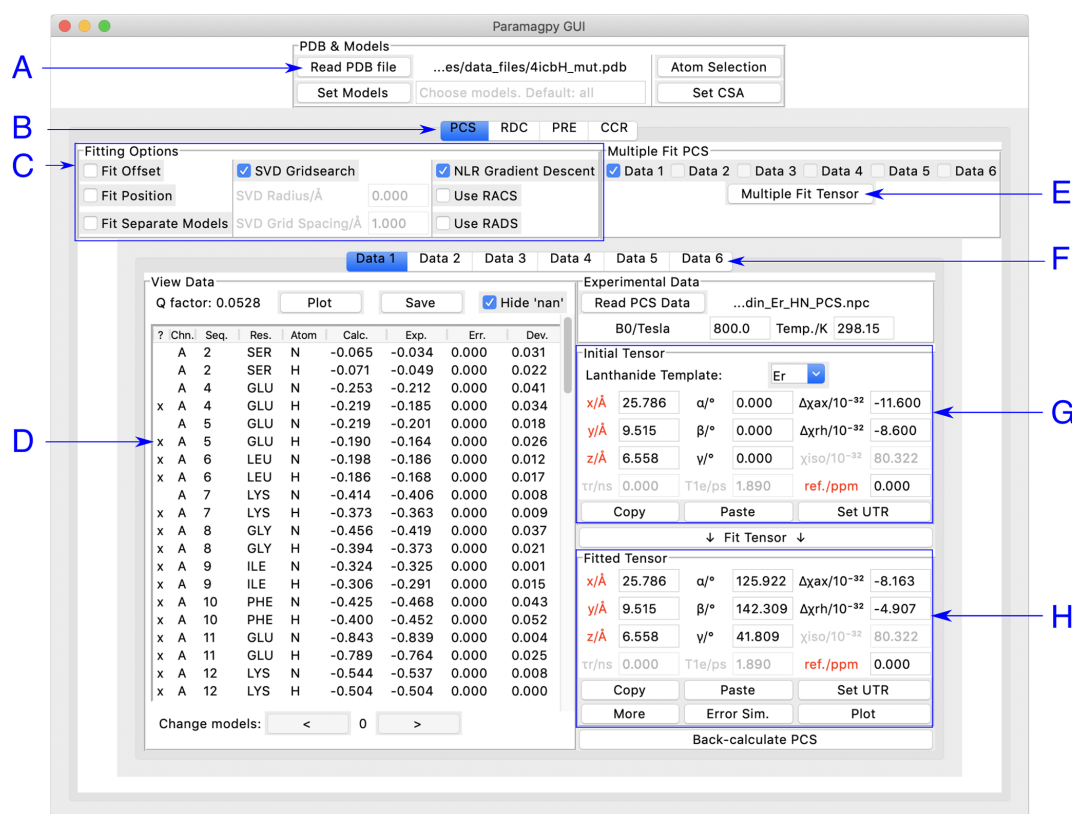
To judge the agreement of tensor fits with the experimental data, a  $Q$  factor can be assigned to a given fit, which Paramagpy calculates using Eq. (31). Here, the experimental and calculated PCS, RDC, PRE or CCR values are denoted  $a^{\text{exp}}$  and  $a^{\text{cal}}$ , respectively, the index  $m$  is for ensemble averaging of common spins between models, and the index  $i$  is for summation over all spins of the molecule. A low  $Q$  factor signifies a good-quality fit.

$$Q = \sqrt{\frac{\sum_i \left[ \left( \sum_m [a_i^{\text{exp}} - a_{m,i}^{\text{cal}}] \right)^2 \right]}{\sum_i \left[ \left( \sum_m [a_i^{\text{exp}}] \right)^2 \right]}} \quad (31)$$

Alternative  $Q$  factors have been proposed (Clare and Garrett, 1999; Bashir et al., 2010). The  $Q$  factor proposed by Bashir et al. (2010), which uses sums of experimental and calculated values in the denominator of Eq. (31) and therefore tends to be 2 times smaller, is supported by the scripted environment of Paramagpy. It is important to note that the fitting algorithm used by Paramagpy targets the minimal root-mean-square deviation between experimental and calculated data rather than the  $Q$  factor. It has been pointed out that  $Q$ -factor evaluations are meaningful only if the number of fitted data greatly exceeds the number of variables (Bax, 2003).

## 9 Graphical user interface

Paramagpy has a graphical user interface (GUI) written for the inbuilt Tk/Tcl interface of Python 3, which can run on Mac OS X, Windows and Linux operating systems. The GUI offers a user-friendly environment for loading and visualising PDB files and experimental PCS, RDC, PRE and CCR data. Two frames display the initial and fitted tensors. The fitted tensor is calculated and displayed by the push of a button. An overview of the PCS fitting tab is shown in Fig. 1. Hovering the mouse over any element in the window displays a useful tool tip to help the user.



**Figure 1.** Paramagpy GUI running on Mac OS X. (A) Frame for loading PDB coordinates. The atoms and models (conformers) of interest can be selected and CSA-tensor parameters set by the user. (B) The user can switch between PCS, RDC, PRE and CCR tabs, where CCR stands for the Curie-spin-dipole-dipole cross-correlated relaxation. (C) Fitting options can be specified by selecting the relevant check box. The “SVD Gridsearch” option searches for the best-fit tensor within a sphere about the initial tensor origin with radius and grid spacing as specified. The “NLR Gradient Descent” option refines the tensor using non-linear least-squares minimisation. (D) Experimental data for atoms in the PDB file are displayed here. The first column contains an “x” if the datum will be used during fitting and may be toggled by pressing the “x” key on the keyboard. Experimental and back-calculated PCS values are also reported and their correlation can be displayed by clicking the “Plot” button above. (E) To utilise multiple PCS data sets to fit different tensors to a common position, the “Multiple Fit Tensor” button can be clicked after selecting the desired data sets. (F) Each tab can contain a different PCS data set, allowing up to 6 to be loaded. If more data sets are required, Paramagpy supports this through the scripted module. (G) The initial tensor parameters can be specified here to define a starting point before fitting. For convenience, the paramagnetic centre can be positioned at any atom in the PDB file by double-clicking on a row of the data view in the frame to the left. Parameters in red are constrained during fitting. Greyed out parameters are not relevant to PCS or RDC calculations, but are used in PRE and CCR calculations. (H) The fitted tensor is displayed here. Clicking the “Copy” button allows the tensor to be pasted into other tabs of the program (see B and F above). The “Plot” button will prompt the user to save an isosurface file for opening in PyMOL. “Error Sim.” will assess the quality of the fit by bootstrap or Monte Carlo methods. The button “Set UTR” is for conversion of the tensor parameters to the unique tensor representation defined by the program Numbat (Schmitz et al., 2008).

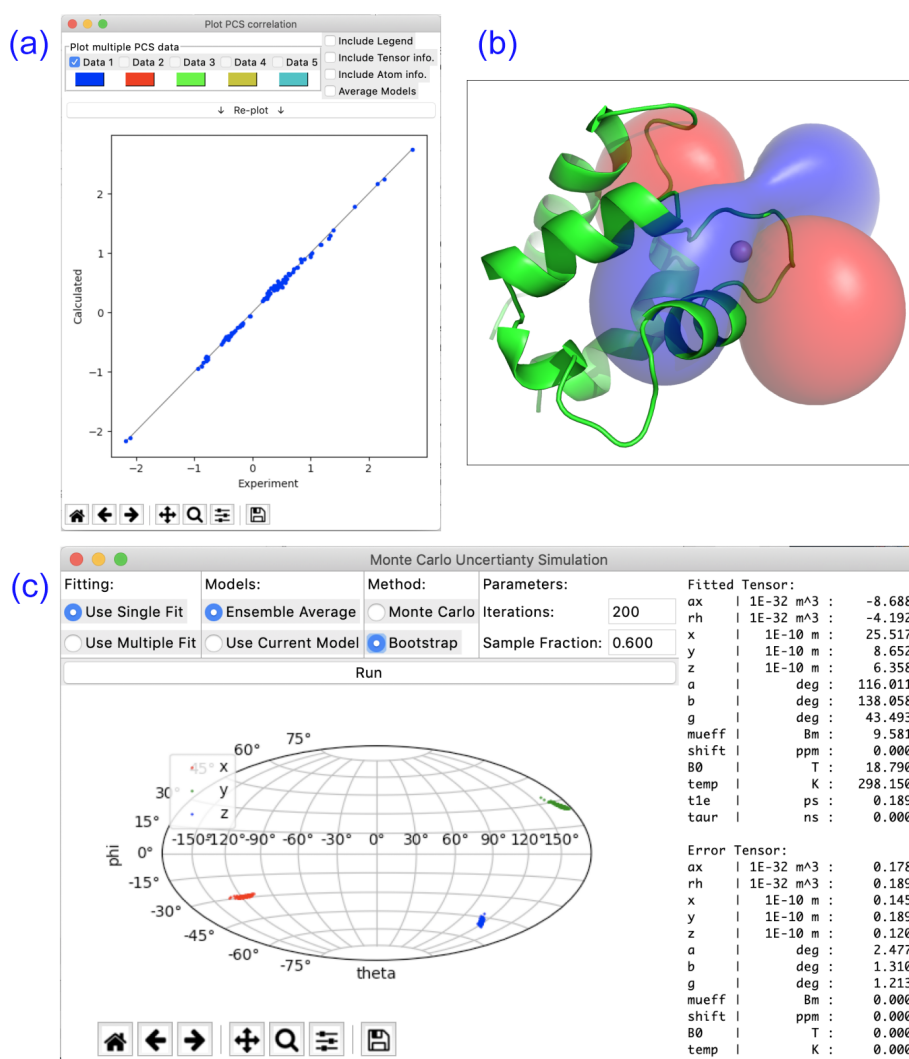
## 10 Visualisation

Paramagpy offers a number of plot options to quickly visualise tensors and quality of fit. The scalar PCS or PRE field can be written to a CCP4 (McNicholas et al., 2011) density map, which can then be visualised as a three-dimensional contour plot in the program PyMOL (Schrödinger, LLC, 2015). The fit quality can be visualised in correlation plots of back-calculated PCS, RDC, PRE and CCR values versus the experimental values. Finally, a scatter plot of the principle axes of the tensors can be viewed in a Sanson-Flamsteed-

style projection following Monte Carlo or bootstrap error analyses. Example plots are summarised in Fig. 2.

## 11 Scripting

Paramagpy is a Python module and can be imported into a scripting environment. The module is split into four major submodules. (i) The “metal” submodule deals with the paramagnetic centre, tensor representations and methods for calculating PCS, RDC, PRE and CCR values. (ii) The “protein” submodule handles the atomic coordinates from the PDB file



**Figure 2.** Plotting options available in Paramagpy illustrated with data of calbindin D9k loaded with Er<sup>3+</sup>. (a) Correlation plot of calculated versus experimental PCS values after fitting of the  $\Delta\chi$  tensor. (b) PCS isosurface plot viewed in PyMOL. (c) Sanson–Flamsteed plot showing the principle axes projections after bootstrap analysis. “Error Tensor” reports the standard deviation in fitted parameters.

and CSA-tensor definitions. (iii) The “dataparse” submodule manages the reading and writing of data files. (iv) The “fit” submodule contains functions for fitting tensors to experimental data. An example script for fitting of a  $\Delta\chi$  tensor to experimental PCS data for calbindin D9k is shown in Fig. 3. It uses only nine lines of code. Some more advanced features of Paramagpy, such as fitting of power spectral density tensors in Eqs. (15) and (16), are only available in the scripted environment. The scripted environment also offers control over which parameters are included for fitting routines and allows calculations for coordinates other than PDB formats.

## 12 NMR software integration

Paramagpy includes macro scripts to interface with popular NMR software: CcpNmr analysis and Sparky (Vranken et al.,

2005; Lee et al., 2014). Currently, these macros allow for the rapid calculation of experimental PCS values from NMR spectra with up to three dimensions, fitting of  $\Delta\chi$  tensors and plotting of back-calculated PCS values onto paramagnetic spectra.

## 13 Tensor conventions and conversions

Paramagpy offers a number of simple routines to convert between tensor representations. In addition to the  $3 \times 3$  matrix representations of tensors, positions, rotation matrices, eigenvalues, axial/rhombic components and Euler angles, alignment tensors and Saupe tensors are available upon clicking the “More” button within the GUI. The axial and rhombic

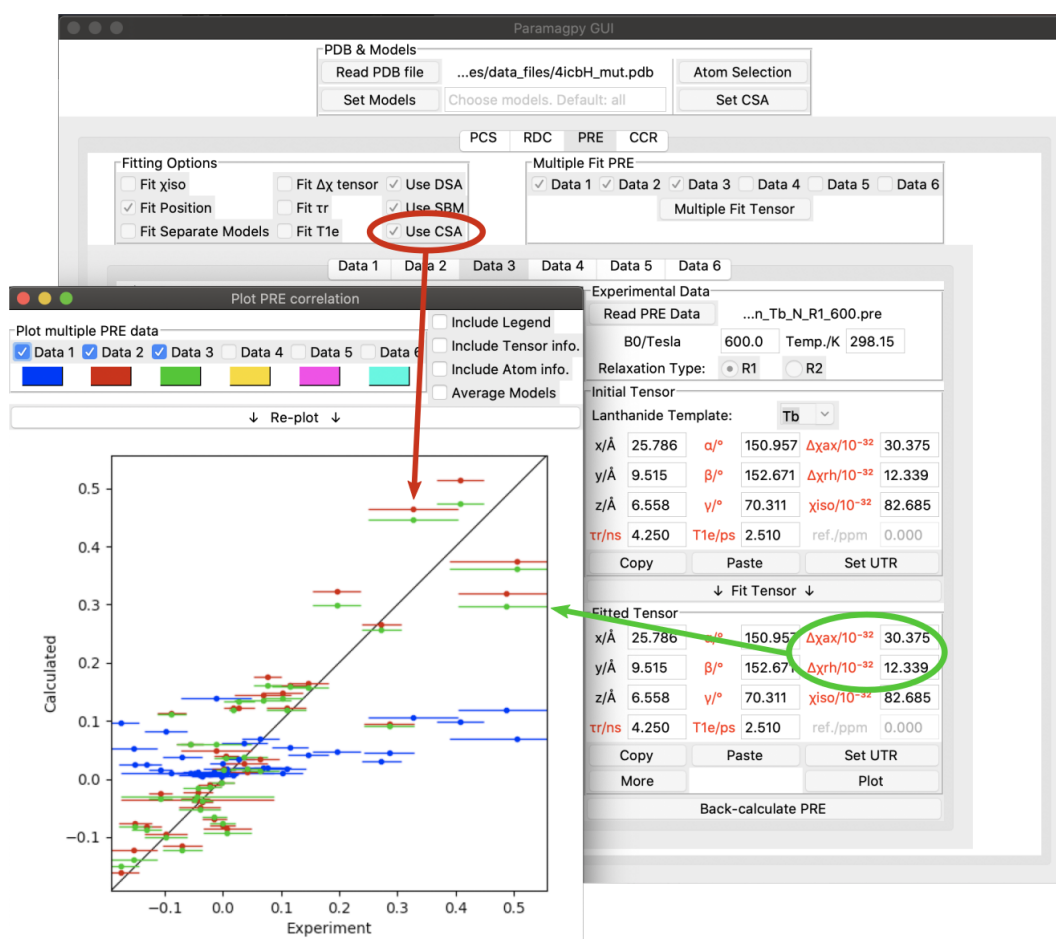
```

1 from paramagpy import protein, fit, dataparse, metal
2 # Load the PDB file
3 prot = protein.load_pdb('../data_files/4icbH_mut.pdb')
4 # Load the PCS data
5 rawData = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS.npc')
6 # Associate PCS data with atoms of the PDB
7 parsedData = prot.parse(rawData)
8 # Define an initial tensor
9 mStart = metal.Metal()
10 # Set the starting position to an atom close to the metal
11 mStart.position = prot[0]['A'][56]['CA'].position
12 # Calculate an initial tensor from an SVD gridsearch
13 mGuess, calc, qfac = fit.svd_gridsearch_fit_metal_from_pcs(
14     [mStart], [parsedData], radius=10, points=10)
15 # Refine the tensor using non-linear regression
16 mFit, calc, qfac = fit.nlr_fit_metal_from_pcs(mGuess, [parsedData])
17 # Save the fitted tensor to file
18 mFit[0].save('calbindin_Er_HN_PCS_tensor.txt')

```

1	ax	1E-32 m <sup>3</sup>	-8.688
2	rh	1E-32 m <sup>3</sup>	-4.192
3	x	1E-10 m	25.517
4	y	1E-10 m	8.652
5	z	1E-10 m	6.358
6	a	deg	116.011
7	b	deg	138.058
8	g	deg	43.492
9	mueff	Bm	0.000
10	shift	ppm	0.000
11	B0	T	18.790
12	temp	K	298.150
13	t1e	ps	0.000
14	taur	ns	0.000

**Figure 3.** Example Python script for fitting a  $\Delta\chi$  tensor to experimental PCS data. The output with fitted tensor parameters is displayed to the right.



**Figure 4.** Paramagpy GUI showing  $R_1(^{15}\text{N})$  PRE data for calbindin D<sub>9k</sub> loaded with Tb<sup>3+</sup>. The correlation plot shows calculated vs. experimental values. Blue: SBM and isotropic Curie-spin theory are used for calculating PREs ( $Q$  factor 1.01). Red: also taking into account the cross-correlation between Curie-spin and CSA relaxation ( $Q$  factor 0.49). Green: including the additional correction arising from the anisotropy of the  $\chi$  tensor ( $Q$  factor 0.47).



components are defined as follows (Eqs. 32 and 33).

$$\Delta\chi_{ax} = \Delta\chi_{zz} - \left( \frac{\Delta\chi_{xx} + \Delta\chi_{yy}}{2} \right) \quad (32)$$

$$\Delta\chi_{rh} = \Delta\chi_{xx} - \Delta\chi_{yy} \quad (33)$$

By default, Paramagpy reports all fitted tensors in the unique tensor representation used by the program Numbat (Schmitz et al., 2008). This requires that the principle axis magnitudes of the  $\Delta\chi$  tensor are ordered  $|\Delta\chi_{zz}| \geq |\Delta\chi_{yy}| \geq |\Delta\chi_{xx}|$ , and all Euler angles are in the range  $[0, \pi]$  using the ZYZ convention.

## 14 Example PRE calculation

PRE calculations that include anisotropy effects and cross-correlation with CSA can be daunting to set up as they require the  $\Delta\chi$  and CSA tensors to possess the correct orientations in the frame of the molecular coordinates. Paramagpy simplifies this for the user by allowing  $\Delta\chi$  tensors fitted from PCS data to be transferred easily to the tab for PRE calculations. Furthermore, CSA-tensor templates are provided for most protein backbone atoms.

As an example, Fig. 4 shows the Paramagpy GUI with  $R_1(^{15}\text{N})$  PRE data for calbindin D<sub>9k</sub> loaded with Tb<sup>3+</sup> (Orton et al., 2016). A  $\Delta\chi$  tensor was fitted using the PCS tab, then transferred to the PRE tab using the “Copy” and “Paste” buttons. Curie-spin–CSA cross-correlation is taken into account simply by checking the box “Use CSA”. This greatly improves the correlation and allows the prediction of negative PREs, resulting in a reduction in the  $Q$  factor from 1.01 to 0.49. The small additional correction arising from the anisotropy of the Curie spin can be included by setting the  $\Delta\chi_{ax}$  and  $\Delta\chi_{rh}$  parameters to the non-zero values obtained from the  $\Delta\chi$  tensor fitted with the help of PCS data yielding a further reduction in the  $Q$  factor to 0.47.

The CSA tensors of  $^{15}\text{N}$  spins are much larger than those of  $^1\text{H}$  spins, so that Curie-spin–CSA cross-correlation effects can dominate the PRE to the point that even negative PREs can be observed (Orton et al., 2016 Fig. 4). These CCR effects are predicted to be most pronounced for  $^{15}\text{N}$  spins located about 10 Å from the metal ion. In contrast, the CSA of  $^1\text{H}$  spins is much smaller, so that their CCR effects are predicted to be most significant in the range of 20–25 Å and therefore too small to be easily observed experimentally (Pintacuda et al., 2004a).

## 15 Conclusions

Paramagpy is an easy-to-use program that integrates the related paramagnetic NMR phenomena of PCS, RDC, PRE and CCR. Paramagpy allows the rapid analysis of NMR spectra of samples containing a single paramagnetic centre, which is particularly useful for data recorded with different paramagnetic lanthanide ions. With an intuitive calculation flow,

Paramagpy can be used, for example, to fit a  $\Delta\chi$  tensor using experimental PCS data and then quickly report the expected PREs of the same complex, informing the user which signals may be too broad to observe. Paramagpy uses efficient fitting algorithms and an up-to-date implementation of paramagnetic NMR theory to capture subtle corrections arising from CSA and anisotropy effects in the PCS and PRE calculations.

**Code availability.** The source code for Paramagpy is available at: <https://doi.org/10.5281/zenodo.3594568> (Orton, 2019).

**Supplement.** The supplement related to this article is available online at: <https://doi.org/10.5194/mr-1-1-2020-supplement>.

**Author contributions.** HWO initiated the project, wrote the source code and the documentation for Paramagpy, and drafted the manuscript. GO and TH contributed advice towards code design and the final versions of the manuscript.

**Competing interests.** The authors declare that they have no conflict of interest.

**Acknowledgements.** Henry William Orton thanks the Westpac Bicentennial foundation for a Future Leaders Scholarship.

**Financial support.** This research has been supported by the Australian Research Council, including a Laureate Fellowship (project no. FL170100019) to Gottfried Otting.

**Review statement.** This paper was edited by Mikael Akke and reviewed by Marcellus Ubbink and one anonymous referee.

## References

- Balayssac, S., Bertini, I., Luchinat, C., Parigi, G., and Piccioli, M.:  $^{13}\text{C}$  direct detected NMR increases the detectability of residual dipolar couplings, *J. Am. Chem. Soc.*, 128, 15042–15043, <https://doi.org/10.1021/ja0645436>, 2006.
- Banci, L., Bertini, I., Cavallaro, G., Giachetti, A., Luchinat, C., and Parigi, G.: Paramagnetism-based restraints for Xplor-NIH, *J. Biomol. NMR*, 28, 249–261, <https://doi.org/10.1023/B:JNMR.0000013703.30623.f7>, 2004.
- Bashir, Q., Volkov, A. N., Ullmann, G. M., and Ubbink, M.: Visualization of the encounter ensemble of the transient electron transfer complex of cytochrome *c* and cytochrome *c* peroxidase, *J. Am. Chem. Soc.*, 132, 241–247, <https://doi.org/10.1021/ja9064574>, 2010.

- Bax, A.: Weak alignment offers new NMR opportunities to study protein structure and dynamics, *Protein Sci.*, 12, 1–16, <https://doi.org/10.1110/ps.0233303>, 2003.
- Bertini, I., Janik, M. B. L., Lee, Y.-M., Luchinat, C., and Rosato, A.: Magnetic susceptibility tensor anisotropies for a lanthanide ion series in a fixed protein matrix, *J. Am. Chem. Soc.*, 123, 4181–4188, <https://doi.org/10.1021/ja0028626>, 2001a.
- Bertini, I., Kowalewski, J., Luchinat, C., and Parigi, G.: Cross correlation between the dipole–dipole interaction and the Curie spin relaxation: the effect of anisotropic magnetic susceptibility, *J. Magn. Reson.*, 152, 103–108, <https://doi.org/10.1006/jmre.2001.2378>, 2001b.
- Bertini, I., Cavallaro, G., Cosenza, M., Kümmerle, R., Piccioli, M., and Poggi, L.: Cross correlation rates between Curie spin and dipole-dipole relaxation in paramagnetic proteins: the case of Cerium substituted calbindin D<sub>9k</sub>, *J. Biomol. NMR*, 23, 115–125, <https://doi.org/10.1023/A:1016341507527>, 2002a.
- Bertini, I., Luchinat, C., and Parigi, G.: Magnetic susceptibility in paramagnetic NMR, *Prog. Nucl. Mag. Res. Sp.*, 40, 249–273, [https://doi.org/10.1016/S0079-6565\(02\)00002-X](https://doi.org/10.1016/S0079-6565(02)00002-X), 2002b.
- Bieri, M., d’Auvergne, E. J., and Gooley, P. R.: relaxGUI: a new software for fast and simple NMR relaxation data analysis and calculation of ps-ns and  $\mu$ s motion of proteins, *J. Biomol. NMR*, 50, 147–155, <https://doi.org/10.1007/s10858-011-9509-1>, 2011.
- Clore, G. M. and Garrett, D. S.: *R*-factor, free *R*, and complete cross-validation for dipolar coupling refinement of NMR structures, *J. Am. Chem. Soc.*, 121, 9008–9012, <https://doi.org/10.1021/ja991789k>, 1999.
- Cornilescu, G. and Bax, A.: Measurement of proton, nitrogen, and carbonyl chemical shielding anisotropies in a protein dissolved in a dilute liquid crystalline phase, *J. Am. Chem. Soc.*, 122, 10143–10154, <https://doi.org/10.1021/ja0016194>, 2000.
- de Vries, S. J., van Dijk, M., and Bonvin, A. M. J. J.: The HADDOCK web server for data-driven biomolecular docking, *Nat. Protoc.*, 5, 883–897, <https://doi.org/10.1038/nprot.2010.32>, 2010.
- Dominguez, C., Boelens, R., and Bonvin, A. M. J. J.: HADDOCK: a protein-protein docking approach based on biochemical or biophysical information, *J. Am. Chem. Soc.*, 125, 1731–1737, <https://doi.org/10.1021/ja026939x>, 2003.
- Dosset, P., Hus, J.-C., Marion, D., and Blackledge, M.: A novel interactive tool for rigid-body modeling of multi-domain macromolecules using residual dipolar couplings, *J. Biomol. NMR*, 20, 223–231, <https://doi.org/10.1023/A:1011206132740>, 2001.
- Fletcher, R.: *Practical methods of optimization*, John Wiley & Sons, New York, 2nd edn., <https://doi.org/10.1002/9781118723203>, 1988.
- Ghose, R. and Prestegard, J. H.: Electron spin–nuclear spin cross-correlation effects on multiplet splittings in paramagnetic proteins, *J. Magn. Reson.*, 128, 138–143, <https://doi.org/10.1006/jmre.1997.1227>, 1997.
- Guéron, M.: Nuclear relaxation in macromolecules by paramagnetic ions: a novel mechanism, *J. Magn. Reson.*, 19, 58–66, [https://doi.org/10.1016/0022-2364\(75\)90029-3](https://doi.org/10.1016/0022-2364(75)90029-3), 1975.
- Hogben, H., Krzystyniak, M., Charnock, G., Hore, P., and Kuprov, I.: Spinach – a software library for simulation of spin dynamics in large spin systems, *J. Magn. Reson.*, 208, 179–194, <https://doi.org/10.1016/j.jmr.2010.11.008>, 2011.
- John, M., Park, A. Y., Pintacuda, G., Dixon, N. E., and Otting, G.: Weak alignment of paramagnetic proteins warrants correction for residual CSA effects in measurements of pseudocontact shifts, *J. Am. Chem. Soc.*, 127, 17190–17191, <https://doi.org/10.1021/ja0564259>, 2005.
- Kervern, G., D’Aléo, A., Toupet, L., Maury, O., Emsley, L., and Pintacuda, G.: Crystal-structure determination of powdered paramagnetic lanthanide complexes by proton NMR spectroscopy, *Angew. Chem. Int. Edit.*, 48, 3082–3086, <https://doi.org/10.1002/anie.200805302>, 2009.
- Kontaxis, G., Clore, G., and Bax, A.: Evaluation of cross-correlation effects and measurement of one-bond couplings in proteins with short transverse relaxation times, *J. Magn. Reson.*, 143, 184–196, <https://doi.org/10.1006/jmre.1999.1979>, 2000.
- Kramer, F., Deshmukh, M., Kessler, H., and Glaser, S.: Residual dipolar coupling constants: an elementary derivation of key equations, *Concepts Magn. Reso. A*, 21, 10–21, <https://doi.org/10.1002/cmr.a.20003>, 2004.
- Lee, W., Tonelli, M., and Markley, J. L.: NMRFAM-SPARKY: enhanced software for biomolecular NMR spectroscopy, *Bioinformatics*, 31, 1325–1327, <https://doi.org/10.1093/bioinformatics/btu830>, 2014.
- McNicholas, S., Potterton, E., Wilson, K. S., and Noble, M. E. M.: Presenting your structures: the *CCP4mg* molecular-graphics software, *Acta Crystallogr. D*, 67, 386–394, <https://doi.org/10.1107/S0907444911007281>, 2011.
- Orton, H. W. and Otting, G.: Accurate electron–nucleus distances from paramagnetic relaxation enhancements, *J. Am. Chem. Soc.*, 140, 7688–7697, <https://doi.org/10.1021/jacs.8b03858>, 2018.
- Orton, H. W., Kuprov, I., Loh, C.-T., and Otting, G.: Using paramagnetism to slow down nuclear relaxation in protein NMR, *J. Phys. Chem. Lett.*, 7, 4815–4818, <https://doi.org/10.1021/acs.jpclett.6b02417>, 2016.
- Orton, H. W.: Paramagpy source code, Version v1.0, Zenodo, <https://doi.org/10.5281/zenodo.3594568>, 2019.
- Pearce, B. J. G., Jabar, S., Loh, C.-T., Szabo, M., Graham, B., and Otting, G.: Structure restraints from heteronuclear pseudocontact shifts generated by lanthanide tags at two different sites, *J. Biomol. NMR*, 68, 19–32, <https://doi.org/10.1007/s10858-017-0111-z>, 2017.
- Pintacuda, G., Hohenthanner, K., Otting, G., and Müller, N.: Angular dependence of dipole-dipole-Curie-spin cross-correlation effects in high-spin and low-spin paramagnetic myoglobin, *J. Biomol. NMR*, 27, 115–135, <https://doi.org/10.1023/A:1024926126239>, 2003.
- Pintacuda, G., Kaikkonen, A., and Otting, G.: Modulation of the distance dependence of paramagnetic relaxation enhancements by CSA  $\times$  DSA cross-correlation, *J. Magn. Reson.*, 171, 233–243, <https://doi.org/10.1016/j.jmr.2004.08.019>, 2004a.
- Pintacuda, G., Keniry, M. A., Huber, T., Park, A. Y., Dixon, N. E., and Otting, G.: Fast structure-based assignment of  $^{15}\text{N}$  HSQC spectra of selectively  $^{15}\text{N}$ -labeled paramagnetic proteins, *J. Am. Chem. Soc.*, 126, 2963–2970, <https://doi.org/10.1021/ja039339m>, 2004b.
- Raman, S., Lange, O. F., Rossi, P., Tyka, M., Wang, X., Aramini, J., Liu, G., Ramelot, T. A., Eletsky, A., Szyperski, T., Kennedy, M. A., Prestegard, J., Montelione, G. T., and Baker, D.: NMR structure determination for larger pro-

- teins using backbone-only data, *Science*, 327, 1014–1018, <https://doi.org/10.1126/science.1183649>, 2010.
- Rinaldelli, M., Carlon, A., Ravera, E., Parigi, G., and Luchinat, C.: FANTEN: a new web-based interface for the analysis of magnetic anisotropy-induced NMR data, *J. Biomol. NMR*, 61, 21–34, <https://doi.org/10.1007/s10858-014-9877-4>, 2015.
- Schmitz, C., Stanton-Cook, M. J., Su, X.-C., Otting, G., and Huber, T.: Numbat: an interactive software tool for fitting  $\Delta\chi$ -tensors to molecular coordinates using pseudocontact shifts, *J. Biomol. NMR*, 41, 179–189, <https://doi.org/10.1007/s10858-008-9249-z>, 2008.
- Schmitz, C., Vernon, R., Otting, G., Baker, D., and Huber, T.: Protein structure determination from pseudocontact shifts using ROSETTA, *J. Mol. Biol.*, 416, 668–677, <https://doi.org/10.1016/j.jmb.2011.12.056>, 2012.
- Schrödinger, LLC: The PyMOL molecular graphics system, version 1.8, 2015.
- Solomon, I.: Relaxation processes in a system of two spins, *Phys. Rev.*, 99, 559–565, <https://doi.org/10.1103/PhysRev.99.559>, 1955.
- Stanton-Cook, M. J., Su, X.-C., Otting, G., and Huber, T.: PyParaTools, available at: <http://comp-bio.anu.edu.au/mscook/PPT/> (last access: 15 October 2019), 2014.
- Suturina, E. A., Mason, K., Geraldes, C. F. G. C., Chilton, N. F., Parker, D., and Kuprov, I.: Lanthanide-induced relaxation anisotropy, *Phys. Chem. Chem. Phys.*, 20, 17676–17686, <https://doi.org/10.1039/C8CP01332B>, 2018.
- Valafar, H. and Prestegard, J. H.: REDCAT: a residual dipolar coupling analysis tool, *J. Magn. Reson.*, 167, 228–241, <https://doi.org/10.1016/j.jmr.2003.12.012>, 2004.
- Vega, A. J. and Fiat, D.: Nuclear relaxation processes of paramagnetic complexes the slow-motion case, *Mol. Phys.*, 31, 347–355, <https://doi.org/10.1080/00268977600100261>, 1976.
- Vranken, W. F., Boucher, W., Stevens, T. J., Fogh, R. H., Pajon, A., Llinas, M., Ulrich, E. L., Markley, J. L., Ionides, J., and Laue, E. D.: The CCPN data model for NMR spectroscopy: development of a software pipeline, *Proteins*, 59, 687–696, <https://doi.org/10.1002/prot.20449>, 2005.
- Zweckstetter, M. and Bax, A.: Prediction of sterically induced alignment in a dilute liquid crystalline phase: aid to protein structure determination by NMR, *J. Am. Chem. Soc.*, 122, 3791–3792, <https://doi.org/10.1021/ja0000908>, 2000.

Supplement of Magn. Reson., 1, 1–12, 2020  
<https://doi.org/10.5194/mr-1-1-2020-supplement>  
© Author(s) 2020. This work is distributed under  
the Creative Commons Attribution 4.0 License.



*Supplement of*

## **Paramagpy: software for fitting magnetic susceptibility tensors using paramagnetic effects measured in NMR spectra**

**Henry William Orton et al.**

*Correspondence to:* Gottfried Otting ([gottfried.otting@anu.edu.au](mailto:gottfried.otting@anu.edu.au))

The copyright of individual parts of the supplement might differ from the CC BY 4.0 License.

---

# **paramagpy Documentation**

***Release 1.0***

**Henry Orton**

**Dec 29, 2019**

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Features</b>	<b>3</b>
<b>3</b>	<b>Documentation</b>	<b>4</b>
<b>4</b>	<b>Citing paramagpy</b>	<b>5</b>
<b>5</b>	<b>Contents</b>	<b>6</b>
5.1	Installation Guide . . . . .	6
5.2	Examples . . . . .	7
5.3	Graphic User Interface (GUI) . . . . .	40
5.4	NMR Software Macros . . . . .	41
5.5	Mathematical Proofs . . . . .	44
5.6	Reference Guide . . . . .	46
	<b>Python Module Index</b>	<b>96</b>
	<b>Index</b>	<b>97</b>

**Release** 1.0

**Date** Dec 29, 2019

## INTRODUCTION

paramagpy is a python module for calculating paramagnetic effects in NMR spectra of proteins. This currently includes fitting of paramagnetic susceptibility tensors to experimental data associated with pseudocontact shifts (PCS) residual dipolar couplings (RDC), paramagnetic relaxation enhancements (PRE) and cross-correlated relaxation (CCR). A GUI allows easy viewing of data and seamless transition between PCS/RDC/PRE/CCR calculations.



Fig. 1: *Please, not the eyes!* - Canberra cyclist



## FEATURES

- Support for PDB protein structures with models
- Combined SVD gridsearch and gradient descent algorithms for solving PCS tensors
- Optional fitting of reference offset parameter for PCS datasets
- Support for Residual Anisotropic Chemical Shielding (RACS) and Residual Anisotropic Dipolar Shielding (RADS) corrections to PCS
- Lanthanide parameter templates available
- Plotting of correlation between experiment/calculated values
- Plotting of tensor isosurfaces compatible with PyMol
- Q-factor calculations
- Error analysis of tensor fit quality by Monte-Carlo or Bootstrap methods
- Optimisation of multiple PCS/PRE/CCR datasets to a common position
- Unique tensor representation compatible with Numbat (program)
- Fitting of RDC tensor by SVD algorithm
- PRE calculations by Solomon and Curie spin mechanisms
- Spectral power density tensor fitting for anisotropic dipolar PREs
- CSA cross-correlation correction to PRE calculations
- Dipole-dipole/Curie spin cross-correlated relaxation calculations
- Fitting of tensor parameters to PRE/CCR data
- Macro scripts for integration with CCPNMR and Sparky

## DOCUMENTATION

- <https://henryorton.github.io/paramagpy/>

## CITING PARAMAGPY

Paramagpy is published in Magnetic Resonance <https://doi.org/10.5194/mr-2019-3>

**CONTENTS**

## 5.1 Installation Guide

### 5.1.1 Requirements

Paramagpy is written for python 3. It requires packages:

- NumPy
- SciPy
- matplotlib
- BioPython

### 5.1.2 Unix/OSX Installation

Install directly using pip:

```
$ pip install paramagpy
```

Or, download the [source code](#) and run:

```
$ python setup.py install
```

within the source directory.

### 5.1.3 Windows Installation

Paramagpy has never been tested on windows, but theoretically it should work. Good luck!

### 5.1.4 Running the GUI

Once you have installed paramagpy, see *Graphic User Interface (GUI)* for how to run the GUI.

## 5.2 Examples

Note that all these examples are for scripted calculations using the Paramagpy python module. Most of this functionality is also available by the GUI *Graphic User Interface (GUI)*

### 5.2.1 PCS data

#### Fit Tensor to PCS Data

This example shows how to fit a  $\Delta\chi$ -tensor to experimental PCS data for the protein calbindin D9k. These data contain amide 1H and 15N chemical shifts between diamagnetic and paramagnetic states with the lanthanide Er3+ bound.

#### Downloads

- Download the data files `4icbH_mut.pdb` and `calbindin_Er_HN_PCS.npc` from [here](#):
- Download the script `pcs_fit.py`

#### Script + Explanation

Firstly, the necessary modules are imported from paramagpy.

```
from paramagpy import protein, fit, dataparse, metal
```

The protein is then loaded from a PDB file using `paramagpy.protein.load_pdb()` into the variable `prot`. This returns a `CustomStructure` object which is closely based on the `Structure` object from `BioPython` and contains the atomic coordinates. The object, and how to access atomic coordinates is discussed at this [link](#).

```
# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')
```

The PCS data is then loaded from a `.npc` file using the function `paramagpy.dataparse.read_pcs()` into the variable `rawData`. This is a dictionary of `(PCS, Error)` tuples which may be accessed by `rawData[(seq, atom)]` where `seq` is an integer specifying the sequence and `atom` is the atom name e.g `(3, 'HA')`. Note that these should match the corresponding sequence and atom in the PDB file.

```
# Load the PCS data
rawData = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS.npc')
```

To associate the experimental PCS value with atoms of the PDB structure, the method `paramagpy.protein.CustomStructure.parse()` is called on `rawData`. The returned array `parsedData` has a row for each atom with columns `[mdl, atm, exp, cal, err, idx]`, where `mdl` is the model number from the PDB file, `atm` is an atom object from the BioPython PDB structure, `exp` and `cal` are the experimental and calculated values, `err` is the experimental uncertainty and `idx` is the atom index, used to define ensemble averaging behaviour.

```
# Associate PCS data with atoms of the PDB
parsedData = prot.parse(rawData)
```

An initial  $\Delta\chi$ -tensor is defined by initialising a `paramagpy.metal.Metal` object. The initial position is known to be near the binding site, which is set to the CA atom of residue 56. Note that the `position` attribute is always in Angstrom units.

```
# Define an initial tensor
mStart = metal.Metal()

# Set the starting position to an atom close to the metal
mStart.position = prot[0]['A'][56]['CA'].position
```

A quick gridsearch is conducted in a sphere of 10 Angstrom with 10 points per radius using the function `paramagpy.fit.svd_gridsearch_fit_metal_from_pcs()`. This requires two lists containing the starting metals `mStart` and parsed experimental data `parsedData`. This function returns lists containing a new fitted metal object, the calculated PCS values from the fitted model.

```
# Calculate an initial tensor from an SVD gridsearch
[mGuess], [data] = fit.svd_gridsearch_fit_metal_from_pcs(
    [mStart], [parsedData], radius=10, points=10)
```

This is then refined using a non-linear regression gradient descent with the function `paramagpy.fit.nlr_fit_metal_from_pcs()`.

```
# Refine the tensor using non-linear regression
[mFit], [data] = fit.nlr_fit_metal_from_pcs([mGuess], [parsedData])
```

The Q-factor is then calculated using the function `paramagpy.fit.qfactor`.

```
# Calculate the Q-factor
qfac = fit.qfactor(data)
```

The fitted tensor parameters are saved by calling the method `paramagpy.metal.Metal.save()`. Alternatively they may be displayed using `print(mFit.info())`

```
# Save the fitted tensor to file
mFit.save('calbindin_Er_HN_PCS_tensor.txt')
```

Output: [calbindin\_Er\_HN\_PCS\_tensor.txt]

```
ax      | 1E-32 m^3 :   -8.688
rh      | 1E-32 m^3 :   -4.192
x       | 1E-10 m   :   25.517
y       | 1E-10 m   :    8.652
z       | 1E-10 m   :    6.358
a       |          deg :  116.011
b       |          deg :  138.058
g       |          deg :   43.492
mueff   |          Bm :    0.000
shift   |          ppm :    0.000
B0      |          T  :   18.790
temp    |          K  :  298.150
tle     |          ps :    0.000
taur    |          ns :    0.000
```

These experimental/calculated PCS values are then plotted in a correlation plot to assess the fit. This is achieved using standard functions of the plotting module `matplotlib`.

```
#### Plot the correlation ####
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data
ax.plot(data['exp'], data['cal'], marker='o', lw=0, ms=3, c='r',
        label="Q-factor = {:.4f}".format(qfac))

# Plot a diagonal
```

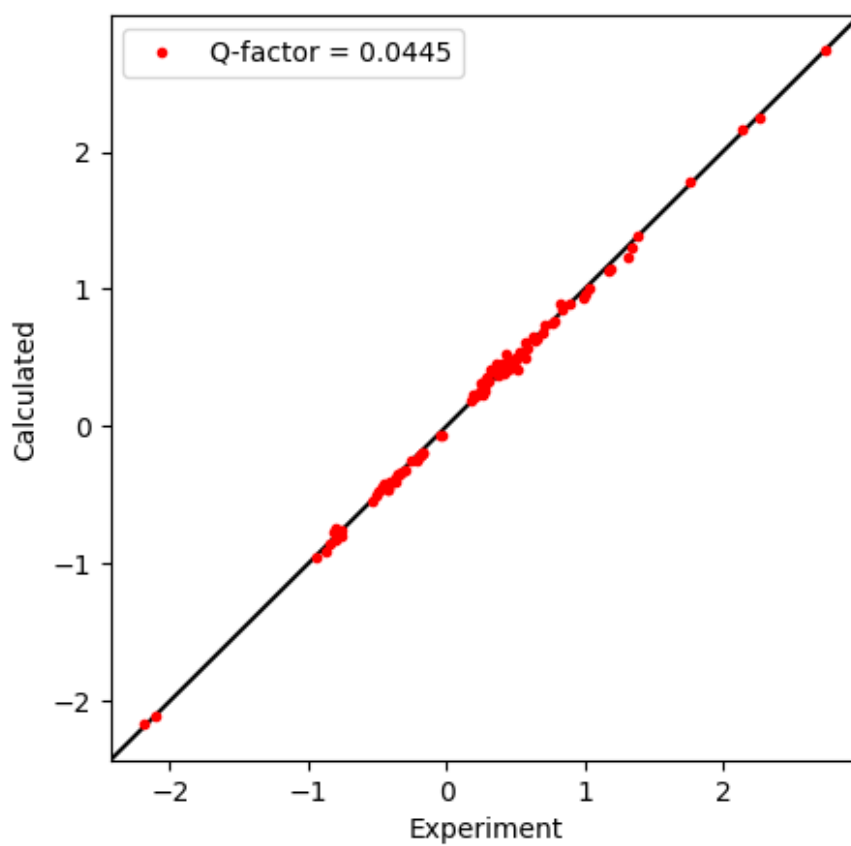
(continues on next page)

(continued from previous page)

```
l, h = ax.get_xlim()
ax.plot([l,h],[l,h], '-k', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pcs_fit.png")
```

Output: [pcs\_fit.png]



## Plot PCS isosurface (PyMol view)

This example shows how to plot the PCS isosurface of a fitted  $\Delta\chi$ -tensor for data from the example *Fit Tensor to PCS Data*. The isosurface can be viewed in [PyMol](#).

## Downloads

- Download the data files `4icbH_mut.pdb` and `calbindin_Er_HN_PCS_tensor.txt` from [here](#):
- Download the script `pcs_plot_isosurface.py`

## Explanation

The protein and tensor are loaded as described previously in.

The isosurface files are generated using the function `paramagpy.metal.Metal.isomap()`. The contour level can be chosen by setting the `isoval` argument. A larger density value will result in a smoother surface. This function writes two files `isomap.pml` and `isomap.pml.ccp4` which are the PyMol script and PCS grid files respectively.

The isosurface can be displayed by executing `pymol isomap.pml` from a terminal, or by selecting `File>Run` and navigating to the script `isomap.pml`.

## Script

[`pcs_plot_isosurface.py`]

```
from paramagpy import protein, fit, dataparse, metal

# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')

# Load the fitted tensor
met = metal.load_tensor('../data_files/calbindin_Er_HN_PCS_tensor.txt')

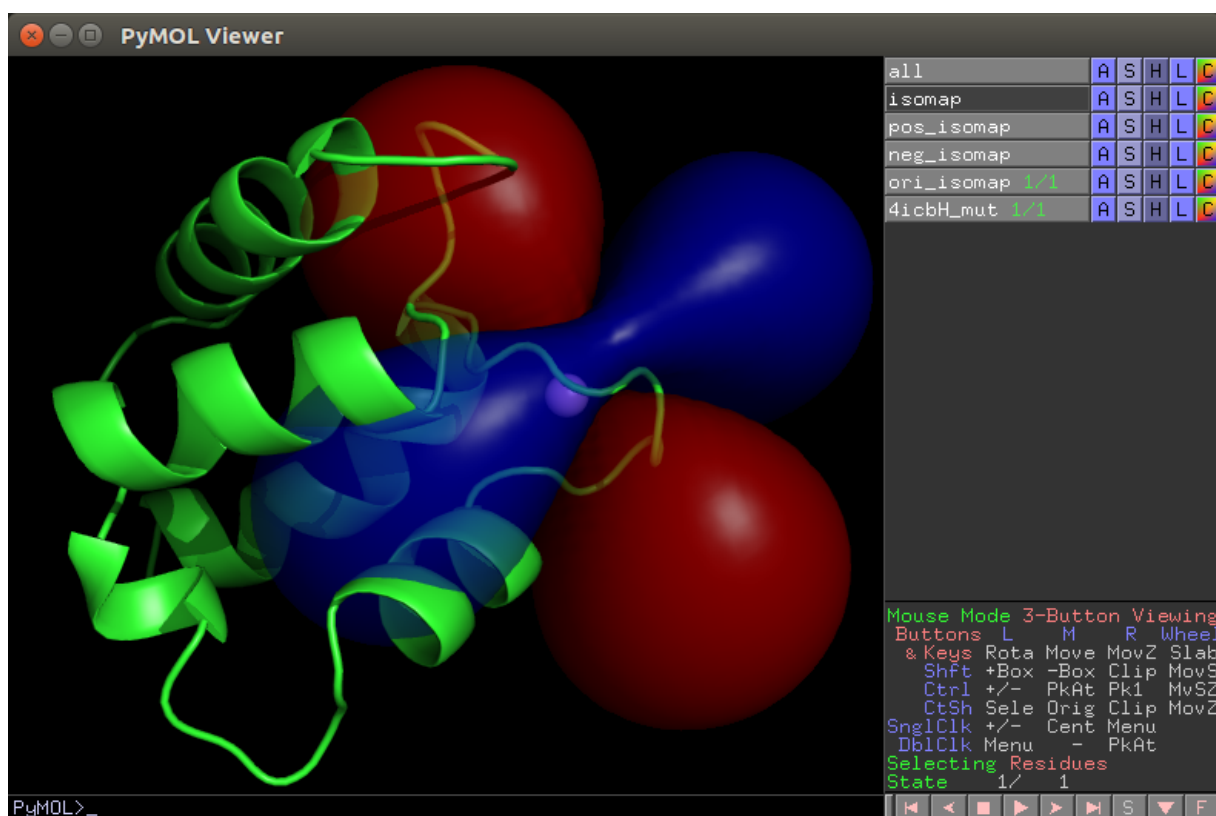
# Plot the isosurface to be opened in PyMol
met.isomap(prot.id, density=1, isoval=1.0)
```

## Output

*PyMol view of isosurface*

[`pcs_plot_isosurface.png`]





## Fit multiple PCS datasets to common position

This example shows how to fit multiple  $\Delta\chi$ -tensors to their respective datasets with a common position, but varied magnitude and orientation. This may arise if several lanthanides were investigated at the same binding site, and the data may be used simultaneously to fit a common position. Data from several PCS datasets for calbindin D9k were used here, and is a generalisation of the previous example: *Fit Tensor to PCS Data*.

## Downloads

- Download the data files `4icbH_mut.pdb`, `calbindin_Tb_HN_PCS.npc`, `calbindin_Er_HN_PCS.npc` and `calbindin_Yb_HN_PCS_tensor.txt` from [here](#):
- Download the script `pcs_fit_multiple.py`

## Explanation

The protein and PCS datasets are loaded and parsed. These are placed into a list `parsedData`, for which each element is a PCS dataset of a given lanthanide.

The two fitting functions:

- `paramagpy.fit.svd_gridsearch_fit_metal_from_pcs()`
- `paramagpy.fit.nlr_fit_metal_from_pcs()`

can accept a list of metal objects and a list of datasets with arbitrary size. If this list contains more than one element, fitting will be performed to a common position. The starting position is taken only from the first metal of the list.

After fitting, a list of fitted metals is returned. The fitted tensor are then written to files and a correlation plot is made.

## Script

[pcs\_fit\_multiple.py]

```
from paramagpy import protein, fit, dataparse, metal

# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')

# Load the PCS data
rawData1 = dataparse.read_pcs('../data_files/calbindin_Tb_HN_PCS.npc')
rawData2 = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS.npc')
rawData3 = dataparse.read_pcs('../data_files/calbindin_Yb_HN_PCS.npc')

# Associate PCS data with atoms of the PDB
parsedData = []
for rd in [rawData1, rawData2, rawData3]:
    parsedData.append(prot.parse(rd))

# Make a list of starting tensors
mStart = [metal.Metal(), metal.Metal(), metal.Metal()]

# Set the starting position to an atom close to the metal
mStart[0].position = prot[0]['A'][56]['CA'].position

# Calculate initial tensors from an SVD gridsearch
mGuess, datas = fit.svd_gridsearch_fit_metal_from_pcs(
    mStart, parsedData, radius=10, points=10)

# Refine the tensors using non-linear regression
fitParameters = ['x', 'y', 'z', 'ax', 'rh', 'a', 'b', 'g']
mFit, datas = fit.nlr_fit_metal_from_pcs(mGuess, parsedData, fitParameters)

# Save the fitted tensors to files
for name, metal in zip(['Tb', 'Er', 'Yb'], mFit):
    metal.save("tensor_{}.txt".format(name))

#### Plot the correlation ####
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data
for d, name, colour in zip(datas, ['Tb', 'Er', 'Yb'], ['r', 'g', 'b']):
    qfactor = fit.qfactor(d)
    ax.plot(d['exp'], d['cal'], marker='o', lw=0, ms=1, c=colour,
            label="{0:} - {1:5.3f}".format(name, qfactor))

# Plot a diagonal
l, h = ax.get_xlim()
ax.plot([l, h], [l, h], '-k', zorder=0)
ax.set_xlim(l, h)
ax.set_ylim(l, h)

# Axis labels
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pcs_fit_multiple.png")
```

## Outputs

### *Tb fitted tensor*

[tensor\_Tb.txt]

ax		1E-32 m <sup>3</sup>	:	31.096
rh		1E-32 m <sup>3</sup>	:	12.328
x		1E-10 m	:	25.937
y		1E-10 m	:	9.481
z		1E-10 m	:	6.597
a		deg	:	151.053
b		deg	:	152.849
g		deg	:	69.821
mueff		Bm	:	0.000
shift		ppm	:	0.000
B0		T	:	18.790
temp		K	:	298.150
t1e		ps	:	0.000
taur		ns	:	0.000

### *Er fitted tensor*

[tensor\_Er.txt]

ax		1E-32 m <sup>3</sup>	:	-8.422
rh		1E-32 m <sup>3</sup>	:	-4.886
x		1E-10 m	:	25.937
y		1E-10 m	:	9.481
z		1E-10 m	:	6.597
a		deg	:	126.015
b		deg	:	142.899
g		deg	:	41.039
mueff		Bm	:	0.000
shift		ppm	:	0.000
B0		T	:	18.790
temp		K	:	298.150
t1e		ps	:	0.000
taur		ns	:	0.000

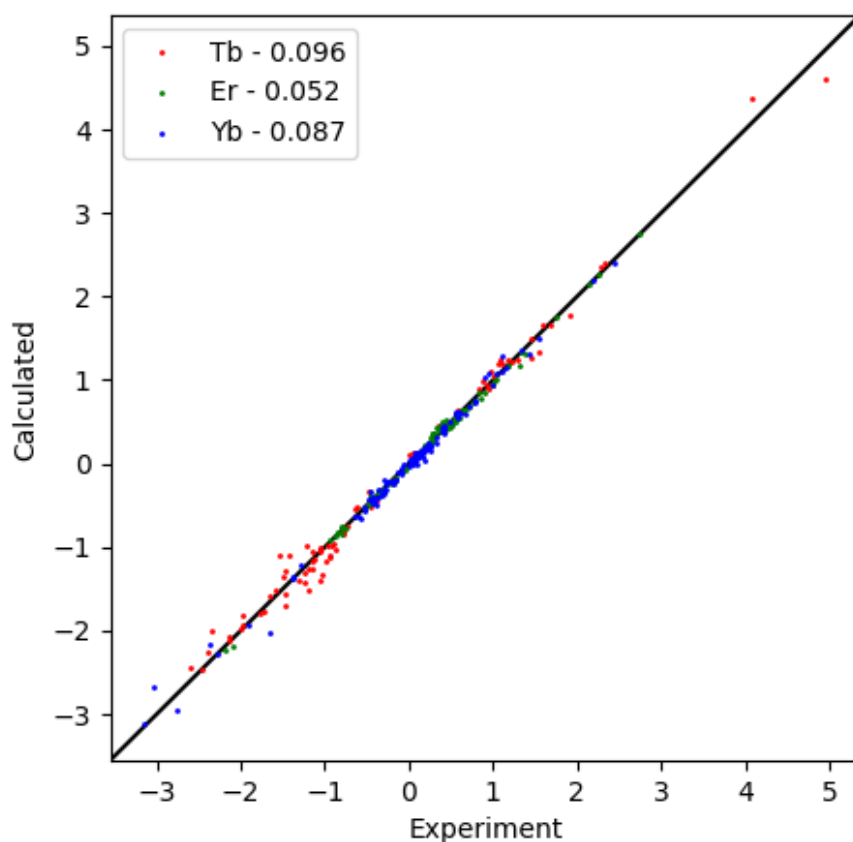
### *Yb fitted tensor*

[tensor\_Yb.txt]

ax		1E-32 m <sup>3</sup>	:	-5.392
rh		1E-32 m <sup>3</sup>	:	-2.490
x		1E-10 m	:	25.937
y		1E-10 m	:	9.481
z		1E-10 m	:	6.597
a		deg	:	129.650
b		deg	:	137.708
g		deg	:	88.796
mueff		Bm	:	0.000
shift		ppm	:	0.000
B0		T	:	18.790
temp		K	:	298.150
t1e		ps	:	0.000
taur		ns	:	0.000

### *Correlation Plot*

[pcs\_fit\_multiple.png]



### Fit Tensor to PDB with Models

This example shows how to fit a  $\Delta\chi$ -tensor to experimental PCS data using an NMR structure that contains many models. The tensor can be fit to ensemble averaged PCS values, or to individual models. An ensemble averaged PCS is the mean calculated PCS of all models. No structural averages are ever taken.

Data for calbindin D9k are used as in the previous example *Fit Tensor to PCS Data*.

### Downloads

- Download the data files `2bcb.pdb` and `calbindin_Er_HN_PCS.npc` from [here](#):
- Download the script `pcs_fit_models.py`

### Script + Explanation

Firstly, the standard preamble and loading of data.

```
from paramagpy import protein, fit, dataparse, metal

# Load data
prot = protein.load_pdb('../data_files/2bcb.pdb')
rawData = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS.npc')
mStart = metal.Metal()
mStart.position = prot[0]['A'][56]['CA'].position
```

The default method of fitting is to fit the tensor independently to each model of the PDB file. To achieve ensemble averaging behaviour, this requires setting the argument `ensembleAverage` to `True` within the fitting function. The default ensemble averaging behaviour is to average atoms with the same serial number in the PDB file. To manipulate ensemble averaging, you can specify the `idx` column of the input `dataArray` for `paramagpy.fit.nlr_fit_metal_from_pcs()`. The `idx` array contains common integers for corresponding atoms to be averaged. After fitting the Q-factor is calculated (with the `ensembleAverage` argument set to `True`), and ensemble averaging of the calculated data values is achieved with the function `paramagpy.fit.ensemble_average()`

```
#### Ensemble average fitting ####
parsedData = prot.parse(rawData)
[mGuess], [data] = fit.svd_gridsearch_fit_metal_from_pcs(
    [mStart], [parsedData], radius=10, points=10, ensembleAverage=True)
[mFit], [data] = fit.nlr_fit_metal_from_pcs([mGuess], [parsedData],
    ↪ensembleAverage=True)

qfac = fit.qfactor(data, ensembleAverage=True)
dataEAv = fit.ensemble_average(data)

mFit.save('calbindin_Er_HN_PCS_tensor_ensemble.txt')
```

Fitting a separate tensor to each model of the PDB is the default behaviour of the fitting functions, the average of all fitted tensors is then returned. The model with the minimum Q-factor is then found by looping over the calculated data and sorting them by the calculated Q-factor.

```
#### Single model fitting ####
[mFitMod], [dataMod] = fit.nlr_fit_metal_from_pcs(
    [mGuess], [parsedData])
qs = {}
for mdl in set(dataMod['mdl']):
    qs[mdl] = fit.qfactor(dataMod[dataMod['mdl']==mdl])

minModel, minQfac = sorted(qs.items(), key=lambda x: x[1])[0]
minData = dataMod[dataMod['mdl']==minModel]
```

Finally we plot three sets of data:

- The ensemble average fit calculated for each model (green)
- The ensemble average of the calculated values of the ensemble fit (red)
- The best fitting single model (blue)

Note that to calculate the ensemble average of the calculated values we use the function `paramagpy.fit.ensemble_average()`. This can take any number of arguments, and will average values based on common serial numbers of the list of atoms in the first argument.

```
# #### Plot the correlation ####
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot all models
ax.plot(data['exp'], data['cal'], marker='o', lw=0, ms=2, c='g',
        alpha=0.5, label="All models: Q = {:.4f}".format(qfac))

# Plot the ensemble average
ax.plot(dataEAv['exp'], dataEAv['cal'], marker='o', lw=0, ms=2, c='r',
        alpha=0.5, label="Ensemble Average: Q = {:.4f}".format(qfac))

# Plot the model with minimum Q-factor
ax.plot(minData['exp'], minData['cal'], marker='o', lw=0, ms=2, c='b',
        alpha=0.5, label="Best Model ({0:}): Q = {:.4f}".format(
            minModel, minQfac))
```

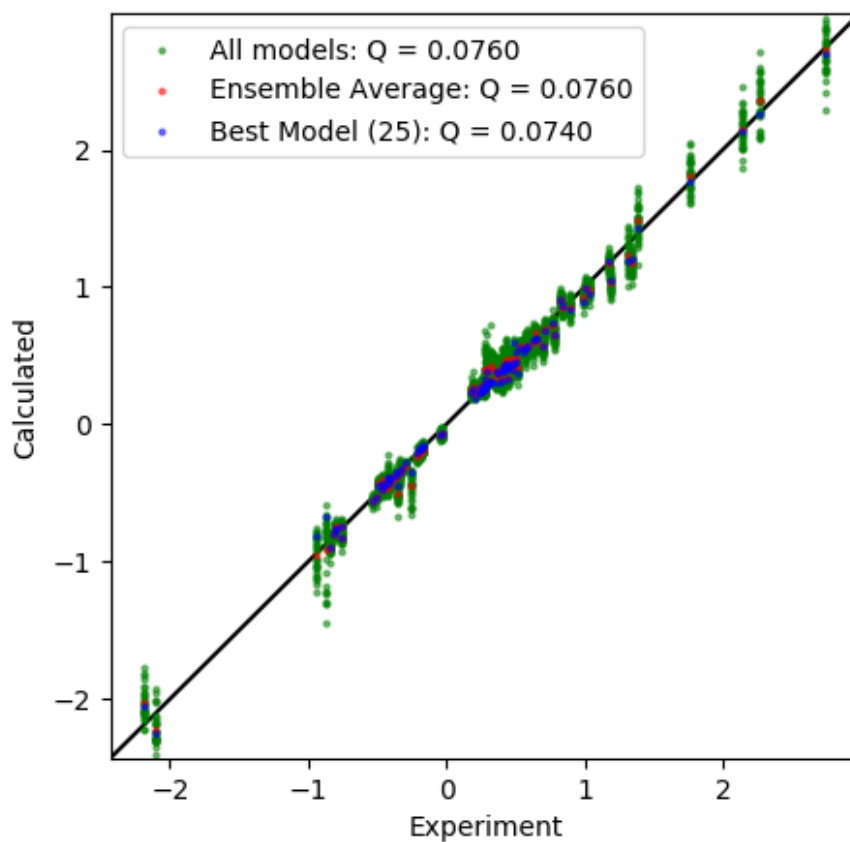
(continues on next page)

(continued from previous page)

```
# Plot a diagonal
l, h = ax.get_xlim()
ax.plot([l,h],[l,h], '-k', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pcs_fit_models.png")
```

Output: [pcs\_fit\_models.png]



## Constrained Fitting

This example shows how to fit a  $\Delta\chi$ -tensor with constraints applied. The two cases here constrain position to fit a tensor to a known metal ion position from an X-ray structure, and fit an axially symmetric tensor with only 6 of the usual 8 parameters.

## Downloads

- Download the data files `4icbH_mut.pdb` and `calbindin_Er_HN_PCS.npc` from [here](#):
- Download the script `pcs_fit_constrained.py`

## Script + Explanation

The necessary modules are imported and data is loaded

```
from paramagpy import protein, fit, dataparse, metal

# Load data
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')
rawData = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS.npc')
parsedData = prot.parse(rawData)
mStart = metal.Metal()
```

The calcium ion from the X-ray structure is contained in a heteroatom of the PDB file. We set the starting position of the tensor to this position.

```
# Set the starting position to Calcium ion heteroatom in PDB
mStart.position = prot[0]['A'][('H_ CA', 77, ' ')]['CA'].position
```

To fit the the anisotropy and orientation without position, the linear PCS equation can be solved analytically by the SVD gridsearch method but using only one point with a radius of zero. The Q-factor is then calculated and the tensor is saved.

```
# Calculate tensor by SVD
[mFit], [data] = fit.svd_gridsearch_fit_metal_from_pcs(
    [mStart], [parsedData], radius=0, points=1)

qfac = fit.qfactor(data)

mFit.save('calbindin_Er_HN_PCS_tensor_position_constrained.txt')
```

Output: `[pcs_fit_constrained.png]`

```
ax | 1E-32 m^3 : -8.152
rh | 1E-32 m^3 : -4.911
x  | 1E-10 m : 25.786
y  | 1E-10 m : 9.515
z  | 1E-10 m : 6.558
a  | deg : 125.841
b  | deg : 142.287
g  | deg : 41.758
mueff | Bm : 0.000
shift | ppm : 0.000
B0 | T : 18.790
temp | K : 298.150
tle | ps : 0.000
taur | ns : 0.000
```

To fit an axially symmetric tensor, we can use the Non-linear regression method and specify exactly which parameters we want to fit. This will be the axiality `ax`, two Euler angles `b` and `g` and the position coordinates. Note that in the output, the rhombic `rh` and alpha `a` parameters are redundant.

```
# Calculate axially symmetric tensor by NRL
[mFitAx], [dataAx] = fit.nlr_fit_metal_from_pcs(
    [mStart], [parsedData], params=('ax', 'b', 'g', 'x', 'y', 'z'))

qfacAx = fit.qfactor(dataAx)

mFitAx.save('calbindin_Er_HN_PCS_tensor_axially_symmetric.txt')
```

Output: [pcs\_fit\_constrained.png]

ax		1E-32 m <sup>3</sup> :	9.510
rh		1E-32 m <sup>3</sup> :	0.000
x		1E-10 m :	24.948
y		1E-10 m :	8.992
z		1E-10 m :	3.205
a		deg :	0.000
b		deg :	134.697
g		deg :	180.000
mueff		Bm :	0.000
shift		ppm :	0.000
B0		T :	18.790
temp		K :	298.150
t1e		ps :	0.000
taur		ns :	0.000

Finally we plot the data.

```
#### Plot the correlation ####
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data
ax.plot(data['exp'], data['cal'], marker='o', lw=0, ms=2, c='r',
        label="Position constrained: Q = {:.4f}".format(qfac))

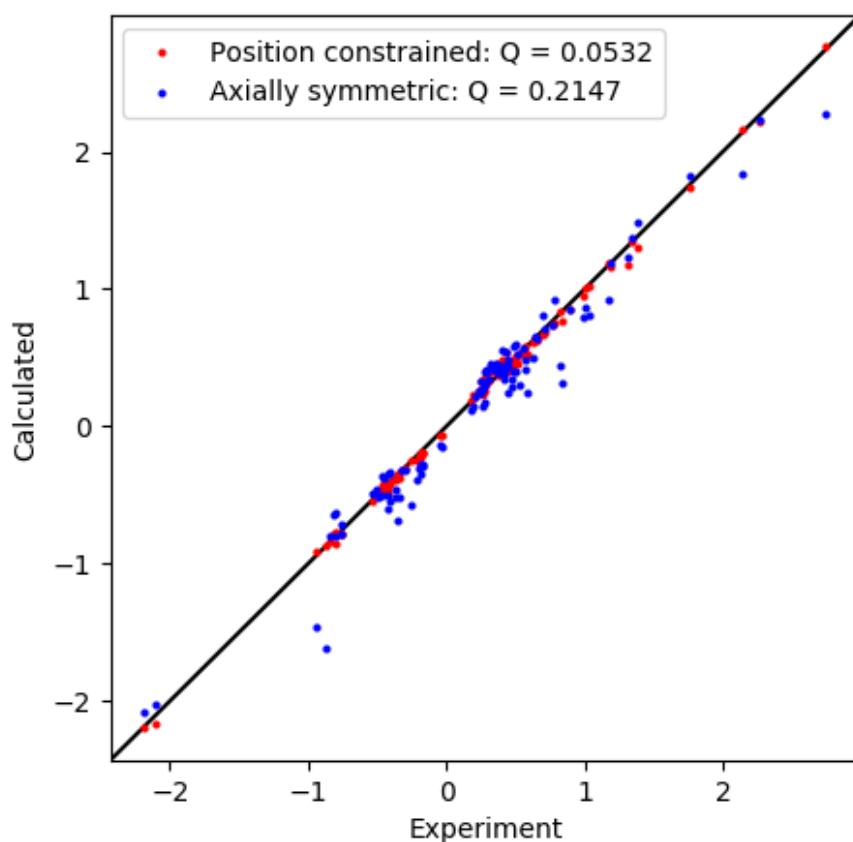
ax.plot(dataAx['exp'], dataAx['cal'], marker='o', lw=0, ms=2, c='b',
        label="Axially symmetric: Q = {:.4f}".format(qfacAx))

# Plot a diagonal
l, h = ax.get_xlim()
ax.plot([l,h],[l,h], '-k', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pcs_fit_constrained.png")
```

Output: [pcs\_fit\_constrained.png]





### Fit a tensor to PCS data with uncertainties

This example shows how to conduct a weighted fit of a  $\Delta\chi$ -tensor to experimental PCS data with experimental errors.

### Downloads

- Download the data files `4icbH_mut.pdb` and `calbindin_Er_HN_PCS_errors.npc` from [here](#):
- Download the script `pcs_fit_error.py`

### Script + Explanation

This script follows very closely the script *Fit Tensor to PCS Data*. The only difference being that errors are included in the fourth column of the `.npc` file and errorbars are included in the plotting routine.

```
from paramagpy import protein, fit, dataparse, metal

# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')

# Load the PCS data
rawData = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS_errors.npc')

# Associate PCS data with atoms of the PDB
```

(continues on next page)

(continued from previous page)

```

parsedData = prot.parse(rawData)

# Define an initial tensor
mStart = metal.Metal()

# Set the starting position to an atom close to the metal
mStart.position = prot[0]['A'][56]['CA'].position

# Calculate an initial tensor from an SVD gridsearch
[mGuess], [data] = fit.svd_gridsearch_fit_metal_from_pcs(
    [mStart], [parsedData], radius=10, points=10)

# Refine the tensor using non-linear regression
[mFit], [data] = fit.nlr_fit_metal_from_pcs([mGuess], [parsedData])

qfac = fit.qfactor(data)

# Save the fitted tensor to file
mFit.save('calbindin_Er_HN_PCS_tensor_errors.txt')

#### Plot the correlation ####
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data
ax.errorbar(data['exp'], data['cal'], xerr=data['err'], fmt='o', c='r', ms=2,
            ecol='k', capsize=3, label="Q-factor = {:.4f}".format(qfac))

# Plot a diagonal
l, h = ax.get_xlim()
ax.plot([l,h],[l,h], 'grey', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pcs_fit_error.png")

```

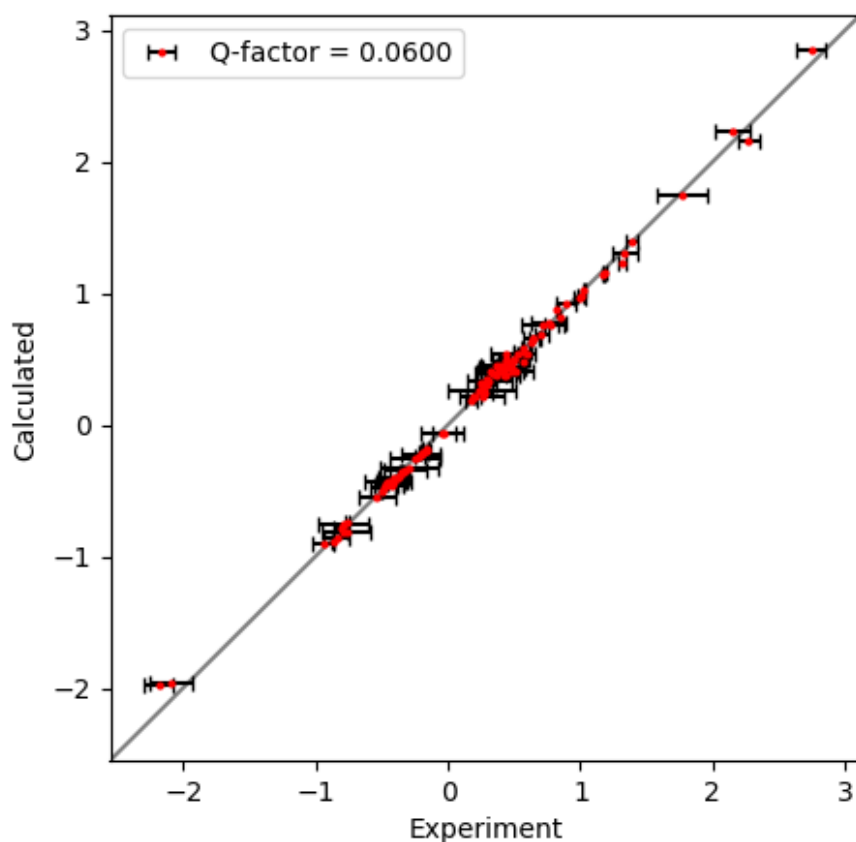
The fitted tensor:

*Output:* [calbindin\_Er\_HN\_PCS\_tensor\_errors.txt]

ax		1E-32 m <sup>3</sup>	:	-8.012
rh		1E-32 m <sup>3</sup>	:	-4.125
x		1E-10 m	:	24.892
y		1E-10 m	:	8.456
z		1E-10 m	:	6.287
a		deg	:	112.440
b		deg	:	135.924
g		deg	:	46.210
mueff		Bm	:	0.000
shift		ppm	:	0.000
B0		T	:	18.790
temp		K	:	298.150
t1e		ps	:	0.000
taur		ns	:	0.000

And correlation plot:

*Output:* [pcs\_fit\_error.png]



## Propagate Uncertainty to Fitted Tensor Parameters

This example shows the various error analysis functions available in paramagpy for estimating the uncertainty in fitted parameters for a paramagnetic center.

## Downloads

- Download the data files `2bcb.pdb` and `calbindin_Er_HN_PCS_errors.npc` from [here](#):
- Download the script `pcs_fit_uncertainty.py`

## Script + Explanation

This start of this script follows the script *Fit Tensor to PCS Data* to fit the tensor.

```
from paramagpy import protein, fit, dataparse, metal
import numpy as np

# Load the PDB file
prot = protein.load_pdb('../data_files/2bcb.pdb')

# Load the PCS data
rawData = dataparse.read_pcs('../data_files/calbindin_Er_HN_PCS_errors.npc')

# Associate PCS data with atoms of the PDB
```

(continues on next page)

(continued from previous page)

```

parsedData = prot.parse(rawData)

# Define an initial tensor
mStart = metal.Metal()

# Set the starting position to an atom close to the metal
mStart.position = prot[0]['A'][56]['CA'].position

# Calculate an initial tensor from an SVD gridsearch
[mGuess], [data] = fit.svd_gridsearch_fit_metal_from_pcs(
    [mStart], [parsedData], radius=10, points=10)

# Refine the tensor using non-linear regression
[mFit], [data] = fit.nlr_fit_metal_from_pcs([mGuess], [parsedData])

```

## Uncertainty from structure models

The PDB file contains models that capture uncertainty in the structure of the protein. This can be propagated to estimate uncertainty in the fitted tensor parameters using the function `paramagpy.fit.fit_error_model()`. This fits a separate tensor to each model and returns all fitted tensors as well as the standard deviation in the fitted parameters.

```

# Estimate uncertainty sourcing noise from the models of the PDB
[mod_all], [mod_std] = fit.fit_error_models(fit.nlr_fit_metal_from_pcs,
    initMetals=[mFit], dataArrays=[parsedData])

mod_std.save('error_tensor_models.txt')

```

The standard deviation in the fitted tensor parameters is found in the variable `mod_std`. This variation in tensor principle axes can be viewed by a Sanson-Flamsteed plot.

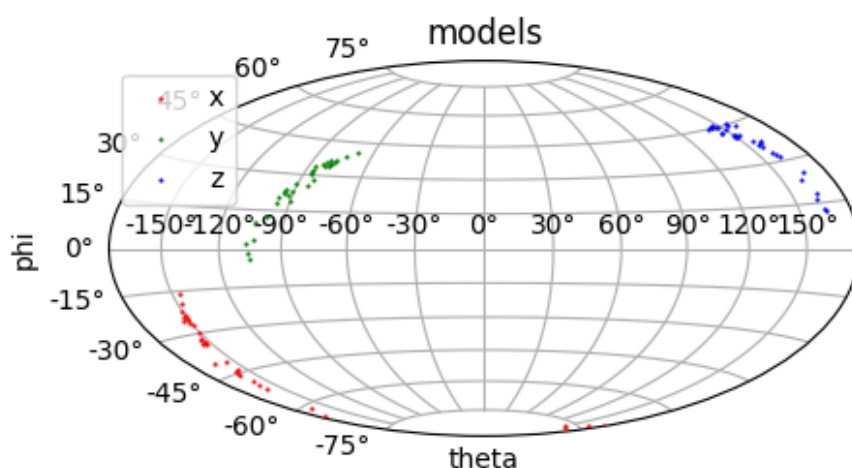
*Output:* [error\_tensor\_models.txt]

```

ax      | 1E-32 m^3 :    0.556
rh      | 1E-32 m^3 :    0.525
x       | 1E-10 m   :    0.756
y       | 1E-10 m   :    0.695
z       | 1E-10 m   :    0.957
a       |          deg :    7.466
b       |          deg :    9.948
g       |          deg :   19.294
mueff   |          Bm :    0.000
shift   |          ppm :    0.000
B0      |          T  :    0.000
temp    |          K  :    0.000
tle     |          ps :    0.000
taur    |          ns :    0.000

```

*Output:* [models.png]



### Uncertainty from experimental uncertainties

Experimental uncertainties can be measured. This may arise due to spectral noise in peak heights for PREs, or spectral noise as uncertainties in chemical shifts for PCSs, as is the case here. The function `paramagpy.fit.fit_error_monte_carlo()` will repeat the fit for many iterations, each time adding random noise from a uniform distribution scaled by the experimental errors present in the `err` column of the `dataArray` `parsedData`.

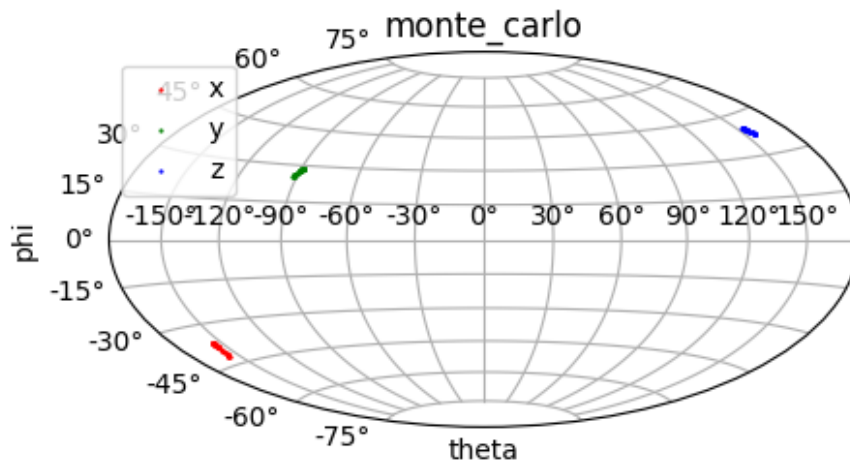
```
# Estimate uncertainty sourcing noise from experimental uncertainties
[mc_all], [mc_std] = fit.fit_error_monte_carlo(fit.nlr_fit_metal_from_pcs,
                                              50, initMetals=[mFit], dataArray=[parsedData])

mod_std.save('error_tensor_monte_carlo.txt')
```

Output: [error\_tensor\_monte\_carlo.txt]

ax		1E-32 m^3 :	0.556
rh		1E-32 m^3 :	0.525
x		1E-10 m :	0.756
y		1E-10 m :	0.695
z		1E-10 m :	0.957
a		deg :	7.466
b		deg :	9.948
g		deg :	19.294
mueff		Bm :	0.000
shift		ppm :	0.000
B0		T :	0.000
temp		K :	0.000
t1e		ps :	0.000
taur		ns :	0.000

Output: [monte\_carlo.png]



### Uncertainty from sample fraction

A final, but generally not recommended method is to source noise from taking a random fraction of the data and conducting the fit for many iterations to then view the deviation in fitted parameters. This method is often called bootstrapping and is desirable if the experimental uncertainties are unknown and the PDB file does not contain models that capture structural uncertainty. The function `paramagpy.fit.fit_error_bootstrap()` will repeat the fit for many iterations, each time sampling the desired amount of the experimental data randomly.

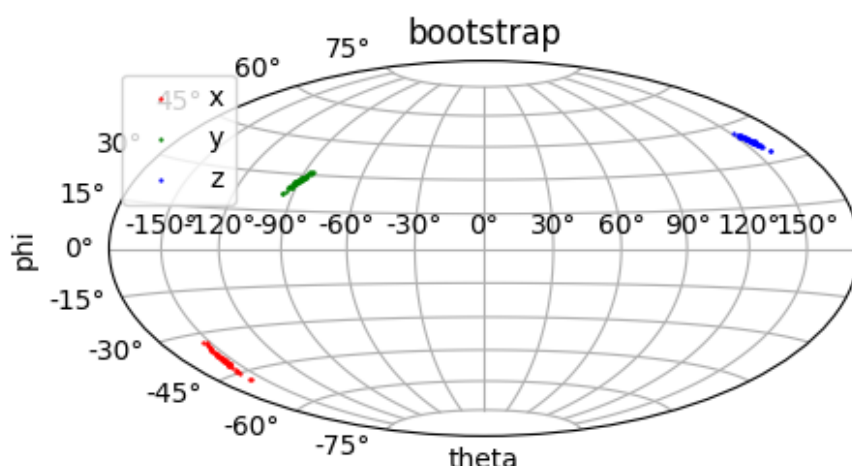
```
# Estimate uncertainty sourcing noise from sample fractions
[bs_all], [bs_std] = fit.fit_error_bootstrap(fit.nlr_fit_metal_from_pcs,
      50, 0.8, initMetals=[mFit], dataArrays=[parsedData])

mod_std.save('error_tensor_bootstrap.txt')
```

Output: [error\_tensor\_bootstrap.txt]

ax		1E-32 m <sup>3</sup>	:	0.556
rh		1E-32 m <sup>3</sup>	:	0.525
x		1E-10 m	:	0.756
y		1E-10 m	:	0.695
z		1E-10 m	:	0.957
a		deg	:	7.466
b		deg	:	9.948
g		deg	:	19.294
mueff		Bm	:	0.000
shift		ppm	:	0.000
B0		T	:	0.000
temp		K	:	0.000
t1e		ps	:	0.000
taur		ns	:	0.000

Output: [bootstrap.png]



This piece of code is used to generate the Sanson-Flamsteed projection plots

```
#### Plot Sanson-Flamsteed ####
from matplotlib import pyplot as plt

def transform(vector):
    x, y, z = vector
    theta = np.arctan2(y, x)
    phi = -np.arccos(z) + np.pi/2.
    return theta, phi

for name, mset in [('models', mod_all), ('monte_carlo', mc_all), ('bootstrap', bs_
    all)]:
    spcoords = []
    for m in mset:
        x, y, z = m.rotationMatrix.T
        spcoords.append(tuple(map(transform, [x,y,z])))
    points = zip(*spcoords)
    fig = plt.figure(figsize=(5, 3), dpi=100)
    ax = fig.add_subplot(111, projection='hammer')
    ax.set_xlabel("theta")
    ax.set_ylabel("phi")
    ax.set_title(name)
    ax.grid()
    for data, col, label in zip(points, ['r', 'g', 'b'], ['x', 'y', 'z']):
        theta, phi = zip(*data)
        ax.scatter(theta, phi, s=0.4, c=col, label=label, zorder=10)
    ax.legend()
    fig.savefig("{}_{}.png".format(name, mset))
```

## Fit to PCS data with offset, RACS and RADS corrections

### 5.2.2 RDC data

#### Fit Tensor to RDC Data

This example shows how to fit a  $\Delta\chi$ -tensor or equivalently, an alignment tensor to experimental RDC data. These data are taken from a Tb<sup>3+</sup> tagged ubiquitin mutant:

Benjamin J. G. Pearce, Shereen Jabar, Choy-Theng Loh, Monika Szabo, Bim Graham, Gottfried Otting (2017) Structure restraints from heteronuclear pseudocontact shifts generated by lanthanide tags at two different sites *J.*

*Biomol. NMR* 68:19-32

## Downloads

- Download the data files 2kox.pdb, ubiquitin\_a28c\_c1\_Tb\_HN.rdc and ubiquitin\_s57c\_c1\_Tb\_HN.rdc from [here](#):
- Download the script rdc\_fit.py

## Script + Explanation

Firstly, the necessary modules are imported from paramagpy. And the two RDC datasets are loaded. Because this PDB contains over 600 models, loading may take a few seconds

```
from paramagpy import protein, fit, dataparse, metal

# Load the PDB file
prot = protein.load_pdb('../data_files/2kox.pdb')

# Load the RDC data
rawData1 = dataparse.read_rdc('../data_files/ubiquitin_a28c_c1_Tb_HN.rdc')
rawData2 = dataparse.read_rdc('../data_files/ubiquitin_s57c_c1_Tb_HN.rdc')

# Associate RDC data with atoms of the PDB
parsedData1 = prot.parse(rawData1)
parsedData2 = prot.parse(rawData2)
```

Two starting metals are initialised. It is important here to set the magnetic field strength and temperature.

```
# Define an initial tensor
mStart1 = metal.Metal(B0=18.8, temperature=308.0)
mStart2 = metal.Metal(B0=18.8, temperature=308.0)
```

The alignment tensor is solved using the function `paramagpy.fit.svd_fit_metal_from_rdc()` which return a tuple of (metal, calculated), where metal is the fitted metal, calculated is the calculated RDC values. The tensors are then saved. Note that we set the argument `ensembleAverage` to `True`. This is important because the PDB structure represents an MD simulation. If set to `False`, a much smaller tensor would be fit.

```
# Calculate the tensor using SVD
[sol1], [data1] = fit.svd_fit_metal_from_rdc([mStart1], [parsedData1],
↪ensembleAverage=True)
[sol2], [data2] = fit.svd_fit_metal_from_rdc([mStart2], [parsedData2],
↪ensembleAverage=True)

# Save the fitted tensor to file
sol1.save('ubiquitin_a28c_c1_Tb_tensor.txt')
sol2.save('ubiquitin_s57c_c1_Tb_tensor.txt')
```

Output: [ubiquitin\_a28c\_c1\_Tb\_tensor.txt]

```
ax | 1E-32 m^3 : -4.776
rh | 1E-32 m^3 : -1.397
x  | 1E-10 m : 0.000
y  | 1E-10 m : 0.000
z  | 1E-10 m : 0.000
a  | deg : 16.022
b  | deg : 52.299
g  | deg : 83.616
```

(continues on next page)



(continued from previous page)

mueff		Bm :	0.000
shift		ppm :	0.000
B0		T :	18.800
temp		K :	308.000
t1e		ps :	0.000
taur		ns :	0.000

*Output:* [ubiquitin\_s57c\_c1\_Tb\_tensor.txt]

ax		1E-32 m^3 :	-5.930
rh		1E-32 m^3 :	-1.899
x		1E-10 m :	0.000
y		1E-10 m :	0.000
z		1E-10 m :	0.000
a		deg :	9.976
b		deg :	99.463
g		deg :	37.410
mueff		Bm :	0.000
shift		ppm :	0.000
B0		T :	18.800
temp		K :	308.000
t1e		ps :	0.000
taur		ns :	0.000

The experimental/calculated correlations are then plotted. The tensor is by default fitted to the ensemble averaged calculated values. Backcalculation of all models is shown here, as well as the ensemble average.

```
#### Plot the correlation ####
from matplotlib import pyplot as plt
fig = plt.figure(figsize=(5,10))
ax1 = fig.add_subplot(211)
ax2 = fig.add_subplot(212)
ax1.set_title('A28C-C1-Tb')
ax2.set_title('S57C-C1-Tb')

for sol, ax, data in zip([sol1,sol2], [ax1,ax2], [data1,data2]):

    # Calculate ensemble averages
    dataEAv = fit.ensemble_average(data)

    # Calculate the Q-factor
    qfac = fit.qfactor(data, ensembleAverage=True)

    # Plot all models
    ax.plot(data['exp'], data['cal'], marker='o', lw=0, ms=2, c='b',
            alpha=0.5, label="All models: Q = {:.4f}".format(qfac))

    # Plot the ensemble average
    ax.plot(dataEAv['exp'], dataEAv['cal'], marker='o', lw=0, ms=2, c='r',
            label="Ensemble Average: Q = {:.4f}".format(qfac))

    # Plot a diagonal
    l, h = ax.get_xlim()
    ax.plot([l,h],[l,h], '-k', zorder=0)
    ax.set_xlim(l,h)
    ax.set_ylim(l,h)

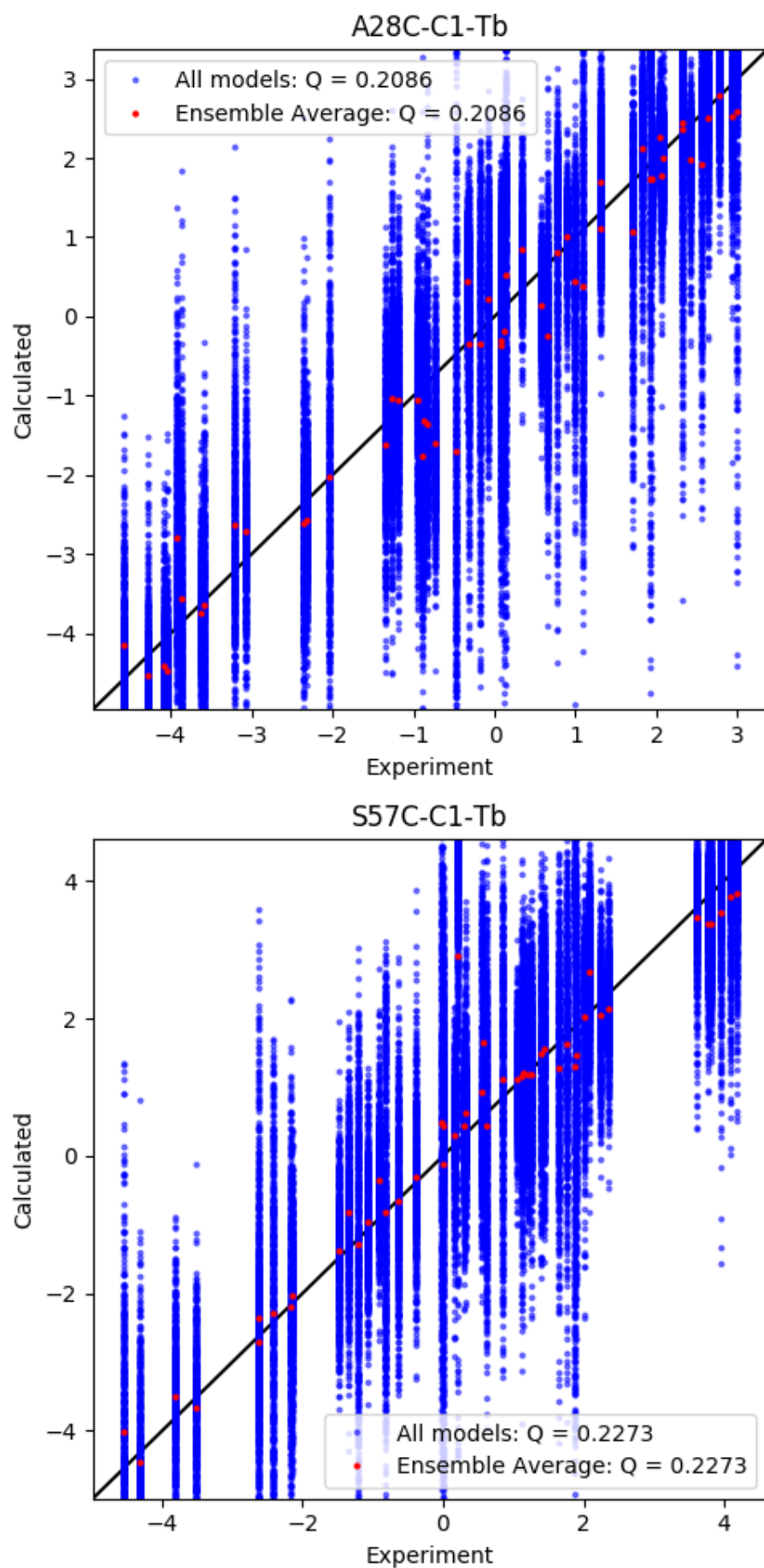
    # Make axis labels and save figure
    ax.set_xlabel("Experiment")
    ax.set_ylabel("Calculated")
    ax.legend()
```

(continues on next page)

(continued from previous page)

```
fig.tight_layout()  
fig.savefig("rdc_fit.png")
```

*Output:* [rdc\_fit.png]



## Calculate RDC from a known Tensor

This example shows how to calculate theoretical RDC values from a known  $\Delta\chi$ -tensor which has been fitted from PCS data. Paramagpy allows seamless calculation of one PCS/PRE/RDC/CCR effect from a tensor fitted from another effect.

## Downloads

- Download the data files `4icbH_mut.pdb` and `calbindin_Er_HN_PCS_tensor.txt` from [here](#):
- Download the script `rdc_calculate.py`

## Script + Explanation

First the relevant modules are loaded, the protein is loaded and the metal is loaded from file. The magnetic field strength and temperature are also set.

```
from paramagpy import protein, metal

# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')

# Load the fitted tensor
met = metal.load_tensor('../data_files/calbindin_Er_HN_PCS_tensor.txt')
met.B0 = 18.8
```

A loop is made over the atoms of the protein. The amide H and N atoms are selected and then the RDC value is calculated. Finally the formatted data is appended to list `forFile`.

```
forFile = []
for atom in prot.get_atoms():
    if atom.name == 'H':
        residue = atom.parent
        seq = residue.id[1]
        if 'N' in residue:
            H = atom
            N = residue['N']
            rdc = met.atom_rdc(H, N)
            line = "{0:2d} {1:^3s} {2:2d} {3:^3s} {4:6.3f} 0.0\n".
↪format(
                                seq, H.name, seq, N.name, rdc)
            forFile.append(line)
```

The formatted data is written to file:

```
with open("calbindin_Er_RDC_calc.rdc", 'w') as f:
    f.writelines(forFile)
```

Output: `[calbindin_Er_RDC_calc.rdc]`

```
0 H 0 N -1.724 0.0
1 H 1 N -6.196 0.0
2 H 2 N -4.993 0.0
4 H 4 N -0.922 0.0
5 H 5 N 1.783 0.0
6 H 6 N 0.280 0.0
7 H 7 N -1.906 0.0
8 H 8 N 1.056 0.0
9 H 9 N 0.713 0.0
```

(continues on next page)

(continued from previous page)

10	H	10	N	0.213	0.0
11	H	11	N	-0.881	0.0
12	H	12	N	2.712	0.0
13	H	13	N	0.614	0.0
14	H	14	N	-2.346	0.0
15	H	15	N	1.659	0.0
16	H	16	N	0.648	0.0
17	H	17	N	0.383	0.0
18	H	18	N	0.420	0.0
19	H	19	N	-7.863	0.0
21	H	21	N	0.973	0.0
22	H	22	N	1.026	0.0
23	H	23	N	-0.613	0.0
24	H	24	N	-5.847	0.0
25	H	25	N	1.761	0.0
26	H	26	N	6.470	0.0
27	H	27	N	5.541	0.0
28	H	28	N	-0.334	0.0
29	H	29	N	3.624	0.0
30	H	30	N	6.673	0.0
31	H	31	N	3.952	0.0
32	H	32	N	1.658	0.0
33	H	33	N	5.449	0.0
34	H	34	N	7.370	0.0
35	H	35	N	1.033	0.0
36	H	36	N	1.136	0.0
38	H	38	N	-7.378	0.0
39	H	39	N	-6.979	0.0
40	H	40	N	-4.810	0.0
41	H	41	N	-3.187	0.0
42	H	42	N	2.415	0.0
43	H	43	N	1.710	0.0
44	H	44	N	-5.977	0.0
45	H	45	N	-5.467	0.0
46	H	46	N	3.243	0.0
47	H	47	N	3.937	0.0
48	H	48	N	7.047	0.0
49	H	49	N	4.577	0.0
50	H	50	N	3.718	0.0
51	H	51	N	4.519	0.0
52	H	52	N	6.077	0.0
53	H	53	N	2.940	0.0
54	H	54	N	2.541	0.0
55	H	55	N	-7.493	0.0
56	H	56	N	-7.159	0.0
57	H	57	N	4.948	0.0
58	H	58	N	-1.078	0.0
59	H	59	N	-0.759	0.0
60	H	60	N	0.161	0.0
61	H	61	N	-1.132	0.0
62	H	62	N	-5.719	0.0
63	H	63	N	4.025	0.0
64	H	64	N	5.929	0.0
65	H	65	N	2.363	0.0
66	H	66	N	2.477	0.0
67	H	67	N	8.265	0.0
68	H	68	N	5.078	0.0
69	H	69	N	3.724	0.0
70	H	70	N	7.743	0.0
71	H	71	N	2.188	0.0
72	H	72	N	4.911	0.0

(continues on next page)

(continued from previous page)

```
73  H   73  N   7.514 0.0
74  H   74  N  -0.001 0.0
75  H   75  N   1.119 0.0
```

## 5.2.3 PRE data

### Fit Tensor to PRE Data

This example demonstrates fitting of the rotational correlation time  $\tau_r$  to 1H PRE data of calbindin D9k. You can fit any parameters of the  $\chi$ -tensor you desire, such as position or magnitude as well.

### Downloads

- Download the data files `4icbH_mut.pdb`, `calbindin_Er_H_R2_600.npc` and `calbindin_Tb_H_R2_800.npc` from [here](#):
- Download the script `pre_fit_proton.py`

### Script + Explanation

Firstly, the necessary modules are imported from paramagpy.

```
from paramagpy import protein, fit, dataparse, metal
```

The protein is then loaded from a PDB file.

```
# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')
```

The PRE data is loaded. Note that the Er data was recorded at 600 MHz and the Tb data was recorded at 800 MHz.

```
rawData_er = dataparse.read_pre('../data_files/calbindin_Er_H_R2_600.pre')
rawData_tb = dataparse.read_pre('../data_files/calbindin_Tb_H_R2_800.pre')
```

The  $\Delta\chi$ -tensors that were fitted from PCS data are loaded from file and the relevant  $B_0$  magnetic field strengths are set.

```
mStart_er = metal.load_tensor('../data_files/calbindin_Er_HN_PCS_tensor.txt')
mStart_tb = metal.load_tensor('../data_files/calbindin_Tb_HN_PCS_tensor.txt')
mStart_er.B0 = 14.1
mStart_tb.B0 = 18.8
```

Fitting of the rotational correlation time is done with the function `paramagpy.fit.nlr_fit_metal_from_pre()`. To fit position or  $\chi$ -tensor magnitude, you can change the `params` argument.

```
(m_er,), (cal_er,) = fit.nlr_fit_metal_from_pre(
    [mStart_er], [data_er], params=['taur'], rtypes=['r2'])
(m_tb,), (cal_tb,) = fit.nlr_fit_metal_from_pre(
    [mStart_tb], [data_tb], params=['taur'], rtypes=['r2'])
```

The fitted tensors are saved to file. Note that the Er dataset gives a reasonable  $\tau_r$  of around 4 ns which is close to the literature value of 4.25 ns. However, the Tb dataset gives an unreasonably large value of 18 ns. This is due to magnetisation attenuation due to 1H-1H RDCs present during the relaxation evolution time as discussed in [literature](#) giving rise to artificially large measured PREs for lanthanides with highly anisotropic  $\Delta\chi$ -tensors. This is also reflected in the correlation plot below.

```
m_er.save('calbindin_Er_H_R2_600_tensor.txt')
m_tb.save('calbindin_Tb_H_R2_800_tensor.txt')
```

*Output:* [calbindin\_Er\_H\_R2\_600\_tensor.txt]

```
ax      | 1E-32 m^3 :    -8.152
rh      | 1E-32 m^3 :    -4.911
x       | 1E-10 m   :    25.786
y       | 1E-10 m   :     9.515
z       | 1E-10 m   :     6.558
a       |          deg :   125.841
b       |          deg :   142.287
g       |          deg :    41.758
mueff   |          Bm :     9.581
shift   |          ppm :     0.000
B0       |          T  :    14.100
temp    |          K  :   298.150
tle     |          ps :     0.189
taur    |          ns :     3.923
```

*Output:* [calbindin\_Tb\_H\_R2\_800\_tensor.txt]

```
ax      | 1E-32 m^3 :    30.375
rh      | 1E-32 m^3 :    12.339
x       | 1E-10 m   :    25.786
y       | 1E-10 m   :     9.515
z       | 1E-10 m   :     6.558
a       |          deg :   150.957
b       |          deg :   152.671
g       |          deg :    70.311
mueff   |          Bm :     9.721
shift   |          ppm :     0.000
B0       |          T  :    18.800
temp    |          K  :   298.150
tle     |          ps :     0.251
taur    |          ns :    18.917
```

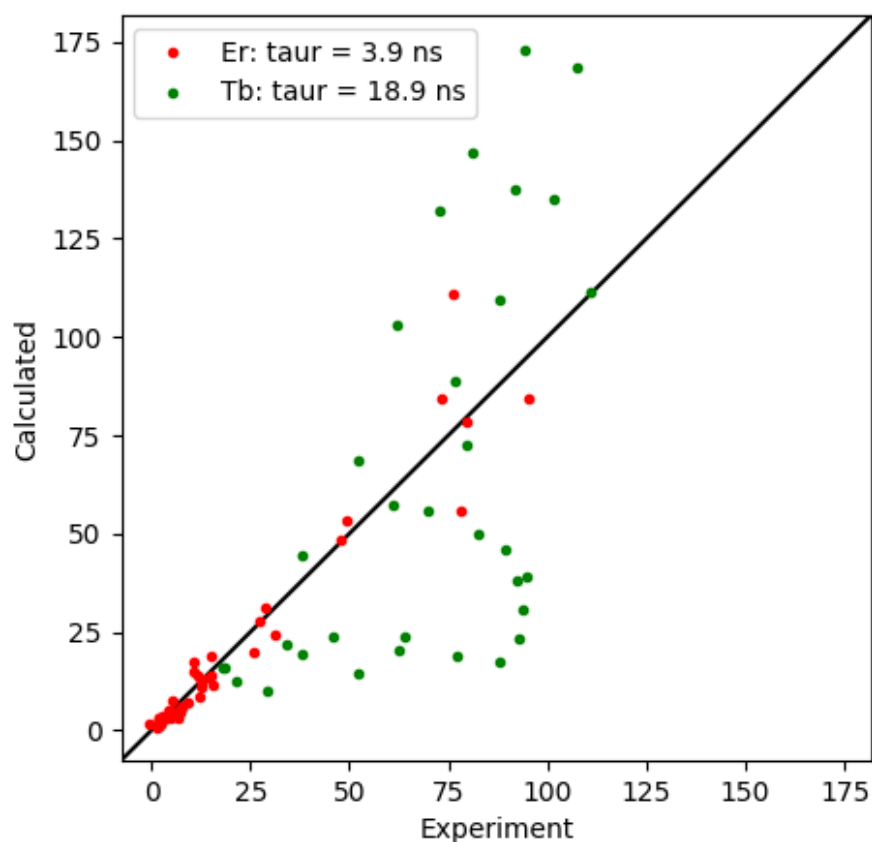
And the results are plotted.

```
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data
ax.plot(cal_er['exp'], cal_er['cal'], marker='o', lw=0, ms=3, c='r',
        label="Er: taur = {:.1f} ns".format(1E9*m_er.taur))
ax.plot(cal_tb['exp'], cal_tb['cal'], marker='o', lw=0, ms=3, c='g',
        label="Tb: taur = {:.1f} ns".format(1E9*m_tb.taur))

# Plot a diagonal
l, h = ax.get_ylim()
ax.plot([l,h],[l,h], '-k', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pre_fit_proton.png")
```



### Calculate $^{15}\text{N}$ PREs with cross-correlation effects

This example shows how to conduct a weighted fit of a  $\Delta\chi$ -tensor to experimental PCS data with experimental errors.

### Downloads

- Download the data files `4icbH_mut.pdb`, `calbindin_Tb_N_R1_600.pre` and `calbindin_Tb_HN_PCS_tensor.txt` from [here](#):
- Download the script `pre_calc_nitrogen.py`

### Script + Explanation

First the relevant modules are loaded, the protein and data are read and the data is parsed by the protein.

```
from paramagpy import protein, metal, dataparse

# Load the PDB file
prot = protein.load_pdb('../data_files/4icbH_mut.pdb')

# Load PRE data
rawData = dataparse.read_pre('../data_files/calbindin_Tb_N_R1_600.pre')
```

(continues on next page)



(continued from previous page)

```
# Parse PRE data
data = prot.parse(rawData)
```

The Tb tensor fitted from PCS data is loaded and the relevant parameters, in this case the magnetic field strength, temperature and rotational correlation time are set.

```
met = metal.load_tensor('../data_files/calbindin_Tb_HN_PCS_tensor.txt')
met.B0 = 14.1
met.T = 298.0
met.taur = 4.25E-9
```

A loop is conducted over the nitrogen atoms that are present in the experimental data. The PRE is calculated using the function `paramagpy.metal.atom_pre()`. Calculations without CSA are appended to the list `cal` and calculations including CSA cross-correlation with the Curie-spin relaxation are appended to the list `cal_csa`.

```
exp = []
cal = []
cal_csa = []
for atom, pre, err in data[['atm', 'exp', 'err']]:
    exp.append(pre)
    cal.append(met.atom_pre(atom, rtype='r1'))
    cal_csa.append(met.atom_pre(atom, rtype='r1', csa=atom.csa))
```

Finally the data are plotted. Clearly CSA cross-correlation is a big effect for backbone nitrogen atoms and should always be taken into account for Curie-spin calculations. Also note the existence and correct prediction of negative PREs!

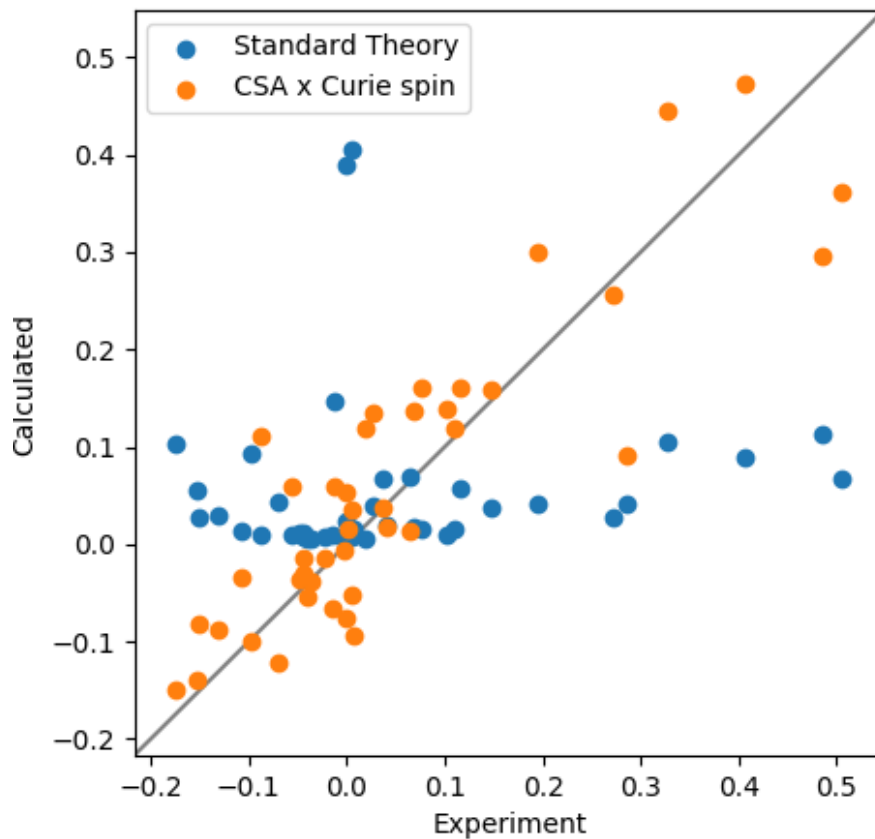
```
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data
ax.scatter(exp, cal, label="Standard Theory")
ax.scatter(exp, cal_csa, label="CSA x Curie spin")

# Plot a diagonal
l, h = ax.get_xlim()
ax.plot([l,h],[l,h], 'grey', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("pre_calc_nitrogen.png")
```

Output: [pre\_calc\_nitrogen.png]



### Fit spectral power density tensor

This example shows how to fit the spectral power density tensor to anisotropic PREs. The data and theory are derived from <https://doi.org/10.1039/C8CP01332B>.

### Downloads

- Download the data files `parashift_Tb.pdb` and `parashift_Tb_R1_exp.pre` from [here](#):
- Download the script `pre_fit_aniso_dipolar.py`

### Script + Explanation

Load the relevant modules, read the PDB coordinates and experimental PRE values. Parse the values.

```
from paramagpy import protein, metal, fit, dataparse
from matplotlib import pyplot as plt
import numpy as np

prot = protein.load_pdb('../data_files/parashift_Tb.pdb')
pre_exp = dataparse.read_pre('../data_files/parashift_Tb_R1_exp.pre')
exp = prot.parse(pre_exp)
```

The spectral power density tensor is written here explicitly and set to the attribute `g_tensor`. The values here are sourced from the original paper, and arise from the robust linear fit to the experimental data. We will use this tensor for comparison to the fit achieved by paramagpy.

```
m = metal.Metal(taur=0.42E-9, B0=1.0, temperature=300.0)
m.set_lanthanide('Tb')

m.g_tensor = np.array([
    [1754.0, -859.0, -207.0],
    [-859.0, 2285.0, -351.0],
    [-207.0, -351.0, -196.0]]) * 1E-60
```

An starting tensor with no parameters is also initialised and will be used for fitting to the experimental data with paramagpy.

```
m0 = metal.Metal(taur=0.42E-9, B0=1.0, temperature=300.0)
m0.set_lanthanide('Tb')
```

The fit is conducted by setting the `usegsbm` flag to `True`. This uses anisotropic SBM theory to fit the spectral power density tensor in place of the isotropic SBM theory. The relevant fitting parameters must be specified as `'tle'`, `'gax'`, `'grh'`, `'a'`, `'b'`, `'g'` which represent the electronic relaxation time, the axial and rhombic componenets of the power spectral density tensor and the 3 Euler angles alpha, beta and gamma respectively. Note that the fitted `tle` parameter is only an estimate of the electronic relaxation time.

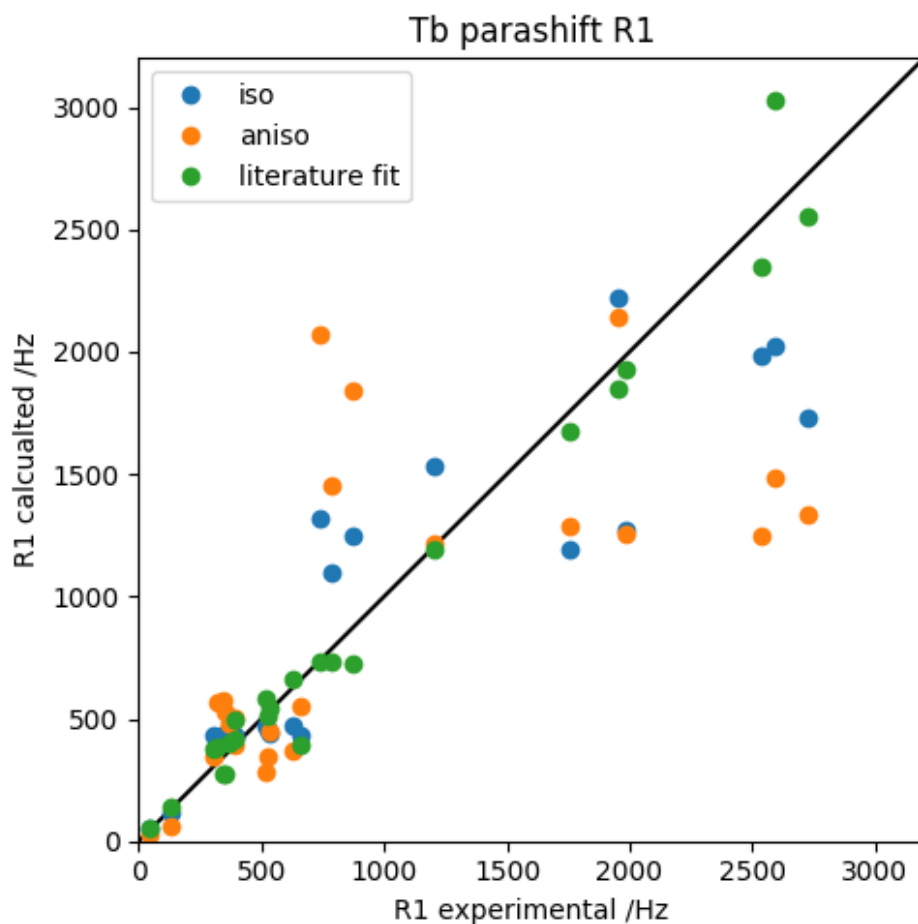
```
[mfit], [data] = fit.nlr_fit_metal_from_pre([m0], [exp], params=('tle', 'gax', 'grh',
↪ 'a', 'b', 'g'),
```

Finally the results of the fit are plotted alongside the isotropic theory and the literature fit. Note that the difference in the fit from paramagpy is small, and probably arises because the original paper uses a *Robust* linear fit, which may include weighting with experimental uncertainties. However paramagpy weights values evenly here because the experimental uncertainties are unknown.

```
pos = np.array([a.position for a in exp['atm']])
gam = np.array([a.gamma for a in exp['atm']])

fig = plt.figure(figsize=(5,5))
ax = fig.add_subplot(111)
ax.plot([0,3200], [0,3200], '-k')
ax.plot(exp['exp'], mfit.fast_sbm_r1(pos, gam), marker='o', lw=0, label='iso')
ax.plot(exp['exp'], mfit.fast_g_sbm_r1(pos, gam), marker='o', lw=0, label='aniso')
ax.plot(exp['exp'], m.fast_g_sbm_r1(pos, gam), marker='o', lw=0, label='literature_
↪ fit')
ax.set_xlim(0,3200)
ax.set_ylim(0,3200)
ax.set_xlabel("R1 experimental /Hz")
ax.set_ylabel("R1 calcualted /Hz")
ax.set_title("Tb parashift R1")
ax.legend()
fig.tight_layout()
fig.savefig("pre_fit_aniso_dipolar.png")
```

Output: [pre\_fit\_aniso\_dipolar.png]



## 5.2.4 CCR data

### Calculate Cross-correlated Relaxation

This example shows how to calculate dipole-dipole/Curie-spin cross-correlated relaxation as measured for data in the literature by [Pintacuda et. al.](#)

### Downloads

- Download the data files `1bzhH.pdb`, `myoglobin_cn.ccr` and `myoglobin_f.ccr` from [here](#):
- Download the script `ccr_calculate.py`

### Script + Explanation

First the relevant modules are loaded, and the iron atom (paramagnetic centre) is identified as the variable `ironAtom`.

```
from paramagpy import protein, fit, dataparse, metal
import numpy as np

# Load the PDB file and get iron centre
prot = protein.load_pdb('../data_files/1bzhH.pdb')
ironAtom = prot[0]['A'][("H_HEM", 154, " ")]['FE']
```

Two paramagnetic centres are defined for the high and low spin iron atom. The positions are set to that of the iron centre along with other relevant parameters. The measured isotropic  $\chi$ -tensor magnitudes are also set.

```
met_cn = metal.Metal(position=ironAtom.position,
                     B0=18.79,
                     temperature=303.0,
                     taur=5.7E-9)

met_f = met_cn.copy()
met_cn.iso = 4.4E-32
met_f.iso = 30.1E-32
```

The experimental data are loaded and parsed by the protein.

```
data_cn = prot.parse(dataparse.read_ccr("../data_files/myoglobin_cn.ccr"))
data_f = prot.parse(dataparse.read_ccr("../data_files/myoglobin_f.ccr"))
```

A loop is conducted over the atoms contained in the experimental data and the CCR rate is calculated using the function `paramagpy.metal.Metal.atom_ccr()`. These are appended to lists `compare_cn` and `compare_f`.

Note that the two H and N atoms are provided. The first atom is the nuclear spin undergoing active relaxation. The second atom is the coupling partner. Thus by swapping the H and N atoms to give `atom_ccr(N, H)`, the differential line broadening can be calculated in the indirect dimension.

```
# Calculate the cross-correlated relaxation
compare_cn = []
for H, N, value, error in data_cn[['atm', 'atx', 'exp', 'err']]:
    delta = met_cn.atom_ccr(H, N)
    compare_cn.append((value, delta*0.5))

compare_f = []
for H, N, value, error in data_f[['atm', 'atx', 'exp', 'err']]:
    delta = met_f.atom_ccr(H, N)
    compare_f.append((value, delta*0.5))
```

Finally a correlation plot is made.

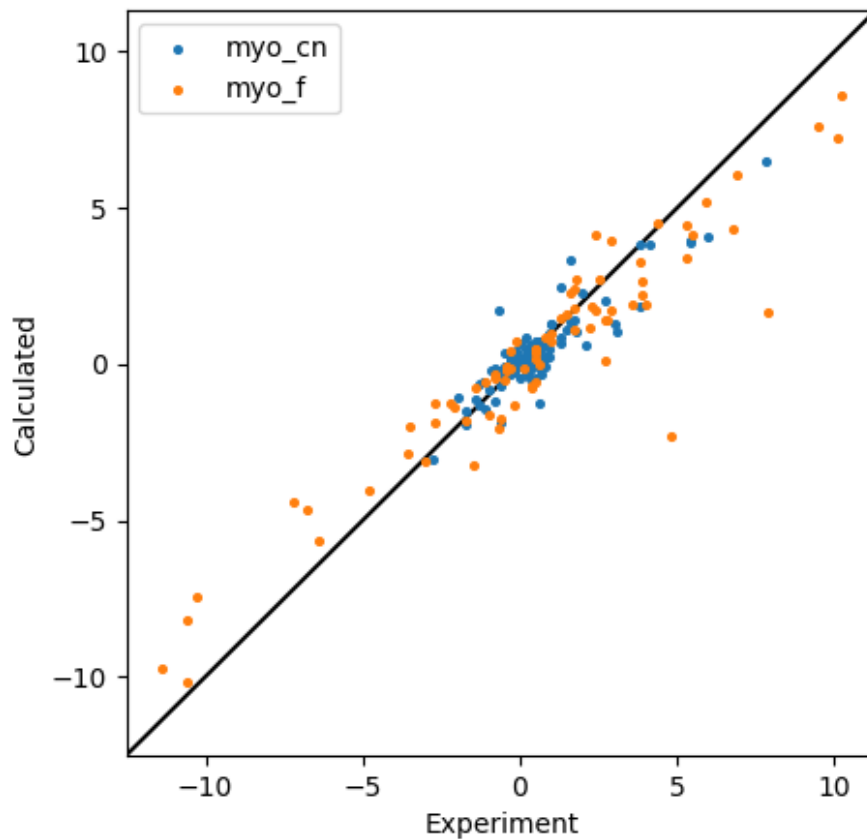
```
#### Plot the correlation ####
from matplotlib import pyplot as plt
fig, ax = plt.subplots(figsize=(5,5))

# Plot the data correlations
ax.scatter(*zip(*compare_cn), s=7, label="myo_cn")
ax.scatter(*zip(*compare_f), s=7, label="myo_f")

# Plot a diagonal
l, h = ax.get_xlim()
ax.plot([l,h],[l,h], '-k', zorder=0)
ax.set_xlim(l,h)
ax.set_ylim(l,h)

# Make axis labels and save figure
ax.set_xlabel("Experiment")
ax.set_ylabel("Calculated")
ax.legend()
fig.savefig("ccr_calculate.png")
```

Output: [ccr\_calculate.png]



## 5.3 Graphic User Interface (GUI)

Paramagpy is equipped with a GUI which is cross-platform and contains most of the functionality of the scripted module. This gives a rapid way for new users to fit and compare PCS, RDC and PRE effects.

### 5.3.1 YouTube Tutorial

[Check out the tutorial on YouTube](#)

### 5.3.2 Running the GUI

To run the GUI, first open the python interpreter in the terminal

```
user@computer:~$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Then import paramagpy and start the gui with `paramagpy.gui.run()`.

```
user@computer:~$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
```

(continues on next page)

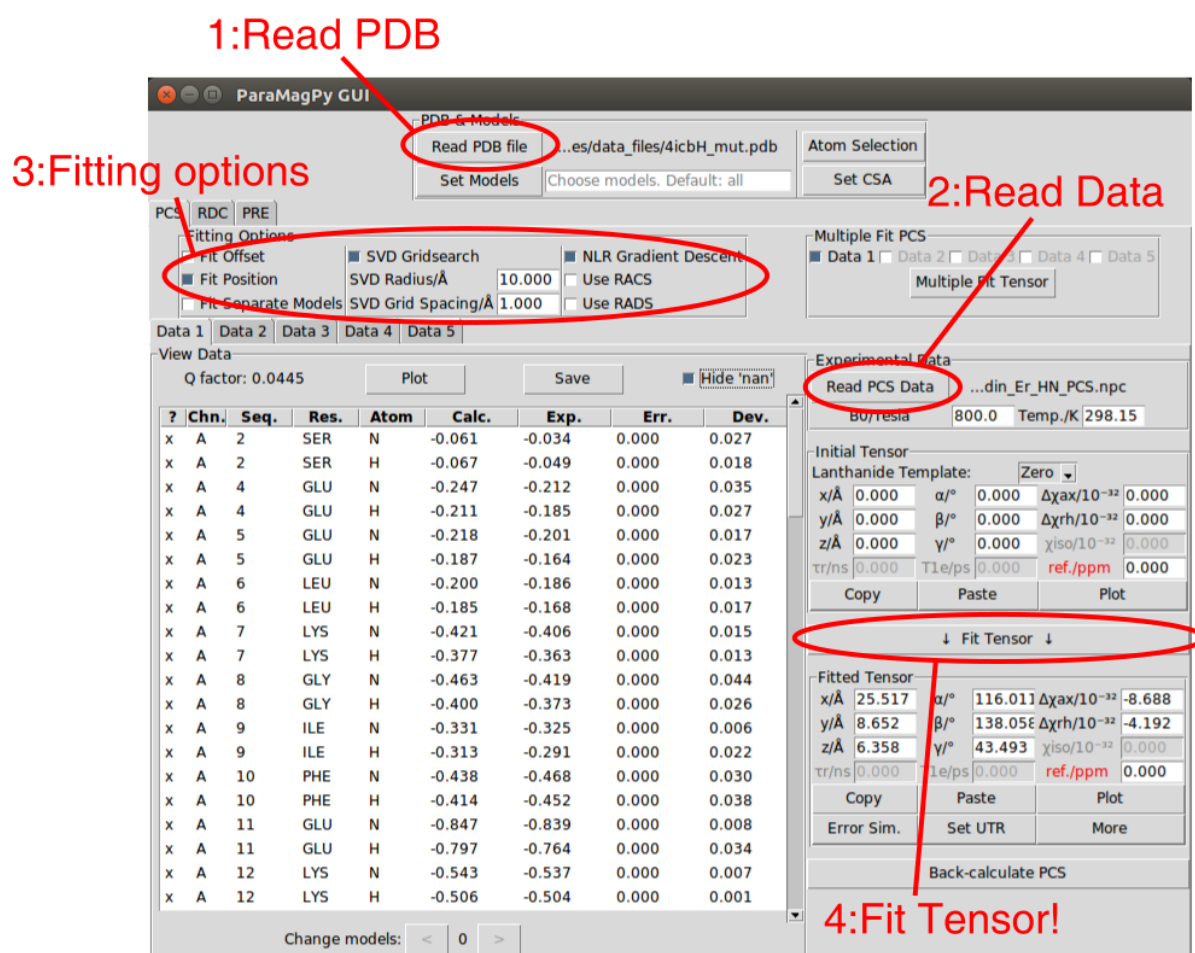
(continued from previous page)

```
Type "help", "copyright", "credits" or "license" for more information.
>>> import paramagpy
>>> paramagpy.gui.run()
```

Alternatively you can simply execute the following from the command line

```
user@computer:~$ echo "import paramagpy; paramagpy.gui.run()" | python3
```

If all this fails, you can contact the author for a prebuilt executable at [henry.orton@anu.edu.au](mailto:henry.orton@anu.edu.au)



## 5.4 NMR Software Macros

Paramagpy includes scripts for reading/writing PCS values directly from popular NMR software. This drastically improves the iterative process of tensor fitting.

### 5.4.1 CCPNMR Analysis 2.4

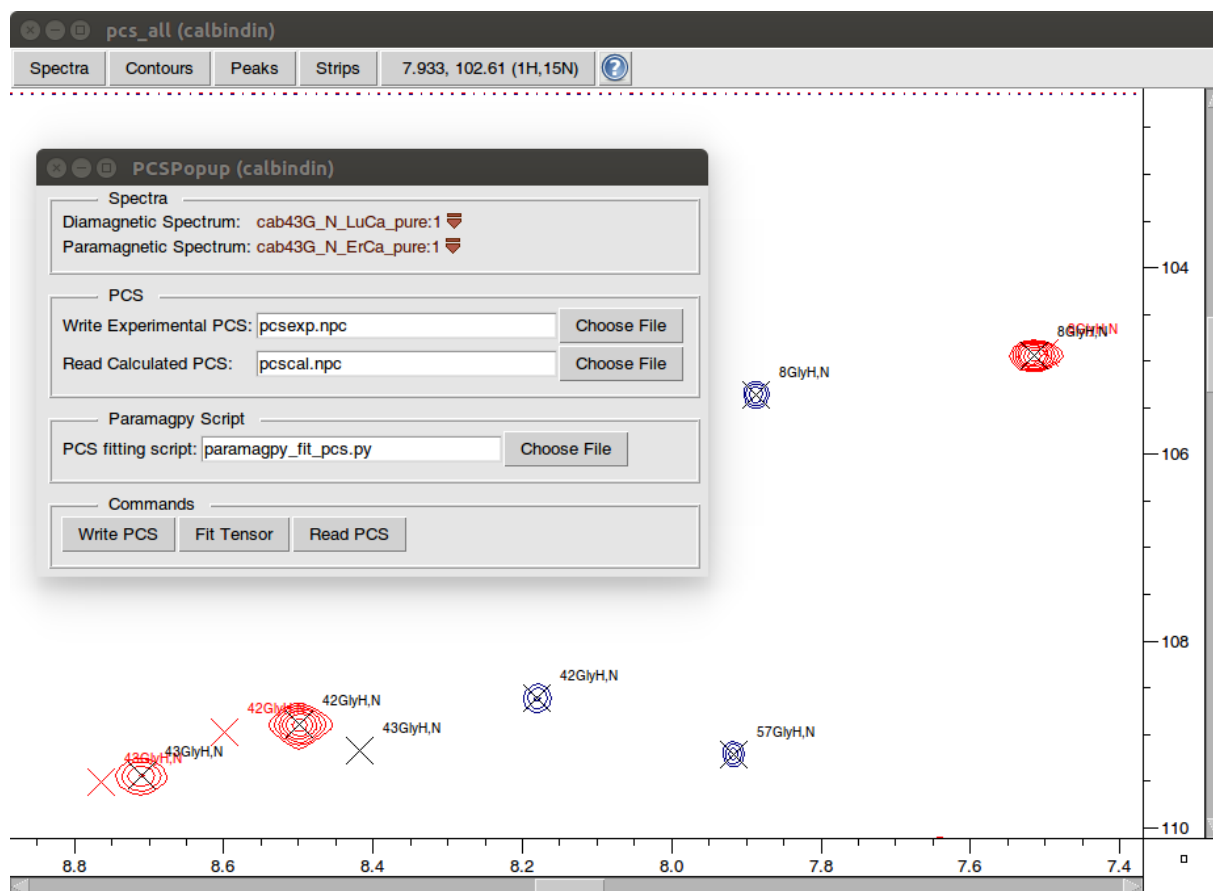
Download the two scripts:

- `paramagpy_ccpnmr_macro.py`
- `paramagpy_fit_pcs.py`

In the first line of the script `paramagpy_fit_pcs.py`, replace the shebang with the path to the python version on your machine that contains the paramagpy installation. On my computer this is set to.

```
#!/usr/bin/python3
```

Open CCPNMR analysis and navigate to Menu->Macro->Organise Macros. At the lower right click *Add Macro* and open the script `paramagpy_ccpnmr_macro.py`, then select `paramagpyMACRO` and click *Load Macro*. You can then select if from the list and click *Run* to reveal the screen below.



The popup window allows you to select a diamagnetic and paramagnetic spectrum and has 3 available buttons:

- **Write PCS:** This will calculate the difference between assigned peaks in the paramagnetic and diamagnetic spectra and write them to a .npc file (as specified in the relevant field).
- **Fit Tensor:** This will call the paramagpy script `paramagpy_fit_pcs.py` to fit the tensor the the written PCS values.
- **Read PCS:** This will read back-calculated PCS values from file (as specified in the relevant field) and plot the expected peaks on the paramagnetic spectrum in red.

Note, to alter the parameters for fitting of the PCS tensor, you can change the values within the script `paramagpy_fit_pcs.py`.



## 5.4.2 Sparky

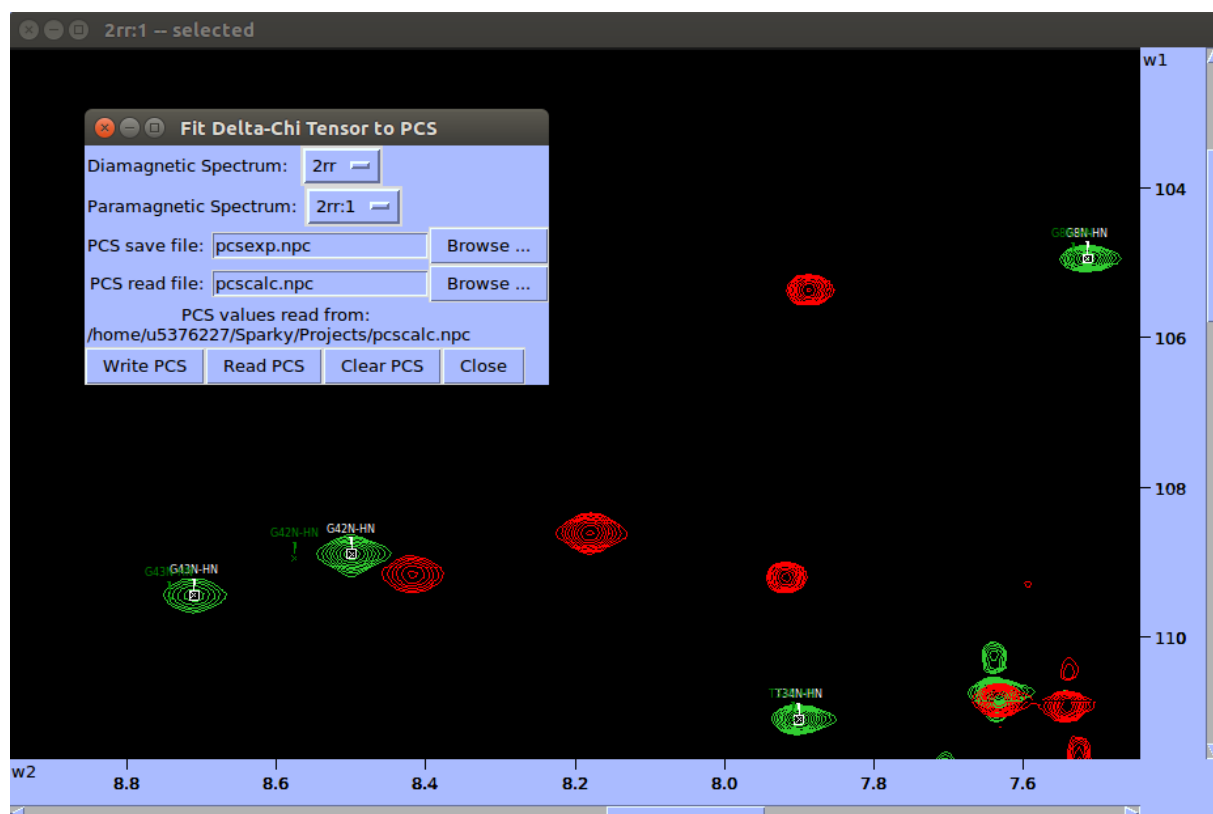
Download the 3 scripts:

- `paramagpy_sparky_macro.py`
- `sparky_init.py`
- `paramagpy_fit_pcs.py`

Place the first two scripts `paramagpy_sparky_macro.py` and `sparky_init.py` in the Sparky directory `~/Sparky/Python`. Note that the Sparky directory usually contains the *Projects*, *Lists* and *Save* folders. You may need to create the *Python* directory here in which to place the two scripts.

Place the third script `paramagpy_fit_pcs.py` in your home directory.

Open Sparky and navigate to Extensions->Read and write PCS files.



The popup window allows you to select a diamagnetic and paramagnetic spectrum and has 3 available buttons:

- **Write PCS:** This will calculate the difference between assigned peaks in the paramagnetic and diamagnetic spectra and write them to a .npc file (as specified in the relevant field).
- **Read PCS:** This will read back-calculated PCS values from file (as specified in the relevant field) and plot the expected peaks on the paramagnetic spectrum in green.
- **Clear PCS:** This will remove all calculated PCS peaks from the spectrum.

Note, to perform the tensor fitting, you will need to execute the paramagpy script in a separate terminal including an argument with the experimental PCS file such as:

```
user@computer:~$ ./paramagpy_fit_pcs.py pcsexp.npc
```

To alter the parameters for fitting of the PCS tensor, you can change the values within the script `paramagpy_fit_pcs.py`.

## 5.5 Mathematical Proofs

### 5.5.1 Proof of equivalence between the matrix representation of Curie-spin dipole–dipole cross-correlated relaxation (Paramagpy implementation) and the original description by Ghose and Prestegard

This proof concerns equations 24-27 of the [article](#) on the software Paramagpy, which describe the cross-correlated relaxation between the Curie spin and dipole–dipole relaxation mechanisms. The equations 24-27 follow the matrix representation of relaxation theory (equations 20-21) by [Suturina et al.](#). It is shown below that these equations are equivalent to the equations established previously by [Ghose and Prestegard](#) and reported in slightly modified form by [Bertini et al.](#). The proof begins with equations 24-27 of the manuscript and describes their rearrangement into the form given by [Bertini et al.](#)

For a specific example, we consider the case of a  $^{15}\text{N}$  -  $^1\text{H}$  group, with the Curie-spin shielding tensor  $\sigma$  at the site of the  $^1\text{H}$  spin located at  $\vec{r} = [x, y, z]$  and distance  $r$  from the paramagnetic centre.

$$\sigma = \frac{1}{4\pi r^5} \begin{bmatrix} (3x^2 - r^2) & 3xy & 3xz \\ 3xy & (3y^2 - r^2) & 3yz \\ 3xz & 3yz & (3z^2 - r^2) \end{bmatrix} \cdot \begin{bmatrix} \chi_{xx} & \chi_{xy} & \chi_{xz} \\ \chi_{xy} & \chi_{yy} & \chi_{yz} \\ \chi_{xz} & \chi_{yz} & \chi_{zz} \end{bmatrix}$$

We choose the orientation of the electron– $^1\text{H}$  vector to be aligned with the  $z$ -axis such that  $z \rightarrow r$ . In the case of an isotropic magnetic susceptibility,  $x = y = 0$  and the  $\chi$  tensor is represented by a diagonal matrix with three identical elements  $\chi_{\text{iso}}$ , yielding the following simplification

$$\begin{aligned} \sigma &= \frac{1}{4\pi r^5} \begin{bmatrix} -r^2 & 0 & 0 \\ 0 & -r^2 & 0 \\ 0 & 0 & 2r^2 \end{bmatrix} \cdot \begin{bmatrix} \chi_{\text{iso}} & 0 & 0 \\ 0 & \chi_{\text{iso}} & 0 \\ 0 & 0 & \chi_{\text{iso}} \end{bmatrix} \\ &= \xi_{\text{DSA}} \begin{bmatrix} -r^2 & 0 & 0 \\ 0 & -r^2 & 0 \\ 0 & 0 & 2r^2 \end{bmatrix} \\ \text{where } \xi_{\text{DSA}} &= \frac{\chi_{\text{iso}}}{4\pi r^5} \end{aligned}$$

The nuclear dipole shielding tensor arising from the  $^{15}\text{N}$  spin can be described in the same coordinate frame for an arbitrary orientation of the bond vector  $\vec{r}_{\text{HN}} = [x, y, z]$  with bond length  $r_{\text{HN}}$  by

$$\begin{aligned} \sigma_N &= \frac{1}{B_0} \frac{\mu_0}{4\pi} \gamma_N \hbar I \left[ 3 \frac{\vec{r}_{\text{HN}} \otimes \vec{r}_{\text{HN}}^T}{r_{\text{HN}}^5} - \frac{\mathbb{I}_3}{r_{\text{HN}}^3} \right] \\ \sigma_N &= \xi_{\text{DD}} \begin{bmatrix} (3x^2 - r_{\text{HN}}^2) & 3xy & 3xz \\ 3xy & (3y^2 - r_{\text{HN}}^2) & 3yz \\ 3xz & 3yz & (3z^2 - r_{\text{HN}}^2) \end{bmatrix} \\ \text{where } \xi_{\text{DD}} &= \frac{1}{B_0} \frac{\mu_0}{4\pi} \frac{\gamma_N \hbar I}{r_{\text{HN}}^5} \end{aligned}$$

and  $x, y, z$  denote the coordinates of the  $^{15}\text{N}$  spin relative to the  $^1\text{H}$  spin.

The effective shielding tensor at the site of the  $^1\text{H}$  spin, when the  $^{15}\text{N}$  partner is in the spin-up state, is given by the sum of the two tensors

$$\begin{aligned} \sigma_{\uparrow} &= \sigma + \sigma_N \\ &= \begin{bmatrix} (3x^2 - r_{\text{HN}}^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}} & (3xy)\xi_{\text{DD}} & (3xz)\xi_{\text{DD}} \\ (3xy)\xi_{\text{DD}} & (3y^2 - r_{\text{HN}}^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}} & (3yz)\xi_{\text{DD}} \\ (3xz)\xi_{\text{DD}} & (3yz)\xi_{\text{DD}} & (3z^2 - r_{\text{HN}}^2)\xi_{\text{DD}} + (2r^2)\xi_{\text{DSA}} \end{bmatrix} \end{aligned}$$

Note that this matrix is symmetric. Therefore we can ignore equation 18 of the main text and only need to substitute matrix elements into equation 19. Expanding and simplifying (via symbolic processing in the program

Mathematica), this yields

$$\begin{aligned}
\Delta(\sigma_{\uparrow})^2 &= \sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx}\sigma_{yy} - \sigma_{xx}\sigma_{zz} - \sigma_{yy}\sigma_{zz} \\
&\quad + \frac{3}{4} [(\sigma_{xy} + \sigma_{yx})^2 + (\sigma_{xz} + \sigma_{zx})^2 + (\sigma_{yz} + \sigma_{zy})^2] \\
&= ((3x^2 - r_{\text{HN}}^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}})^2 \\
&\quad + ((3y^2 - r^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}})^2 \\
&\quad + ((3z^2 - r_{\text{HN}}^2)\xi_{\text{DD}} + (2r^2)\xi_{\text{DSA}})^2 \\
&\quad - ((3x^2 - r_{\text{HN}}^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}})((3y^2 - r^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}}) \\
&\quad - ((3x^2 - r_{\text{HN}}^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}})((3z^2 - r_{\text{HN}}^2)\xi_{\text{DD}} + (2r^2)\xi_{\text{DSA}}) \\
&\quad - ((3y^2 - r^2)\xi_{\text{DD}} - (r^2)\xi_{\text{DSA}})((3z^2 - r_{\text{HN}}^2)\xi_{\text{DD}} + (2r^2)\xi_{\text{DSA}}) \\
&\quad + \frac{3}{4} [(6xy\xi_{\text{DD}})^2 + (6xz\xi_{\text{DD}})^2 + (6yz\xi_{\text{DD}})^2] \\
&= 9\xi_{\text{DD}}^2 r_{\text{HN}}^4 + 9\xi_{\text{DSA}}^2 r^4 - 9\xi_{\text{DD}}(r_{\text{HN}}^2 - 3z^2)\xi_{\text{DSA}}r^2
\end{aligned}$$

The angle  $\theta$  between the electron-nuclear vector  $\vec{r}$  and the nuclear bond vector  $\vec{r}_{\text{HN}}$  is captured by the dot product formula

$$\begin{aligned}
\vec{r} \cdot \vec{r}_{\text{HN}} &= |\vec{r}| |\vec{r}_{\text{HN}}| \cos \theta \\
\Rightarrow [0, 0, r] \cdot [x, y, z] &= r r_{\text{HN}} \cos \theta \\
\Rightarrow z &= r_{\text{HN}} \cos \theta
\end{aligned}$$

Using above equation to substitute  $z$  yields

$$\Delta(\sigma_{\uparrow})^2 = 9\xi_{\text{DD}}^2 r_{\text{HN}}^4 + 9\xi_{\text{DSA}}^2 r^4 - 9\xi_{\text{DD}} r_{\text{HN}}^2 (1 - 3 \cos^2 \theta) \xi_{\text{DSA}} r^2$$

where the first two terms account for the dipolar and Curie spin auto-relaxation terms respectively, and the last term accounts for their cross-correlation. The  $R_2$  relaxation rate can be calculated by substitution of  $\Delta^2$  into equation 21 of the main text.

$$R_2^{\text{Curie}}(\sigma_{\uparrow}) = \frac{1}{45} \omega^2 [4\mathbf{J}(0) + 3\mathbf{J}(\omega)] (9\xi_{\text{DD}}^2 r_{\text{HN}}^4 + 9\xi_{\text{DSA}}^2 r^4 + 9\xi_{\text{DD}} r_{\text{HN}}^2 (3 \cos^2 \theta - 1) \xi_{\text{DSA}} r^2)$$

The same derivation for  $\sigma_{\downarrow}$  yields the same result except for a sign change in the cross term:

$$R_2^{\text{Curie}}(\sigma_{\downarrow}) = \frac{1}{45} B_0^2 \gamma_H^2 [4\mathbf{J}(0) + 3\mathbf{J}(\omega)] (9\xi_{\text{DD}}^2 r_{\text{HN}}^4 + 9\xi_{\text{DSA}}^2 r^4 - 9\xi_{\text{DD}} r_{\text{HN}}^2 (3 \cos^2 \theta - 1) \xi_{\text{DSA}} r^2)$$

Taking the difference we obtain

$$\begin{aligned}
R_2^{\text{Curie} \times \text{DD}} &= R_2^{\text{Curie}}(\sigma_{\uparrow}) - R_2^{\text{Curie}}(\sigma_{\downarrow}) \\
&= \frac{1}{45} B_0^2 \gamma_H^2 (18\xi_{\text{DD}} r_{\text{HN}}^2 (3 \cos^2 \theta - 1) \xi_{\text{DSA}} r^2) [4\mathbf{J}(0) + 3\mathbf{J}(\omega)] \\
&= \frac{18}{45} \frac{\mu_0}{4\pi} \frac{B_0 \gamma_H^2 \gamma_N \hbar I}{r_{\text{HN}}^3} \frac{\chi_{\text{iso}}}{4\pi r^3} (3 \cos^2 \theta - 1) [4\mathbf{J}(0) + 3\mathbf{J}(\omega)]
\end{aligned}$$

Substituting  $\chi_{\text{iso}}$  and the spin of  $^{15}\text{N}$  as  $I=1/2$  yields

$$\begin{aligned}
R_2^{\text{Curie} \times \text{DD}} &= \frac{9}{45} \left( \frac{\mu_0}{4\pi} \right)^2 \frac{B_0 \gamma_H^2 \gamma_N \hbar}{r_{\text{HN}}^3} \frac{\mu_B^2 g^2 J(J+1)}{3kTr^3} (3 \cos^2 \theta - 1) [4\mathbf{J}(0) + 3\mathbf{J}(\omega)] \\
&= \frac{2}{15} \left( \frac{\mu_0}{4\pi} \right)^2 \frac{B_0 \gamma_H^2 \gamma_N \hbar}{r_{\text{HN}}^3} \frac{\mu_B^2 g^2 J(J+1)}{kTr^3} \frac{(3 \cos^2 \theta - 1)}{2} [4\mathbf{J}(0) + 3\mathbf{J}(\omega)]
\end{aligned}$$

The differential line width can be calculated from the relaxation rate as  $\Delta\nu = R_2^{\text{Curie} \times \text{DD}} / \pi$  and thus this equation matches equation 7 from Bertini et al..

## 5.6 Reference Guide

### 5.6.1 Paramagnetic module

This module handles the paramagnetic centre by defining the magnetic susceptibility tensor and methods for PCS, RDC and PRE calculations.

#### paramagpy.metal

##### Functions

<code>euler_to_matrix(eulers)</code>	Calculate a rotation matrix from euler angles using ZYZ convention
<code>matrix_to_euler(M)</code>	Calculate Euler angles from a rotation matrix using ZYZ convention
<code>unique_eulers(eulers)</code>	Calculate Euler angles in unique tensor representation.
<code>make_tensor(x, y, z, axial, rhombic, alpha, ...)</code>	Make a ChiTensor instance from given parameters.

#### paramagpy.metal.euler\_to\_matrix

`paramagpy.metal.euler_to_matrix(eulers)`

Calculate a rotation matrix from euler angles using ZYZ convention

**Parameters** `eulers` (*array of floats*) – the euler angles [alpha,beta,gamma] in radians by ZYZ convention.

**Returns** `matrix` – the rotation matrix

**Return type** 3x3 numpy ndarray

##### Examples

```
>>> eulers = np.array([0.5,1.2,0.8])
>>> euler_to_matrix(eulers)
array([[ -0.1223669 , -0.5621374 ,  0.81794125],
       [  0.75057357,  0.486796  ,  0.44684334],
       [ -0.64935788,  0.66860392,  0.36235775]])
```

#### paramagpy.metal.matrix\_to\_euler

`paramagpy.metal.matrix_to_euler(M)`

Calculate Euler angles from a rotation matrix using ZYZ convention

**Parameters** `M` (*3x3 numpy ndarray*) – a rotation matrix

**Returns** `eulers` – the euler angles [alpha,beta,gamma] in radians by ZYZ convention

**Return type** array of floats

## Examples

```
>>> matrix = array([[ -0.1223669 , -0.5621374 ,  0.81794125],
                    [  0.75057357,  0.486796  ,  0.44684334],
                    [-0.64935788,  0.66860392,  0.36235775]])
>>> matrix_to_euler(matrix)
np.array([0.5, 1.2, 0.8])
```

## paramagpy.metal.unique\_eulers

paramagpy.metal.**unique\_eulers** (*eulers*)

Calculate Euler angles in unique tensor representation.

Given general Euler angles by ZYZ convention, this function accounts for the symmetry of a second rank symmetric tensor to map all angles within the range  $[0, \pi]$ .

**Parameters** *eulers* (*array of float*) – the three Euler angles in radians

**Returns** *eulers\_utr* – the euler angles  $[\alpha, \beta, \gamma]$  in radians by ZYZ convention

**Return type** array of floats

## Examples

```
>>> eulers = np.array([-5.2, 10.3, 0.1])
>>> unique_eulers(eulers)
np.array([1.08318531  0.87522204  3.04159265])
```

## paramagpy.metal.make\_tensor

paramagpy.metal.**make\_tensor** (*x, y, z, axial, rhombic, alpha, beta, gamma, lanthanide=None, temperature=298.15*)

Make a ChiTensor instance from given parameters. This is designed to use pdb coordinates (*x, y, z*) and euler angles from an output like Numbat.

**Parameters**

- *y, z* (*x*,) – tensor position in pdb coordiante in Angstroms
- *rhombic* (*axial*,) – the tensor anisotropies in units  $10^{-32}$
- *beta, gamma* (*alpha*,) – the euler angles in degrees that maps the tensor to the pdb (I think?)

**Returns** **ChiTensor** – a tensor object for calulating paramagnetic effects on nuclear spins in the pdb coordinate

**Return type** object *paramagpy.metal.Metal*

## Classes

<code>Metal</code> ([position, eulers, axrh, mueff, ...])	An object for paramagnetic chi tensors and delta-chi tensors.
---	---

### paramagpy.metal.Metal

**class** paramagpy.metal.**Metal** (*position*=(0, 0, 0), *eulers*=(0, 0, 0), *axrh*=(0, 0), *mueff*=0.0, *g\_axrh*=(0, 0), *t1e*=0.0, *shift*=0.0, *temperature*=298.15, *B0*=18.79, *taur*=0.0)

An object for paramagnetic chi tensors and delta-chi tensors. This class has basic attributes that specify position, axially/rhombicity, isotropy and euler angles. It also has methods for calculating PCS, RDC, PRE and CCR values.

**\_\_init\_\_** (*position*=(0, 0, 0), *eulers*=(0, 0, 0), *axrh*=(0, 0), *mueff*=0.0, *g\_axrh*=(0, 0), *t1e*=0.0, *shift*=0.0, *temperature*=298.15, *B0*=18.79, *taur*=0.0)  
 Instantiate ChiTensor object

#### Parameters

- **position** (*array of floats, optional*) – the (x,y,z) position in meters. Default is (0,0,0) stored as a np.matrix object.
- **eulers** (*array of floats, optional*) – the euler angles [alpha,beta,gamma] in radians by ZYZ convention. Default is (0,0,0)
- **axrh** (*array of floats, optional*) – the axial and rhombic values defining the magnetic susceptibility anisotropy
- **g\_axrh** (*array of floats, optional*) – the axial and rhombic values defining the power spectral density tensor
- **mueff** (*float*) – the effective magnetic moment in units of A.m<sup>2</sup>
- **shift** (*float*) – a bulk shift value applied to all PCS calculations. This is a correction parameter that may arise due to an offset between diamagnetic and paramagnetic PCS datasets.
- **temperature** (*float*) – the temperature in Kelvin
- **t1e** (*float*) – the longitudinal electronic relaxation time
- **B0** (*float*) – the magnetic field in Telsa
- **taur** (*float*) – the rotational correlation time in seconds

#### Methods

<code>atom_ccr</code> (atom, atomPartner)	Calculate R2 cross-correlated relaxation due to DDxDSA
<code>atom_pcs</code> (atom[, racs, rads])	Calculate the psuedo-contact shift at the given atom
<code>atom_pre</code> (atom[, rtype, dsa, sbm, csa])	Calculate the PRE for an atom
<code>atom_rdc</code> (atom1, atom2)	Calculate the residual dipolar coupling between two atoms
<code>atom_set_position</code> (atom)	Set the position of the Metal object to that of an atom
<code>average</code> (metals)	Set the attributes of the current instance to the average of a list of provided tensor objects

Continued on next page

Table 3 – continued from previous page

<code>ccr(position, gamma, dipole_shift_tensor)</code>	Calculate R2 cross-correlated relaxation due to DDxDSA
<code>copy()</code>	Copy the current Metal object to a new instance
<code>dipole_shift_tensor(position)</code>	Calculate the chemical shift tensor at the given position
<code>dsa_r1(position, gamma[, csa])</code>	Calculate R1 relaxation due to Curie Spin
<code>dsa_r2(position, gamma[, csa])</code>	Calculate R2 relaxation due to Curie Spin
<code>fast_ccr(posarray, gammaarray, dstarray)</code>	Vectorised version of <code>paramagpy.metal.Metal.ccr()</code>
<code>fast_dipole_shift_tensor(posarray)</code>	A vectorised version of <code>paramagpy.metal.Metal.dipole_shift_tensor()</code>
<code>fast_dsa_r1(posarray, gammaarray[, csaarray])</code>	Vectorised version of <code>paramagpy.metal.Metal.dsa_r1()</code>
<code>fast_dsa_r2(posarray, gammaarray[, csaarray])</code>	Vectorised version of <code>paramagpy.metal.Metal.dsa_r2()</code>
<code>fast_first_invariant_squared(t)</code>	Vectorised version of <code>paramagpy.metal.Metal.first_invariant_squared()</code>
<code>fast_g_sbm_r1(posarray, gammaarray)</code>	Vectorised version of <code>paramagpy.metal.Metal.g_sbm_r1()</code>
<code>fast_pcs(posarray)</code>	A vectorised version of <code>paramagpy.metal.Metal.pcs()</code>
<code>fast_pre(posarray, gammaarray, rtype[, dsa, ...])</code>	Calculate the PRE for a set of spins using Curie and or SBM theory
<code>fast_racs(csaarray)</code>	A vectorised version of <code>paramagpy.metal.Metal.racs()</code>
<code>fast_rads(posarray)</code>	A vectorised version of <code>paramagpy.metal.Metal.rads()</code>
<code>fast_rdc(vecarray, gammaProdArray)</code>	A vectorised version of <code>paramagpy.metal.Metal.rdc()</code> method.
<code>fast_sbm_r1(posarray, gammaarray)</code>	Vectorised version of <code>paramagpy.metal.Metal.sbm_r1()</code>
<code>fast_sbm_r2(posarray, gammaarray)</code>	Vectorised version of <code>paramagpy.metal.Metal.sbm_r2()</code>
<code>fast_second_invariant_squared(t)</code>	Vectorised version of <code>paramagpy.metal.Metal.second_invariant_squared()</code>
<code>first_invariant_squared(t)</code>	Calculate the antisymmetric contribution to relaxation via the first invariant of a tensor.
<code>g_sbm_r1(position, gamma)</code>	Calculate R1 relaxation due to Solomon-Bloembergen-Morgan theory from anisotropic power spectral density tensor
<code>get_params(params)</code>	Get tensor parameters that have been scaled appropriately
<code>info([comment])</code>	Get basic information about the Metal object
<code>isomap([protein, isoval])</code>	
<code>make_mesh([density, size])</code>	Construct a 3D grid of points to map an isosurface
<code>pcs(position)</code>	Calculate the psuedo-contact shift at the given position
<code>pcs_mesh(mesh)</code>	Calculate a PCS value at each location of cubic grid of points
<code>pre(position, gamma, rtype[, dsa, sbm, ...])</code>	Calculate the PRE for a set of spins using Curie and or SBM theory
<code>pre_mesh(mesh[, gamma, rtype, dsa, sbm])</code>	Calculate a PRE value at each location of cubic grid of points

Continued on next page

Table 3 – continued from previous page

<code>racs(csa)</code>	Calculate the residual anisotropic chemical shift at the given position.
<code>rads(position)</code>	Calculate the residual anisotropic dipolar shift at the given position.
<code>rdc(vector, gammaProd)</code>	Calculate Residual Dipolar Coupling (RDC)
<code>save([fileName])</code>	
<code>sbm_r1(position, gamma)</code>	Calculate R1 relaxation due to Solomon-Bloembergen-Morgan theory
<code>sbm_r2(position, gamma)</code>	Calculate R2 relaxation due to Solomon-Bloembergen-Morgan theory
<code>second_invariant_squared(t)</code>	Calculate the second invariant squared of a tensor.
<code>set_Jg(J, g)</code>	Set the magnetic susceptibility absolute magnitude from J/g.
<code>set_lanthanide(lanthanide[, set_dchi])</code>	Set the anisotropy, isotropy and T1e parameters from literature values
<code>set_params(paramValues)</code>	Set tensor parameters that have been scaled appropriately
<code>set_utr()</code>	Modify current tensor parameters to unique tensor representation (UTR)
<code>spec_dens(tau, omega)</code>	A spectral density function with Lorentzian shape:
<code>write_isomap(mesh, bounds[, fileName])</code>	Write a PyMol script to file which allows loading of the isosurface file
<code>write_pymol_script([isoval, surface-Name, ...])</code>	Write a PyMol script to file which allows loading of the isosurface file

### paramagpy.metal.Metal.atom\_ccr

`Metal.atom_ccr(atom, atomPartner)`

Calculate R2 cross-correlated relaxation due to DDxDSA

#### Parameters

- **atom** (`paramagpy.protein.CustomAtom`) – the active nuclear spin for which relaxation will be calculated must have attributes ‘position’ and ‘gamma’
- **atomPartner** (`paramagpy.protein.CustomAtom`) – the coupling partner nuclear spin must have method ‘dipole\_shift\_tensor’

**Returns value** – the CCR differential line broadening in Hz

**Return type** float

### paramagpy.metal.Metal.atom\_pcs

`Metal.atom_pcs(atom, racs=False, rads=False)`

Calculate the pseudo-contact shift at the given atom

#### Parameters

- **atom** (*biopython atom object*) – must have ‘position’ attribute
- **racs** (*bool (optional)*) – when True, RACS (residual anisotropic chemical shielding) correction is included. Default is False
- **rads** (*bool (optional)*) – when True, RADS (residual anisotropic dipolar shielding) correction is included. Default is False

**Returns pcs** – the pseudo-contact shift in parts-per-million (ppm)

**Return type** float



**paramagpy.metal.Metal.atom\_pre**

`Metal.atom_pre(atom, rtype='r2', dsa=True, sbm=True, csa=0.0)`

Calculate the PRE for an atom

**Parameters**

- **atom** (`paramagpy.protein.CustomAtom`) – the active nuclear spin for which relaxation will be calculated must have attributes 'position' and 'gamma'
- **rtype** (*str*) – either 'r1' or 'r2', the relaxation type
- **dsa** (*bool (optional)*) – when True (default), DSA or Curie spin relaxation is included
- **sbm** (*bool (optional)*) – when True (default), SBM spin relaxation is included
- **csa** (*array with shape (3,3) (optional)*) – CSA tensor of the spin. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns** **rate** – The PRE rate in /s

**Return type** float

**paramagpy.metal.Metal.atom\_rdc**

`Metal.atom_rdc(atom1, atom2)`

Calculate the residual dipolar coupling between two atoms

**Parameters**

- **atom1** (*biopython atom object*) – must have 'position' and 'gamma' attribute
- **atom2** – must have 'position' and 'gamma' attribute

**Returns** **rdc** – the RDC values in Hz

**Return type** float

**paramagpy.metal.Metal.atom\_set\_position**

`Metal.atom_set_position(atom)`

Set the position of the Metal object to that of an atom

**Parameters** **atom** (*biopython atom object*) – must have 'position' attribute

**paramagpy.metal.Metal.average**

`Metal.average(metals)`

Set the attributes of the current instance to the average of a list of provided tensor objects

WARNING: averaging is unstable for spectral power density <g\_tensor>

**Parameters** **metals** (*a list of Metal objects*) – the average of attributes of this list will be taken

**paramagpy.metal.Metal.ccr**

`Metal.ccr(position, gamma, dipole_shift_tensor)`

Calculate R2 cross-correlated relaxation due to DDxDSA

If the metal has an anisotropic magnetic susceptibility, this is taken into account.

**Parameters**

- **position** (*array of floats*) – three coordinates (x,y,z) this is the position of the nuclear spin
- **gamma** (*float*) – the gyromagnetic ratio of the relaxing spin
- **dipole\_shift\_tensor** (*3x3 array of floats*) – this is the dipole shift tensor arising from the nuclear spin of the coupling partner

**Returns value** – The R2 differential line broadening rate in /s

**Return type** float

### paramagpy.metal.Metal.copy

`Metal.copy()`

Copy the current Metal object to a new instance

**Returns new\_tensor** – a new Metal instance with the same parameters

**Return type** Metal object

### paramagpy.metal.Metal.dipole\_shift\_tensor

`Metal.dipole_shift_tensor(position)`

Calculate the chemical shift tensor at the given position

This arises due to the paramagnetic dipole tensor field

**Parameters position** (*array floats*) – the position (x, y, z) in meters

**Returns dipole\_shift\_tensor** – the tensor describing chemical shift at the nuclear position

**Return type** 3x3 array

### paramagpy.metal.Metal.dsa\_r1

`Metal.dsa_r1(position, gamma, csa=0.0)`

Calculate R1 relaxation due to Curie Spin

If the metal has an anisotropic magnetic susceptibility, this is taken into account, resulting in orientation dependent PRE as predicted by Vega and Fiat. CSA cross-correlated relaxation may be included by providing an appropriate CSA tensor.

**Parameters**

- **position** (*array of floats*) – three coordinates (x,y,z) in meters
- **gamma** (*float*) – the gyromagnetic ratio of the spin
- **csa** (*3x3 matrix (optional)*) – the CSA tensor of the given spin. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns value** – The R1 relaxation rate in /s

**Return type** float

### paramagpy.metal.Metal.dsa\_r2

`Metal.dsa_r2(position, gamma, csa=0.0)`

Calculate R2 relaxation due to Curie Spin

If the metal has an anisotropic magnetic susceptibility, this is taken into account, resulting in orientation dependent PRE as predicted by Vega and Fiat. CSA cross-correlated relaxation may be included by providing an appropriate CSA tensor.

**Parameters**

- **position** (*array of floats*) – three coordinates (x,y,z)
- **gamma** (*float*) – the gyromagnetic ratio of the spin
- **csa** (*3x3 matrix (optional)*) – the CSA tensor of the given spin. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns value** – The R2 relaxation rate in /s

**Return type** float

**paramagpy.metal.Metal.fast\_ccr**

`Metal.fast_ccr(posarray, gammaarray, dstarray)`

Vectorised version of `paramagpy.metal.Metal.ccr()`

This is generally used for speed in fitting DDxDSA data

If the metal has an anisotropic magnetic susceptibility, this is taken into account.

**Parameters**

- **posarray** (array with shape  $(n, 3)$ ) – array of positions in meters
- **gammaarray** (array with shape  $(n, 3)$ ) – array of gyromagnetic ratios of the spins
- **dstarray** (array with shape  $(n, 3, 3)$ ) – array of nuclear dipole shift tensors arising from the coupling partners

**Returns** **rates** – The R2 differential line broadening rates in /s

**Return type** array with shape  $(n, 1)$

**paramagpy.metal.Metal.fast\_dipole\_shift\_tensor**

`Metal.fast_dipole_shift_tensor(posarray)`

A vectorised version of `paramagpy.metal.Metal.dipole_shift_tensor()`

This is generally used for fast calculations.

**Parameters** **posarray** (array) – an array of positions with shape  $(n, 3)$

**Returns** **dipole\_shift\_tensor\_array** – and array of dipole shift tensors at corresponding positions. This has shape  $(n, 3, 3)$

**Return type** array

**paramagpy.metal.Metal.fast\_dsa\_r1**

`Metal.fast_dsa_r1(posarray, gammaarray, csaarray=0.0)`

Vectorised version of `paramagpy.metal.Metal.dsa_r1()`

This is generally used for speed in fitting PRE data

**Parameters**

- **posarray** (array with shape  $(n, 3)$ ) – array of positions in meters
- **gammaarray** (array with shape  $(n, 3)$ ) – array of gyromagnetic ratios of the spins
- **csaarray** (array with shape  $(m, 3, 3)$  (optional)) – array of CSA tensors of the spins. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns** **rates** – The R1 relaxation rates in /s

**Return type** array with shape  $(n, 1)$

**paramagpy.metal.Metal.fast\_dsa\_r2**

`Metal.fast_dsa_r2(posarray, gammaarray, csaarray=0.0)`

Vectorised version of `paramagpy.metal.Metal.dsa_r2()`

This is generally used for speed in fitting PRE data.

**Parameters**

- **posarray** (array with shape  $(n, 3)$ ) – array of positions in meters
- **gammaarray** (array with shape  $(n, 3)$ ) – array of gyromagnetic ratios of the spins
- **csaarray** (array with shape  $(m, 3, 3)$  (optional)) – array of CSA tensors of the spins. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns** **rates** – The R2 relaxation rates in /s

**Return type** array with shape (n,1)

### paramagpy.metal.Metal.fast\_first\_invariant\_squared

**static** `Metal.fast_first_invariant_squared(t)`

Vectorised version of `paramagpy.metal.Metal.first_invariant_squared()`

This is generally used for speed in fitting PRE data

**Parameters** **tensorarray** (array with shape (n, 3, 3)) – array of shielding tensors

**Returns** **firstInvariantSquared** – the first invariants squared of the tensors

**Return type** array with shape (n,1)

### paramagpy.metal.Metal.fast\_g\_sbm\_r1

`Metal.fast_g_sbm_r1(posarray, gammaarray)`

Vectorised version of `paramagpy.metal.Metal.g_sbm_r1()`

This is generally used for speed in fitting PRE data

**Parameters**

- **posarray** (array with shape (n, 3)) – array of positions in meters
- **gammaarray** (array with shape (n, 3)) – array of gyromagnetic ratios of the spins

**Returns** **rates** – The R1 relaxation rates in /s

**Return type** array with shape (n,1)

### paramagpy.metal.Metal.fast\_pcs

`Metal.fast_pcs(posarray)`

A vectorised version of `paramagpy.metal.Metal.pcs()`

This efficient algorithm calculates the PCSs for an array of positions and is best used where speed is required for fitting.

**Parameters** **posarray** (array with shape (n, 3)) – array of ‘n’ positions (x, y, z) in meters

**Returns** **pcs** – the pseudo-contact shift in parts-per-million (ppm)

**Return type** array of floats with shape (n,1)

### paramagpy.metal.Metal.fast\_pre

`Metal.fast_pre(posarray, gammaarray, rtype, dsa=True, sbm=True, gsbm=False, csaarray=0.0)`

Calculate the PRE for a set of spins using Curie and or SBM theory

**Parameters**

- **posarray** (array with shape (n, 3)) – array of positions in meters
- **gammaarray** (array with shape (n, 3)) – array of gyromagnetic ratios of the spins
- **rtype** (str) – either ‘r1’ or ‘r2’, the relaxation type
- **dsa** (bool (optional)) – when True (default), DSA or Curie spin relaxation is included
- **sbm** (bool (optional)) – when True (default), SBM spin relaxation is included
- **gsbm** (bool (optional)) – when True (default=False), anisotropic dipolar relaxation is included using the spectral power density tensor <g\_tensor> NOTE:

when true, ignores relaxation of type SBM NOTE: only implemented for R1 relaxation calculations

- **csaarray** (array with shape  $(m, 3, 3)$  (optional)) – array of CSA tensors of the spins. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns** **rates** – The PRE rates in /s

**Return type** array with shape  $(n, 1)$

### paramagpy.metal.Metal.fast\_racs

`Metal.fast_racs(csaarray)`

A vectorised version of `paramagpy.metal.Metal.racs()`

This is generally used when speed is required for fitting

**Parameters** **csaarray** (array with shape  $(n, 3, 3)$ ) – array of chemical shift anisotropy tensors

**Returns** **racs\_array** – the residual anisotropic chemical shift in parts-per-million (ppm)

**Return type** array of floats with shape  $(n, 1)$

### paramagpy.metal.Metal.fast\_rads

`Metal.fast_rads(posarray)`

A vectorised version of `paramagpy.metal.Metal.rads()`

This is generally used when speed is required for fitting

**Parameters** **posarray** (array with shape  $(n, 3)$ ) – an array of ‘n’ positions (x, y, z) in meters

**Returns** **rads\_array** – the residual anisotropic dipole shift in parts-per-million (ppm)

**Return type** array of floats with shape  $(n, 1)$

### paramagpy.metal.Metal.fast\_rdc

`Metal.fast_rdc(vecarray, gammaProdArray)`

A vectorised version of `paramagpy.metal.Metal.rdc()` method.

This is generally used for speed in fitting RDC data

**Parameters**

- **vecarray** (array with shape  $(n, 3)$ ) – array of internuclear vectors in meters

- **gammaProdArray** (array with shape  $(n, 1)$ ) – the products of gyromagnetic ratios of spins A and B where each has units of rad/s/T

**Returns** **rdc\_array** – the RDC values in Hz

**Return type** array with shape  $(n, 1)$

### paramagpy.metal.Metal.fast\_sbm\_r1

`Metal.fast_sbm_r1(posarray, gammaarray)`

Vectorised version of `paramagpy.metal.Metal.sbm_r1()`

This is generally used for speed in fitting PRE data

**Parameters**

- **posarray** (array with shape  $(n, 3)$ ) – array of positions in meters
- **gammaarray** (array with shape  $(n, 3)$ ) – array of gyromagnetic ratios of the spins

**Returns** **rates** – The R1 relaxation rates in /s

**Return type** array with shape  $(n, 1)$

**paramagpy.metal.Metal.fast\_sbm\_r2**

`Metal.fast_sbm_r2(posarray, gammaarray)`

Vectorised version of `paramagpy.metal.Metal.sbm_r2()`

This is generally used for speed in fitting PRE data

**Parameters**

- **posarray** (*array with shape (n, 3)*) – array of positions in meters
- **gammaarray** (*array with shape (n, 3)*) – array of gyromagnetic ratios of the spins

**Returns** **rates** – The R2 relaxation rates in /s

**Return type** array with shape (n,1)

**paramagpy.metal.Metal.fast\_second\_invariant\_squared**

**static** `Metal.fast_second_invariant_squared(t)`

Vectorised version of `paramagpy.metal.Metal.second_invariant_squared()`

This is generally used for speed in fitting PRE data

**Parameters** **tensorarray** (*array with shape (n, 3, 3)*) – array of shielding tensors

**Returns** **secondInvariantSquared** – the second invariants squared of the tensors

**Return type** array with shape (n,1)

**paramagpy.metal.Metal.first\_invariant\_squared**

**static** `Metal.first_invariant_squared(t)`

Calculate the antisymmetric contribution to relaxation via the first invariant of a tensor.

This is required for PRE calculations using the shilding tensor

**Parameters** **tensor** (*3x3 matrix*) – a second rank tensor

**Returns** **firstInvariantSquared** – the first invariant squared of the shift tensor

**Return type** float

**paramagpy.metal.Metal.g\_sbm\_r1**

`Metal.g_sbm_r1(position, gamma)`

Calculate R1 relaxation due to Solomon-Bloembergen-Morgan theory from anisotropic power spectral density tensor

**Parameters**

- **position** (*array of floats*) – three coordinates (x,y,z)
- **gamma** (*float*) – the gyromagnetic ratio of the spin

**Returns** **value** – The R1 relaxation rate in /s

**Return type** float

## paramagpy.metal.Metal.get\_params

`Metal.get_params(params)`

Get tensor parameters that have been scaled appropriately

This is often used to get parameter values during fitting where floating point errors would otherwise occur on the small values encountered.

**Parameters** `params` (*list of str*) – each element of the list is a string that corresponds to an attribute of the Metal to be retrieved.

**Returns** `scaled_params` – a list with respective scaled parameter values from the input.

**Return type** list

### Examples

```
>>> metal = Metal(axrh=[20E-32, 3E-32], position=[0.0, 10E-10, -5E-10])
>>> metal.get_params(['ax', 'rh', 'x', 'y', 'z'])
[20.0, 3.0, 0.0, 10.0, -5.0]
```

## paramagpy.metal.Metal.info

`Metal.info(comment=True)`

Get basic information about the Metal object

This is returned as a string in human readable units This is also the file format for saving the tensor

**Parameters** `comment` (*bool (optional)*) – if True, each line has a '#' placed at the front

**Returns** `information` – a string containing basic information about the Metal

**Return type** str

### Examples

```
>>> metal = Metal()
>>> metal.set_lanthanide('Er')
>>> metal.info()
# ax      | 1E-32 m^3 :   -11.600
# rh      | 1E-32 m^3 :    -8.600
# x       | 1E-10 m   :    0.000
# y       | 1E-10 m   :    0.000
# z       | 1E-10 m   :    0.000
# a       |          deg :    0.000
# b       |          deg :    0.000
# g       |          deg :    0.000
# mueff   |          Bm :    9.581
# shift   |          ppm :    0.000
# B0      |          T  :   18.790
# temp    |          K  :   298.150
# tle     |          ps :    0.189
# taur    |          ns :    0.000
```

**paramagpy.metal.Metal.isomap**

`Metal.isomap` (*protein=None, isoval=1.0, \*\*kwargs*)

**paramagpy.metal.Metal.make\_mesh**

`Metal.make_mesh` (*density=2, size=40.0*)

Construct a 3D grid of points to map an isosurface

This is contained in a cube

**Parameters**

- **density** (*int (optional)*) – the points per Angstrom in the grid
- **size** (*float (optional)*) – the length of one edge of the cube

**Returns**

- **mesh** (*cubic grid array*) – This has shape (n,n,n,3) where n is the number of points along one edge of the grid. Units are meters
- **origin** (*array of floats,*) – the (x,y,z) location of mesh vertex
- **low** (*array of ints, the integer location of the first*) – point in each dimension
- **high** (*array of ints, the integer location of the last*) – point in each dimension
- **points** (*array of ints,*) – the number of points along each dimension

**paramagpy.metal.Metal.pcs**

`Metal.pcs` (*position*)

Calculate the psuedo-contact shift at the given postition

**Parameters** **position** (*array floats*) – the position (x, y, z) in meters

**Returns** **pcs** – the pseudo-contact shift in parts-per-million (ppm)

**Return type** float

**Examples**

```
>>> metal = Metal()
>>> metal.set_lanthanide('Er')
>>> metal.pcs([0., 0., 10E-10])
-6.153991132886608
```

**paramagpy.metal.Metal.pcs\_mesh**

`Metal.pcs_mesh` (*mesh*)

Calculate a PCS value at each location of cubic grid of points

**Parameters** **mesh** (*array with shape (n, n, n, 3)*) – a cubic grid as generated by the method <make\_mesh>

**Returns** **pcs\_mesh** – The same grid shape, with PCS values at the respective locations

**Return type** array with shape (n,n,n,1)



**paramagpy.metal.Metal.pre**

`Metal.pre` (*position*, *gamma*, *rtype*, *dsa=True*, *sbm=True*, *gsbm=False*, *csa=0.0*)

Calculate the PRE for a set of spins using Curie and or SBM theory

**Parameters**

- **position** (*array of floats*) – position in meters
- **gamma** (*float*) – gyromagnetic ratio of the spin
- **rtype** (*str*) – either 'r1' or 'r2', the relaxation type
- **dsa** (*bool (optional)*) – when True (default), DSA or Curie spin relaxation is included
- **sbm** (*bool (optional)*) – when True (default), SBM spin relaxation is included
- **gsbm** (*bool (optional)*) – when True (default=False), anisotropic dipolar relaxation is included using the spectral power density tensor <g\_tensor> NOTE: when true, ignores relaxation of type SBM NOTE: only implemented for R1 relaxation calculations
- **csa** (*array with shape (3,3) (optional)*) – CSA tensor of the spin. This defaults to 0.0, meaning CSAxDSA crosscorrelation is not accounted for.

**Returns** **rate** – The PRE rate in /s

**Return type** float

**paramagpy.metal.Metal.pre\_mesh**

`Metal.pre_mesh` (*mesh*, *gamma=267512897.63847807*, *rtype='r2'*, *dsa=True*, *sbm=True*)

Calculate a PRE value at each location of cubic grid of points

**Parameters**

- **mesh** (*array with shape (n,n,n,3)*) – a cubic grid as generated by the method <make\_mesh>
- **gamma** (*float*) – the gyromagnetic ratio of the spin
- **rtype** (*str*) – either 'r1' or 'r2', the relaxation type
- **dsa** (*bool (optional)*) – when True (default), DSA or Curie spin relaxation is included
- **sbm** (*bool (optional)*) – when True (default), SBM spin relaxation is included

**Returns** **pre\_mesh** – The same grid shape, with PRE values at the respective locations

**Return type** array with shape (n,n,n,1)

**paramagpy.metal.Metal.racs**

`Metal.racs` (*csa*)

Calculate the residual anisotropic chemical shift at the given position.

The partial alignment induced by an anisotropic magnetic susceptibility causes the chemical shift tensor at a nuclear position to average to a value different to the isotropic value.

**Parameters** **csa** (*3 x 3 array*) – the chemical shift anisotropy tensor

**Returns** **racs** – the residual anisotropic chemical shift in parts-per-million (ppm)

**Return type** float

**paramagpy.metal.Metal.rads****Metal.rads** (*position*)

Calculate the residual anisotropic dipolar shift at the given position.

The partial alignment induced by an anisotropic magnetic susceptibility causes the dipole shift tensor at a nuclear position to average to a value different to the PCS.

**Parameters** **position** (*array floats*) – the position (x, y, z) in meters

**Returns** **rads** – the residual anisotropic dipole shift in parts-per-million (ppm)

**Return type** float

**paramagpy.metal.Metal.rdc****Metal.rdc** (*vector, gammaProd*)

Calculate Residual Dipolar Coupling (RDC)

**Parameters**

- **vector** (*array of floats*) – internuclear vector (x,y,z) in meters
- **gammaProd** (*float*) – the product of gyromagnetic ratios of spin A and B where each has units of rad/s/T

**Returns** **rdc** – the RDC in Hz

**Return type** float

**paramagpy.metal.Metal.save****Metal.save** (*fileName='tensor.txt'*)**paramagpy.metal.Metal.sbm\_r1****Metal.sbm\_r1** (*position, gamma*)

Calculate R1 relaxation due to Solomon-Bloembergen-Morgan theory

**Parameters**

- **position** (*array of floats*) – three coordinates (x,y,z)
- **gamma** (*float*) – the gyromagnetic ratio of the spin

**Returns** **value** – The R1 relaxation rate in /s

**Return type** float

**paramagpy.metal.Metal.sbm\_r2****Metal.sbm\_r2** (*position, gamma*)

Calculate R2 relaxation due to Solomon-Bloembergen-Morgan theory

**Parameters**

- **position** (*array of floats*) – three coordinates (x,y,z)
- **gamma** (*float*) – the gyromagnetic ratio of the spin

**Returns** **value** – The R2 relaxation rate in /s

**Return type** float

**paramagpy.metal.Metal.second\_invariant\_squared****static** Metal.**second\_invariant\_squared**(*t*)

Calculate the second invariant squared of a tensor.

This is required for PRE calculations using the shielding tensor

**Parameters** **tensor** (*3x3 matrix*) – a second rank tensor**Returns** **secondInvariantSquared** – the second invariant squared of the shift tensor**Return type** float**paramagpy.metal.Metal.set\_Jg**Metal.**set\_Jg**(*J*, *g*)

Set the magnetic susceptibility absolute magnitude from J/g.

This is achieved using the following formula:

$$\mu_{eff} = g\mu_B\sqrt{J(J+1)}$$

**Parameters**

- **J** (*str*) – the total spin angular momentum quantum number
- **g** (*bool*, *optional*) – the Lande g-factor

**paramagpy.metal.Metal.set\_lanthanide**Metal.**set\_lanthanide**(*lanthanide*, *set\_dchi=True*)

Set the anisotropy, isotropy and T1e parameters from literature values

**Parameters**

- **lanthanide** (*str*) – one of 'Ce','Pr','Nd','Pm','Sm','Eu','Gd','Tb','Dy','Ho','Er','Tm','Yb'
- **set\_dichi** (*bool (optional)*) – if True (default), the tensor anisotropy is set. Otherwise only the isotropy and T1e values are set

**paramagpy.metal.Metal.set\_params**Metal.**set\_params**(*paramValues*)

Set tensor parameters that have been scaled appropriately

This is the inverse of the method &lt;get\_params&gt;

**Parameters** **paramValues** (*list of tuple*) – each element is a tuple (variable, value) where 'variable' is the string identifying the attribute to be set, and 'value' is the corresponding value

**Examples**

```
>>> metal = Metal()
>>> metal.set_params([('ax', 20.0), ('rh', 3.0)])
>>> metal.axrh
[2.e-31 3.e-32]
```

**paramagpy.metal.Metal.set\_utr****Metal.set\_utr()**

Modify current tensor parameters to unique tensor representation (UTR)

Note that multiple axial/rhombic and euler angles can give congruent tensors. This method ensures that identical tensors may always be compared by using Numbat style representation.

**paramagpy.metal.Metal.spec\_dens****static Metal.spec\_dens** (*tau, omega*)

A spectral density function with Lorentzian shape:

$$\mathbb{J}(\tau, \omega) = \frac{\tau}{1 + (\omega\tau)^2}$$

**Parameters**

- **tau** (*float*) – correlation time
- **omega** (*float*) – frequency

**Returns value** – the value of the spectral density**Return type** float**paramagpy.metal.Metal.write\_isomap****Metal.write\_isomap** (*mesh, bounds, fileName='isomap.pml.ccp4'*)

Write a PyMol script to file which allows loading of the isosurface file

**Parameters**

- **mesh** (*3D scalar np.ndarray of floats*) – the scalar field of PCS or PRE values in a cubic grid
- **bounds** (*tuple (origin, low, high, points)*) – as generated by `paramagpy.metal.Metal.make_mesh()`
- **fileName** (*str (optional)*) – the filename of the isosurface file

**paramagpy.metal.Metal.write\_pymol\_script**

**Metal.write\_pymol\_script** (*isoval=1.0, surfaceName='isomap', scriptName='isomap.pml', meshName='./isomap.pml.ccp4', pdbFile=None*)

Write a PyMol script to file which allows loading of the isosurface file

**Parameters**

- **isoval** (*float (optional)*) – the contour level of the isosurface
- **surfaceName** (*str (optional)*) – the name of the isosurface file within PyMol
- **scriptName** (*str (optional)*) – the name of the PyMol script to load the tensor isosurface
- **meshName** (*str (optional)*) – the name of the binary isosurface file
- **pdbFile** (*str (optional)*) – if not <None>, the file name of the PDB file to be loaded with the isosurface.

**Attributes**

<i>B0_MHz</i>	1H NMR frequency for the given field in MHz
<i>GAMMA</i>	
<i>HBAR</i>	
<i>K</i>	
<i>MU0</i>	
<i>MUB</i>	
<i>a</i>	alpha euler anglue
<i>alignment_factor</i>	Factor for conversion between magnetic susceptibility and alignment tensors
<i>ax</i>	axiality
<i>b</i>	beta euler anglue
<i>eigenvalues</i>	The eigenvalues defining the magnitude of the principle axes
<i>fit_scaling</i>	
<i>fundamental_attributes</i>	
<i>g</i>	gamma euler anglue
<i>g_eigenvalues</i>	The eigenvalues defining the magnitude of the principle axes
<i>g_isotropy</i>	Estimate of the spectral power density tensor isotropy
<i>g_tensor</i>	The magnetic susceptibility tensor matrix representation
<i>gax</i>	axial componenet of spectral power density tensor
<i>grh</i>	axial componenet of spectral power density tensor
<i>iso</i>	isotropy
<i>isotropy</i>	The magnidue of the isotropic component of the tensor
<i>lanth_axrh</i>	
<i>lanth_lib</i>	
<i>lower_coords</i>	
<i>rh</i>	rhombicity
<i>rotationMatrix</i>	The rotation matrix as defined by the euler angles
<i>saupe_factor</i>	Factor for conversion between magnetic susceptibility and saupe tensors
<i>tauc</i>	The effective rotational correlation time.
<i>tensor</i>	The magnetic susceptibility tensor matrix representation
<i>tensor_alignment</i>	The alignment tensor matrix representation
<i>tensor_saupe</i>	The saupe tensor matrix representation
<i>tensor_traceless</i>	The traceless magnetic susceptibility tensor matrix representation
<i>upper_coords</i>	
<i>upper_triang</i>	Fetch 5 unique matrix element defining the magnetic susceptibility tensor
<i>upper_triang_alignment</i>	Fetch 5 unique matrix element defining the alignment tensor
<i>upper_triang_saupe</i>	Fetch 5 unique matrix element defining the magnetic susceptibility tensor
<i>x</i>	x coordinate
<i>y</i>	y coordinate

Continued on next page

Table 4 – continued from previous page

<i>z</i>	<i>z</i> coordinate
----------	---------------------

**paramagpy.metal.Metal.B0\_MHz****Metal.B0\_MHz**

1H NMR frequency for the given field in MHz

**paramagpy.metal.Metal.GAMMA****Metal.GAMMA = 176085964400.0****paramagpy.metal.Metal.HBAR****Metal.HBAR = 1.0546e-34****paramagpy.metal.Metal.K****Metal.K = 1.381e-23****paramagpy.metal.Metal.MU0****Metal.MU0 = 1.2566370614359173e-06****paramagpy.metal.Metal.MUB****Metal.MUB = 9.274e-24****paramagpy.metal.Metal.a****Metal.a**

alpha euler anglue

**paramagpy.metal.Metal.alignment\_factor****Metal.alignment\_factor**

Factor for conversion between magnetic susceptibility and alignment tensors

**paramagpy.metal.Metal.ax****Metal.ax**

axiality

**paramagpy.metal.Metal.b**

**Metal.b**  
beta euler anglue

**paramagpy.metal.Metal.eigenvalues**

**Metal.eigenvalues**  
The eigenvalues defining the magnitude of the principle axes

**paramagpy.metal.Metal.fit\_scaling**

**Metal.fit\_scaling** = {'a': 57.29577951308232, 'ax': 1e+32, 'b': 57.29577951308232}

**paramagpy.metal.Metal.fundamental\_attributes**

**Metal.fundamental\_attributes** = ('position', 'eulers', 'axrh', 'mueff', 'g\_axrh')

**paramagpy.metal.Metal.g**

**Metal.g**  
gamma euler anglue

**paramagpy.metal.Metal.g\_eigenvalues**

**Metal.g\_eigenvalues**  
The eigenvalues defining the magnitude of the principle axes

**paramagpy.metal.Metal.g\_isotropy**

**Metal.g\_isotropy**  
Estimate of the spectral power density tensor isotropy

**paramagpy.metal.Metal.g\_tensor**

**Metal.g\_tensor**  
The magnetic susceptibility tensor matrix representation

**paramagpy.metal.Metal.gax**

**Metal.gax**  
axial componenet of spectral power density tensor

**paramagpy.metal.Metal.grh**

**Metal.grh**  
axial componenet of spectral power density tensor

**paramagpy.metal.Metal.iso**

**Metal.iso**  
isotropy

**paramagpy.metal.Metal.isotropy**

**Metal.isotropy**  
The magnidue of the isotropic component of the tensor

**paramagpy.metal.Metal.lanth\_axrh**

**Metal.lanth\_axrh** = {'Ce': (2.1, 0.7), 'Dy': (34.7, 20.3), 'Er': (-11.6, -8.6)}

**paramagpy.metal.Metal.lanth\_lib**

**Metal.lanth\_lib** = {'Ce': (2.5, 0.8571428571428571, 1.33e-13), 'Dy': (7.5, 1.33e-13)}

**paramagpy.metal.Metal.lower\_coords**

**Metal.lower\_coords** = ((0, 1, 1, 2, 2), (0, 1, 0, 0, 1))

**paramagpy.metal.Metal.rh**

**Metal.rh**  
rhombicity

**paramagpy.metal.Metal.rotationMatrix**

**Metal.rotationMatrix**  
The rotation matrix as defined by the euler angles

**paramagpy.metal.Metal.saupe\_factor**

**Metal.saupe\_factor**  
Factor for conversion between magnetic susceptibility and saupe tensors



### **paramagpy.metal.Metal.tauc**

#### **Metal.tauc**

The effective rotational correlation time.

This is calculated by combining the rotational correlation time and the electronic relaxation time:

$$\tau_c = \frac{1}{\frac{1}{\tau_r} + \frac{1}{T_{1e}}}$$

### **paramagpy.metal.Metal.tensor**

#### **Metal.tensor**

The magnetic susceptibility tensor matrix representation

### **paramagpy.metal.Metal.tensor\_alignment**

#### **Metal.tensor\_alignment**

The alignment tensor matrix representation

### **paramagpy.metal.Metal.tensor\_saupe**

#### **Metal.tensor\_saupe**

The saupe tensor matrix representation

### **paramagpy.metal.Metal.tensor\_traceless**

#### **Metal.tensor\_traceless**

The traceless magnetic susceptibility tensor matrix representation

### **paramagpy.metal.Metal.upper\_coords**

**Metal.upper\_coords = ((0, 1, 0, 0, 1), (0, 1, 1, 2, 2))**

### **paramagpy.metal.Metal.upper\_triangular**

#### **Metal.upper\_triangular**

Fetch 5 unique matrix element defining the magnetic susceptibility tensor

### **paramagpy.metal.Metal.upper\_triangular\_alignment**

#### **Metal.upper\_triangular\_alignment**

Fetch 5 unique matrix element defining the alignment tensor

**paramagpy.metal.Metal.upper\_triang\_saupe****Metal.upper\_triang\_saupe**

Fetch 5 unique matrix element defining the magnetic susceptibility tensor

**paramagpy.metal.Metal.x****Metal.x**

x coordinate

**paramagpy.metal.Metal.y****Metal.y**

y coordinate

**paramagpy.metal.Metal.z****Metal.z**

z coordinate

## 5.6.2 Protein module

This module handles the protein structure coordinates and includes methods for loading a PDB file and calculating atomic properities such as CSA or gyromagnetic ratio

**paramagpy.protein****Functions**

<code>load_pdb(fileName[, ident])</code>	Read PDB from file into biopython structure object
<code>rotation_matrix(axis, theta)</code>	Return the rotation matrix associated with counter-clockwise rotation about the given axis by theta radians.

**paramagpy.protein.load\_pdb****paramagpy.protein.load\_pdb** (*fileName*, *ident=None*)

Read PDB from file into biopython structure object

**Parameters**

- **fileName** (*str*) – the path to the file
- **ident** (*str* (*optional*)) – the desired identity of the structure object

**Returns values** – a structure object containing the atomic coordinates**Return type** `paramagpy.protein.CustomStructure`

## paramagpy.protein.rotation\_matrix

paramagpy.protein.rotation\_matrix(*axis*, *theta*)

Return the rotation matrix associated with counterclockwise rotation about the given axis by theta radians.

**Parameters** *axis* (*array of floats*) – the [x,y,z] axis for rotation.

**Returns** *matrix* – the rotation matrix

**Return type** numpy 3x3 matrix object

## Classes

<code>CustomAtom(*arg, **kwargs)</code>	
<code>CustomStructure(*arg, **kwargs)</code>	This is an overload hack of the BioPython Structure object
<code>CustomStructureBuilder(*arg, **kwargs)</code>	This is an overload hack of BioPython's Custom-StructureBuilder

## paramagpy.protein.CustomAtom

**class** paramagpy.protein.CustomAtom(\*arg, \*\*kwargs)

**\_\_init\_\_**(\*arg, \*\*kwargs)

Initialize Atom object.

### Parameters

- **name** (*string*) – atom name (eg. "CA"). Note that spaces are normally stripped.
- **coord** (*Numeric array (Float0, size 3)*) – atomic coordinates (x,y,z)
- **bfactor** (*number*) – isotropic B factor
- **occupancy** (*number*) – occupancy (0.0-1.0)
- **altloc** (*string*) – alternative location specifier for disordered atoms
- **fullname** (*string*) – full atom name, including spaces, e.g. " CA ". Normally these spaces are stripped from the atom name.
- **element** (*uppercase string (or None if unknown)*) – atom element, e.g. "C" for Carbon, "HG" for mercury,

### Methods

<code>copy()</code>	Create a copy of the Atom.
<code>detach_parent()</code>	Remove reference to parent.
<code>dipole_shift_tensor(position)</code>	Calculate the magnetic field shielding tensor at the given position due to the nuclear dipole
<code>flag_disorder()</code>	Set the disordered flag to 1.
<code>get_altloc()</code>	Return alternative location specifier.
<code>get_anisou()</code>	Return anisotropic B factor.
<code>get_bfactor()</code>	Return B factor.
<code>get_coord()</code>	Return atomic coordinates.
<code>get_full_id()</code>	Return the full id of the atom.

Continued on next page

Table 7 – continued from previous page

<code>get_fullname()</code>	Return the atom name, including leading and trailing spaces.
<code>get_id()</code>	Return the id of the atom (which is its atom name).
<code>get_level()</code>	Return level.
<code>get_name()</code>	Return atom name.
<code>get_occupancy()</code>	Return occupancy.
<code>get_parent()</code>	Return parent residue.
<code>get_serial_number()</code>	Return the serial number.
<code>get_sigatm()</code>	Return standard deviation of atomic parameters.
<code>get_siguij()</code>	Return standard deviations of anisotropic temperature factors.
<code>get_vector()</code>	Return coordinates as Vector.
<code>is_disordered()</code>	Return the disordered flag (1 if disordered, 0 otherwise).
<code>set_altloc(altloc)</code>	Set alternative location specifier.
<code>set_anisou(anisou_array)</code>	Set anisotropic B factor.
<code>set_bfactor(bfactor)</code>	Set isotropic B factor.
<code>set_coord(coord)</code>	Set coordinates.
<code>set_occupancy(occupancy)</code>	Set occupancy.
<code>set_parent(parent)</code>	Set the parent residue.
<code>set_serial_number(n)</code>	Set serial number.
<code>set_sigatm(sigatm_array)</code>	Set standard deviation of atomic parameters.
<code>set_siguij(siguij_array)</code>	Set standard deviations of anisotropic temperature factors.
<code>top()</code>	
<code>transform(rot, tran)</code>	Apply rotation and translation to the atomic coordinates.

### paramagpy.protein.CustomAtom.copy

`CustomAtom.copy()`

Create a copy of the Atom.

Parent information is lost.

### paramagpy.protein.CustomAtom.detach\_parent

`CustomAtom.detach_parent()`

Remove reference to parent.

### paramagpy.protein.CustomAtom.dipole\_shift\_tensor

`CustomAtom.dipole_shift_tensor(position)`

Calculate the magnetic field shielding tensor at the given position due to the nuclear dipole

Assumes nuclear spin 1/2

**Parameters** `position` (*array floats*) – the position (x, y, z) in meters

**Returns** `dipole_shielding_tensor` – the tensor describing magnetic shielding at the given position

**Return type** 3x3 array

**paramagpy.protein.CustomAtom.flag\_disorder**

`CustomAtom.flag_disorder()`

Set the disordered flag to 1.

The disordered flag indicates whether the atom is disordered or not.

**paramagpy.protein.CustomAtom.get\_altloc**

`CustomAtom.get_altloc()`

Return alternative location specifier.

**paramagpy.protein.CustomAtom.get\_anisou**

`CustomAtom.get_anisou()`

Return anisotropic B factor.

**paramagpy.protein.CustomAtom.get\_bfactor**

`CustomAtom.get_bfactor()`

Return B factor.

**paramagpy.protein.CustomAtom.get\_coord**

`CustomAtom.get_coord()`

Return atomic coordinates.

**paramagpy.protein.CustomAtom.get\_full\_id**

`CustomAtom.get_full_id()`

Return the full id of the atom.

The full id of an atom is the tuple (structure id, model id, chain id, residue id, atom name, altloc).

**paramagpy.protein.CustomAtom.get\_fullname**

`CustomAtom.get_fullname()`

Return the atom name, including leading and trailing spaces.

**paramagpy.protein.CustomAtom.get\_id**

`CustomAtom.get_id()`

Return the id of the atom (which is its atom name).

**paramagpy.protein.CustomAtom.get\_level**

`CustomAtom.get_level()`  
Return level.

**paramagpy.protein.CustomAtom.get\_name**

`CustomAtom.get_name()`  
Return atom name.

**paramagpy.protein.CustomAtom.get\_occupancy**

`CustomAtom.get_occupancy()`  
Return occupancy.

**paramagpy.protein.CustomAtom.get\_parent**

`CustomAtom.get_parent()`  
Return parent residue.

**paramagpy.protein.CustomAtom.get\_serial\_number**

`CustomAtom.get_serial_number()`  
Return the serial number.

**paramagpy.protein.CustomAtom.get\_sigatm**

`CustomAtom.get_sigatm()`  
Return standard deviation of atomic parameters.

**paramagpy.protein.CustomAtom.get\_siguij**

`CustomAtom.get_siguij()`  
Return standard deviations of anisotropic temperature factors.

**paramagpy.protein.CustomAtom.get\_vector**

`CustomAtom.get_vector()`  
Return coordinates as Vector.  
**Returns** coordinates as 3D vector  
**Return type** Bio.PDB.Vector class

**paramagpy.protein.CustomAtom.is\_disordered**

`CustomAtom.is_disordered()`

Return the disordered flag (1 if disordered, 0 otherwise).

**paramagpy.protein.CustomAtom.set\_altloc**

`CustomAtom.set_altloc(altloc)`

Set alternative location specifier.

**paramagpy.protein.CustomAtom.set\_anisou**

`CustomAtom.set_anisou(anisou_array)`

Set anisotropic B factor.

**Parameters** `anisou_array` (*Numeric array (length 6)*) – anisotropic B factor.

**paramagpy.protein.CustomAtom.set\_bfactor**

`CustomAtom.set_bfactor(bfactor)`

Set isotropic B factor.

**paramagpy.protein.CustomAtom.set\_coord**

`CustomAtom.set_coord(coord)`

Set coordinates.

**paramagpy.protein.CustomAtom.set\_occupancy**

`CustomAtom.set_occupancy(occupancy)`

Set occupancy.

**paramagpy.protein.CustomAtom.set\_parent**

`CustomAtom.set_parent(parent)`

Set the parent residue.

**Parameters** `parent` – Residue object (-) –

**paramagpy.protein.CustomAtom.set\_serial\_number**

`CustomAtom.set_serial_number(n)`

Set serial number.

### paramagpy.protein.CustomAtom.set\_sigatm

CustomAtom.**set\_sigatm**(*sigatm\_array*)

Set standard deviation of atomic parameters.

The standard deviation of atomic parameters consists of 3 positional, 1 B factor and 1 occupancy standard deviation.

**Parameters** **sigatm\_array** (*Numeric array (length 5)*) – standard deviations of atomic parameters.

### paramagpy.protein.CustomAtom.set\_siguij

CustomAtom.**set\_siguij**(*siguij\_array*)

Set standard deviations of anisotropic temperature factors.

**Parameters** **siguij\_array** (*Numeric array (length 6)*) – standard deviations of anisotropic temperature factors.

### paramagpy.protein.CustomAtom.top

CustomAtom.**top**()

### paramagpy.protein.CustomAtom.transform

CustomAtom.**transform**(*rot, tran*)

Apply rotation and translation to the atomic coordinates.

**Parameters**

- **rot** (*3x3 Numeric array*) – A right multiplying rotation matrix
- **tran** (*size 3 Numeric array*) – the translation vector

### Examples

This is an incomplete but illustrative example:

```

from numpy import pi, array
from Bio.PDB.vectors import Vector, rotmat
rotation = rotmat(pi, Vector(1, 0, 0))
translation = array((0, 0, 1), 'f')
atom.transform(rotation, translation)

```

### Attributes

<i>HBAR</i>	
<i>MU0</i>	
<i>csa</i>	Get the CSA tensor at the nuclear position This uses the geometry of neighbouring atoms and a standard library from Bax J.
<i>csa_lib</i>	docstring for CustomAtom
<i>gyro_lib</i>	
<i>position</i>	



**paramagpy.protein.CustomAtom.HBAR**

CustomAtom.HBAR = 1.0546e-34

**paramagpy.protein.CustomAtom.MU0**

CustomAtom.MU0 = 1.2566370614359173e-06

**paramagpy.protein.CustomAtom.csa**

CustomAtom.csa

Get the CSA tensor at the nuclear position This uses the geometry of neighbouring atoms and a standard library from Bax J. Am. Chem. Soc. 2000

**Returns** **matrix** – the CSA tensor in the PDB frame if appropriate nuclear positions are not available <None> is returned.

**Return type** 3x3 array

**paramagpy.protein.CustomAtom.csa\_lib**

CustomAtom.csa\_lib = {'C': (array([-8.65e-05, 1.18e-05, 7.47e-05]), 0.663225115)  
docstring for CustomAtom

**paramagpy.protein.CustomAtom.gyro\_lib**

CustomAtom.gyro\_lib = {'C': 67261498.71335746, 'H': 267512897.63847807, 'N': -2

**paramagpy.protein.CustomAtom.position**

CustomAtom.position

**paramagpy.protein.CustomStructure**

**class** paramagpy.protein.CustomStructure (\*arg, \*\*kwargs)

This is an overload hack of the BioPython Structure object

**\_\_init\_\_** (\*arg, \*\*kwargs)

Initialize the class.

**Methods**

<code>add(entity)</code>	Add a child to the Entity.
<code>copy()</code>	Copy entity recursively.
<code>detach_child(id)</code>	Remove a child.
<code>detach_parent()</code>	Detach the parent.
<code>get_atoms()</code>	Return atoms from residue.
<code>get_chains()</code>	Return chains from models.
<code>get_full_id()</code>	Return the full id.
<code>get_id()</code>	Return the id.
<code>get_iterator()</code>	Return iterator over children.
<code>get_level()</code>	Return level in hierarchy.

Continued on next page

Table 9 – continued from previous page

<code>get_list()</code>	Return a copy of the list of children.
<code>get_models()</code>	Return models.
<code>get_parent()</code>	Return the parent Entity object.
<code>get_residues()</code>	Return residues from chains.
<code>has_id(id)</code>	Check if a child with given id exists.
<code>insert(pos, entity)</code>	Add a child to the Entity at a specified position.
<code>parse(dataValues[, models])</code>	Associate experimental data with atoms of the PDB file This method takes a DataContainer instance from the dataparse module
<code>set_parent(entity)</code>	Set the parent Entity object.
<code>transform(rot, tran)</code>	Apply rotation and translation to the atomic coordinates.

**paramagpy.protein.CustomStructure.add**

`CustomStructure.add(entity)`  
Add a child to the Entity.

**paramagpy.protein.CustomStructure.copy**

`CustomStructure.copy()`  
Copy entity recursively.

**paramagpy.protein.CustomStructure.detach\_child**

`CustomStructure.detach_child(id)`  
Remove a child.

**paramagpy.protein.CustomStructure.detach\_parent**

`CustomStructure.detach_parent()`  
Detach the parent.

**paramagpy.protein.CustomStructure.get\_atoms**

`CustomStructure.get_atoms()`  
Return atoms from residue.

**paramagpy.protein.CustomStructure.get\_chains**

`CustomStructure.get_chains()`  
Return chains from models.

**paramagpy.protein.CustomStructure.get\_full\_id**

`CustomStructure.get_full_id()`

Return the full id.

The full id is a tuple containing all id's starting from the top object (Structure) down to the current object. A full id for a Residue object e.g. is something like:

`("1abc", 0, "A", (" ", 10, "A"))`

This corresponds to:

Structure with id "1abc" Model with id 0 Chain with id "A" Residue with id (" ", 10, "A")

The Residue id indicates that the residue is not a hetero-residue (or a water) because it has a blank hetero field, that its sequence identifier is 10 and its insertion code "A".

**paramagpy.protein.CustomStructure.get\_id**

`CustomStructure.get_id()`

Return the id.

**paramagpy.protein.CustomStructure.get\_iterator**

`CustomStructure.get_iterator()`

Return iterator over children.

**paramagpy.protein.CustomStructure.get\_level**

`CustomStructure.get_level()`

Return level in hierarchy.

A - atom R - residue C - chain M - model S - structure

**paramagpy.protein.CustomStructure.get\_list**

`CustomStructure.get_list()`

Return a copy of the list of children.

**paramagpy.protein.CustomStructure.get\_models**

`CustomStructure.get_models()`

Return models.

**paramagpy.protein.CustomStructure.get\_parent**

`CustomStructure.get_parent()`

Return the parent Entity object.

**paramagpy.protein.CustomStructure.get\_residues**

`CustomStructure.get_residues()`  
Return residues from chains.

**paramagpy.protein.CustomStructure.has\_id**

`CustomStructure.has_id(id)`  
Check if a child with given id exists.

**paramagpy.protein.CustomStructure.insert**

`CustomStructure.insert(pos, entity)`  
Add a child to the Entity at a specified position.

**paramagpy.protein.CustomStructure.parse**

`CustomStructure.parse(dataValues, models=None)`  
Associate experimental data with atoms of the PDB file This method takes a DataContainer instance from the dataparse module

**Parameters** `dataValues` (*DataContainer instance*) – a dictionary containing the experimental values

**Returns** `dataArray` – the returned array has a row for each relevant atom in the PDB file. The columns contain model, experimental/calculated data, errors and indexes.

**Return type** numpy structured array

**paramagpy.protein.CustomStructure.set\_parent**

`CustomStructure.set_parent(entity)`  
Set the parent Entity object.

**paramagpy.protein.CustomStructure.transform**

`CustomStructure.transform(rot, tran)`  
Apply rotation and translation to the atomic coordinates.

**Parameters**

- `rot` (*3x3 Numeric array*) – A right multiplying rotation matrix
- `tran` (*size 3 Numeric array*) – the translation vector

**Examples**

This is an incomplete but illustrative example:

```
from numpy import pi, array
from Bio.PDB.vectors import Vector, rotmat
rotation = rotmat(pi, Vector(1, 0, 0))
translation = array((0, 0, 1), 'f')
entity.transform(rotation, translation)
```

## Attributes

<code>id</code>	Return identifier.
-----------------	--------------------

### paramagpy.protein.CustomStructure.id

`CustomStructure.id`  
Return identifier.

### paramagpy.protein.CustomStructureBuilder

**class** `paramagpy.protein.CustomStructureBuilder` (*\*arg, \*\*kwargs*)

This is an overload hack of BioPython's CustomStructureBuilder

`__init__` (*\*arg, \*\*kwargs*)  
Initialize the class.

## Methods

<code>get_structure()</code>	Return the structure.
<code>init_atom(name, coord, b_factor, occupancy, ...)</code>	Create a new Atom object.
<code>init_chain(chain_id)</code>	Create a new Chain object with given id.
<code>init_model(model_id[, serial_num])</code>	Create a new Model object with given id.
<code>init_residue(resname, field, resseq, icode)</code>	Create a new Residue object.
<code>init_seg(segid)</code>	Flag a change in segid.
<code>init_structure(structure_id)</code>	Initialize a new Structure object with given id.
<code>set_anisou(anisou_array)</code>	Set anisotropic B factor of current Atom.
<code>set_header(header)</code>	Set header.
<code>set_line_counter(line_counter)</code>	Tracks line in the PDB file that is being parsed.
<code>set_sigatm(sigatm_array)</code>	Set standard deviation of atom position of current Atom.
<code>set_siguij(siguij_array)</code>	Set standard deviation of anisotropic B factor of current Atom.
<code>set_symmetry(spacegroup, cell)</code>	Set symmetry.

### paramagpy.protein.CustomStructureBuilder.get\_structure

`CustomStructureBuilder.get_structure()`  
Return the structure.

### paramagpy.protein.CustomStructureBuilder.init\_atom

`CustomStructureBuilder.init_atom` (*name, coord, b\_factor, occupancy, altloc, fullname, serial\_number=None, element=None*)

Create a new Atom object. :param - name - string, atom name, e.g. CA, spaces should be stripped:  
:param - coord - Numeric array: :type - coord - Numeric array: Float0, size 3 :param - b\_factor - float, B factor: :param - occupancy - float: :param - altloc - string, alternative location specifier:  
:param - fullname - string, atom name including spaces, e.g. " CA ": :param - element - string, upper case, e.g. "HG" for mercury:

### paramagpy.protein.CustomStructureBuilder.init\_chain

CustomStructureBuilder.**init\_chain**(*chain\_id*)

Create a new Chain object with given id.

**Parameters** *chain\_id* - string (-) -

### paramagpy.protein.CustomStructureBuilder.init\_model

CustomStructureBuilder.**init\_model**(*model\_id*, *serial\_num=None*)

Create a new Model object with given id.

**Parameters**

- *id* - int (-) -
- *serial\_num* - int (-) -

### paramagpy.protein.CustomStructureBuilder.init\_residue

CustomStructureBuilder.**init\_residue**(*resname*, *field*, *resseq*, *icode*)

Create a new Residue object.

**Parameters**

- *resname* - string, e.g. "ASN" (-) -
- *field* - hetero flag, "W" for waters, "H" for (-) - hetero residues, otherwise blank.
- *resseq* - int, sequence identifier (-) -
- *icode* - string, insertion code (-) -

### paramagpy.protein.CustomStructureBuilder.init\_seg

CustomStructureBuilder.**init\_seg**(*segid*)

Flag a change in segid.

**Parameters** *segid* - string (-) -

### paramagpy.protein.CustomStructureBuilder.init\_structure

CustomStructureBuilder.**init\_structure**(*structure\_id*)

Initialize a new Structure object with given id.

**Parameters** *id* - string (-) -

### paramagpy.protein.CustomStructureBuilder.set\_anisou

CustomStructureBuilder.**set\_anisou**(*anisou\_array*)

Set anisotropic B factor of current Atom.

**paramagpy.protein.CustomStructureBuilder.set\_header**

`CustomStructureBuilder.set_header(header)`  
Set header.

**paramagpy.protein.CustomStructureBuilder.set\_line\_counter**

`CustomStructureBuilder.set_line_counter(line_counter)`  
Tracks line in the PDB file that is being parsed.  
**Parameters** `line_counter` – `int` (–) –

**paramagpy.protein.CustomStructureBuilder.set\_sigatm**

`CustomStructureBuilder.set_sigatm(sigatm_array)`  
Set standard deviation of atom position of current Atom.

**paramagpy.protein.CustomStructureBuilder.set\_siguij**

`CustomStructureBuilder.set_siguij(siguij_array)`  
Set standard deviation of anisotropic B factor of current Atom.

**paramagpy.protein.CustomStructureBuilder.set\_symmetry**

`CustomStructureBuilder.set_symmetry(spacegroup, cell)`  
Set symmetry.

### 5.6.3 Data I/O module

This module handles the reading and writing of experimental data.

**paramagpy.dataparse****Functions**

<code>read_pcs(fileName)</code>	Read pseudo contact shift values from file.
<code>read_rdc(fileName)</code>	Read residual dipolar coupling values from file.
<code>read_pre(fileName)</code>	Read paramagnetic relaxation enhancement values from file.
<code>read_ccr(fileName)</code>	Read cross-correlated relaxation values from file.

## paramagpy.dataparse.read\_pcs

paramagpy.dataparse.**read\_pcs** (*fileName*)

Read pseudo contact shift values from file. The returned object is a dicationary. They keys are tuples of (sequence, atomName) The values are tuples of (value, error)

**Parameters** **fileName** (*str*) – the path to the file

**Returns** **values** – a dictionary containing the parsed data

**Return type** *paramagpy.dataparse.DataContainer*

### Examples

```

>>> values = paramagpy.dataparse.read_pcs("calbindin_Er_HN_PCS_errors.npc")
>>> for v in values.items():
...     print(v)
...
((2, 'H'), (-0.04855485, 0.0016))
((2, 'N'), (-0.03402764, 0.0009))
((4, 'H'), (-0.18470315, 0.0004))
...
((75, 'H'), (0.19553661, 0.0005))
((75, 'N'), (0.17840666, 0.0004))

```

## paramagpy.dataparse.read\_rdc

paramagpy.dataparse.**read\_rdc** (*fileName*)

Read residual dipolar coupling values from file. The returned object is a dicationary. They keys are frozensets of tuples of the form: frozenset(({sequence1, atomName1), (sequence2, atomName2)}) The frozenset only allows unordered unique atom identification pairs The values are tuples of (value, error)

**Parameters** **fileName** (*str*) – the path to the file

**Returns** **values** – a dictionary containing the parsed data

**Return type** *paramagpy.dataparse.DataContainer*

### Examples

```

>>> values = paramagpy.dataparse.read_rdc("ubiquitin_a28c_c1_Tb_HN.rdc")
>>> for v in values.items():
...     print(v)
...
(frozenset(({2, 'N'}, (2, 'H'))), (-2.35, 0.32))
(frozenset(({3, 'N'}, (3, 'H'))), (-4.05, 0.38))
(frozenset(({4, 'H'}, (4, 'N'))), (-3.58, 0.42))
...
(frozenset(({73, 'N'}, (73, 'H'))), (-0.47, 0.75))
(frozenset(({76, 'H'}, (76, 'N'))), (0.14, 0.3))

```



## paramagpy.dataparse.read\_pre

`paramagpy.dataparse.read_pre(fileName)`

Read paramagnetic relaxation enhancement values from file. The returned object is a dicationary. They keys are tuples of (sequence, atomName) The values are tuples of (value, error)

**Parameters** `fileName` (*str*) – the path to the file

**Returns** `values` – a dictionary containing the parsed data

**Return type** `paramagpy.dataparse.DataContainer`

### Examples

see `paramagpy.dataparse.read_pcs()` which has the same file structure

## paramagpy.dataparse.read\_ccr

`paramagpy.dataparse.read_ccr(fileName)`

Read cross-correlated relaxation values from file. These are typically Curie-spin cross Dipole-dipole relaxation rates The returned object is a dicationary. They keys are tuples of the form: ((sequence1, atomName1), (sequence2, atomName2)) Note that the first column is for the active nucleus undergoing relaxation and the second column is for the partner spin. The values are tuples of (value, error)

**Parameters** `fileName` (*str*) – the path to the file

**Returns** `values` – a dictionary containing the parsed data

**Return type** `paramagpy.dataparse.DataContainer`

### Examples

see `paramagpy.dataparse.read_rdc()` which has the similar file structure

## Classes

---

<code>DataContainer(*args, **kwargs)</code>	A dictionary-like container for storing PCS, RDC, PRE and CCR data Has an additional attribute 'dtype' to define datatype
---	---

---

## paramagpy.dataparse.DataContainer

**class** `paramagpy.dataparse.DataContainer(*args, **kwargs)`

A dictionary-like container for storing PCS, RDC, PRE and CCR data Has an additional attribute 'dtype' to define datatype

**\_\_init\_\_** (*\*args, \*\*kwargs*)

Initialize self. See `help(type(self))` for accurate signature.

## Methods

<code>clear()</code>	
<code>copy()</code>	
<code>fromkeys</code>	Create a new ordered dictionary with keys from iterable and values set to value.
<code>get</code>	Return the value for key if key is in the dictionary, else default.
<code>items()</code>	
<code>keys()</code>	
<code>move_to_end</code>	Move an existing element to the end (or beginning if last is false).
<code>pop(k[,d])</code>	value.
<code>popitem</code>	Remove and return a (key, value) pair from the dictionary.
<code>setdefault</code>	Insert key with a value of default if key is not in the dictionary.
<code>update([E, ]**F)</code>	If E is present and has a .keys() method, then does: for k in E: D[k] = E[k] If E is present and lacks a .keys() method, then does: for k, v in E: D[k] = v In either case, this is followed by: for k in F: D[k] = F[k]
<code>values()</code>	

### paramagpy.dataparse.DataContainer.clear

`DataContainer.clear()` → None. Remove all items from od.

### paramagpy.dataparse.DataContainer.copy

`DataContainer.copy()` → a shallow copy of od

### paramagpy.dataparse.DataContainer.fromkeys

`DataContainer.fromkeys()`

Create a new ordered dictionary with keys from iterable and values set to value.

### paramagpy.dataparse.DataContainer.get

`DataContainer.get()`

Return the value for key if key is in the dictionary, else default.

**paramagpy.dataparse.DataContainer.items**

`DataContainer.items()` → a set-like object providing a view on D's items

**paramagpy.dataparse.DataContainer.keys**

`DataContainer.keys()` → a set-like object providing a view on D's keys

**paramagpy.dataparse.DataContainer.move\_to\_end**

`DataContainer.move_to_end()`

Move an existing element to the end (or beginning if last is false).

Raise `KeyError` if the element does not exist.

**paramagpy.dataparse.DataContainer.pop**

`DataContainer.pop(k[, d])` → v, remove specified key and return the corresponding value. If key is not found, d is returned if given, otherwise `KeyError` is raised.

**paramagpy.dataparse.DataContainer.popitem**

`DataContainer.popitem()`

Remove and return a (key, value) pair from the dictionary.

Pairs are returned in LIFO order if last is true or FIFO order if false.

**paramagpy.dataparse.DataContainer.setdefault**

`DataContainer.setdefault()`

Insert key with a value of default if key is not in the dictionary.

Return the value for key if key is in the dictionary, else default.

**paramagpy.dataparse.DataContainer.update**

`DataContainer.update([E], **F)` → None. Update D from dict/iterable E and F.

If E is present and has a `.keys()` method, then does: for k in E: D[k] = E[k] If E is present and lacks a `.keys()` method, then does: for k, v in E: D[k] = v In either case, this is followed by: for k in F: D[k] = F[k]

**paramagpy.dataparse.DataContainer.values**

`DataContainer.values()` → an object providing a view on D's values

## 5.6.4 Fitting module

This module handles the fitting of paramagnetic objects to experimental data.

---

*paramagpy.fit*

---

### paramagpy.fit

#### Functions

<i>ensemble_average</i> (dataArray)	Calculate the ensemble average for the calculated values in the column 'cal' of the argument <dataArray> over models of the PDB file.
<i>extract_atom_data</i> (data[, csa, separateModels])	Extract values required for PCS/PRE calculations
<i>extract_ccr_data</i> (data[, separateModels])	Extract values required for CCR calculations
<i>extract_rdc_data</i> (data[, separateModels])	Extract values required for RDC calculations
<i>fit_error_bootstrap</i> (fittingFunction, ...)	Perform uncertainty analysis sourcing noise from fractioning the experimental data.
<i>fit_error_models</i> (fittingFunction, **kwargs)	Perform uncertainty analysis sourcing noise from coordinates as defined by models of the PDB structure.
<i>fit_error_monte_carlo</i> (fittingFunction, ...)	Perform uncertainty analysis sourcing noise from experimental uncertainties This function takes a fitting routine <fittingFunction> and repeats it for the specified iterations in a Monte-Carlo approach.
<i>metal_standard_deviation</i> (metals, params)	Calculate the standard deviation in parameters <params> for a list of metal objects <metals>.
<i>nlr_fit_metal_from_ccr</i> (initMetals, dataArrays)	Fit Chi tensor to CCR values using non-linear regression.
<i>nlr_fit_metal_from_pcs</i> (initMetals, dataArrays)	Fit deltaChi tensor to PCS values using non-linear regression.
<i>nlr_fit_metal_from_pre</i> (initMetals, ...[, ...])	Fit Chi tensor to PRE values using non-linear regression.
<i>qfactor</i> (dataArray[, ensembleAverage, ...])	Calculate the Q-factor to judge tensor fit quality
<i>sphere_grid</i> (origin, radius, points)	Make a grid of cartesian points within a sphere
<i>svd_calc_metal_from_pcs</i> (pos, pcs, idx, errors)	Solve PCS equation by single value decomposition.
<i>svd_calc_metal_from_pcs_offset</i> (pos, pcs, ...)	Solve PCS equation by single value decomposition with offset.
<i>svd_calc_metal_from_rdc</i> (vec, ...)	Solve RDC equation by single value decomposition.
<i>svd_fit_metal_from_rdc</i> (initMetals, dataArrays)	Fit deltaChi tensor to RDC values using Single Value Decomposition.
<i>svd_gridsearch_fit_metal_from_pcs</i> (...[, ...])	Fit deltaChi tensor to PCS values using Single Value Decomposition over a grid of points in a sphere.

### paramagpy.fit.ensemble\_average

`paramagpy.fit.ensemble_average(dataArray)`

Calculate the ensemble average for the calculated values in the column 'cal' of the argument <dataArray> over models of the PDB file. Ensemble averaging behaviour is determined by the column 'idx' of the input array.

**Parameters** `dataArray` (*numpy array*) – the input array for ensemble averaging

**Returns** `data` – a smaller dataArray with ensemble averaged values

**Return type** `numpy array`

### paramagpy.fit.extract\_atom\_data

`paramagpy.fit.extract_atom_data(data, csa=False, separateModels=True)`

Extract values required for PCS/PRE calculations

**Parameters**

- **data** (*numpy.ndarray*) – a numpy structured array containing atomic information and experimental data values This is returned from. `paramagpy.protein.CustomStructure.parse()`
- **csa** (*bool, optional*) – when True, calculates the CSA tensor for each atom this may be required for RACS and CSAxDSA calculations
- **separateModels** (*bool, optional*) – when True, separates data into separate lists by their model number. When False, returns only one list

**Returns** `arr` – this has fields specified by structdtype this array is core to all fitting algorithms

**Return type** `numpy.ndarray`

### paramagpy.fit.extract\_ccr\_data

`paramagpy.fit.extract_ccr_data(data, separateModels=True)`

Extract values required for CCR calculations

**Parameters**

- **data** (*numpy.ndarray*) – a numpy structured array containing atomic information and experimental data values This is returned from `paramagpy.protein.CustomStructure.parse()`
- **separateModels** (*bool, optional*) – when True, separates data into separate lists by their model number. When False, returns only one list

**Returns** `arr` – this has fields specified by structdtype this array is core to all fitting algorithms

**Return type** `numpy.ndarray`

## paramagpy.fit.extract\_rdc\_data

`paramagpy.fit.extract_rdc_data(data, separateModels=True)`

Extract values required for RDC calculations

### Parameters

- **data** (*numpy.ndarray*) – a numpy structured array containing atomic information and experimental data values This is returned from `paramagpy.protein.CustomStructure.parse()`
- **separateModels** (*bool, optional*) – when True, separates data into separate lists by their model number. When False, returns only one list

**Returns** **arr** – this has fields specified by structdtype this array is core to all fitting algorithms

**Return type** `numpy.ndarray`

## paramagpy.fit.fit\_error\_bootstrap

`paramagpy.fit.fit_error_bootstrap(fittingFunction, iterations, fraction, **kwargs)`

Perform uncertainty analysis sourcing noise from fractioning the experimental data. This function takes a fitting routine `<fittingFunction>` and repeats it for the specified iterations in a Bootstrap approach. With each iteration, a random subset of the experimental data is sampled as specified by the `<fraction>` argument. The standard deviation in the fitted parameters is then returned.

### Parameters

- **fittingFunction** (*function*) – the fitting routine to be used. This could be `'nlr_fit_metal_from_ccr'` for example
- **iterations** (*int*) – the number of iterations for the Monte-Carlo simulation
- **kwargs** (*dict*) – all key-word arguments will be bundled into this variable and parsed to the fittingFunction.

### Returns

- **sample\_metals** (*list of list of metals*) – the metals fitted to the data with noise at each iteration
- **std\_metals** (*list of metals*) – the standard deviation in fitted parameters over all iterations of the Monte Carlo simulation. These are stored within the metal object. All unfitted parameters are zero.

## paramagpy.fit.fit\_error\_models

`paramagpy.fit.fit_error_models(fittingFunction, **kwargs)`

Perform uncertainty analysis sourcing noise from coordinates as defined by models of the PDB structure. This function takes a fitting routine `<fittingFunction>` and repeats it for each model. The standard deviation in the fitted parameters is then returned.

### Parameters

- **fittingFunction** (*function*) – the fitting routine to be used. This could be `'nlr_fit_metal_from_ccr'` for example
- **kwargs** (*dict*) – all key-word arguments will be bundled into this variable and parsed to the fittingFunction.

### Returns

- **sample\_metals** (*list of list of metals*) – the metals fitted to the data with noise at each iteration

- **std\_metals** (*list of metals*) – the standard deviation in fitted parameters over all iterations of the Monte Carlo simulation. These are stored within the metal object. All unfitted parameters are zero.

### paramagpy.fit.fit\_error\_monte\_carlo

paramagpy.fit.**fit\_error\_monte\_carlo** (*fittingFunction, iterations, \*\*kwargs*)

Perform uncertainty analysis sourcing noise from experimental uncertainties This function takes a fitting routine <fittingFunction> and repeats it for the specified iterations in a Monte-Carlo approach. With each iteration, random noise sourced from a uniform distribution scaled by the experimental uncertainties is added to the experimental values. The standard deviation in the fitted parameters is then returned.

NOTE: the 'err' column of the dataArrays must be set to non-zero values for this method to work.

#### Parameters

- **fittingFunction** (*function*) – the fitting routine to be used. This could be 'nlr\_fit\_metal\_from\_ccr' for example
- **iterations** (*int*) – the number of iterations for the Monte-Carlo simulation
- **kwargs** (*dict*) – all key-word arguments will be bundled into this variable and parsed to the fittingFunction.

#### Returns

- **sample\_metals** (*list of list of metals*) – the metals fitted to the data with noise at each iteration
- **std\_metals** (*list of metals*) – the standard deviation in fitted parameters over all iterations of the Monte Carlo simulation. These are stored within the metal object. All unfitted parameters are zero.

### paramagpy.fit.metal\_standard\_deviation

paramagpy.fit.**metal\_standard\_deviation** (*metals, params*)

Calculate the standard deviation in parameters <params> for a list of metal objects <metals>.

#### Parameters

- **metals** (*list of Metal objects*) – the metals for which the standard deviation in parameters will be calculated
- **params** (*list of str*) – the parameters for the standard deviation calculation. For example ['x','y','z','ax','rh','a','b','g','shift']

**Returns** **std\_metal** – the returned metal object has attributes equal to the standard deviation in the given parameter. All other attributes are zero.

**Return type** metal object

### paramagpy.fit.nlr\_fit\_metal\_from\_ccr

paramagpy.fit.**nlr\_fit\_metal\_from\_ccr** (*initMetals, dataArrays, params=('x', 'y', 'z'), ensembleAverage=False, progress=None*)

Fit Chi tensor to CCR values using non-linear regression. This algorithm applies to CSA/Curie spin cross-correlated relaxation for R2 differential line broadening.

#### Parameters

- **initMetals** (*list of Metal objects*) – a list of metals used as starting points for fitting. a list must always be provided, but may also contain only one element. If multiple metals are provided, each metal is fitted to their respective PRE dataset by index in <dataArray>, but all are fitted to a common position.
- **dataArray** (*list of PRE dataArray*) – each PRE dataArray must correspond to an associated metal for fitting. each PRE dataArray has structure determined by `paramagpy.protein.CustomStructure.parse()`.
- **params** (*list of str*) – the parameters to be fit. For example ['x','y','z','ax','rh','a','b','g']
- **ensembleAverage** (*bool, optional*) – when False, each model of the structure is fit independently. The parameters for each fitted tensor are then averaged before returning the final averaged tensor. When True, the structure models are treated as an ensemble and ensemble averaging of calculated PCS/PRE/RDC/CCR values is conducted at all stages of fitting to fit a single tensor to all models simultaneously. The 'idx' column of the dataArray determines the ensemble averaging behaviour with common indices for atoms between models resulting in their summation.
- **progress** (*object, optional*) – to keep track of the calculation, `progress.set(x)` is called each iteration and varies from 0.0 -> 1.0 when the calculation is complete.

#### Returns

- **fitMetals** (*list of metals*) – a list of the fitted tensors.
- **dataArray** (*list of dataArray*) – each dataArray is copy of the original dataArray with the 'cal' column populated with back-calculated values from the fitted tensor.

### paramagpy.fit.nlr\_fit\_metal\_from\_pcs

```
paramagpy.fit.nlr_fit_metal_from_pcs (initMetals, dataArray, params=('x', 'y', 'z', 'ax',
                                'rh', 'a', 'b', 'g'), ensembleAverage=False, user-
                                ads=False, useracs=False, progress=None)
```

Fit deltaChi tensor to PCS values using non-linear regression.

#### Parameters

- **initMetals** (*list of Metal objects*) – a list of metals used as starting points for fitting. a list must always be provided, but may also contain only one element. If multiple metals are provided, each metal is fitted to their respective PCS dataset by index in <dataArray>, but all are fitted to a common position.
- **dataArray** (*list of PCS dataArray*) – each PCS dataArray must correspond to an associated metal for fitting. each PCS dataArray has structure determined by `paramagpy.protein.CustomStructure.parse()`.
- **params** (*list of str*) – the parameters to be fit. For example ['x','y','z','ax','rh','a','b','g','shift']
- **ensembleAverage** (*bool, optional*) – when False, each model of the structure is fit independently. The parameters for each fitted tensor are then averaged before returning the final averaged tensor. When True, the structure models are treated as an ensemble and ensemble averaging of calculated PCS/PRE/RDC/CCR values is conducted at all stages of fitting to fit a single tensor to all models simultaneously. The 'idx' column of the dataArray determines the ensemble averaging behaviour with common indices for atoms between models resulting in their summation.
- **userads** (*bool, optional*) – include residual anisotropic dipolar shielding (RADS) during fitting
- **useracs** (*bool, optional*) – include residual anisotropic chemical shielding (RACS) during fitting. CSA tensors are taken using the <csa> method of atoms.



- **progress** (*object, optional*) – to keep track of the calculation, `progress.set(x)` is called each iteration and varies from 0.0 -> 1.0 when the calculation is complete.

#### Returns

- **fitMetals** (*list of metals*) – a list of the fitted tensors.
- **dataArrays** (*list of dataArray*) – each dataArray is copy of the original dataArray with the 'cal' column populated with back-calculated values from the fitted tensor.

### paramagpy.fit.nlr\_fit\_metal\_from\_pre

```
paramagpy.fit.nlr_fit_metal_from_pre (initMetals, dataArrays, rtypes, params=('x', 'y', 'z'),
                                     usesbm=True, usegsbm=False, usedsa=True,
                                     usecsa=False, ensembleAverage=False,
                                     progress=None)
```

Fit Chi tensor to PRE values using non-linear regression.

#### Parameters

- **initMetals** (*list of Metal objects*) – a list of metals used as starting points for fitting. a list must always be provided, but may also contain only one element. If multiple metals are provided, each metal is fitted to their respective PRE dataset by index in <dataArrays, but all are fitted to a common position.
- **dataArrays** (*list of PRE dataArray*) – each PRE dataArray must correspond to an associated metal for fitting. each PRE dataArray has structure determined by `paramagpy.protein.CustomStructure.parse()`.
- **rtypes** (*list of str, optional*) – the relaxation type, either 'r1' or 'r2'. A list must be provided with each element corresponding to an associated dataset. Defaults to 'r2' for all datasets of None is specified.
- **params** (*list of str*) – the parameters to be fit. For example ['x','y','z','ax','rh','a','b','g','shift']
- **usesbm** (*bool, optional*) – include Solomon-Bloembergen-Morgan (Dipole-dipole) relaxation theory. default is True
- **usegsbm** (*bool, optional*) – include anisotropic dipolar relaxation theory. note that the g-tensor must be set for this default is False
- **usedsa** (*bool, optional*) – include Dipolar-Shielding-Anisotropy (Curie Spin) relaxation theory. default is True
- **usecsa** (*bool, optional*) – include Chemical-Shift-Anisotropy cross-correlated relaxation theory. default is False
- **ensembleAverage** (*bool, optional*) – when False, each model of the structure is fit independently. The parameters for each fitted tensor are then averaged before returning the final averaged tensor. When True, the structure models are treated as an ensemble and ensemble averaging of calculated PCS/PRE/RDC/CCR values is conducted at all stages of fitting to fit a single tensor to all models simultaneously. The 'idx' column of the dataArray determines the ensemble averaging behaviour with common indices for atoms between models resulting in their summation.
- **progress** (*object, optional*) – to keep track of the calculation, `progress.set(x)` is called each iteration and varies from 0.0 -> 1.0 when the calculation is complete.

#### Returns

- **fitMetals** (*list of metals*) – a list of the fitted tensors.
- **dataArrays** (*list of dataArray*) – each dataArray is copy of the original dataArray with the 'cal' column populated with back-calculated values from the fitted tensor.

## paramagpy.fit.qfactor

`paramagpy.fit.qfactor(dataArray, ensembleAverage=False, calDenominator=False)`

Calculate the Q-factor to judge tensor fit quality

A lower value indicates a better fit. The Q-factor is calculated using the following equation:

$$Q = \sqrt{\frac{\sum_i \left[ \left( \sum_m [PCS_{m,i}^{exp} - PCS_{m,i}^{calc}] \right)^2 \right]}{\sum_i \left[ \left( \sum_m [PCS_{m,i}^{exp}] \right)^2 \right]}}$$

where  $m$  and  $i$  are usually indexed over models and atoms respectively.

### Parameters

- **dataArray** (*numpy array*) – the dataArray must contain the columns ‘exp’, ‘cal’ and ‘idx’ corresponding to the experimental, calculated and index values respectively. The index value determines the ensemble averaging behaviour, and can be ignored if the argument <ensembleAverage> is False.
- **ensembleAverage** (*bool, optional*) – when False, the q-factor calculation squares each difference independently. When True, the q-factor calculates an ensemble average before taking the square of differences. The ‘idx’ column of the dataArray determines the ensemble averaging behaviour with common indices for atoms between models resulting in their summation.
- **calDenominator** (*bool, optional*) – when False, the standard Q-factor is calculated with only the sum of squares for the experimental values used in the denominator when True, the Q-factor established by Ubbink et al. is calculated which has a sum of absolute values of exp and cal values squared in the denominator.

**Returns** **qfactor** – the Q-factor

**Return type** float

## paramagpy.fit.sphere\_grid

`paramagpy.fit.sphere_grid(origin, radius, points)`

Make a grid of cartesian points within a sphere

### Parameters

- **origin** (*float*) – the centre of the sphere
- **radius** (*float*) – the radius of the sphere
- **points** (*int*) – the number of points per radius

**Returns** **array** – the points within the sphere

**Return type** array of [x,y,z] coordinates

## paramagpy.fit.svd\_calc\_metal\_from\_pcs

`paramagpy.fit.svd_calc_metal_from_pcs(pos, pcs, idx, errors)`

Solve PCS equation by single value decomposition. This function is generally called by higher methods like <svd\_gridsearch\_fit\_metal\_from\_pcs>

### Parameters

- **pos** (*array of [x,y,z] floats*) – the atomic positions in meters
- **pcs** (*array of floats*) – the PCS values in ppm. Note these should be weighted by the experimental uncertainties.

- **idx** (*array of ints*) – an index assigned to each atom. Common indices determine summation between models for ensemble averaging.
- **errors** (*array of floats*) – the standard deviation representing experimental uncertainty in the measured value

#### Returns

- **calc** (*array of floats*) – the calculated PCS values from the fitted tensor
- **sol** (*array of floats*) – solution to the linearised PCS equation and consists of the tensor 5 matrix elements

### paramagpy.fit.svd\_calc\_metal\_from\_pcs\_offset

`paramagpy.fit.svd_calc_metal_from_pcs_offset(pos, pcs, idx, errors)`

Solve PCS equation by single value decomposition with offset. An offset arising from referencing errors between diamagnetic and paramagnetic datasets can be accounted for using this method. This function is generally called by higher methods like `<svd_gridsearch_fit_metal_from_pcs>`

NOTE: the factor of 1E26 is required for floating point error mitigation

#### Parameters

- **pos** (*array of [x,y,z] floats*) – the atomic positions in meters
- **pcs** (*array of floats*) – the PCS values in ppm. Note these should be weighted by the experimental uncertainties.
- **idx** (*array of ints*) – an index assigned to each atom. Common indices determine summation between models for ensemble averaging.
- **errors** (*array of floats*) – the standard deviation representing experimental uncertainty in the measured value

#### Returns

- **calc** (*array of floats*) – the calculated PCS values from the fitted tensor
- **sol** (*array of floats*) – solution to the linearised PCS equation and consists of the tensor 5 matrix elements and offset values

### paramagpy.fit.svd\_calc\_metal\_from\_rdc

`paramagpy.fit.svd_calc_metal_from_rdc(vec, rdc_parameterised, idx, errors)`

Solve RDC equation by single value decomposition. This function is generally called by higher methods like `<svd_fit_metal_from_rdc>`

#### Parameters

- **vec** (*array of [x,y,z] floats*) – the internuclear vectors in meters
- **rdc\_parameterised** (*array of floats*) – the experimental RDC values, normalised by a prefactor
- **idx** (*array of ints*) – an index assigned to each atom. Common indices determine summation between models for ensemble averaging.
- **errors** (*array of floats*) – the standard deviation representing experimental uncertainty in the measured value

#### Returns

- **calc** (*array of floats*) – the calculated RDC values from the fitted tensor
- **sol** (*array of floats*) – sol is the solution to the linearised PCS equation and consists of the tensor matrix elements

## paramagpy.fit.svd\_fit\_metal\_from\_rdc

```
paramagpy.fit.svd_fit_metal_from_rdc (initMetals, dataArrays, params=('ax', 'rh', 'a', 'b',
                                                                    'g'), ensembleAverage=False, progress=None)
```

Fit deltaChi tensor to RDC values using Single Value Decomposition. Note this is a weighted SVD calculation which takes into account experimental errors.

### Parameters

- **initMetals** (*list of Metal objects*) – a list of metals used as starting points for fitting. a list must always be provided, but may also contain only one element. If multiple metals are provided, each metal is fitted to their respective RDC dataset by index in <dataArrays>.
- **dataArrays** (*list of PRE dataArray*) – each RDC dataArray must correspond to an associated metal for fitting. each RDC dataArray has structure determined by `paramagpy.protein.CustomStructure.parse()`.
- **params** (*list of str*) – the parameters to be fit. NOTE: This is a dummy argument and does not influence the fitting. The default parameters ('ax','rh','a','b','g') are the only option.
- **ensembleAverage** (*bool, optional*) – when False, each model of the structure is fit independently. The parameters for each fitted tensor are then averaged before returning the final averaged tensor. When True, the structure models are treated as an ensemble and ensemble averaging of calculated PCS/PRE/RDC/CCR values is conducted at all stages of fitting to fit a single tensor to all models simultaneously. The 'idx' column of the dataArray determines the ensemble averaging behaviour with common indices for atoms between models resulting in their summation.
- **progress** (*object, optional*) – to keep track of the calculation, `progress.set(x)` is called each iteration and varies from 0.0 -> 1.0 when the calculation is complete.

### Returns

- **fitMetals** (*list of metals*) – a list of the fitted tensors.
- **dataArrays** (*list of dataArray*) – each dataArray is copy of the original dataArray with the 'cal' column populated with back-calculated values from the fitted tensor.

## paramagpy.fit.svd\_gridsearch\_fit\_metal\_from\_pcs

```
paramagpy.fit.svd_gridsearch_fit_metal_from_pcs (initMetals, dataArrays, ensembleAverage=False, origin=None,
                                                  radius=20.0, points=16, offsetShift=False, progress=None)
```

Fit deltaChi tensor to PCS values using Single Value Decomposition over a grid of points in a sphere. Note this uses a weighted SVD fit which takes into account experimental errors

### Parameters

- **initMetals** (*list of Metal objects*) – a list of metals used as starting points for fitting. a list must always be provided, but may also contain only one element. If multiple metals are provided, each metal is fitted to their respective PCS dataset by index in <dataArrays>, but all are fitted to a common position.
- **dataArrays** (*list of PCS dataArray*) – each PCS dataArray must correspond to an associated metal for fitting. each PCS dataArray has structure determined by `paramagpy.protein.CustomStructure.parse()`.
- **ensembleAverage** (*bool, optional*) – when False, each model of the structure is fit independently. The parameters for each fitted tensor are then averaged before returning the final averaged tensor. When True, the structure models are treated as

an ensemble and ensemble averaging of calculated PCS/PRE/RDC/CCR values is conducted at all stages of fitting to fit a single tensor to all models simultaneously. The 'idx' column of the dataArray determines the ensemble averaging behaviour with common indices for atoms between models resulting in their summation.

- **origin** (*float, optional*) – the centre of the gridsearch of positions in Angstroms. If None, the position of the first metal is used
- **radius** (*float, optional*) – the radius of the gridsearch in Angstroms.
- **points** (*int, optional*) – the number of points per radius in the gridsearch
- **offsetShift** (*bool, optional*) – if True, an offset value added to all PCS values is included in the SVD fitting. This may arise due to a referencing error between diamagnetic and paramagnetic PCS datasets and may be used when many data points are available. Default False, no offset is included in the fitting.
- **progress** (*object, optional*) – to keep track of the calculation, progress.set(x) is called each iteration and varies from 0.0 -> 1.0 when the calculation is complete.

#### Returns

- **fitMetals** (*list of metals*) – a list of the fitted tensors.
- **dataArrays** (*list of dataArray*) – each dataArray is copy of the original dataArray with the 'cal' column populated with back-calculated values from the fitted tensor.

## PYTHON MODULE INDEX

### p

`paramagpy.dataparse`, [81](#)  
`paramagpy.fit`, [86](#)  
`paramagpy.metal`, [46](#)  
`paramagpy.protein`, [68](#)

## Symbols

`__init__()` (*paramagpy.dataparse.DataContainer method*), 83  
`__init__()` (*paramagpy.metal.Metal method*), 48  
`__init__()` (*paramagpy.protein.CustomAtom method*), 69  
`__init__()` (*paramagpy.protein.CustomStructure method*), 75  
`__init__()` (*paramagpy.protein.CustomStructureBuilder method*), 79

## A

`a` (*paramagpy.metal.Metal attribute*), 64  
`add()` (*paramagpy.protein.CustomStructure method*), 76  
`alignment_factor` (*paramagpy.metal.Metal attribute*), 64  
`atom_ccr()` (*paramagpy.metal.Metal method*), 50  
`atom_pcs()` (*paramagpy.metal.Metal method*), 50  
`atom_pre()` (*paramagpy.metal.Metal method*), 51  
`atom_rdc()` (*paramagpy.metal.Metal method*), 51  
`atom_set_position()` (*paramagpy.metal.Metal method*), 51  
`average()` (*paramagpy.metal.Metal method*), 51  
`ax` (*paramagpy.metal.Metal attribute*), 64

## B

`b` (*paramagpy.metal.Metal attribute*), 65  
`B0_MHz` (*paramagpy.metal.Metal attribute*), 64

## C

`ccr()` (*paramagpy.metal.Metal method*), 51  
`clear()` (*paramagpy.dataparse.DataContainer method*), 84  
`copy()` (*paramagpy.dataparse.DataContainer method*), 84  
`copy()` (*paramagpy.metal.Metal method*), 52  
`copy()` (*paramagpy.protein.CustomAtom method*), 70  
`copy()` (*paramagpy.protein.CustomStructure method*), 76  
`csa` (*paramagpy.protein.CustomAtom attribute*), 75  
`csa_lib` (*paramagpy.protein.CustomAtom attribute*), 75  
`CustomAtom` (*class in paramagpy.protein*), 69  
`CustomStructure` (*class in paramagpy.protein*), 75

`CustomStructureBuilder` (*class in paramagpy.protein*), 79

## D

`DataContainer` (*class in paramagpy.dataparse*), 83  
`detach_child()` (*paramagpy.protein.CustomStructure method*), 76  
`detach_parent()` (*paramagpy.protein.CustomAtom method*), 70  
`detach_parent()` (*paramagpy.protein.CustomStructure method*), 76  
`dipole_shift_tensor()` (*paramagpy.metal.Metal method*), 52  
`dipole_shift_tensor()` (*paramagpy.protein.CustomAtom method*), 70  
`dsa_r1()` (*paramagpy.metal.Metal method*), 52  
`dsa_r2()` (*paramagpy.metal.Metal method*), 52

## E

`eigenvalues` (*paramagpy.metal.Metal attribute*), 65  
`ensemble_average()` (*in module paramagpy.fit*), 87  
`euler_to_matrix()` (*in module paramagpy.metal*), 46  
`extract_atom_data()` (*in module paramagpy.fit*), 87  
`extract_ccr_data()` (*in module paramagpy.fit*), 87  
`extract_rdc_data()` (*in module paramagpy.fit*), 88

## F

`fast_ccr()` (*paramagpy.metal.Metal method*), 53  
`fast_dipole_shift_tensor()` (*paramagpy.metal.Metal method*), 53  
`fast_dsa_r1()` (*paramagpy.metal.Metal method*), 53  
`fast_dsa_r2()` (*paramagpy.metal.Metal method*), 53  
`fast_first_invariant_squared()` (*paramagpy.metal.Metal static method*), 54  
`fast_g_sbm_r1()` (*paramagpy.metal.Metal method*), 54  
`fast_pcs()` (*paramagpy.metal.Metal method*), 54

fast\_pre() (*paramagpy.metal.Metal method*), 54  
 fast\_racs() (*paramagpy.metal.Metal method*), 55  
 fast\_rads() (*paramagpy.metal.Metal method*), 55  
 fast\_rdc() (*paramagpy.metal.Metal method*), 55  
 fast\_sbm\_r1() (*paramagpy.metal.Metal method*), 55  
 fast\_sbm\_r2() (*paramagpy.metal.Metal method*), 56  
 fast\_second\_invariant\_squared() (*paramagpy.metal.Metal static method*), 56  
 first\_invariant\_squared() (*paramagpy.metal.Metal static method*), 56  
 fit\_error\_bootstrap() (*in module paramagpy.fit*), 88  
 fit\_error\_models() (*in module paramagpy.fit*), 88  
 fit\_error\_monte\_carlo() (*in module paramagpy.fit*), 89  
 fit\_scaling (*paramagpy.metal.Metal attribute*), 65  
 flag\_disorder() (*paramagpy.protein.CustomAtom method*), 71  
 fromkeys() (*paramagpy.dataparse.DataContainer method*), 84  
 fundamental\_attributes (*paramagpy.metal.Metal attribute*), 65

## G

g (*paramagpy.metal.Metal attribute*), 65  
 g\_eigenvalues (*paramagpy.metal.Metal attribute*), 65  
 g\_isotropy (*paramagpy.metal.Metal attribute*), 65  
 g\_sbm\_r1() (*paramagpy.metal.Metal method*), 56  
 g\_tensor (*paramagpy.metal.Metal attribute*), 65  
 GAMMA (*paramagpy.metal.Metal attribute*), 64  
 gax (*paramagpy.metal.Metal attribute*), 65  
 get() (*paramagpy.dataparse.DataContainer method*), 84  
 get\_altloc() (*paramagpy.protein.CustomAtom method*), 71  
 get\_anisou() (*paramagpy.protein.CustomAtom method*), 71  
 get\_atoms() (*paramagpy.protein.CustomStructure method*), 76  
 get\_bfactor() (*paramagpy.protein.CustomAtom method*), 71  
 get\_chains() (*paramagpy.protein.CustomStructure method*), 76  
 get\_coord() (*paramagpy.protein.CustomAtom method*), 71  
 get\_full\_id() (*paramagpy.protein.CustomAtom method*), 71  
 get\_full\_id() (*paramagpy.protein.CustomStructure method*), 77  
 get\_fullname() (*paramagpy.protein.CustomAtom method*), 71

get\_id() (*paramagpy.protein.CustomAtom method*), 71  
 get\_id() (*paramagpy.protein.CustomStructure method*), 77  
 get\_iterator() (*paramagpy.protein.CustomStructure method*), 77  
 get\_level() (*paramagpy.protein.CustomAtom method*), 72  
 get\_level() (*paramagpy.protein.CustomStructure method*), 77  
 get\_list() (*paramagpy.protein.CustomStructure method*), 77  
 get\_models() (*paramagpy.protein.CustomStructure method*), 77  
 get\_name() (*paramagpy.protein.CustomAtom method*), 72  
 get\_occupancy() (*paramagpy.protein.CustomAtom method*), 72  
 get\_params() (*paramagpy.metal.Metal method*), 57  
 get\_parent() (*paramagpy.protein.CustomAtom method*), 72  
 get\_parent() (*paramagpy.protein.CustomStructure method*), 77  
 get\_residues() (*paramagpy.protein.CustomStructure method*), 78  
 get\_serial\_number() (*paramagpy.protein.CustomAtom method*), 72  
 get\_sigatm() (*paramagpy.protein.CustomAtom method*), 72  
 get\_siguij() (*paramagpy.protein.CustomAtom method*), 72  
 get\_structure() (*paramagpy.protein.CustomStructureBuilder method*), 79  
 get\_vector() (*paramagpy.protein.CustomAtom method*), 72  
 grh (*paramagpy.metal.Metal attribute*), 66  
 gyro\_lib (*paramagpy.protein.CustomAtom attribute*), 75

## H

has\_id() (*paramagpy.protein.CustomStructure method*), 78  
 HBAR (*paramagpy.metal.Metal attribute*), 64  
 HBAR (*paramagpy.protein.CustomAtom attribute*), 75

## I

id (*paramagpy.protein.CustomStructure attribute*), 79  
 info() (*paramagpy.metal.Metal method*), 57  
 init\_atom() (*paramagpy.protein.CustomStructureBuilder method*), 79



`init_chain()` (*paramagpy.protein.CustomStructureBuilder method*), 80  
`init_model()` (*paramagpy.protein.CustomStructureBuilder method*), 80  
`init_residue()` (*paramagpy.protein.CustomStructureBuilder method*), 80  
`init_seg()` (*paramagpy.protein.CustomStructureBuilder method*), 80  
`init_structure()` (*paramagpy.protein.CustomStructureBuilder method*), 80  
`insert()` (*paramagpy.protein.CustomStructure method*), 78  
`is_disordered()` (*paramagpy.protein.CustomAtom method*), 73  
`iso` (*paramagpy.metal.Metal attribute*), 66  
`isomap()` (*paramagpy.metal.Metal method*), 58  
`isotropy` (*paramagpy.metal.Metal attribute*), 66  
`items()` (*paramagpy.dataparse.DataContainer method*), 85

## K

`K` (*paramagpy.metal.Metal attribute*), 64  
`keys()` (*paramagpy.dataparse.DataContainer method*), 85

## L

`lanth_axrh` (*paramagpy.metal.Metal attribute*), 66  
`lanth_lib` (*paramagpy.metal.Metal attribute*), 66  
`load_pdb()` (*in module paramagpy.protein*), 68  
`lower_coords` (*paramagpy.metal.Metal attribute*), 66

## M

`make_mesh()` (*paramagpy.metal.Metal method*), 58  
`make_tensor()` (*in module paramagpy.metal*), 47  
`matrix_to_euler()` (*in module paramagpy.metal*), 46  
`Metal` (*class in paramagpy.metal*), 48  
`metal_standard_deviation()` (*in module paramagpy.fit*), 89  
`move_to_end()` (*paramagpy.dataparse.DataContainer method*), 85  
`MU0` (*paramagpy.metal.Metal attribute*), 64  
`MU0` (*paramagpy.protein.CustomAtom attribute*), 75  
`MUB` (*paramagpy.metal.Metal attribute*), 64

## N

`nlr_fit_metal_from_ccr()` (*in module paramagpy.fit*), 89  
`nlr_fit_metal_from_pcs()` (*in module paramagpy.fit*), 90

`nlr_fit_metal_from_pre()` (*in module paramagpy.fit*), 91

## P

`paramagpy.dataparse` (*module*), 81  
`paramagpy.fit` (*module*), 86  
`paramagpy.metal` (*module*), 46  
`paramagpy.protein` (*module*), 68  
`parse()` (*paramagpy.protein.CustomStructure method*), 78  
`pcs()` (*paramagpy.metal.Metal method*), 58  
`pcs_mesh()` (*paramagpy.metal.Metal method*), 58  
`pop()` (*paramagpy.dataparse.DataContainer method*), 85  
`popitem()` (*paramagpy.dataparse.DataContainer method*), 85  
`position` (*paramagpy.protein.CustomAtom attribute*), 75  
`pre()` (*paramagpy.metal.Metal method*), 59  
`pre_mesh()` (*paramagpy.metal.Metal method*), 59

## Q

`qfactor()` (*in module paramagpy.fit*), 92

## R

`racs()` (*paramagpy.metal.Metal method*), 59  
`rads()` (*paramagpy.metal.Metal method*), 60  
`rdc()` (*paramagpy.metal.Metal method*), 60  
`read_ccr()` (*in module paramagpy.dataparse*), 83  
`read_pcs()` (*in module paramagpy.dataparse*), 82  
`read_pre()` (*in module paramagpy.dataparse*), 83  
`read_rdc()` (*in module paramagpy.dataparse*), 82  
`rh` (*paramagpy.metal.Metal attribute*), 66  
`rotation_matrix()` (*in module paramagpy.protein*), 69  
`rotationMatrix` (*paramagpy.metal.Metal attribute*), 66

## S

`saupe_factor` (*paramagpy.metal.Metal attribute*), 66  
`save()` (*paramagpy.metal.Metal method*), 60  
`sbm_r1()` (*paramagpy.metal.Metal method*), 60  
`sbm_r2()` (*paramagpy.metal.Metal method*), 60  
`second_invariant_squared()` (*paramagpy.metal.Metal static method*), 61  
`set_altloc()` (*paramagpy.protein.CustomAtom method*), 73  
`set_anisou()` (*paramagpy.protein.CustomAtom method*), 73  
`set_anisou()` (*paramagpy.protein.CustomStructureBuilder method*), 80  
`set_bfactor()` (*paramagpy.protein.CustomAtom method*), 73  
`set_coord()` (*paramagpy.protein.CustomAtom method*), 73

set\_header() (paramagpy.protein.CustomStructureBuilder method), 81  
 set\_Jg() (paramagpy.metal.Metal method), 61  
 set\_lanthanide() (paramagpy.metal.Metal method), 61  
 set\_line\_counter() (paramagpy.protein.CustomStructureBuilder method), 81  
 set\_occupancy() (paramagpy.protein.CustomAtom method), 73  
 set\_params() (paramagpy.metal.Metal method), 61  
 set\_parent() (paramagpy.protein.CustomAtom method), 73  
 set\_parent() (paramagpy.protein.CustomStructure method), 78  
 set\_serial\_number() (paramagpy.protein.CustomAtom method), 73  
 set\_sigatm() (paramagpy.protein.CustomAtom method), 74  
 set\_sigatm() (paramagpy.protein.CustomStructureBuilder method), 81  
 set\_siguij() (paramagpy.protein.CustomAtom method), 74  
 set\_siguij() (paramagpy.protein.CustomStructureBuilder method), 81  
 set\_symmetry() (paramagpy.protein.CustomStructureBuilder method), 81  
 set\_utr() (paramagpy.metal.Metal method), 62  
 setdefault() (paramagpy.dataparse.DataContainer method), 85  
 spec\_dens() (paramagpy.metal.Metal static method), 62  
 sphere\_grid() (in module paramagpy.fit), 92  
 svd\_calc\_metal\_from\_pcs() (in module paramagpy.fit), 92  
 svd\_calc\_metal\_from\_pcs\_offset() (in module paramagpy.fit), 93  
 svd\_calc\_metal\_from\_rdc() (in module paramagpy.fit), 93  
 svd\_fit\_metal\_from\_rdc() (in module paramagpy.fit), 94  
 svd\_gridsearch\_fit\_metal\_from\_pcs() (in module paramagpy.fit), 94

**T**

tau\_c (paramagpy.metal.Metal attribute), 67  
 tensor (paramagpy.metal.Metal attribute), 67  
 tensor\_alignment (paramagpy.metal.Metal attribute), 67  
 tensor\_saupe (paramagpy.metal.Metal attribute), 67

tensor\_traceless (paramagpy.metal.Metal attribute), 67  
 top() (paramagpy.protein.CustomAtom method), 74  
 transform() (paramagpy.protein.CustomAtom method), 74  
 transform() (paramagpy.protein.CustomStructure method), 78

**U**

unique\_eulers() (in module paramagpy.metal), 47  
 update() (paramagpy.dataparse.DataContainer method), 85  
 upper\_coors (paramagpy.metal.Metal attribute), 67  
 upper\_triangular (paramagpy.metal.Metal attribute), 67  
 upper\_triangular\_alignment (paramagpy.metal.Metal attribute), 67  
 upper\_triangular\_saupe (paramagpy.metal.Metal attribute), 68

**V**

values() (paramagpy.dataparse.DataContainer method), 85

**W**

write\_isomap() (paramagpy.metal.Metal method), 62  
 write\_pymol\_script() (paramagpy.metal.Metal method), 62

**X**

x (paramagpy.metal.Metal attribute), 68

**Y**

y (paramagpy.metal.Metal attribute), 68

**Z**

z (paramagpy.metal.Metal attribute), 68

## **Appendix B**

# **Accurate Electron-Nucleus Distances from Paramagnetic Relaxation Enhancements**



Australian  
National  
University

## Statement of Contribution

This thesis is submitted as a Thesis by Compilation in accordance with:

[https://policies.anu.edu.au/ppl/document/ANUP\\_003405](https://policies.anu.edu.au/ppl/document/ANUP_003405)

I declare that the research presented in this Thesis represents original work that I carried out during my candidature at the Australian National University, except for contributions to multi-author papers incorporated in the Thesis where my contributions are specified in this Statement of Contribution.

**Title:** Accurate electron-nucleus distances from paramagnetic relaxation enhancements

**Authors:** Henry W. Orton and Gottfried Otting

**Publication Outlet:** Journal of the American Chemical Society

**Current status of paper:** Published

**Contribution:** I contributed all aspects of this publication including cloning, expression and purification of samples, design, acquisition and processing of NMR experiments and writing of the manuscript.

Henry William Orton

CANDIDATE

SIGNATURE

10/02/2020

DATE

Endorsed

Gottfried Otting

SENIOR AUTHOUR

SIGNATURE

10/02/2020

DATE

Gottfried Otting

PRIMARY SUPERVISOR

SIGNATURE

10/02/2020

DATE

Kate Connell

DELEGATED AUTHORITY

SIGNATURE

11/02/2020

DATE

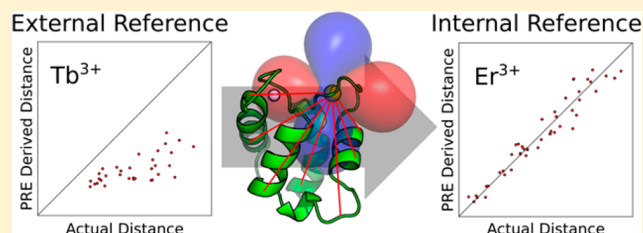
# Accurate Electron–Nucleus Distances from Paramagnetic Relaxation Enhancements

Henry W. Orton and Gottfried Otting\*

Research School of Chemistry, Australian National University, Canberra, Australian Capital Territory 2601, Australia

**S** Supporting Information

**ABSTRACT:** Measurements of paramagnetic relaxation enhancements (PREs) in  $^1\text{H}$  NMR spectra are an important tool to obtain long-range distance information in proteins, but quantitative interpretation is easily compromised by nonspecific intermolecular PREs. Here we show that PREs generated by lanthanides with anisotropic magnetic susceptibilities offer a route to accurate calibration-free distance measurements. As these lanthanides change  $^1\text{H}$  chemical shifts due to pseudocontact shifts, the relaxation rates in the paramagnetic and diamagnetic state can be measured with a single sample that simultaneously contains the protein labeled with a paramagnetic and a diamagnetic lanthanide ion. Nonspecific intermolecular PREs are thus automatically subtracted when calculating the PREs as the difference in nuclear relaxation rates between paramagnetic and diamagnetic protein. Although PREs from lanthanides with anisotropic magnetic susceptibilities are complicated by additional cross-correlation effects and residual dipolar couplings (RDCs) in the paramagnetic state, these effects can be controlled by the choice of lanthanide ion and experimental conditions. Using calbindin  $\text{D}_{9k}$  with erbium, we succeeded in measuring intramolecular PREs with unprecedented accuracy, resulting in distance predictions with a root-mean-square-deviation of  $<0.9$  Å in the range 11–24 Å.



## INTRODUCTION

Relaxation rate enhancements of nuclear spins induced by paramagnetic probes have long played an important role for NMR spectroscopic investigations of proteins to obtain long-range distance restraints and refine molecular structure.<sup>1–15</sup> Most proteins are not paramagnetic but can be furnished with a paramagnetic tag such as a nitroxide spin label. In view of the popularity of paramagnetic relaxation enhancements (PREs), it is surprising that distances derived from PREs have rarely been compared with those found in crystal structures in a quantitative manner. In the case of nitroxide tags, flexibility of the tether between nitroxide group and protein backbone poses an intrinsic difficulty to translate PREs into distances between the nitroxide and the nuclear spins. Where PREs generated by nitroxide tags have been calibrated against electron–nucleus distances, the uncertainties were large and distances outside the range 15–24 Å proved inaccessible in solution.<sup>3,16,17</sup> Similar results have also been obtained for RNA.<sup>18–20</sup> In the example of a copper protein, PREs determined from the longitudinal relaxation rates of protons,  $R_1(^1\text{H})$ , correlated reasonably well with  $\text{Cu}^{2+}$ – $^1\text{H}$  distances measured in a solution NMR structure, but with systematic deviations for distances greater than 10 Å.<sup>21</sup> For perdeuterated proteins in the solid state, distance measurements up to 32 Å have been reported, but with very large outliers.<sup>22</sup>

Alternatively, metal ions such as lanthanide ions can serve as paramagnetic probes. Using lanthanides to generate PREs is attractive because lanthanide ions are chemically very similar while their paramagnetism varies greatly.<sup>23</sup> Therefore, once a

lanthanide binding site has been established, either directly in the protein or by tagging the protein with a suitable complex,<sup>24</sup> different lanthanide ions can be used to generate easily measurable PREs at different distances from the paramagnetic center. While this concept may appear attractive, it has only rarely been applied and the results have been discouraging.

An early investigation of carp parvalbumin loaded with  $\text{Yb}^{3+}$  derived 12 electron–nucleus distances from PREs of unassigned protons, but these were never confirmed by comparison with the crystal structure.<sup>25</sup> More recently, a 3D structure determination of calbindin  $\text{D}_{9k}$  was assisted by 26 distance restraints derived from  $R_1(^1\text{H})$  PREs measured with  $\text{Ce}^{3+}$ , but the conversion of PREs to distance restraints was adjusted during the structure computation rather than determined from first principles.<sup>26</sup> More recently, a lanthanide binding peptide was engineered into an immunoglobulin G binding protein and loaded with either  $\text{Yb}^{3+}$  or  $\text{Dy}^{3+}$  to yield distance restraints from PREs of the transverse  $^1\text{H}$  relaxation rate,  $R_2(^1\text{H})$ . It was noted that the experimental  $R_2(^1\text{H})$  PRE values determined with  $\text{Dy}^{3+}$  were much larger than expected.<sup>27</sup> The situation is complicated further by theoretical considerations, which predict that lanthanide ions other than  $\text{Gd}^{3+}$  can also produce *negative* PREs, where the nuclear relaxation is slower in the paramagnetic than in the diamagnetic state.<sup>28,29</sup> Experimental examples of this have recently been demonstrated for  $^{15}\text{N}$  relaxation in lanthanide-loaded calbindin  $\text{D}_{9k}$ .<sup>30</sup>

Received: April 10, 2018

Published: May 23, 2018



Among the paramagnetic lanthanide ions, the  $\text{Gd}^{3+}$  ion stands out for its large magnetic susceptibility, which is isotropic and driven by seven unpaired electrons with relatively slow electronic relaxation rates. Nuclear relaxation arising from the paramagnetism of a  $\text{Gd}^{3+}$  ion is described by the Solomon–Bloembergen–Morgan (SBM) theory.<sup>31</sup> Any other paramagnetic lanthanide ion is characterized by a magnetic susceptibility tensor  $\chi$ , which is anisotropic; its electronic relaxation times are short, and nuclear relaxation arising from these ions is described by both Curie-spin relaxation theory, which depends on the magnetic susceptibility tensor  $\chi$ ,<sup>32</sup> and SBM relaxation.<sup>28</sup> The problem associated with interpreting PREs arising from lanthanide ions other than  $\text{Gd}^{3+}$  lies in the additional effects caused by a nonvanishing  $\Delta\chi$  tensor. This tensor anisotropy produces, in particular, pseudocontact shifts (PCSs), which change the chemical shifts in the paramagnetic state,<sup>33</sup> and residual dipolar couplings (RDCs), which arise from partial molecular alignment induced by the  $\Delta\chi$  tensor.<sup>34</sup>

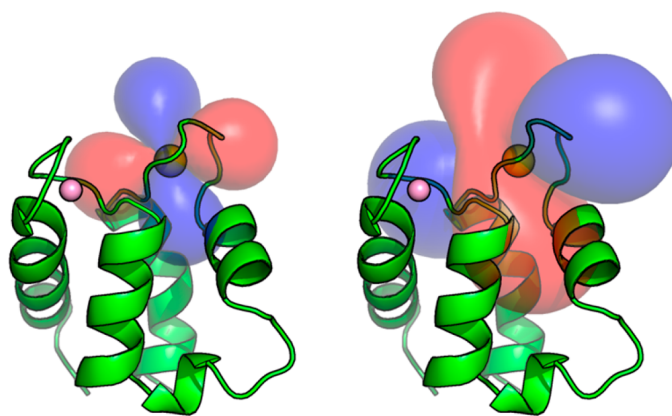
PREs generated by lanthanide ions with rapid electron relaxation arise from both the Curie-spin mechanism<sup>32</sup> and the dipole–dipole or SBM mechanism.<sup>31</sup> In the case of slowly tumbling proteins,  $R_2(^1\text{H})$  PREs are predominantly governed by Curie-spin relaxation, whereas  $R_1(^1\text{H})$  PREs feature significant contributions from the SBM relaxation mechanism.<sup>28</sup> As both Curie-spin and SBM relaxation effects decrease with the inverse sixth power of the distance  $r$  from the paramagnetic center, PREs carry valuable structural information regardless of the relative contributions of these two relaxation mechanisms.<sup>5,35,36</sup>

Curie-spin relaxation can be described by a dipolar shielding anisotropy (DSA) at the location of the nuclear spin.<sup>33</sup> As molecular tumbling drives both DSA and chemical shift anisotropy (CSA) relaxation, DSA–CSA cross-correlation effects can substantially alter the overall PRE, when the CSA tensor is large as in the case of  $^{15}\text{N}$  spins. In this case, any conversion of PREs to distance restraints is impractical.<sup>30</sup> In the case of aliphatic  $^1\text{H}$  spins, however, the CSA effects are much less dominant and substantial deviations between experimental and predicted PREs need a different explanation. To explore the utility of  $^1\text{H}$  PREs generated by paramagnetic lanthanide ions other than  $\text{Gd}^{3+}$  (which causes SBM relaxation only), we used calbindin  $\text{D}_{9k}$ .

Calbindin  $\text{D}_{9k}$  is a protein with two calcium binding sites, where the  $\text{Ca}^{2+}$  ion at site-II is readily replaced by a lanthanide ion.<sup>37</sup> The present work used the double mutant P43G/N56D of calbindin  $\text{D}_{9k}$ . The Pro43Gly mutation prevents cis–trans isomerization of the peptide bond between residues 42 and 43 and has been used in previous NMR studies.<sup>26,38</sup> The Asn56Asp mutation abolishes deamidation of residue 56 by internal isomerization.<sup>30,39</sup> With site-I filled with a  $\text{Ca}^{2+}$  ion and site-II filled with a lanthanide ion ( $\text{Ln}^{3+}$ ; Figure 1), we refer to the double mutant P43G/N56D as CbCaLn. Using CbCaLn with  $\text{Ln} = \text{Er}$  or  $\text{Tb}$ , we show that the influence of PCSs and RDCs on  $R_2(^1\text{H})$  PRE measurements can be controlled and, most importantly, that the PCSs open a way to eliminate the effects from nonspecific intermolecular relaxation, leading to more accurate distance measurements over a greater range of distances from the metal ion.

## RESULTS

**Samples.** Initially, we used two-dimensional (2D)  $[^{15}\text{N}, ^1\text{H}]$ -HSQC-type spectra to measure the chemical shifts and  $^1\text{H}$  relaxation rates of the double mutant P43G/N56D of calbindin

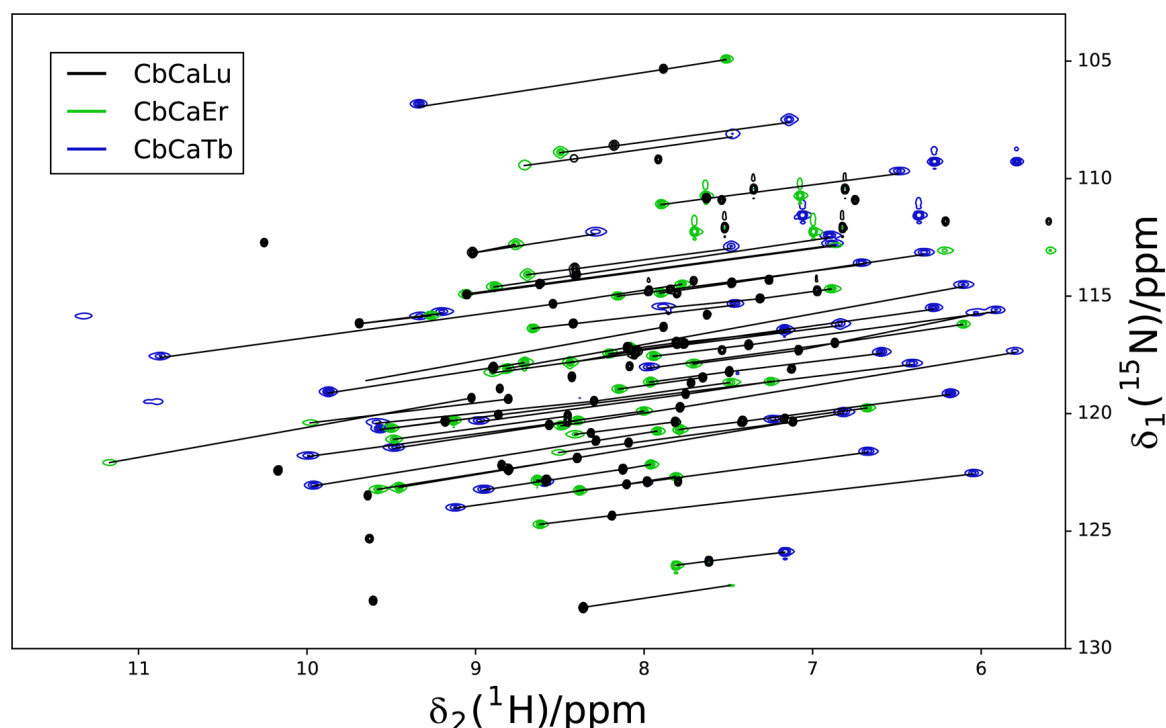


**Figure 1.** Crystal structure of calbindin  $\text{D}_{9k}$  (PDB ID: 4ICB<sup>40</sup>) with PCS isosurfaces. The  $\text{Ca}^{2+}$  ion in the N-terminal binding site-I is shown in pink. Red and blue surfaces represent PCSs of  $-4$  ppm and  $+4$  ppm, respectively. The PCS isosurfaces are centered about the  $\text{Ln}^{3+}$  ion represented by a larger sphere. Tensor parameters are from Table S1. Left: Isosurfaces for CbCaEr. Right: Isosurfaces for CbCaTb.

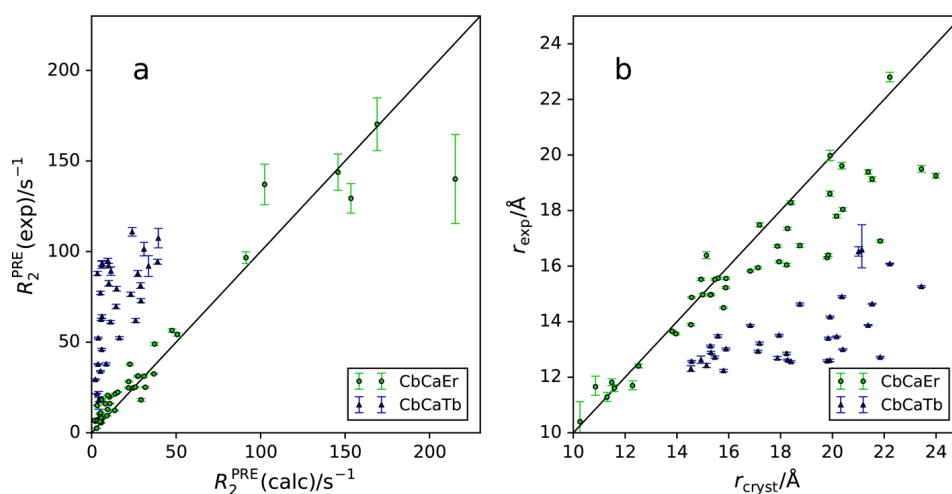
$\text{D}_{9k}$  loaded with  $\text{Tb}^{3+}$ ,  $\text{Er}^{3+}$ , or  $\text{Lu}^{3+}$  (CbCaTb, CbCaEr, and CbCaLu, respectively) in separate experiments. In addition, explorative data were collected with  $\text{Ho}^{3+}$ ,  $\text{Dy}^{3+}$ , and  $\text{Tm}^{3+}$ . These paramagnetic lanthanides possess magnetic susceptibilities of similar magnitudes<sup>23</sup> and therefore should produce similar PREs. Their  $\Delta\chi$  tensors, however, vary significantly, allowing an assessment of any impact of the correspondingly varied PCSs and RDCs on the PRE measurements. Calbindin  $\text{D}_{9k}$  binds calcium ions with submicromolar affinity at neutral pH,<sup>41</sup> and preferential binding of lanthanide ions has been demonstrated at pH 6.<sup>37</sup> The current study used samples at pH 6.5.

Example  $[^{15}\text{N}, ^1\text{H}]$ -HSQC spectra are shown in Figure 2. Cross-peaks were observed for 72 backbone amides in CbCaLu. In the paramagnetic samples with  $\text{Er}^{3+}$  and  $\text{Tb}^{3+}$ , cross-peaks from amides closer than  $10 \text{ \AA}$  to the paramagnetic center were broadened beyond detection because of the PRE effect. After removing overlapping peaks, CbCaEr and CbCaTb yielded 45 and 33 PCSs, respectively. Using the crystal structure of calbindin  $\text{D}_{9k}$  (PDB ID: 4ICB<sup>40</sup>) to fit the  $\Delta\chi$  tensors revealed a  $\Delta\chi_{\text{ax}}$  parameter about 4-fold larger for CbCaTb than for CbCaEr (Table S1). Figure 1 displays the tensors in an isosurface representation. The larger  $\Delta\chi$  tensor associated with the  $\text{Tb}^{3+}$  ion is also reflected by the  $[^{15}\text{N}, ^1\text{H}]$ -HSQC spectrum, which showed overall larger PCSs for CbCaTb. The magnitudes of the axial and rhombic components of the fitted tensors were about one-third smaller than those reported previously for calbindin  $\text{D}_{9k}$ ,<sup>37</sup> which may be attributed to the mutation N56D at the lanthanide binding site or the difference in pH.

**$R_2(^1\text{H})$  Relaxation Measurements.** Initial measurements of the transverse relaxation rates  $R_2(^1\text{H})$  conducted with CbCaEr, CbCaTb, and CbCaLu yielded disappointing correlations between experimental and expected PREs, resulting in correspondingly poor distance measurements (Figure 3). In particular, the apparent PREs measured with  $\text{Tb}^{3+}$  were much too large, and while more reliable, the PRE measurements with  $\text{Er}^{3+}$  systematically underestimated distances longer than  $\sim 16 \text{ \AA}$ , resulting in an overall root-mean-squared deviation (RMSD) of  $1.8 \text{ \AA}$ . These results point to the effects from RDCs, which dephase transverse magnetization during the transverse relaxation delay, and intermolecular PREs.



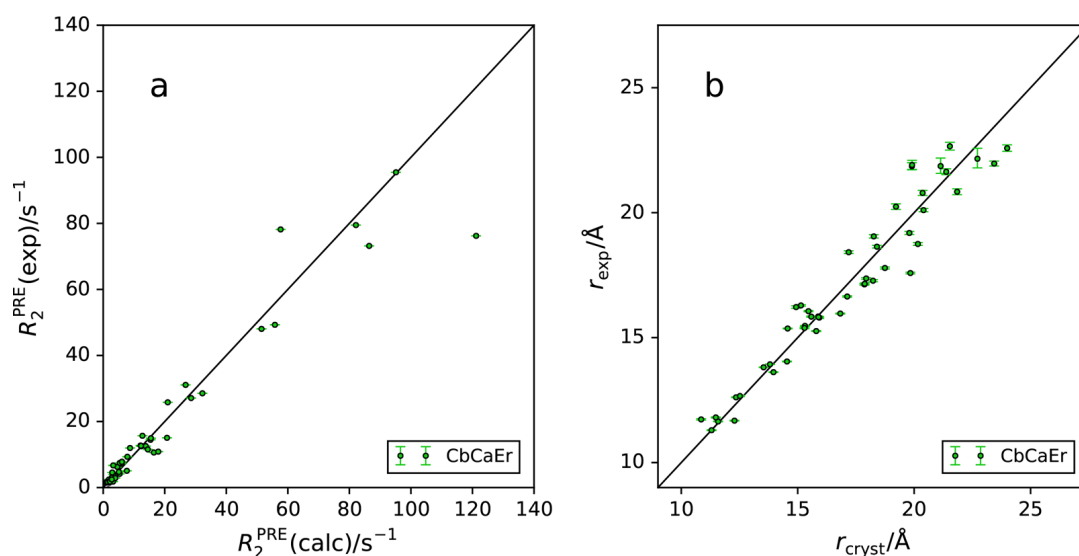
**Figure 2.**  $^{15}\text{N}, ^1\text{H}$ -HSQC spectra of a 0.2 mM solution of calbindin  $\text{D}_{9\text{k}}$  P43G/N56D loaded with a  $\text{Lu}^{3+}$  (black),  $\text{Er}^{3+}$  (green), or  $\text{Tb}^{3+}$  (blue) ion. Black lines connect cross-peaks belonging to the same backbone amide group. Each spectrum was acquired in 30 min in 20 mM MES buffer, pH 6.5, at 25 °C on a Bruker Avance III spectrometer at a  $^1\text{H}$  NMR frequency of 800 MHz.



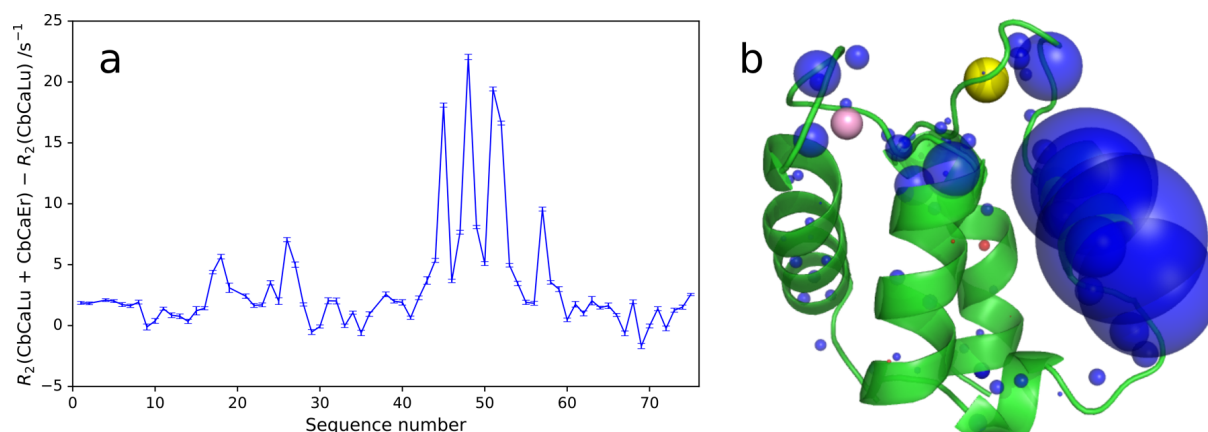
**Figure 3.**  $R_2(^1\text{H})$  relaxation data and corresponding distances derived using Curie-spin relaxation theory (eq 2) applied to CbCaEr and CbCaTb. The experimental data, recorded on an 800 MHz NMR spectrometer, are plotted versus expected values based on the crystal structure 4ICB.<sup>40</sup> (a)  $R_2(^1\text{H})$  PREs. Error bars reflect the uncertainties in experimental  $R_2$  values and represent 95% confidence intervals. (b) Distances from the paramagnetic center derived from the  $R_2(^1\text{H})$  PRE data versus the corresponding distances in the crystal structure. The asymmetric error bars were calculated by propagating  $R_2$  values through eq 2 and represent 95% confidence intervals.

Effects from RDCs can be minimized by measuring the relaxation rates using short relaxation delays, choosing a lanthanide ion with smaller  $\Delta\chi$  tensor ( $\text{Er}^{3+}$  rather than  $\text{Tb}^{3+}$ ), and measuring at a lower magnetic field strength. Effects from intermolecular PREs can be minimized by using dilute samples, but the data of Figure 3 were already recorded at a low protein concentration (0.2 mM). Clearly, the degree of acceptable dilution is fundamentally limited by the minimal sensitivity required for the NMR measurements. Ultimately, accurate measurements of small PREs requires accurate relaxation rate measurements of the paramagnetic as well as the diamagnetic state, which must take into account any nonspecific intermolecular PREs.

Accurate PRE measurements were obtained by mixing the diamagnetic reference sample (CbCaLu) with the paramagnetic species (CbCaEr) in a 1:1 ratio. The peaks of both species were well separated by PCSs. To circumvent the increased signal overlap in the  $^{15}\text{N}, ^1\text{H}$ -HSQC spectrum, we recorded the  $R_2(^1\text{H})$  relaxation rates of the amide protons in a pseudo-4D HNCO experiment, where the 3D HNCO pulse sequence was preceded by a spin-echo relaxation delay (Figure 7). The experiment simultaneously yielded the relaxation rates of the paramagnetic and diamagnetic species and enabled resolution of the PREs of 47 amide protons. To test the efficiency of the experiment at removing intermolecular PREs, the sample was prepared at a high total protein concentration (5 mM).



**Figure 4.**  $R_2(^1\text{H})$  relaxation data and corresponding distances derived using Curie-spin relaxation theory (eq 2) applied to a sample simultaneously containing paramagnetic CbCaEr and diamagnetic CbCaLu at 5 mM total protein concentration. The experimental PREs are plotted versus corresponding theoretical data derived from the crystal structure 4ICB.<sup>40</sup> (a)  $R_2(^1\text{H})$  PREs. Error bars reflect the uncertainties in experimental  $R_2$  values and represent 95% confidence intervals. (b) Distances from the paramagnetic center derived from the  $R_2(^1\text{H})$  PRE data versus the corresponding distances in the crystal structure. The asymmetric error bars were calculated by propagating  $R_2$  values through eq 2 and represent 95% confidence intervals.



**Figure 5.** Intermolecular PRE effects measured for CbCaLu at high concentration by the HNC0  $R_2(^1\text{H})$  relaxation experiment. (a) Amide proton  $R_2(^1\text{H})$  relaxation rates of a 2.5 mM solution of CbCaLu in a sample containing an equimolar concentration of CbCaEr minus the corresponding relaxation rates of an isolated 2.5 mM solution of CbCaLu, plotted versus residue number. Error bars were derived from random spectral noise only, disregarding baseline artifacts. (b) Crystal structure of calbindin D<sub>9k</sub> with the radii of amide protons plotted in proportion to the intermolecular PRE data shown in (a). Blue and red spheres indicate positive and negative values, respectively. The pink and yellow spheres indicate the  $\text{Ca}^{2+}$  and  $\text{Lu}^{3+}$  ions, respectively.

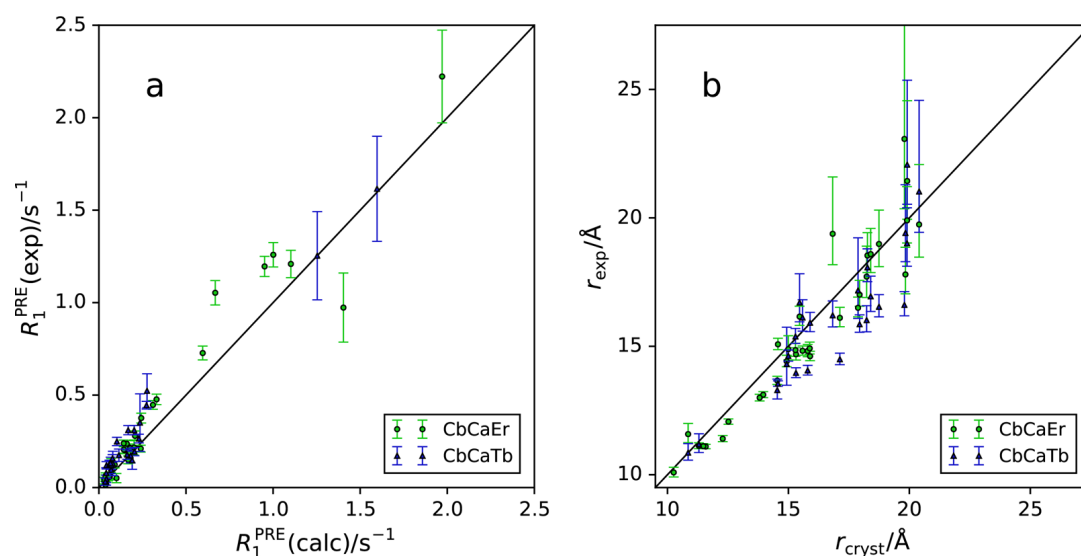
Conducted on a 600 MHz NMR spectrometer, the experiment used relaxation delays of 0, 5, and 10 ms and was acquired over 36 h, which was sufficient to result in very small error bars. The metal–nuclear distances calculated from the PREs were in excellent agreement with expectations from the crystal structure in a range of 11–24 Å with an RMSD of 0.9 Å (Figure 4). The chemical shifts of the diamagnetic and paramagnetic proteins did not change significantly between separate and mixed samples.

Intermolecular PRE effects were quantified by comparing  $R_2(^1\text{H})$  values obtained for CbCaLu in the mixture with CbCaEr (Figure 4) to those of an isolated 2.5 mM CbCaLu sample (Figure 5). In the presence of CbCaEr, the  $R_2(^1\text{H})$  values of CbCaLu increased on average by  $3.1 \text{ s}^{-1}$ . This small intermolecular effect would bias distance measurements beyond 20 Å from the paramagnetic center by at least 12%. The discrepancies in relaxation rates were particularly large for

solvent-exposed amides of residues 45–53 in the second-last helix. As this region is close to the lanthanide binding site, these intermolecular effects are small relative to the intramolecular PRE. Measured at 0.2 mM CbCaLu concentration, the intermolecular PREs decreased by <10-fold, to  $1.4 \text{ s}^{-1}$  on average. Notably, slightly different sample conditions may also contribute to different diamagnetic relaxation rates, as the mixed and isolated CbCaLu samples were prepared from different stock solutions. Mixed samples offer a convenient way of ensuring identical conditions for the paramagnetic and diamagnetic species.

Evolution of  $^1\text{H}$ – $^1\text{H}$  RDCs during the  $R_2(^1\text{H})$  relaxation delay modulates signal intensities by a product of cosine functions, and this is difficult to separate from PREs. To assess the influence of RDCs on PRE measurements, we also measured  $R_2(^1\text{H})$  relaxation rates of CbCaEr and CbCaTb on a 800 MHz NMR spectrometer using very short relaxation





**Figure 6.**  $R_1(^1\text{H})$  relaxation data and corresponding distances determined by Curie-spin and SBM relaxation theory (eqs 2 and 4) for CbCaTb and CbCaEr. The experimental data are plotted versus corresponding theoretical data derived from the crystal structure 4ICB.<sup>40</sup> Error bars were determined as described in Figure 3. (a)  $R_1(^1\text{H})$  PREs. (b) Corresponding distances.

delays (0 and 5 ms). As expected for greater magnetic alignment, the RMSD between distances derived from PREs and observed in the crystal structure increased to 1.7 Å for the CbCaEr sample, and the CbCaTb sample yielded an unacceptably poor correlation (Figure S4). The large  $\Delta\chi$  tensor associated with the  $\text{Tb}^{3+}$  ion easily produces  $^1\text{H}$ – $^1\text{H}$  RDCs of 15 Hz between backbone atoms, which results in 10% signal attenuation between 0 and 5 ms. As expected, sampling longer relaxation delays of 0, 5, 15, and 30 ms produced correspondingly larger signal attenuations, increasing the RMSD of distance measurements to 1.8 Å for CbCaEr (Figure S5).

Attempts to refocus the RDCs between amide protons and protons with chemical shifts outside the amide region were only partially successful, as  $^1\text{H}$ – $^1\text{H}$  RDCs between amide protons cannot be decoupled by a semiselective  $180^\circ$  refocusing pulse covering the range of amide protons. Furthermore, the ensuing refocusing of  $^3J(\text{H}^{\text{N}}, \text{H}^{\alpha})$  couplings requires that the same scheme is applied to the diamagnetic reference, but this results in inconsistencies for resonances shifted by PCs inside and outside of the excitation profile of the semiselective pulse. Tests with CbCaLn, where Ln = Ho, Dy, Tb, or Tm, measured with relaxation delays of 0, 5, 10, and 20 ms on a 600 MHz NMR spectrometer resulted in RMSDs of distance measurements of 1.6, 1.9, 3.3, and 3.6 Å, respectively (Figure S7).

We subsequently attempted to improve the correlations of Figure 4 by using more detailed relaxation theory. First, the Curie-spin relaxation theory can be refined by including the anisotropy of the magnetic susceptibility.<sup>42</sup> For CbCaEr, this changed the predicted  $R_2(^1\text{H})$  PREs by up to 6% compared with the isotropic theory, and the effect on the distance derived was at most 1%, resulting in no significant improvement of the overall correlation (data not shown). Second, cross-correlation between DSA and amide  $^1\text{H}$  CSA could affect the PREs.<sup>28,30</sup> Calculating the cross-correlation effect using the average tensor anisotropies determined for amides of ubiquitin<sup>43</sup> improved the RMSDs of the distance measurements by only 0.01 Å (data not shown). The improvement is insignificantly small because the amide  $^1\text{H}$  CSA tensor is small. Third, we considered the effect of rotational anisotropy on the relaxation rates. Modeling this effect for Curie-spin relaxation using the program Spinach<sup>44</sup>

again produced negligible corrections as calbindin D<sub>9k</sub> has a mostly isotropic rotational diffusion tensor with a  $D_{\parallel}/D_{\perp}$  ratio of 1.08.<sup>45</sup> Finally, the quality of the correlation for small PREs proved to be more sensitive to the baseline correction algorithm used during spectral processing than to any of the refinements in relaxation theory. Best results were obtained when no baseline correction was applied.

**$R_1(^1\text{H})$  Relaxation Measurements and Effects from Multiexponential Relaxation.**  $R_1(^1\text{H})$  PRE measurements are not affected by RDCs and may thus yield better distance measurements. To test this hypothesis, we measured  $R_1(^1\text{H})$  PREs in saturation-recovery experiments with CbCaEr and CbCaTb, using CbCaLu as the internal reference. In these measurements, the choice of relaxation delays proved to be critical, as residues in flexible parts of the protein (the terminal residues 1–5 and 74–75 as well as residues 18–19 and 40–44 in loop regions)<sup>46</sup> featured relatively fast diamagnetic  $R_1(^1\text{H})$  rates of the backbone amides. Consequently, these residues violated the initial rate condition even for relaxation delays as short as 100 ms and compromised fits by a monoexponential recovery curve (Figure S9), whereas good fits were obtained for the other residues in the protein, which feature more slowly relaxing amides. Ignoring the cross-peaks from these flexible regions and measuring paramagnetic and diamagnetic relaxation rates in separate samples by 2D [ $^{15}\text{N}$ ,  $^1\text{H}$ ]-HSQC-type spectra, the analysis of  $R_1(^1\text{H})$  rates yielded RMSDs of the distance measurements of 1.1 and 1.4 Å for CbCaEr and CbCaTb, respectively (Figure 6). While these RMSD values are competitive with those obtained from  $R_2(^1\text{H})$  rates, they exclude data from flexible residues; the PRE measurements took three times longer than the data of Figure 3, as an experiment with a very long recovery delay (15 s) had to be included to determine the equilibrium magnetization. The lesser sensitivity of the experiment is manifested by the large error bars.

While  $R_2(^1\text{H})$  PREs are overwhelmingly governed by Curie-spin relaxation, the SBM relaxation mechanism contributes significantly to  $R_1(^1\text{H})$  PREs.<sup>28</sup> To calculate the SBM contribution, we set the electronic relaxation time  $T_{1e}$  of the  $\text{Er}^{3+}$  and  $\text{Tb}^{3+}$  ions to 0.21 and 0.25 ps, respectively, as reported for the DTPA complex.<sup>47</sup> As the precise  $T_{1e}$  value is sensitive to

the ligand environment,<sup>48</sup> we also used  $T_{1e}$  values as fitting parameters in a best fit of the experimental  $R_1(^1\text{H})$  PRE values to the crystal structure. This resulted in  $T_{1e}$  values of 0.24 and 0.71 ps for CbCaEr and CbCaTb, respectively, and reduced the RMSDs of the distance measurements to 1.0 Å for both data sets. The small magnitude of this improvement indicates that useful results can still be obtained with approximate values of  $T_{1e}$ .

Finally, we tested the impact of intermolecular PREs on  $R_1(^1\text{H})$  measurements by rerecording the experiment for CbCaEr in the presence of the internal reference CbCaLu. For a fair comparison with previous  $R_2(^1\text{H})$  experiments, the spectrum was acquired over 24 h. The RMSD of the distance measurements was 1.4 Å. The lack of improvement in the predicted distances indicates that intermolecular PREs have less of an impact on  $R_1(^1\text{H})$  PRE measurements and that  $R_2(^1\text{H})$  experiments can give more precise distance measurements. In summary, the best distance measurements were obtained from  $R_2(^1\text{H})$  PREs measured with  $\text{Er}^{3+}$ , with the diamagnetic species provided as an internal reference.

## DISCUSSION

The present work introduces lanthanide ions as a tool for quantitative measurement of long-range distances in proteins by PREs. The results show that an  $\text{Er}^{3+}$  ion is outstandingly suited for distance measurements by PREs. Endowed with a large magnetic susceptibility that is not very anisotropic, it causes significant PREs without too large RDCs. At the same time, the PCSs induced are sufficiently large to separate the NMR signals of the paramagnetic protein from those of the diamagnetic species provided as an internal reference, allowing automatic subtraction of nonspecific intermolecular PREs, which is critically important for accurate measurements of small PREs.

**$R_2(^1\text{H})$  versus  $R_1(^1\text{H})$  PREs.** Our work highlights the importance of considering RDCs when measuring  $R_2(^1\text{H})$  PREs. Although  $R_1(^1\text{H})$  PREs were observed to be free of RDC and less sensitive toward intermolecular PRE effects, they are affected by multiexponential recovery caused by cross-relaxation.<sup>49</sup> This makes it difficult to find relaxation delays that are optimal for all residues and adds to the overall duration of the experiment. Nonetheless,  $R_1(^1\text{H})$  measurements are an option for PRE measurements using strongly anisotropic lanthanides like  $\text{Tb}^{3+}$  and  $\text{Yb}^{3+}$ . As opposed to  $R_2(^1\text{H})$  PREs, where SBM relaxation hardly contributes to the overall relaxation enhancement caused by lanthanide ions heavier than gadolinium,  $R_1(^1\text{H})$  PREs are to a large degree governed by SBM relaxation. This makes it difficult to predict  $R_1(^1\text{H})$  PREs from the protein structure, calling for calibration of the PREs against a known distance. In contrast, the  $R_2(^1\text{H})$  experiment does not require calibration as it is practically independent of the electronic relaxation time, giving it a distinct advantage over the  $R_1(^1\text{H})$  experiment.

We obtained the most accurate distances using a  $R_2(^1\text{H})$  pseudo-4D HNCU experiment with a CbCaEr sample containing CbCaLu as internal reference combined with short relaxation delays <10 ms and a relatively low magnetic field (600 MHz  $^1\text{H}$  NMR frequency) to minimize the effects from RDCs. This allowed excellent distance predictions in the range 11–24 Å for 47 amide resonances with an RMSD of 0.9 Å.

**Comparison with PREs from Transition Metals and Nitroxides.** It is interesting to compare these results to PRE measurements performed with the transition metal ions  $\text{Cu}^{2+}$

and  $\text{Mn}^{2+}$  or by commonly used nitroxide spin labels. In a previous study,  $R_1(^1\text{H})$  or  $R_2(^1\text{H})$  PREs generated by a  $\text{Mn}^{2+}$ -EDTA tag were shown to give good quality distances in the range between 16 and 23 Å.<sup>36</sup> That same study also reported good quality distances from the same EDTA tag loaded with  $\text{Cu}^{2+}$  in the range between 9 and 18 Å.  $\text{Cu}^{2+}$  tags have also successfully been used for distance measurements in the solid state in the range between 11 and 19 Å, with limited accuracy.<sup>50–53</sup> The data presented here allow more accurate distance measurements over a larger range than either of these transition metals. Furthermore, a high degree of accuracy is maintained for long distances, so that the useful range of distance measurements may extend beyond 23 Å in a larger protein complex. The PREs generated by  $\text{Cu}^{2+}$  and  $\text{Mn}^{2+}$  ions are governed by SBM relaxation (eq 5), which depends on the electronic relaxation time. The electronic relaxation time in turn depends on the ligand field and rotational correlation time of the metal–ligand complex, which can be different from that of the protein in the case of a flexible tether.<sup>48</sup> Therefore, conversion of PREs from  $\text{Cu}^{2+}$  and  $\text{Mn}^{2+}$  ions to distances usually requires calibration of the PREs against known distances in the protein fold.<sup>36</sup>

For the popular nitroxide tag MTSL, the intrinsic electron relaxation time is very long (~100 ns)<sup>54</sup> and the PRE is governed by the correlation time of the nitroxide moiety. Distances with an RMSD of ~2 Å have been determined in the range 15–24 Å from  $R_2(^1\text{H})$  PREs induced by MTSL tags.<sup>16</sup> For rigid tags, the rotational correlation time of the protein can be used for calculating the PRE<sup>55</sup> and can be accurately estimated from the molecular mass.<sup>56</sup> Intrinsic flexibility of the paramagnetic tag therefore requires calibration against known distances.<sup>57</sup> Curie-spin relaxation generated by lanthanides is also governed by the correlation time of the metal complex and therefore is most easily interpreted for rigid tags. The RMSD of the distance correlation of Figure 4 only increased from 0.9 to 1.0 Å when experimental distances were derived using an incorrect correlation time miset by  $\pm 20\%$ .

**Taking into Account RDCs.** The present work presents clear evidence that RDCs resulting from  $\chi$  tensor anisotropies lead to significant signal attenuation and artificially large apparent PREs in  $R_2(^1\text{H})$  measurements. To first approximation, the effects from RDCs can be ignored if the ratio  $\chi_{\text{iso}}^2/|\Delta\chi_{\text{ax}}|$  is large. For the  $\Delta\chi$  tensors reported for calbindin  $\text{D}_{9k}$ <sup>37</sup> and standard  $\chi_{\text{iso}}$  values,<sup>23</sup> this condition is fulfilled in the following descending order: Er, Ho, Dy, Tb, Tm, Nd, Eu, Yb, Pr, Ce, Sm. In agreement with this order and disregarding the lanthanide ions with very small  $\chi$  tensors (Nd, Eu, Yb, Pr, Ce, Sm), we obtained the most accurate distance measurements from  $R_2(^1\text{H})$  PREs measured with  $\text{Er}^{3+}$ . Significant contributions from RDCs thus explain the previously reported observation that  $R_2(^1\text{H})$  PREs measured for an immunoglobulin G binding protein tagged with  $\text{Dy}^{3+}$  were much larger than expected.<sup>27</sup>

Even with the use of  $\text{Er}^{3+}$ , RDCs still interfere significantly with distance measurements by  $R_2(^1\text{H})$  PREs if the relaxation delay is long and the magnetic field strength is high. The effect is particularly detrimental for long distances, for which the PREs are weak, whereas the paramagnetically induced RDCs are distance-independent. Attempts to refocus RDCs between backbone amide protons and  $\alpha$ -protons by an amide-selective REBURP pulse only yielded very minor improvements. Because the selectivity profile of the REBURP pulse will not refocus RDCs between amide protons and unexpected signal

attenuations arise from scalar  $H^{\alpha}-H^N$  couplings when PCSs bring the chemical shifts of  $\alpha$ -protons into the excitation range of the REBURP pulse, we found that use of a REBURP pulse was detrimental for some of the short-range distance measurements. For best overall reliability, we therefore propose to measure  $R_2(^1H)$  PREs using nonselective spin echoes and accept the larger relative uncertainties associated with long-range distances.

**Taking into Account Scalar Couplings.** For quantitative measurements of  $R_2(^1H)$  PREs, a two-time-point strategy has been recommended, where the cross-peak intensities of the diamagnetic and paramagnetic samples are measured for two different relaxation delays, which are optimally set to 0 and  $1.15/R_2$  seconds.<sup>58</sup> The method has the advantage of compensating for scalar couplings. In the present study, experimental relaxation rates ranged from 15 to 200  $s^{-1}$ , which made it difficult to choose the optimal second time-point. To assess a wide range of distances and also avoid the evolution of RDCs, best results were obtained with three short relaxation delays (0, 5, and 10 ms), from which PREs were determined by nonlinear regression with an equation that takes into account signal attenuation by scalar couplings (eqs 7 and 8).

**Suppression of Nonspecific Intermolecular PREs.** The interpretation of PRE data depends on separating paramagnetic from diamagnetic effects, as well as distinction from nonspecific PREs associated with measuring in a paramagnetic solution. When the cross-peaks of paramagnetic and diamagnetic species have the same chemical shifts, incomplete yields in the paramagnetic tagging reaction make it difficult to extract the pure paramagnetic relaxation rates. This is a regular problem for samples with isotropic  $\chi$  tensor (e.g., nitroxide radicals or samples with  $Mn^{2+}$  or  $Gd^{3+}$  ions). In this case it is also difficult to identify nonspecific intermolecular PREs, unless measurements performed at different concentrations yield unambiguously interpretable results. Both of these problems are circumvented, when the paramagnetic relaxation rates are determined from cross-peaks that are shifted by PCSs and diamagnetic relaxation rates are determined from cross-peaks of an internal reference. This new experimental approach greatly improves the agreement of PRE-derived distances with distances measured in the crystal structure. In the case of calbindin  $D_{9k}$ , intermolecular PREs were apparent even at concentrations as low as 0.2 mM. Furthermore, our scheme of providing the diamagnetic species as an internal reference is compatible with high sample concentrations, allowing drastically diminished acquisition times without compromising the accuracy of the measured PREs.

Intermolecular PRE effects were more pronounced in  $R_2(^1H)$  than in  $R_1(^1H)$  relaxation experiments.  $R_1$  Curie and SBM relaxation mechanisms depend on the spectral density at the Larmor frequency, while  $R_2$  is governed by the spectral density at zero frequency. In the case of calbindin  $D_{9k}$ , the dynamics of the intermolecular interactions may not occur on a time scale that matches the Larmor frequency of  $^1H$  spins, making the  $R_1(^1H)$  PRE less efficient.

To overcome the increased spectral overlap arising from the simultaneous presence of the peaks from paramagnetic and diamagnetic species, we propose to use a pseudo-4D HNCO experiment, where the cross-peaks are dispersed in a set of 3D NMR spectra recorded with different relaxation delays. The pseudo-4D HNCO experiment recorded with internal reference delivers metal–nuclear distances in the absence of any

structural information, in contrast to the evaluation of PCS data that require  $\Delta\chi$  tensor fits to known protein structures. In the case of calbindin  $D_{9k}$ , the experiment measured the complete set of PREs within 36 h. If additional resolution or shorter recording times are required, nonuniform sampling can be applied as shown previously.<sup>59</sup> The improved spectral resolution of 3D NMR spectra also benefits the performance of baseline correction algorithms, which matter when measuring small PREs.

**PREs from Lanthanide Tags.** Most proteins do not contain a metal binding site, and PRE measurements with a lanthanide ion require labeling with a suitable lanthanide tag.<sup>24</sup> The lanthanide ion in chemically synthesized tags tends to be at least as solvent-accessible as in calbindin  $D_{9k}$ , which highlights the importance of accounting for nonspecific intermolecular PREs as in the present work. If the lanthanide tag is flexible, the effective  $\Delta\chi$  tensor experienced by the protein is reduced, leading to smaller PCSs and RDCs.<sup>60</sup> The flexibility of the tag would thus facilitate the measurement of accurate PREs also with a  $Tb^{3+}$  ion. As the isotropic magnetic susceptibility remains unaffected by the mobility of the tag, long distances could be measured in this way, but the uncertainty of the metal position would also affect the quality of the distances measured.

## CONCLUSION

In summary, PREs measured with the use of lanthanide ions can deliver more accurate electron–nucleus distances over a greater range of distances than any previous approach that uses either transition metal ions or nitroxide tags. In particular, the possibility of providing the diamagnetic species as an internal reference renders the approach insensitive to incomplete tagging yields and nonspecific intermolecular interactions. With an estimate of the rotational correlation time of the protein,  $R_2(^1H)$  PREs from rigid lanthanide probes can be interpreted in terms of electron–nucleus distances without calibration, allowing absolute distance restraints to be derived for proteins of unknown structure. These advantages greatly enhance the attractiveness of already popular PRE measurements.

## EXPERIMENTAL SECTION

**Protein Samples.** Protein samples were produced following previously published protocols.<sup>30,61</sup> Details are reported in the Supporting Information.

**Predicted Relaxation Rates.** Theoretical Curie-spin relaxation rates were calculated using eqs 1 and 2, which assume an isotropic magnetic susceptibility (eq 3), and relaxation rates arising from SBM relaxation were calculated using eqs 4 and 5.<sup>31,32</sup>

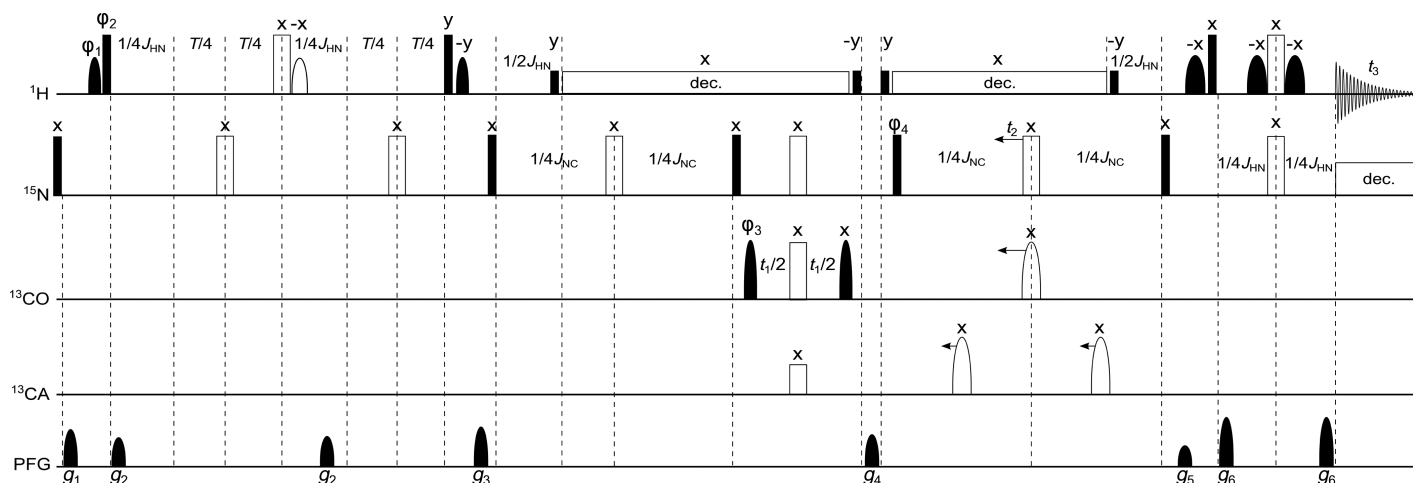
$$R_1^{\text{Curie}} = \frac{\omega_I^2}{45} \left( \frac{3}{4\pi} \frac{\chi_{\text{iso}}}{r^3} \right)^2 \left[ \frac{6\tau_r}{1 + \omega_I^2 \tau_r^2} \right] \quad (1)$$

$$R_2^{\text{Curie}} = \frac{\omega_I^2}{45} \left( \frac{3}{4\pi} \frac{\chi_{\text{iso}}}{r^3} \right)^2 \left[ 4\tau_r + \frac{3\tau_r}{1 + \omega_I^2 \tau_r^2} \right] \quad (2)$$

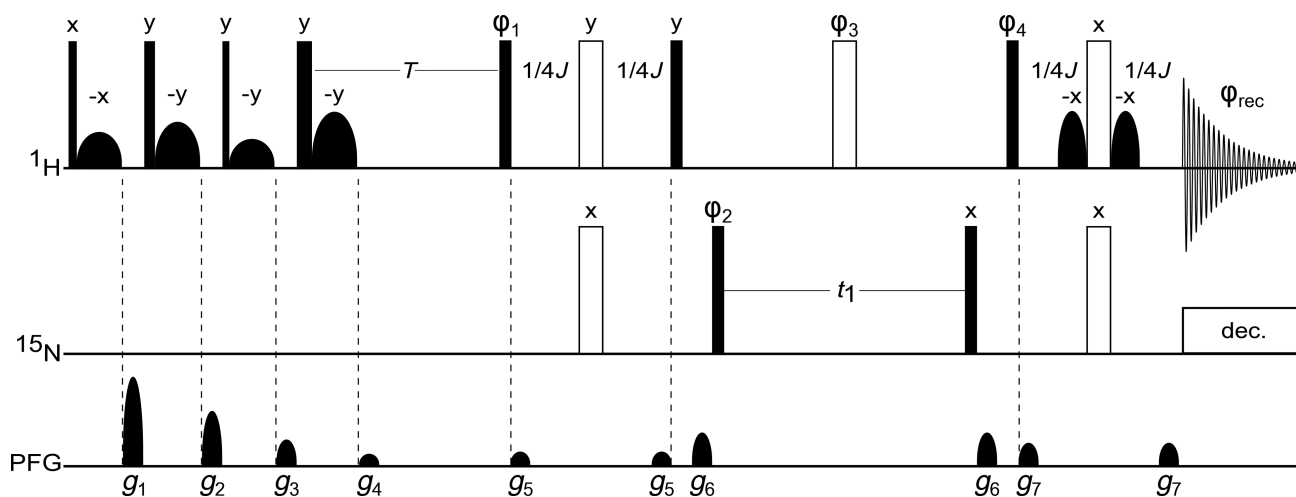
$$\chi_{\text{iso}} = \frac{\mu_0 \mu_B g^2 J(J+1)}{3kT} \quad (3)$$

$$R_1^{\text{SBM}} = \frac{2}{15} \left( \frac{\mu_0}{4\pi} \frac{\gamma_I \mu_B g \sqrt{J(J+1)}}{r^3} \right)^2 \left[ \frac{3\tau_c}{1 + \omega_I^2 \tau_c^2} + \frac{7\tau_c}{1 + \omega_S^2 \tau_c^2} \right] \quad (4)$$





**Figure 7.** HNCO pulse sequence for the measurement of  $R_2(^1\text{H})$  relaxation rates of amide protons with relaxation delay  $T$ . The  $^1\text{H}$ – $^{15}\text{N}$  INEPT delays were set to  $1/2J_{\text{HN}}$  assuming  $J_{\text{HN}} = 95$  Hz. The  $^{15}\text{N}$ – $^{13}\text{C}$  INEPT delays were set to  $1/2J_{\text{NC}}$  assuming  $J_{\text{NC}} = 83$  Hz. Open and filled shapes represent  $180^\circ$  and  $90^\circ$  pulses, respectively. All shaped pulses on the proton channel were applied on the water resonance as sinc-pulses of 1 ms duration. Low-power proton pulses were  $60\ \mu\text{s}$  in length, and Waltz decoupling was applied during  $^{13}\text{C}$  and  $^{15}\text{N}$  evolution periods.  $90^\circ$  shaped pulses on the carbon channel were Q5 pulses, and  $180^\circ$  pulses were Q3 pulses, with the exception of the rectangular  $180^\circ$  pulse applied on CA during the  $t_1$  evolution period, which was adjusted to  $58\ \mu\text{s}$  to achieve a  $360^\circ$  rotation of CO spins. Pulsed field gradients were applied for 0.6 ms with the following magnitudes:  $g_1 = 20$  G/cm;  $g_2 = 15$  G/cm;  $g_3 = 15$  G/cm;  $g_4 = 25$  G/cm;  $g_5 = 5$  G/cm;  $g_6 = 10$  G/cm. Phase cycle:  $\phi_1 = (-x, -x, -x, -x, x, x, x, x)$ ;  $\phi_2 = -\phi_1$ ;  $\phi_3 = (x, -x)_4$ ;  $\phi_4 = (x, x, -x, -x)_2$ ; receiver =  $(x, -x, -x, x, -x, x, x, -x)$ . Experiments of calbindin D<sub>9k</sub> samples were recorded with  $t_{1\text{max}} = 28$  ms,  $t_{2\text{max}} = 15$  ms, and  $t_{3\text{max}} = 100$  ms. A recycle delay of 1 s was used.



**Figure 8.**  $[^{15}\text{N}, ^1\text{H}]$ -HSQC pulse sequence preceded by a modified saturation–recovery sequence for  $R_1(^1\text{H})$  measurements. The relaxation delay  $T$  is varied to measure the longitudinal relaxation. See Figure 7 for a description of the pulse symbols and INEPT delays, except that the pulse lengths in the initial WET scheme followed the recommendation by Smallcombe et al.<sup>62</sup> Each selective sinc-shaped pulse of the WET scheme was of 3.5 ms duration, with the pulse amplitude adjusted to achieve water flip-back. The final Watergate scheme used selective sinc pulses of 1 ms duration. A recycle delay of 1 s was used. Pulsed field gradients were applied for 0.6 ms with the following magnitudes:  $g_1 = 40$  G/cm;  $g_2 = 20$  G/cm;  $g_3 = 10$  G/cm;  $g_4 = 5$  G/cm;  $g_5 = 7.5$  G/cm;  $g_6 = 17.5$  G/cm;  $g_7 = 12.5$  G/cm. Phase cycle:  $\phi_1 = x, x, x, x, -x, -x, -x, -x$ ;  $\phi_2 = (x, -x)_4$ ;  $\phi_3 = (y, y, -y, -y)_2$ ;  $\phi_4 = -\phi_1$ ; receiver =  $(x, -x, x, -x)_2$ .

$$R_2^{\text{SBM}} = \frac{1}{15} \left( \frac{\mu_0 \gamma_I \mu_B g \sqrt{J(J+1)}}{4\pi r^3} \right)^2 \times \left[ 4\tau_c + \frac{3\tau_c}{1 + \omega_I^2 \tau_c^2} + \frac{13\tau_c}{1 + \omega_S^2 \tau_c^2} \right] \quad (5)$$

$\mu_0$  is the magnetic permeability of free space,  $\mu_B$  is the Bohr magneton,  $g$  is the Landé  $g$  factor,  $J$  is the total angular momentum quantum number,  $k$  is the Boltzmann constant,  $T$  is the absolute temperature,  $\omega_I$  and  $\omega_S$  are the angular Larmor frequencies of the nuclear and electron spins, respectively,  $r$  is the electron–nucleus distance,  $\tau_c$  is the rotational correlation time of the molecule,  $\gamma_I$  is the gyromagnetic ratio of the nuclear spin, and  $\tau_c$  is the effective correlation time given by  $\tau_c^{-1} = (T_{1e}^{-1} + \tau_r^{-1})$ , where  $T_{1e}$  is the electronic longitudinal relaxation

time. The PRE predictions assumed a rotational correlation time of 4.25 ns as determined previously for calbindin D<sub>9k</sub> at 300 K.<sup>46</sup>

RDC values were predicted using eq 6,<sup>33</sup>

$$\text{RDC} = -\frac{1}{4\pi} \frac{B_0^2}{15kT} \frac{\gamma_A \gamma_B \hbar}{2\pi r_{\text{AB}}^3} \times \left[ \Delta\chi_{\text{ax}} (3 \cos^2 \theta - 1) + \frac{3}{2} \Delta\chi_{\text{rh}} \sin^2 \theta \cos(2\varphi) \right] \quad (6)$$

where  $r_{\text{AB}}$  is the internuclear distance between spins A and B with gyromagnetic ratios  $\gamma_A$  and  $\gamma_B$ , respectively,  $\Delta\chi_{\text{ax}}$  and  $\Delta\chi_{\text{rh}}$  denote the axial and rhombic components of the  $\Delta\chi$  tensor, and  $\theta$  and  $\varphi$  define the spherical polar angles of the internuclear vector  $\vec{r}_{\text{AB}}$  in the coordinate frame of the  $\Delta\chi$  tensor.

**$R_2(^1\text{H})$  Relaxation Measurements.**  $R_2(^1\text{H})$  relaxation experiments were performed using a pseudo-3D [ $^{15}\text{N}$ ,  $^1\text{H}$ ]-HSQC (Figure S1, modified from ref 58) and a novel pseudo-4D HNCQ experiment (Figure 7) recorded with relaxation delay  $T$ . Both experiments used the relaxation block described by Iwahara et al.<sup>58</sup> except that two water-selective pulses were included to keep the water magnetization longitudinal during the entire relaxation delay.

$R_2(^1\text{H})$  PREs were measured by monitoring peak heights and using a single exponential decay model (see eq 7). In addition to the  $R_2$  value, the scaling parameter  $a$  was fitted by nonlinear regression by minimizing eq 8, where the index of the sum refers to each relaxation delay  $T_i$ .

$$I^{\text{para}}(T) = aI^{\text{dia}}(T) \exp(-R_2^{\text{PRE}}T) \quad (7)$$

$$f(a, R_2^{\text{PRE}}) = \sum_i (aI^{\text{dia}}(T_i) \exp(-R_2^{\text{PRE}}T_i) - I^{\text{para}}(T_i))^2 \quad (8)$$

This method of fitting  $R_2$  PREs was applied to the pairs of diamagnetic and paramagnetic peak intensities of each amide. It compensates for common sources of signal attenuation such as scalar couplings.

**$R_1(^1\text{H})$  Relaxation Measurements.**  $R_1(^1\text{H})$  relaxation measurements were acquired using a saturation–recovery pseudo-3D [ $^{15}\text{N}$ ,  $^1\text{H}$ ]-HSQC experiment (Figure 8). Complete saturation of solvent magnetization proved difficult due to limitations in permissible pulse power and pulse duration on the cryoprobe of our NMR spectrometer. Specifically, the commonly used WET sequence<sup>62</sup> resulted in sporadic signal overload due to accidental refocusing of water magnetization between subsequent scans. Therefore, we opted to preserve the water magnetization by use of a modified WET scheme that included flip-back pulses to keep the water magnetization aligned with the magnetic field. The WET sequence thus saturated all protein magnetization except for the water magnetization and protein resonances at the chemical shift of the water, which acquired at least some degree of saturation by spin diffusion within the protein and transverse relaxation during the relatively long water-selective flip-back pulses. The modified WET scheme achieved 99% water flip-back, and no amide cross-peaks were observed for  $T = 0$  s.

To minimize artifacts due to cross-relaxation, the initial rate of magnetization recovery was captured by five linearly spaced recovery delays  $T = 20, 40, 60, 80$ , and  $100$  ms. In addition, a single point was sampled at  $T = 15$  s to record a time point, where the magnetization had fully recovered. The experiments used  $t_{1\text{max}} = 75$  ms and  $t_{2\text{max}} = 100$  ms.  $R_1(^1\text{H})$  PREs were measured by fitting peak intensities to the saturation–recovery model with two parameters (relaxation rate and peak amplitude following complete relaxation).

## ■ ASSOCIATED CONTENT

### Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/jacs.8b03858.

Sample preparation, NMR measurements, resonance assignments, relaxation and PCS data, and pulse programs (PDF)

## ■ AUTHOR INFORMATION

### Corresponding Author

\*gottfried.otting@anu.edu.au

### ORCID

Gottfried Otting: 0000-0002-0563-0146

### Notes

The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENTS

H.W.O. thanks the Westpac Bicentennial foundation for a Future Leaders Scholarship. Financial support by the Australian

Research Council, including a Laureate Fellowship to G.O., is gratefully acknowledged.

## ■ REFERENCES

- (1) Bertini, I.; Couture, M. M. J.; Donaire, A.; Eltis, L. D.; Felli, I. C.; Luchinat, C.; Piccioli, M.; Rosato, A. *Eur. J. Biochem.* **1996**, *241*, 440–452.
- (2) Bertini, I.; Donaire, A.; Luchinat, C.; Rosato, A. *Proteins: Struct., Funct., Genet.* **1997**, *29*, 348–358.
- (3) Battiste, J. L.; Wagner, G. *Biochemistry* **2000**, *39*, 5355–5365.
- (4) Bertini, I.; Felli, I. C.; Luchinat, C.; Parigi, G.; Pierattelli, R. *ChemBioChem* **2007**, *8*, 1422–1429.
- (5) Clore, G. M.; Iwahara, J. *Chem. Rev.* **2009**, *109*, 4108–4139.
- (6) Kim, H. J.; Howell, S. C.; Van Horn, W. D.; Jeon, Y. H.; Sanders, C. R. *Prog. Nucl. Magn. Reson. Spectrosc.* **2009**, *55*, 335–360.
- (7) Otting, G. *Annu. Rev. Biophys.* **2010**, *39*, 387–405.
- (8) Keizers, P. H. J.; Ubbink, M. *Prog. Nucl. Magn. Reson. Spectrosc.* **2011**, *58*, 88–96.
- (9) Kleckner, I. R.; Foster, M. P. *Biochim. Biophys. Acta, Proteins Proteomics* **2011**, *1814*, 942–968.
- (10) Bertini, I.; Luchinat, C.; Nagulapalli, M.; Parigi, G.; Ravera, E. *Phys. Chem. Chem. Phys.* **2012**, *14*, 9149.
- (11) Göbl, C.; Madl, T.; Simon, B.; Sattler, M. *Prog. Nucl. Magn. Reson. Spectrosc.* **2014**, *80*, 26–63.
- (12) Jensen, M. R.; Zweckstetter, M.; Huang, J.; Blackledge, M. *Chem. Rev.* **2014**, *114*, 6632–6660.
- (13) Liu, W.-M.; Overhand, M.; Ubbink, M. *Coord. Chem. Rev.* **2014**, *273–274*, 2–12.
- (14) Anthis, N. J.; Clore, G. M. *Q. Rev. Biophys.* **2015**, *48*, 35–116.
- (15) Andralojć, W.; Hiruma, Y.; Liu, W.-M.; Ravera, E.; Nojiri, M.; Parigi, G.; Luchinat, C.; Ubbink, M. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114*, 1840–1847.
- (16) Liang, B.; Bushweller, J. H.; Tamm, L. K. *J. Am. Chem. Soc.* **2006**, *128*, 4389–4397.
- (17) Matei, E.; Gronenborn, A. M. *Angew. Chem.* **2016**, *128*, 158–162.
- (18) Helmling, C.; Bessi, I.; Wacker, A.; Schnorr, K. A.; Jonker, H. R. A.; Richter, C.; Wagner, D.; Kreibich, M.; Schwalbe, H. *ACS Chem. Biol.* **2014**, *9*, 1330–1339.
- (19) Schnorr, K. A.; Gophane, D. B.; Helmling, C.; Cetiner, E.; Pasemann, K.; Fürtig, B.; Wacker, A.; Qureshi, N. S.; Gränz, M.; Barthelme, D.; Jonker, H. R. A.; Stirmal, E.; Sigurdsson, S. Th.; Schwalbe, H. *J. Biomol. NMR* **2017**, *68*, 53–63.
- (20) Wunderlich, C. H.; Huber, R. G.; Spitzer, R.; Liedl, K. R.; Klobner, K.; Kreutz, C. *ACS Chem. Biol.* **2013**, *8*, 2697–2706.
- (21) Ma, L.; Jørgensen, A.-M. M.; Sørensen, G. O.; Ulstrup, J.; Led, J. *J. Am. Chem. Soc.* **2000**, *122*, 9473–9485.
- (22) Rovó, P.; Grohe, K.; Giller, K.; Becker, S.; Linser, R. *ChemPhysChem* **2015**, *16*, 3791–3796.
- (23) Bleaney, B. J. *Magn. Reson.* **1972**, *8*, 91–100.
- (24) Nitsche, C.; Otting, G. *Prog. Nucl. Magn. Reson. Spectrosc.* **2017**, *98–99*, 20–49.
- (25) Lee, L.; Sykes, B. D. *Biochemistry* **1980**, *19*, 3208–3214.
- (26) Bertini, I.; Donaire, A.; Jiménez, B.; Luchinat, C.; Parigi, G.; Piccioli, M.; Poggi, L. *J. Biomol. NMR* **2001**, *21*, 85–98.
- (27) Barb, A. W.; Ho, T. G.; Flanagan-Steet, H.; Prestegard, J. H. *Protein Sci.* **2012**, *21*, 1456–1466.
- (28) Pintacuda, G.; Kaikkonen, A.; Otting, G. *J. Magn. Reson.* **2004**, *171*, 233–243.
- (29) We note a mistake in Table 2 of ref 28, where the values reported for a rotational correlation time of 1 ns were in fact calculated for a rotational correlation time of 10 ns.
- (30) Orton, H. W.; Kuprov, I.; Loh, C.-T.; Otting, G. *J. Phys. Chem. Lett.* **2016**, *7*, 4815–4818.
- (31) (a) Solomon, I. *Phys. Rev.* **1955**, *99*, 559–565. (b) Bloembergen, N.; Morgan, L. O. *Chem. Phys.* **1961**, *34*, 842–850.
- (32) Guéron, M. *J. Magn. Reson.* **1975**, *19*, 58–66.
- (33) Bertini, I.; Luchinat, C.; Parigi, G. *Prog. Nucl. Magn. Reson. Spectrosc.* **2002**, *40*, 249–273.

- (34) Tolman, J. R.; Flanagan, J. M.; Kennedy, M. A.; Prestegard, J. H. *Proc. Natl. Acad. Sci. U. S. A.* **1995**, *92*, 9279–9283.
- (35) Donaldson, L. W.; Skrynnikov, N. R.; Choy, W.-Y.; Muhandiram, D. R.; Sarkar, B.; Forman-Kay, J. D.; Kay, L. E. *J. Am. Chem. Soc.* **2001**, *123*, 9843–9847.
- (36) Pintacuda, G.; Moshref, A.; Leonchiks, A.; Sharipo, A.; Otting, G. *J. Biomol. NMR* **2004**, *29*, 351–361.
- (37) Bertini, I.; Janik, M. B. L.; Lee, Y.-M.; Luchinat, C.; Rosato, A. *J. Am. Chem. Soc.* **2001**, *123*, 4181–4188.
- (38) Skelton, N. J.; Kördel, J.; Chazin, W. J. *J. Mol. Biol.* **1995**, *249*, 441–462.
- (39) Johansson, C.; Brodin, P.; Grundstrom, T.; Thulin, E.; Forsen, S.; Drakenberg, T. *Eur. J. Biochem.* **1990**, *187*, 455–460.
- (40) Svensson, L. A.; Thulin, E.; Forsén, S. *J. Mol. Biol.* **1992**, *223*, 601–606.
- (41) Kesvatera, T.; Jönsson, B.; Telling, A.; Töugu, V.; Vija, H.; Thulin, E.; Linse, S. *Biochemistry* **2001**, *40*, 15334–15340.
- (42) Vega, A. J.; Fiat, D. *Mol. Phys.* **1976**, *31*, 347–355.
- (43) Yao, L.; Grishaev, A.; Cornilescu, G.; Bax, A. *J. Am. Chem. Soc.* **2010**, *132*, 10866–10875.
- (44) Hogben, H. J.; Krzystyniak, M.; Charnock, G. T. P.; Hore, P. J.; Kuprov, I. *J. Magn. Reson.* **2011**, *208*, 179–194.
- (45) Lee, L. K.; Rance, M.; Chazin, W. J.; Palmer, A. G., III. *J. Biomol. NMR* **1997**, *9*, 287–298.
- (46) Koedel, J.; Skelton, N. J.; Akke, M.; Palmer, A. G.; Chazin, W. J. *Biochemistry* **1992**, *31*, 4856–4866.
- (47) Alsaadi, B. M.; Rossotti, F. J. C.; Williams, R. J. P. *J. Chem. Soc., Dalton Trans.* **1980**, *11*, 2151–2154.
- (48) Bertini, I.; Luchinat, C.; Parigi, G. *Solution NMR of paramagnetic molecules: applications to metallobiomolecules and models*; Elsevier: 1996; Vol. 2.
- (49) Kalk, A.; Berendsen, H. J. C. *J. Magn. Reson.* **1976**, *24*, 343–366.
- (50) Mukhopadhyay, D.; Nadaud, P. S.; Shannon, M. D.; Jaroniec, C. P. *J. Phys. Chem. Lett.* **2017**, *8*, 5871–5877.
- (51) Nadaud, P. S.; Helmus, J. J.; Sengupta, I.; Jaroniec, C. P. *J. Am. Chem. Soc.* **2010**, *132*, 9561–9563.
- (52) Nadaud, P. S.; Helmus, J. J.; Kall, S. L.; Jaroniec, C. P. *J. Am. Chem. Soc.* **2009**, *131*, 8108–8120.
- (53) Sengupta, I.; Gao, M.; Arachchige, R. J.; Nadaud, P. S.; Cunningham, T. F.; Saxena, S.; Schwieters, C. D.; Jaroniec, C. P. *J. Biomol. NMR* **2015**, *61*, 1–6.
- (54) Kosen, P. A. *Methods Enzymol.* **1989**, *177*, 86–121.
- (55) Fawzi, N. L.; Fleissner, M. R.; Anthis, N. J.; Kálai, T.; Hideg, K.; Hubbell, W. L.; Clore, G. M. *J. Biomol. NMR* **2011**, *51*, 105–114.
- (56) Ryabov, Y. E.; Geraghty, C.; Varshney, A.; Fushman, D. J. *J. Am. Chem. Soc.* **2006**, *128*, 15432–15444.
- (57) Iwahara, J.; Schwieters, C. D.; Clore, G. M. *J. Am. Chem. Soc.* **2004**, *126*, 5879–5896.
- (58) Iwahara, J.; Tang, C.; Marius Clore, G. *J. Magn. Reson.* **2007**, *184*, 185–195.
- (59) Stetz, M. A.; Wand, A. J. *J. Biomol. NMR* **2016**, *65*, 157–170.
- (60) Shishmarev, D.; Otting, G. *J. Biomol. NMR* **2013**, *56*, 203–216.
- (61) Hitchman, A. J. W.; Kerr, M.-K.; Harrison, J. E. *Arch. Biochem. Biophys.* **1973**, *155*, 221–222.
- (62) Smallcombe, S. H.; Patt, S. L.; Keifer, P. A. *J. Magn. Reson., Ser. A* **1995**, *117*, 295–303.

## Supporting Information

Accurate electron–nucleus distances from paramagnetic relaxation enhancements

Henry W. Orton, Gottfried Otting

### Sample preparation

#### *Cloning*

Mutations at two different sites were introduced into wild-type calbindin D<sub>9k</sub> to simplify the NMR spectra and guard against protein degradation. The mutation Pro43Gly removes the *cis-trans* isomerization equilibrium otherwise present in the protein and the mutation Asn56Asp was made to prevent any backbone rearrangement.<sup>1–3</sup> The gene of the double mutant P43G/N56D of calbindin D<sub>9k</sub> was cloned into the vector pETMCSI.<sup>4</sup> The mutant P43M has also been reported to suppress *cis-trans* isomerization<sup>5</sup> and we used the double mutant P43M/N56D in earlier experiments that explored the use of RE-BURP pulses to refocus RDCs. These samples were prepared with identical expression, purification, and sample preparation protocols.

#### *Calbindin D<sub>9k</sub> expression and purification*

A 15 mL overnight starter culture of transformed *E. coli* BL21(DE3) cells was used to inoculate 1.0 L of ampicillin supplemented LB-media and incubated at 37 °C with shaking. An OD<sub>600</sub> of 0.6 was achieved after 4 hours, after which the cells were pelleted and resuspended in 0.5 L of M9 minimal media supplemented with 0.6 g/L <sup>15</sup>N ammonium chloride or 6.0 g/L <sup>13</sup>C glucose. Following incubation at 37 °C for 1 hour, IPTG was added to 1 mM and incubation at 25 °C continued overnight. Cells were pelleted and resuspended in 50 mM HEPES (pH 7.5), 20 mM NaCl.

The purification of the calbindin D<sub>9k</sub> samples followed an established protocol<sup>6</sup> with modifications. Cells were lysed by a French Press, then heat-treated in an 80 °C water bath for 8 minutes. The lysate was then clarified by centrifugation at 30000 g for 1 hour. EDTA (pH 8.0) was added to the supernatant to 5 mM to increase affinity to a DEAE column. The samples were purified by a DEAE column using 50 mM Tris (pH 7.5) as running buffer, using a gradient from 0 to 0.5 M NaCl. Calbindin D<sub>9k</sub> eluted at about 150 mM NaCl. Following concentration to 2.5 mL, calcium chloride (15 mM) was added, then the sample was loaded onto an S75 size exclusion column, and eluted with 50 mM Tris (pH 7.5), 400 mM NaCl to give 50 mg of purified uniformly <sup>15</sup>N-labeled product.

### *Lanthanide substitution for NMR sample preparation*

10 equivalents of lanthanide salt were added to an 0.2 mM stock solution of calbindin D<sub>9k</sub>. This was dialysed overnight with 20 mM MES (pH 6.5), 400 mM NaCl. The dialysis buffer was then changed to 20 mM MES (pH 6.5) and dialysed overnight. The protein solution was concentrated to the desired concentration and 50  $\mu$ L deuterium oxide were added to make a 0.5 mL sample for NMR.

### **NMR measurements**

#### *PCS measurement and $\Delta\chi$ tensor fitting*

NMR spectra were recorded of protein solutions in 20 mM MES buffer, pH 6.5, at 25 °C. The assignment of the amide resonances was confirmed for uniformly <sup>15</sup>N-labeled diamagnetic CbCaLu using 3D NOESY-<sup>15</sup>N-HSQC and HNCO spectra by reference to the BMRB entry 16340.<sup>7</sup> PCSs in paramagnetic samples were measured in [<sup>15</sup>N,<sup>1</sup>H]-HSQC spectra using CbCaLu or CbCaY as the diamagnetic reference. (Spectra of CbCaLu or CbCaY were indistinguishable for amide protons not in the immediate vicinity of the lanthanide binding site.) The PCSs of the amide protons together with the crystal structure (PDB ID 4ICB<sup>2</sup>) were used to fit  $\Delta\chi$  tensors using the software Numbat.<sup>8</sup>

**Table S1.**  $\chi$  tensor parameters for calbindin D<sub>9k</sub> P43G/N56D<sup>a</sup>

Sample	$\chi_{\text{iso}}^{\text{b}}$	$\Delta\chi_{\text{ax}}^{\text{b}}$	$\Delta\chi_{\text{rh}}^{\text{b}}$	$\alpha/^{\circ}$	$\beta/^{\circ}$	$\gamma/^{\circ}$	$Q\text{-factor}^{\text{c}}$
CbCaTb	82.7	29.7	12.2	152	153	71	0.08
CbCaEr	80.4	-8.1	-4.9	126	142	42	0.05

<sup>a</sup> The isotropic magnitudes of the  $\chi$  tensors were calculated using Eq. 3 with parameters from Bleaney<sup>9</sup> at 298 K. The anisotropy components were experimentally determined from PCSs of backbone amide protons by fitting  $\Delta\chi$  tensors to the crystal structure 4ICB<sup>2</sup> while constraining the position of the lanthanide ion to the coordinates of the Ca<sup>2+</sup> ion in site-II.

<sup>b</sup> In units of 10<sup>-32</sup> m<sup>3</sup>.

<sup>c</sup> Quality factors were calculated as the root-mean-square deviation between experimental and back-calculated PCSs divided by the root-mean-square of the experimental PCSs.



### *Additional $\Delta\chi$ tensor determinations*

$\Delta\chi$  tensors were also determined for the P43M/N56D mutant of calbindin D<sub>9k</sub> loaded with Tb<sup>3+</sup>, Dy<sup>3+</sup>, Ho<sup>3+</sup>, Er<sup>3+</sup>, or Tm<sup>3+</sup>, and PCSs of backbone amides could be measured for 41, 40, 48, 50, and 53 residues, respectively. The resonance assignment was verified by a 3D NOESY-<sup>15</sup>N-HSQC spectrum of CbCaY and this sample was also used as the diamagnetic reference for PCS calculations. The fitted  $\Delta\chi$  tensors are reported in Table S2.

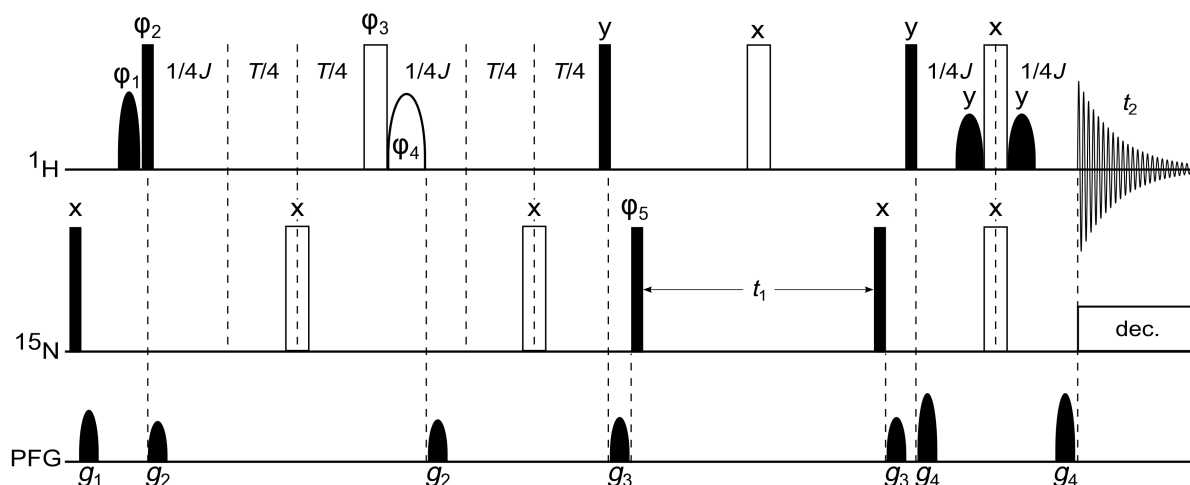
Where both mutants had been measured with the same lanthanide ion (Tb<sup>3+</sup> or Er<sup>3+</sup>), the  $\Delta\chi$  tensors were practically indistinguishable, indicating conservation of the 3D structure and paramagnetic properties (Tables S1 and S2).

**Table S2.**  $\chi$  tensor parameters for the calbindin D<sub>9k</sub> mutants P43M/N56D<sup>a</sup>

Sample	$\chi_{\text{iso}}$	$\Delta\chi_{\text{ax}}$	$\Delta\chi_{\text{rh}}$	$\alpha/^{\circ}$	$\beta/^{\circ}$	$\gamma/^{\circ}$	$Q$ -factor
CbCaTb	82.7	29.7	12.5	155	152	75	0.08
CbCaDy	99.2	22.4	8.0	148	150	106	0.06
CbCaHo	98.5	14.7	4.7	138	143	98	0.04
CbCaEr	80.4	-8.2	-4.7	126	142	42	0.04
CbCaTm	50.4	-20.4	-9.2	125	140	65	0.05

<sup>a</sup> Tensor values reported in units of  $10^{-32} \text{ m}^3$ . See footnotes of Table S1 for further details.

### Pseudo-3D $R_2(^1\text{H})$ relaxation pulse sequence



**Figure S1.** [ $^{15}\text{N}$ ,  $^1\text{H}$ ]-HSQC-type pulse sequence for the measurement of  $R_2(^1\text{H})$  relaxation rates of amide protons. The sequence is that of Iwahara et al. (ref. 10) supplemented with water-selective pulses to retain the water magnetisation as longitudinal magnetisation during the relaxation delay  $T$ . The INEPT delay was set to  $1/2J_{\text{HN}}$  assuming  $J_{\text{HN}} = 95$  Hz. Open and filled shapes represent  $180^\circ$  and  $90^\circ$  pulses, respectively. All shaped pulses were applied on the water resonance as sinc-pulses of 1 ms duration. Pulsed field gradients were applied for 0.6 ms with the following magnitudes:  $g_1 = 20$  G/cm;  $g_2 = 15$  G/cm;  $g_3 = 15$  G/cm;  $g_4 = 25$  G/cm. Phase cycle:  $\phi_1 = (-x, -x, x, x)_2$ ;  $\phi_2 = -\phi_1$ ;  $\phi_3 = y, y, y, y, -y, -y, -y, -y$ ;  $\phi_4 = -\phi_3$ ;  $\phi_5 = (x, -x)_4$ ; receiver =  $(x, -x, -x, x)_2$ . Experiments were conducted with a recycle delay of 1 s.

### Control experiments

A large number of control experiments were conducted to establish the best experiments for PRE measurements and test the influence of RDCs and non-specific intermolecular PREs. Table S3 provides a summary. The designs and results of individual experiments are described below. The success of different experimental schemes was judged by the RMSD between PRE-derived distances and distances measured in the crystal structure 4ICB. The  $R_2(^1\text{H})$  experiments A, F and G are discussed in the main text; experiment A provided the best correlations of distances derived from PREs versus distances measured in the crystal structure.  $R_1(^1\text{H})$  experiments L and M are also discussed in the main text; they were conducted to compare the performance of  $R_1(^1\text{H})$  PREs with  $R_2(^1\text{H})$  PREs. All other control experiments are discussed below.

**Table S3.** Summary of relaxation experiments<sup>a</sup>

Label	Sample <sup>b</sup>	Conc. (mM) <sup>c</sup>	Relaxation delays (ms)	Experiment <sup>d</sup>	Total time (hrs)	<sup>1</sup> H NMR frequency (MHz)	No. peaks	RMSD (Å)
A	Cb(NC)CaEr+Lu	5.0	0, 5, 10	<i>R</i> <sub>2</sub> HNCO	36	600	47	0.9
B	Cb(N)CaEr+Lu	0.4	0, 5, 10	<i>R</i> <sub>2</sub> HSQC	24	600	37	0.9
C	Cb(N)CaEr,Lu	0.2	0, 5, 10	<i>R</i> <sub>2</sub> HSQC	24	600	45	1.5
D	Cb(N)CaEr,Lu	0.2	0, 5	<i>R</i> <sub>2</sub> HSQC	24	800	43	1.7
E	Cb(N)CaTb,Lu	0.2	0, 5	<i>R</i> <sub>2</sub> HSQC	24	800	33	4.8
F	Cb(N)CaEr,Lu	0.2	0, 5, 15, 30	<i>R</i> <sub>2</sub> HSQC	24	800	43	1.8
G	Cb(N)CaTb,Lu	0.2	0, 5, 15, 30	<i>R</i> <sub>2</sub> HSQC	24	800	33	5.2
H	Cb(N)CaHo,Y	0.2	0, 5, 10, 20	<i>R</i> <sub>2</sub> HSQC-reb	24	600	48	1.6
I	Cb(N)CaDy,Y	0.2	0, 5, 10, 20	<i>R</i> <sub>2</sub> HSQC-reb	24	600	40	1.9
J	Cb(N)CaTb,Y	0.2	0, 5, 10, 20	<i>R</i> <sub>2</sub> HSQC-reb	24	600	41	3.3
K	Cb(N)CaTm,Y	0.2	0, 5, 10, 20	<i>R</i> <sub>2</sub> HSQC-reb	24	600	53	3.6
L	Cb(N)CaTb,Lu	0.2	[20-100] <sup>e</sup>	<i>R</i> <sub>1</sub> HSQC-sat	72	800	25	1.4
M	Cb(N)CaEr,Lu	0.2	[20-100] <sup>e</sup>	<i>R</i> <sub>1</sub> HSQC-sat	72	800	33	1.1
N	Cb(N)CaEr+Lu	0.4	[20-100] <sup>e</sup>	<i>R</i> <sub>1</sub> HSQC-sat	24	600	29	1.4

<sup>a</sup> Parameters explored by specific experiments are highlighted in red.

<sup>b</sup> Sample names encode the isotope labeling, lanthanoid ion, and diamagnetic reference. For example, Cb(NC)CaEr,Lu refers to uniformly <sup>15</sup>N/<sup>13</sup>C labeled calbindin D<sub>9k</sub> loaded with calcium in site-I and erbium in site-II of the paramagnetic protein; the diamagnetic relaxation rates were measured separately using lutetium in the diamagnetic reference. Cb(N)CaEr+Lu refers to <sup>15</sup>N labeled protein and loading with erbium and lutetium in 1:1 ratio to provide an internal diamagnetic reference.

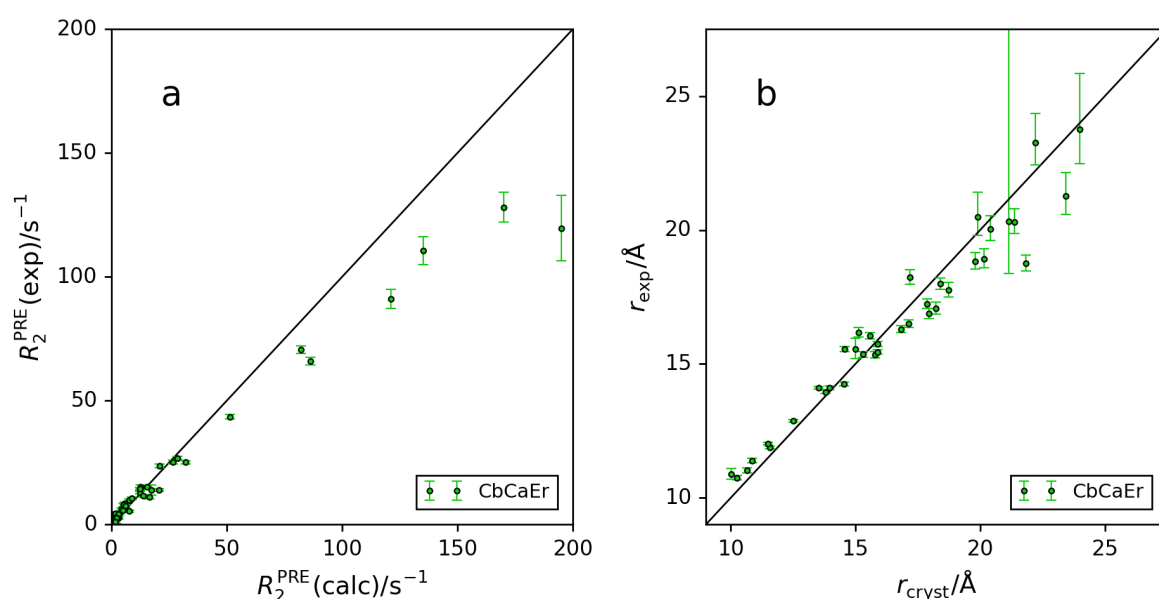
<sup>c</sup> For separated diamagnetic and paramagnetic samples, this is the concentration of each individual sample. For samples with an internal diamagnetic reference, this is the total protein concentration.

<sup>d</sup> *R*<sub>2</sub> HNCO refers to the pseudo-4D HNCO relaxation experiment. *R*<sub>2</sub> HSQC represents the pseudo-3D [<sup>15</sup>N,<sup>1</sup>H]-HSQC experiment. *R*<sub>2</sub> HSQC-reb is the same experiment but with a REBURP pulse applied to the amide protons to refocus RDCs. *R*<sub>1</sub> HSQC-sat is the saturation-recovery experiment.

<sup>e</sup> Six linearly spaced relaxation delays between 20 ms and 100 ms, and an additional delay at 15 s for complete recovery.

*Control experiment B: PREs measured with an internal diamagnetic reference are independent of sample concentration*

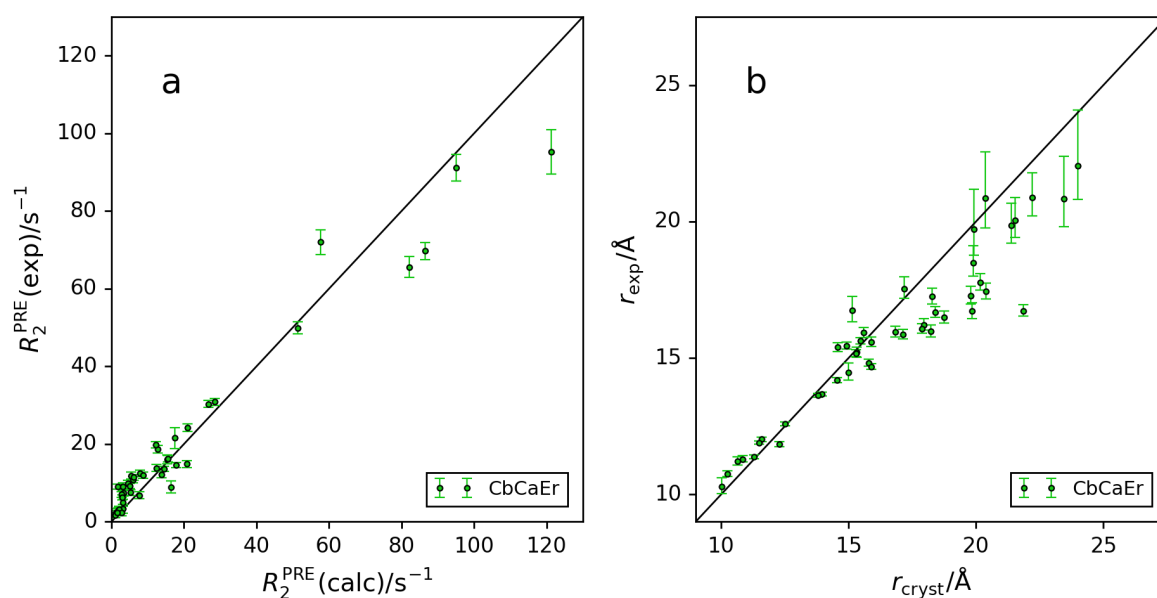
Experiment B was recorded with a total protein concentration of 0.4 mM in contrast to experiment A, which was performed with 5 mM protein concentration. The distance correlation plot yielded the same RMSD of 0.9 Å for the 37 non-overlapping amide cross-peaks that could be resolved in the pseudo-3D  $R_2(^1\text{H})$  experiment (Figure S2). 10 additional amide cross-peaks could be evaluated in the pseudo-4D experiment A.



**Figure S2.**  $R_2(^1\text{H})$  PRE data for mixture of CbCaEr and CbCaLu with corresponding distances. 0.4 mM total protein concentration. Acquired at 600 MHz sampling relaxation delays 0, 5 and 10 ms. In this figure as in all following figures, the experimental data are plotted versus the corresponding calculated data derived from the crystal structure 4ICB.<sup>2</sup> Experimental PREs were converted into metal-nuclear distances using the isotropic Curie-spin relaxation theory only (Eq. 2 of the main text). Error bars reflect the uncertainties in experimental  $R_2$  values and represent 95% confidence intervals. The asymmetric error bars for distances were obtained by propagating  $R_2$  values through the relevant PRE theory and represent 95% confidence intervals. (a)  $R_2(^1\text{H})$  PREs. (b) Distances from the paramagnetic center derived from the  $R_2(^1\text{H})$  PRE data versus the corresponding distances in the crystal structure.

*Control experiment C: Effect of non-specific intermolecular PREs when paramagnetic and diamagnetic samples are measured separately*

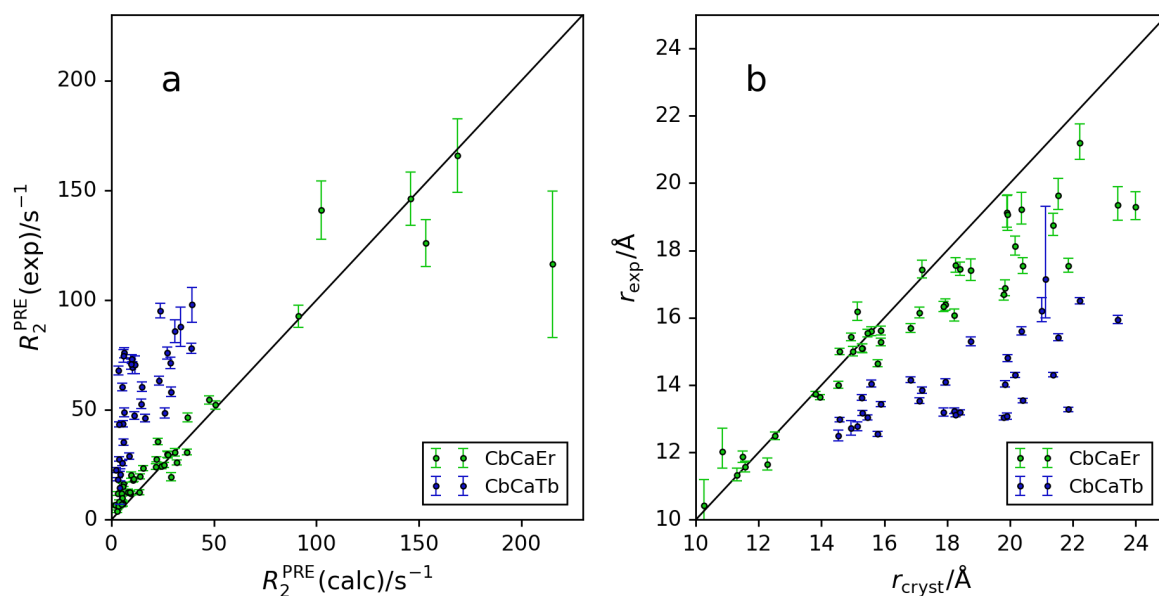
Even at low protein concentrations (in this case 0.2 mM), non-specific intermolecular PREs contribute when pseudo-3D  $R_2$  HSQC relaxation experiments are recorded separately for the paramagnetic and diamagnetic samples. The effect is unimportant for nuclear spins close to the metal (between 10 and 15 Å), but results in too short distances for spins far from the metal center. The RMSD of the distance correlation increased to 1.5 Å (Figure S3). The improved spectral resolution in the absence of an internal reference allowed PRE measurement for 45 amide cross-peaks.



**Figure S3.**  $R_2(^1\text{H})$  PRE data obtained from separate spectra recorded of 0.2 mM solutions of CbCaEr and CbCaLu (experiment C of Table S3: 600 MHz; relaxation delays 0, 5, and 10 ms). See legend of Figure S2 for further details. (a)  $R_2(^1\text{H})$  PREs. (b) Corresponding distances.

*Control experiments D and E: RDC artefacts at high magnetic field and dependence on magnetic susceptibility anisotropy*

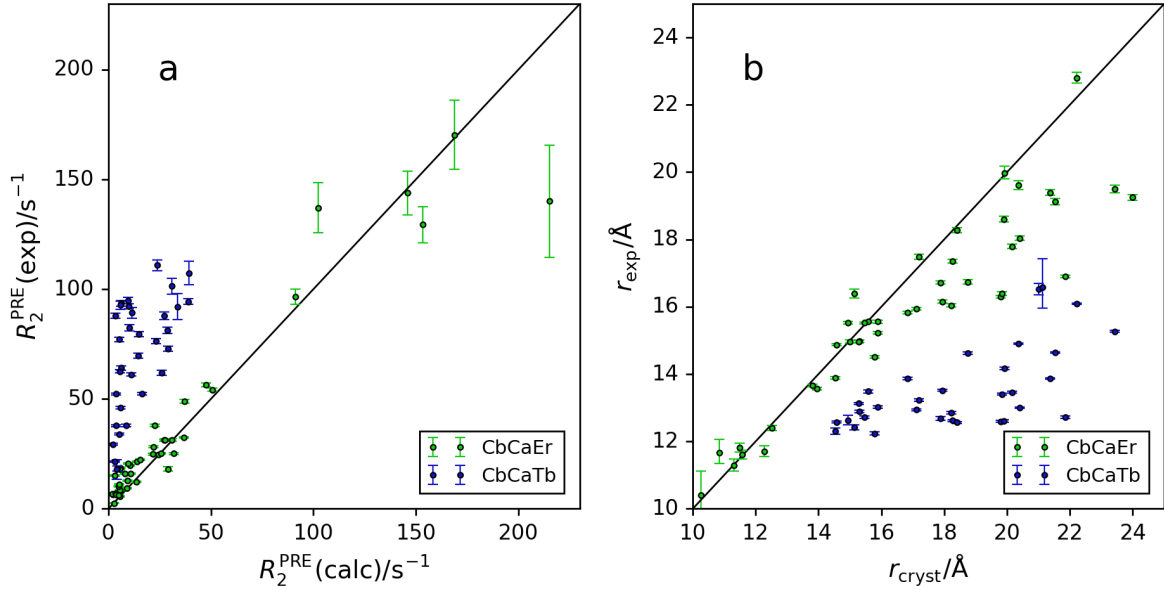
$^1\text{H}$ - $^1\text{H}$  RDCs evolving during the spin-echo period of the  $R_2(^1\text{H})$  relaxation delay result in signal attenuation in the paramagnetic but not in the diamagnetic protein. As RDCs increase with the square of the magnetic field, the effect is more detrimental at higher field strengths. The coupling evolutions are cosine modulated and therefore less pronounced for short relaxation delays. It is thus of interest to explore measurements at a higher magnetic field (800 MHz  $^1\text{H}$  NMR frequency) conducted with a very short relaxation delay (0 and 5 ms; note that RDCs evolve during the INEPT delay even for  $T = 0$ , Figure S1). In addition, RDCs are expected to be less problematic for a lanthanide ion with a small rather than large  $\Delta\chi$  tensor ( $\text{Er}^{3+}$  versus  $\text{Tb}^{3+}$ ). Figure S4 compares  $R_2(^1\text{H})$  PRE data for CbCaEr and CbCaTb recorded on a 800 MHz NMR spectrometer. For CbCaEr, the RMSD of the distance correlation increased to 1.7 Å. For CbCaTb, the apparent PREs were much larger than expected, leading to a very poor RMSD of the distance correlation (4.8 Å). For the  $\Delta\chi$  tensor determined for CbCaTb (Table S1) and the crystal structure of calbindin  $\text{D}_{9\text{k}}$ , we predict that, on average, each amide proton has  $^1\text{H}$ - $^1\text{H}$  RDCs of 15 Hz due to interactions with other protons (predominantly  $\text{H}^{\text{N}}$  and  $\text{H}^{\alpha}$  protons). A 15 Hz RDC attenuates the signal for the relaxation delay  $T = 5$  ms by 10%. This is a greater source of signal attenuation than small PREs, which are of the order of  $2 \text{ s}^{-1}$  at larger distances from the metal.



**Figure S4.**  $R_2(^1\text{H})$  PRE data obtained from separate spectra recorded of CbCaEr, CbCaTb, and CbCaLu (experiments D and E of Table S3: 800 MHz; relaxation delays 0 and 5 ms; 0.2 mM protein concentration). See legend of Figure S2 for further details. (a)  $R_2(^1\text{H})$  PREs. (b) Corresponding distances.

*Control experiments F and G: Allowing RDC evolution during longer relaxation delays and correcting signal attenuations by predicted RDCs*

Adding the data measured with longer relaxation delays (15 and 30 ms) to the data of experiments D and E resulted in larger apparent PREs, and the RMSDs of the distance correlations increased to 1.8 Å and 5.2 Å for CbCaEr and CbCaTb, respectively (Figure S4).

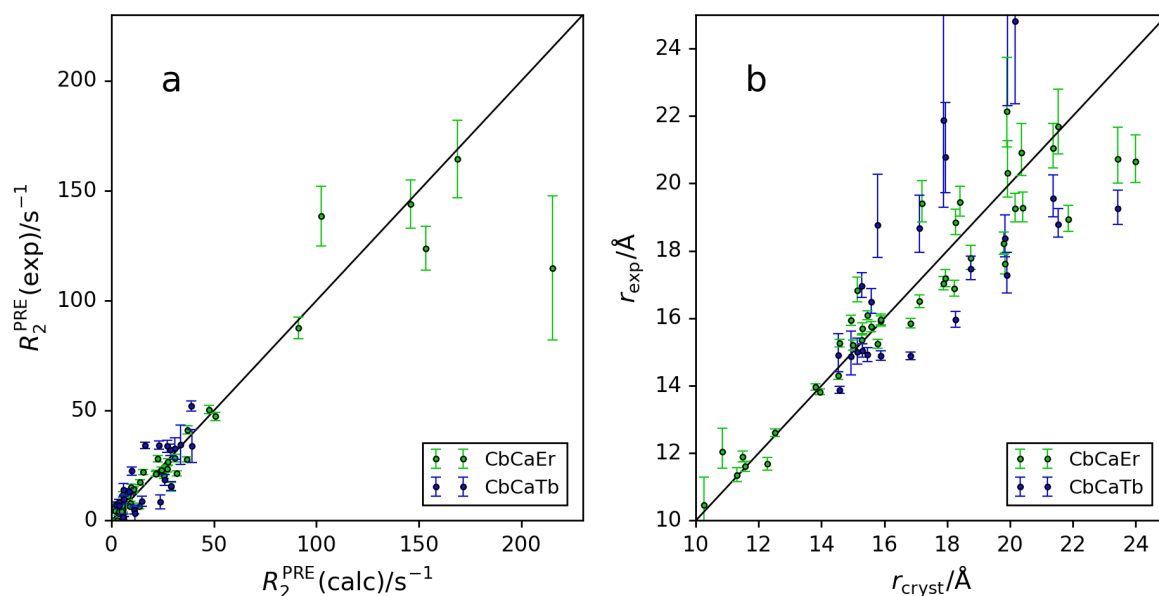


**Figure S5.**  $R_2(^1\text{H})$  PRE data obtained from separate spectra recorded of CbCaEr, CbCaTb, and CbCaLu (experiments F and G of Table S3: 800 MHz; relaxation delays 0, 5, 15, and 30 ms; 0.2 mM protein concentration). See legend of Figure S2 for further details. (a)  $R_2(^1\text{H})$  PREs. (b) Corresponding distances.

Knowledge of the  $\Delta\chi$  tensors and 3D structure of the protein allows prediction of the  $^1\text{H}$ - $^1\text{H}$  RDCs and compensation of their effect on signal attenuation during the  $R_2$  relaxation delays. Using the crystal structure 4ICB, we calculated the RDCs of each amide proton with backbone amide and  $\alpha$ -protons within a 10 Å radius (Eq. 7). To estimate the impact of the RDCs on the PRE measurements, the cross-peak intensities observed for paramagnetic calbindin D<sub>9k</sub> samples,  $I^{\text{para}}$ , were fitted using Eq. S1 with  $R_2^{\text{PRE}}$  and a scaling factor  $a$  between the paramagnetic and diamagnetic spectra as fitting parameters,

$$I^{\text{para}}(T) = a I^{\text{dia}}(T) \exp(-R_2^{\text{PRE}}(T + \frac{1}{2})) \prod_i \cos(\pi J_i^{\text{RDC}} T) \quad (\text{S1})$$

where  $I^{\text{dia}}(T)$  is the cross-peak intensity observed for the diamagnetic reference for the relaxation delay  $T$  and the index  $i$  of the product of cosine functions runs over all  $^1\text{H}^{\text{N}}\text{-}^1\text{H}$  RDCs. The initial time point at  $T = 0$  captures RDC evolution during the INEPT delay  $\frac{1}{2J}$ . Only data from experiments with relaxation delays  $T = 0$  and 5 ms (the data of experiments D and E) were used for the fitting, as inclusion of the data recorded with  $T = 15$  and 30 ms gave significantly poorer fits. Conversion of these corrected PREs into distances greatly improved the correlation between experimental distances and distances in the crystal structure for both CbCaTb and CbCaEr, giving RMSDs of 1.2 and 1.8 Å, respectively, and much better agreements particularly for longer distances (Figure S6). This indicates that RDC artefacts are the main source of error in PRE measurements with lanthanide ions that feature large  $\Delta\chi$  tensors. In the case of CbCaTb, however, the RDC correction resulted in negative PRE values for some of the residues in the flexible regions of the protein and therefore were omitted from the correlation plots of Figure 6. It is well known that RDCs are averaged to smaller values in dynamic polypeptide segments. Therefore the negative PREs likely reflect overcompensation by too large RDCs predicted for a static protein structure.



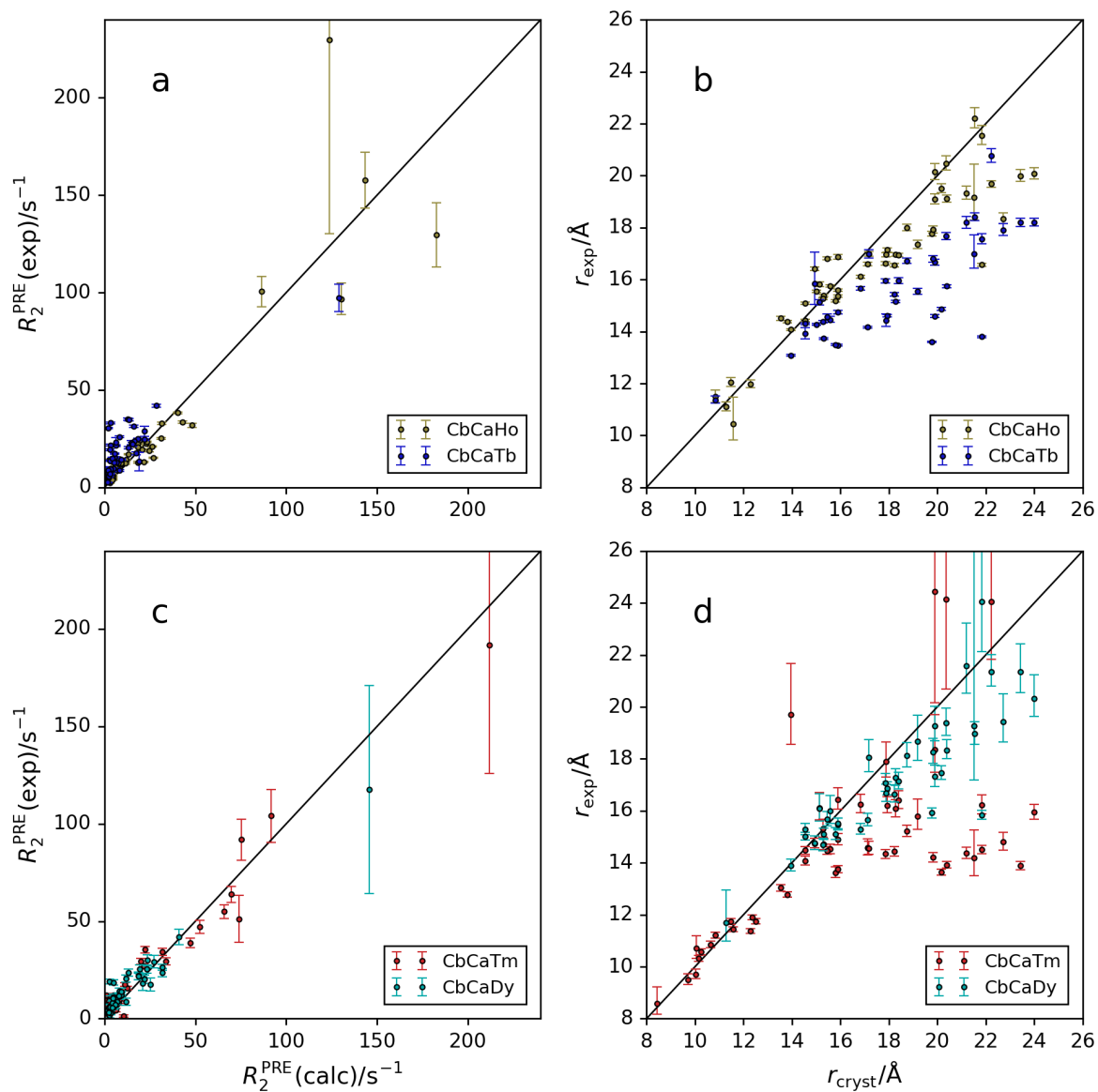
**Figure S6.** Same as Figure S5, but correcting the apparent  $R_2(^1\text{H})$  PRE data by the predicted RDC evolution during the relaxation delay. (a)  $R_2(^1\text{H})$  PREs. (b) Corresponding distances.



*Control experiments H–K: Refocussing of RDCs by a semi-selective RE-BURP pulse*

In an attempt to refocus the RDCs between amide protons and protons with chemical shifts outside the amide region, we substituted the hard  $180^\circ$  refocussing pulse during the relaxation delay  $T$  by a semi-selective RE-BURP pulse covering the  $H^N$  region of the  $^1H$  NMR spectrum. (As the RE-BURP pulse did not excite the water resonance, we also removed the following water-selective flip-back pulse in the  $R_2(^1H)$  HSQC pulse sequence of Figure S1.) The pulse sequence was tested on calbindin D<sub>9k</sub> loaded with  $Tb^{3+}$ ,  $Dy^{3+}$ ,  $Ho^{3+}$ , or  $Tm^{3+}$ . As the RE-BURP pulse also refocused the scalar  $^3J(H^N, H^\alpha)$  couplings, the same modified pulse sequence was also applied to the diamagnetic reference, CbCaY. Spectra were measured with relaxation delays of 0, 5, 10, and 20 ms at a  $^1H$  NMR frequency of 600 MHz. The double-mutant P43M/N56D was used for the experiments conducted with RE-BURP pulse.

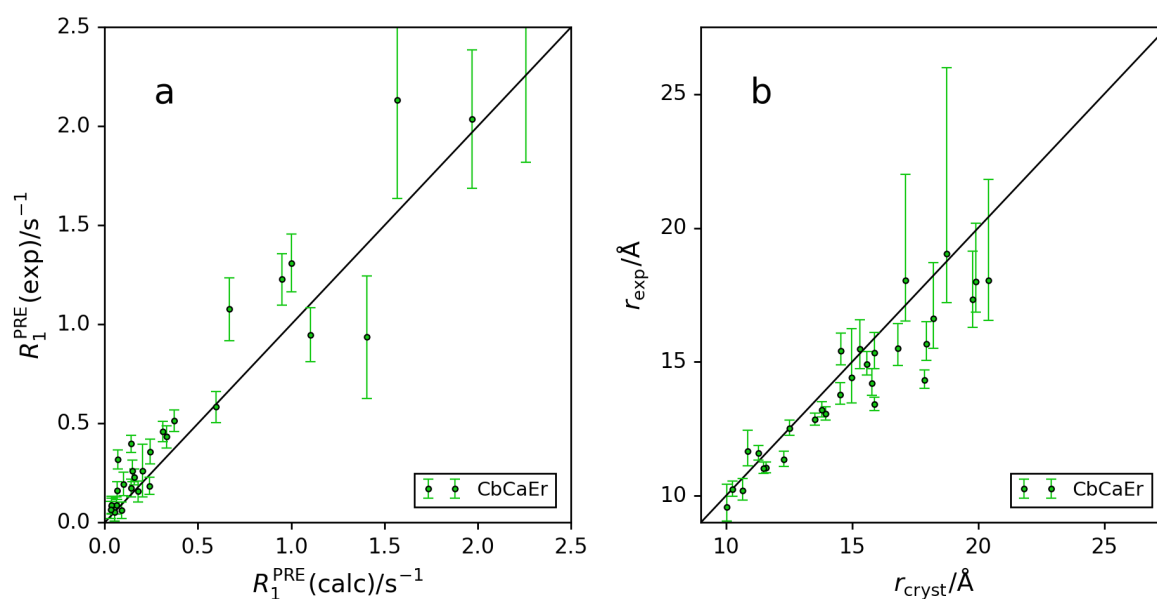
The RMSDs of the distance correlations obtained with  $Ho^{3+}$ ,  $Dy^{3+}$ ,  $Tb^{3+}$ , and  $Tm^{3+}$  were 1.6 Å, 1.9 Å, 3.3 Å, and 3.6 Å, respectively (Figure S7). For CbCaTb, the use of the RE-BURP pulse yielded an improvement in RMSD of the distance correlation from 4.8 Å to 3.3 Å (Table S3). The improvement was limited because the RE-BURP pulse did not refocus RDCs between amide protons and some PREs were overestimated due to large PCSs shifting  $H^\alpha$  resonances into the excitation range of the RE-BURP pulse, which were outside the RE-BURP excitation profile in the diamagnetic sample. This effect greatly affected the PREs determined for some of the amide protons in CbCaTm and CbCaTb.



**Figure S7.**  $R_2(^1\text{H})$  PRE data obtained from separate spectra recorded of CbCaHo, CbCaTb, CbCaDy, CbCaTm, and CbCaY (experiments H–K of Table S3: 600 MHz; relaxation delays 0, 5, 10, and 20 ms; 0.2 mM protein concentration). See legend of Figure S2 for further details. (a)  $R_2(^1\text{H})$  PREs. (b) Corresponding distances.

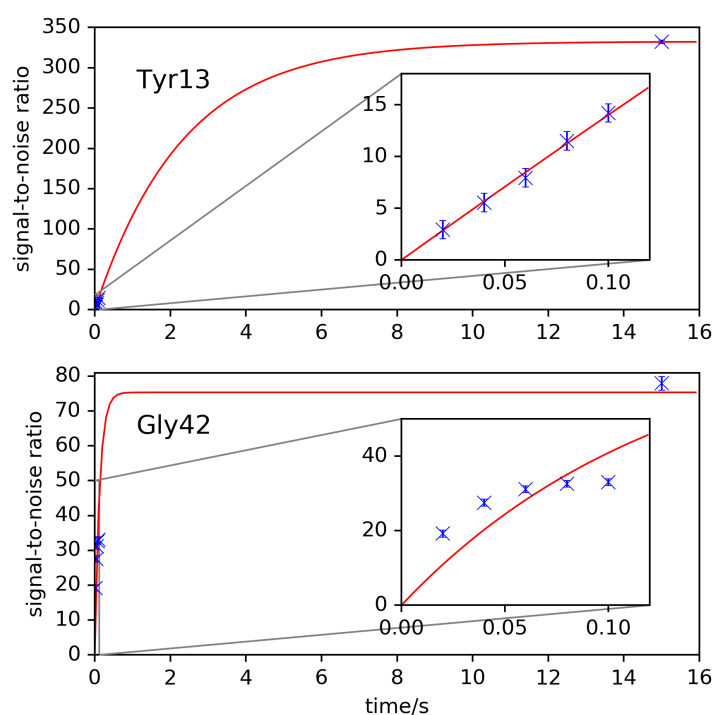
*Control experiment N: Effect of non-specific intermolecular PREs on  $R_1(^1\text{H})$  PREs*

To assess the impact of non-specific intermolecular PREs on  $R_1(^1\text{H})$  PRE data, the  $R_1(^1\text{H})$  saturation–recovery experiment was recorded with a 1:1 mixture of CbCaEr and the diamagnetic reference CbCaLu. The resulting RMSD in distance correlation was 1.4 Å, which is no improvement when compared to the data obtained from separate spectra of diamagnetic and paramagnetic species (Figure S8; see experiments M and N in Table S3). The increase in RMSD can be attributed to a loss in precision due to a shorter total recording time used (24 versus 72 h).



**Figure S8.**  $R_1(^1\text{H})$  PRE data for a 1:1 mixture of CbCaEr and CbCaLu (experiments M and N of Table S3: 600 MHz; relaxation delays 20 ms, 40 ms, 60 ms, 80 ms, 100 ms, and 15 s; 0.4 mM total protein concentration). (a)  $R_1(^1\text{H})$  PREs. (b) Corresponding distances.

*Cross-relaxation effects in  $R_1(^1\text{H})$  measurements*



**Figure S9.** Cross-peak intensities measured in the saturation–recovery experiment plotted versus the recovery time for the amides of Tyr13 and Gly42 in CbCaLu. The insert shows an enlarged view of the initial rate. The red lines indicate the best fits to a single exponential recovery. The fits yielded  $R_1 = 0.43 \text{ s}^{-1}$  for Tyr13 and  $R_1 = 9.13 \text{ s}^{-1}$  for Gly42. The data points of Gly42 do not conform to a model of single exponential relaxation.

**Table S4:** CbCaLu P43G/N56D assignment with uniform  $^{15}\text{N}$  labeling

	H(ppm)	N(ppm)
1Lys	8.581	122.832
2Ser	9.180	120.324
4Glu	8.896	118.039
5Glu	8.126	122.367
6Leu	8.564	120.485
7Lys	8.286	121.152
8Gly	7.887	105.358
9Ile	8.105	123.006
10Phe	8.455	120.340
11Glu	8.541	115.346
12Lys	7.755	119.175
13Tyr	7.314	115.113
14Ala	8.298	119.475
15Ala	6.869	117.002
16Lys	7.165	120.224
17Glu	9.690	116.172
18Gly	9.018	113.168
19Asp	8.362	128.246
21Asn	9.052	114.946
22Gln	7.261	114.316
23Leu	9.629	125.318
24Ser	10.171	122.422
25Lys	8.846	122.197
26Glu	8.056	117.444
27Glu	7.818	120.355
28Leu	8.806	119.384
29Lys	8.455	120.110
30Leu	7.653	118.467
31Leu	8.191	124.340
32Leu	8.865	120.058
33Gln	8.621	114.496
34Thr	7.632	110.852
35Glu	8.422	116.180
36Phe	7.846	114.729
38Ser	8.413	113.842
39Leu	7.983	122.898
40Leu	7.764	117.022
41Lys	7.417	120.335
42Gly	8.179	108.614
43Gly	8.417	109.165
44Ser	8.044	117.328
45Thr	8.406	114.110
46Leu	8.806	122.368
47Asp	8.102	117.192
48Glu	7.788	119.742
49Leu	8.398	121.887
50Phe	9.025	119.341
51Glu	7.887	116.323
52Glu	7.723	118.700

53Leu	7.807	114.893
54Asp	8.317	120.829
55Lys	7.800	122.894
56Asp	7.708	114.362
57Gly	7.917	109.207
58Asp	8.088	117.995
59Gly	10.254	112.740
60Glu	7.539	117.306
61Val	11.028	133.032
62Ser	9.608	127.951
63Phe	9.640	123.482
64Glu	8.429	118.436
65Glu	8.093	121.240
66Phe	8.857	118.946
67Gln	7.629	115.807
68Val	7.125	118.094
69Leu	7.119	120.345
70Val	7.085	117.312
71Lys	7.496	118.217
72Lys	7.381	117.078
73Ile	7.482	114.444
74Ser	7.809	116.988
75Gln	7.617	126.275

**Table S5:** CbCaLu P43G/N56D assignment with uniform  $^{15}\text{N}$  and  $^{13}\text{C}$  labeling

	H (ppm)	N (ppm)	C (ppm)
1Lys	8.577	122.778	176.025
2Ser	9.169	120.236	179.669
4Glu	8.881	117.949	179.893
5Glu	8.114	122.281	180.201
6Leu	8.553	120.400	178.411
7Lys	8.277	121.047	177.916
8Gly	7.877	105.306	177.150
9Ile	8.094	122.898	176.683
10Phe	8.444	120.291	176.375
11Glu	8.532	115.257	179.238
12Lys	7.745	119.073	178.172
13Tyr	7.303	114.996	176.375
14Ala	8.293	119.374	178.948
15Ala	6.863	116.925	177.569
16Lys	7.155	120.113	177.148
17Glu	9.674	116.049	176.113
18Gly	9.007	113.067	174.089
19Asp	8.329	128.095	174.661
21Asn	9.033	114.851	174.005
22Gln	7.248	114.192	174.166
23Leu	9.618	125.244	176.798
24Ser	10.160	122.308	175.237
25Lys	8.832	122.103	178.077
26Glu	8.037	117.209	179.723

27Glu	7.803	120.245	178.994
28Leu	8.791	119.293	177.639
29Lys	8.441	120.019	177.638
30Leu	7.637	118.363	179.033
31Leu	8.176	124.222	179.335
32Leu	8.856	119.936	178.588
33Gln	8.612	114.400	177.507
34Thr	7.617	110.773	175.486
35Glu	8.409	116.072	176.833
36Phe	7.834	114.624	179.505
38Ser	8.403	113.779	176.514
39Leu	7.975	122.818	178.364
40Leu	7.749	116.921	176.971
41Lys	7.401	120.200	176.793
42Gly	8.174	108.525	174.762
43Gly	8.407	109.087	174.874
44Ser	8.043	117.378	174.484
45Thr	8.374	113.951	175.230
46Leu	8.798	122.328	178.166
47Asp	8.109	117.092	178.765
48Glu	7.769	119.588	179.545
49Leu	8.393	121.806	178.397
50Phe	9.016	119.260	176.494
51Glu	7.872	116.234	178.214
52Glu	7.721	118.619	177.982
53Leu	7.804	114.868	179.466
54Asp	8.298	120.665	177.575
55Lys	7.801	122.890	176.857
56Asp	7.697	114.289	177.340
57Gly	7.904	109.116	174.948
58Asp	8.085	117.944	177.114
59Gly	10.247	112.669	172.872
60Glu	7.535	117.176	176.107
61Val	11.017	132.968	174.431
62Ser	9.590	127.834	174.991
63Phe	9.627	123.405	177.696
64Glu	8.421	118.336	180.536
65Glu	8.076	121.140	178.786
66Phe	8.844	118.836	175.958
67Gln	7.616	115.716	177.440
68Val	7.110	117.967	177.838
69Leu	7.116	120.229	177.960
70Val	7.072	117.171	177.890
71Lys	7.482	118.143	177.952
72Lys	7.371	116.947	178.127
73Ile	7.469	114.389	176.360
74Ser	7.804	116.912	173.289
75Gln	7.605	126.178	NaN

**Table S6:** CbCaY P43M/N56D assignment with uniform  $^{15}\text{N}$  labeling

	H(ppm)	N(ppm)
1Lys	8.570	122.750
2Ser	9.157	120.281
4Glu	8.879	118.019
5Glu	8.115	122.315
6Leu	8.557	120.457
7Lys	8.280	121.201
8Gly	7.888	105.398
9Ile	8.100	122.945
10Phe	8.461	120.409
11Glu	8.530	115.358
12Lys	7.755	119.179
13Tyr	7.312	115.112
14Ala	8.316	119.494
15Ala	6.868	117.046
16Lys	7.171	120.198
17Glu	9.684	116.175
18Gly	9.008	113.166
19Asp	8.350	128.152
21Asn	9.069	115.109
22Gln	7.257	114.469
23Leu	9.630	125.503
24Ser	10.156	122.341
25Lys	8.828	122.202
26Glu	8.051	117.341
27Glu	7.804	120.295
28Leu	8.774	119.402
29Lys	8.425	120.035
30Leu	7.656	118.494
31Leu	8.190	124.309
32Leu	8.805	120.036
33Gln	8.623	114.564
34Thr	7.643	110.882
35Glu	8.409	116.174
36Phe	7.862	114.873
38Ser	8.393	113.688
39Leu	7.962	122.687
40Leu	7.724	116.993
41Lys	7.452	120.405
42Gly	8.071	108.853
43Met	8.350	120.037
44Ser	8.245	115.167
45Thr	8.104	113.850
46Leu	8.678	122.502
47Asp	8.114	117.529
48Glu	7.788	119.570
49Leu	8.375	121.783
50Phe	8.979	119.345
51Glu	7.865	116.149
52Glu	7.709	118.673



53Leu	7.838	114.964
54Asp	8.289	120.566
55Lys	7.760	122.896
56Asp	7.697	114.282
57Gly	7.900	109.112
58Asp	8.075	118.108
59Gly	10.305	112.741
60Glu	7.550	117.376
61Val	10.936	132.617
62Ser	9.603	127.760
63Phe	9.618	123.415
64Glu	8.424	118.395
65Glu	8.081	121.161
66Phe	8.848	119.040
67Gln	7.630	115.895
68Val	7.111	118.045
69Leu	7.114	120.256
70Val	7.089	117.301
71Lys	7.494	118.196
72Lys	7.377	117.010
73Ile	7.476	114.398
74Ser	7.802	116.954
75Gln	7.614	126.264

**Table S7:** Data from experiment A (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS(H)	PCS(N)	PCS(C)
1Lys	1.76	0.05	0.043	-0.041	-0.013
2Ser	1.49	0.04	-0.061	-0.045	-0.259
4Glu	1.46	0.06	-0.194	-0.248	-0.197
5Glu	2.42	0.07	-0.172	-0.215	-0.210
6Leu	3.97	0.06	-0.178	-0.216	-0.304
7Lys	7.22	0.08	-0.372	-0.408	-0.573
8Gly	6.25	0.08	-0.379	-0.464	-0.376
9Ile	7.45	0.10	-0.300	-0.318	-0.325
10Phe	12.74	0.12	-0.459	-0.517	-0.879
11Glu	14.45	0.07	-0.770	-0.829	-0.687
12Lys	9.32	0.09	-0.510	-0.536	-0.448
13Tyr	15.68	0.12	-0.429	-0.388	-0.492
14Ala	31.08	0.17	-0.812	-0.754	-1.035
16Lys	14.94	0.15	-0.490	-0.459	-0.304
17Glu	15.06	0.08	-0.424	-0.334	-0.428
18Gly	10.61	0.12	-0.251	-0.319	-0.668
19Asp	76.21	1.06	-0.867	-0.945	-1.658
21Asn	95.47	0.71	-2.173	-2.093	5.382
26Glu	49.28	0.22	0.852	0.952	0.616
27Glu	48.05	0.45	0.589	0.527	0.592
28Leu	79.48	0.42	1.173	1.000	1.044
29Lys	28.54	0.11	1.026	0.991	0.589
30Leu	12.51	0.08	0.494	0.493	0.356
31Leu	12.55	0.13	0.422	0.366	0.426

32Leu	12.64	0.10	0.627	0.596	0.456
33Gln	4.14	0.09	0.434	0.439	0.244
34Thr	1.81	0.07	0.262	0.238	0.165
35Glu	1.79	0.09	0.232	0.201	0.175
36Phe	6.70	0.07	0.304	0.275	0.302
38Ser	1.67	0.19	0.270	0.230	0.370
39Leu	3.00	0.06	0.392	0.323	0.485
40Leu	4.56	0.07	0.435	0.449	0.401
41Lys	1.93	0.06	0.369	0.371	0.294
43Gly	1.81	0.15	0.289	0.275	0.339
44Ser	2.88	0.08	0.380	0.343	0.475
45Thr	7.84	0.08	0.492	0.528	0.746
46Leu	5.07	0.09	0.645	0.750	1.132
47Asp	11.55	0.10	0.705	0.903	0.938
48Glu	10.87	0.18	0.718	0.812	1.165
49Leu	27.11	0.11	1.180	1.313	2.675
50Phe	73.17	0.64	2.147	2.721	2.480
69Leu	78.17	0.50	1.365	1.290	0.750
70Val	25.81	0.13	0.608	0.589	0.360
71Lys	11.97	0.08	0.459	0.398	0.389
72Lys	7.77	0.08	0.553	0.488	0.379
73Ile	4.72	0.07	0.413	0.385	0.237
74Ser	2.45	0.06	0.261	0.281	0.122

**Table S8:** Data from experiment B (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	2.14	0.46	0.051	0.016
2Ser	1.10	0.43	-0.051	-0.039
5Glu	4.54	0.42	-0.163	-0.198
6Leu	4.45	0.43	-0.170	-0.189
7Lys	8.58	0.54	-0.362	-0.407
8Gly	6.30	0.57	-0.370	-0.420
9Ile	7.98	0.64	-0.290	-0.322
11Glu	15.12	0.62	-0.753	-0.830
12Lys	9.80	0.50	-0.496	-0.524
13Tyr	15.22	0.75	-0.428	-0.401
14Ala	25.19	0.69	-0.795	-0.775
15Ala	14.04	2.00	-0.758	-0.791
17Glu	14.03	0.52	-0.415	-0.333
18Gly	11.04	0.74	-0.249	-0.350
19Asp	91.05	3.87	-0.859	-0.908
24Ser	119.67	13.20	-0.109	0.169
27Glu	43.55	0.99	0.600	0.453
28Leu	70.58	1.62	1.168	0.993
29Lys	25.13	0.61	1.022	0.965
30Leu	13.01	0.50	0.496	0.479
31Leu	11.62	0.53	0.422	0.364
32Leu	14.58	0.51	0.625	0.544
35Glu	2.67	0.62	0.235	0.198
39Leu	3.06	0.42	0.392	0.364

40Leu	4.31	0.47	0.432	0.429
41Lys	2.83	0.38	0.368	0.339
43Gly	2.80	2.35	0.286	0.245
46Leu	5.39	0.48	0.640	0.741
49Leu	26.85	0.61	1.163	1.296
50Phe	65.99	1.59	2.118	2.693
51Glu	128.15	6.03	1.746	2.233
68Val	110.61	5.74	1.153	0.958
70Val	23.71	0.69	0.599	0.546
71Lys	10.58	0.48	0.454	0.414
72Lys	7.55	0.45	0.545	0.455
73Ile	5.85	0.41	0.407	0.393
75Gln	1.25	0.30	0.187	0.174

**Table S9:** Data from experiment C (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	2.42	0.84	0.054	0.027
2Ser	1.72	0.71	-0.048	-0.039
4Glu	3.05	0.65	-0.184	-0.218
5Glu	9.02	0.70	-0.162	-0.200
6Leu	7.43	0.85	-0.169	-0.186
7Lys	10.87	0.81	-0.360	-0.395
8Gly	9.88	0.80	-0.368	-0.424
9Ile	11.89	0.96	-0.288	-0.319
11Glu	16.05	1.14	-0.752	-0.831
12Lys	12.41	0.82	-0.494	-0.521
13Tyr	18.65	0.94	-0.413	-0.397
14Ala	30.35	0.92	-0.792	-0.775
15Ala	21.55	2.76	-0.756	-0.799
16Lys	16.22	0.79	-0.475	-0.460
17Glu	14.85	0.93	-0.419	-0.329
18Gly	8.97	1.49	-0.248	-0.363
19Asp	95.27	5.70	-0.862	-0.919
21Asn	91.12	3.38	-2.150	-2.065
24Ser	166.73	28.44	-0.108	0.173
27Glu	49.92	1.52	0.600	0.531
28Leu	65.58	2.71	1.172	0.986
30Leu	13.82	0.92	0.496	0.469
31Leu	12.15	0.87	0.424	0.367
32Leu	19.76	0.80	0.629	0.544
33Gln	7.52	0.74	0.435	0.397
34Thr	4.94	0.88	0.268	0.240
35Glu	3.37	1.17	0.239	0.198
36Phe	9.08	0.96	0.310	0.265
39Leu	7.03	0.70	0.395	0.376
40Leu	6.29	0.61	0.439	0.434
41Lys	3.23	0.70	0.372	0.329
46Leu	6.79	0.91	0.644	0.732
47Asp	13.60	0.61	0.712	0.861
48Glu	14.59	0.70	0.699	0.757

49Leu	30.90	0.88	1.168	1.301
50Phe	69.73	2.17	2.123	2.686
51Glu	127.53	7.06	1.750	2.232
68Val	99.18	7.51	1.156	0.942
69Leu	72.02	3.16	1.341	1.250
70Val	24.30	0.98	0.598	0.541
71Lys	11.99	0.84	0.452	0.409
72Lys	11.46	0.71	0.547	0.435
73Ile	9.19	0.65	0.411	0.402
74Ser	2.41	0.90	0.265	0.279
75Gln	2.38	0.52	0.191	0.173

**Table S10:** Data from experiment D (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS(H)	PCS(N)
1Lys	6.69	1.02	0.054	0.005
2Ser	6.80	0.92	-0.048	-0.029
4Glu	6.13	0.87	-0.184	-0.207
5Glu	12.05	0.88	-0.163	-0.199
6Leu	16.24	1.01	-0.167	-0.185
7Lys	18.15	1.17	-0.363	-0.405
8Gly	12.66	1.38	-0.373	-0.421
9Ile	20.37	1.37	-0.290	-0.325
11Glu	29.81	1.46	-0.762	-0.843
12Lys	19.79	1.16	-0.503	-0.536
13Tyr	35.64	1.44	-0.423	-0.415
14Ala	54.58	1.96	-0.806	-0.797
15Ala	30.81	1.73	-0.761	-0.797
16Lys	29.66	1.37	-0.488	-0.463
17Glu	30.88	1.16	-0.424	-0.342
18Gly	19.58	1.99	-0.253	-0.362
19Asp	116.43	33.32	-0.871	-0.935
21Asn	165.79	17.22	-2.184	-2.092
27Glu	92.73	5.02	0.592	0.528
28Leu	146.18	11.18	1.175	1.010
30Leu	27.60	1.17	0.497	0.488
31Leu	24.36	1.27	0.427	0.370
32Leu	24.22	1.24	0.636	0.564
33Gln	12.00	0.89	0.440	0.417
34Thr	7.21	0.95	0.267	0.257
35Glu	7.30	1.17	0.239	0.198
36Phe	15.21	1.27	0.311	0.275
39Leu	12.08	0.88	0.400	0.368
40Leu	9.91	0.97	0.442	0.433
41Lys	8.09	0.82	0.374	0.339
46Leu	12.54	1.15	0.648	0.771
47Asp	25.02	1.04	0.713	0.892
48Glu	26.06	1.24	0.703	0.774
49Leu	52.35	1.87	1.182	1.334
50Phe	126.16	10.79	2.145	2.742
51Glu	273.75	99.48	1.762	2.272

69Leu	141.06	13.28	1.387	1.312
70Val	46.59	2.02	0.622	0.570
71Lys	23.56	1.08	0.470	0.442
72Lys	18.57	0.99	0.564	0.482
73Ile	12.46	0.91	0.423	0.424
74Ser	7.00	1.06	0.271	0.258
75Gln	3.88	0.56	0.195	0.179

**Table S11:** Data from experiment E (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	22.66	1.09	0.011	0.088
4Glu	27.70	1.15	0.977	1.085
5Glu	67.94	1.99	0.826	0.881
6Leu	76.12	2.40	0.920	0.993
7Lys	47.47	1.78	1.677	1.926
8Gly	29.05	1.51	1.450	1.587
9Ile	69.37	2.57	1.017	1.015
11Glu	71.35	2.72	2.329	2.272
12Lys	60.55	2.28	1.227	1.171
13Tyr	95.24	2.95	0.578	0.398
16Lys	58.24	2.15	0.073	0.062
17Glu	78.08	2.42	-0.486	-0.434
18Gly	85.89	4.84	-0.726	-0.806
31Leu	48.64	2.47	-2.142	-1.771
32Leu	63.31	2.09	-2.447	-2.138
33Gln	73.08	2.14	-1.730	-1.647
34Thr	74.58	3.03	-1.140	-1.057
35Glu	35.42	1.54	-0.955	-0.790
36Phe	48.90	1.93	-1.130	-1.059
39Leu	60.49	1.64	-1.306	-1.247
40Leu	43.63	1.38	-1.477	-1.456
41Lys	43.46	1.18	-1.229	-1.147
42Gly	20.53	2.70	-1.036	-1.002
43Gly	14.68	7.54	-0.937	-0.939
46Leu	52.80	1.96	-1.984	-2.381
47Asp	75.98	2.65	-1.992	-2.595
48Glu	87.94	8.87	-1.988	-2.340
70Val	97.92	7.33	-1.049	-1.537
71Lys	46.34	1.66	-0.900	-0.774
72Lys	70.61	3.96	-1.466	-1.413
73Ile	70.92	2.24	-1.137	-1.213
74Ser	25.84	1.19	-0.639	-0.482
75Gln	18.42	0.62	-0.452	-0.380

**Table S12:** Data from experiment F (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	6.40	0.23	0.054	0.005
2Ser	6.91	0.19	-0.048	-0.029

4Glu	7.18	0.21	-0.184	-0.207
5Glu	15.08	0.23	-0.163	-0.199
6Leu	18.73	0.26	-0.167	-0.185
7Lys	19.78	0.32	-0.363	-0.405
8Gly	15.99	0.39	-0.373	-0.421
9Ile	20.64	0.42	-0.290	-0.325
11Glu	31.18	0.50	-0.762	-0.843
12Lys	21.43	0.35	-0.503	-0.536
13Tyr	37.83	0.57	-0.423	-0.415
14Ala	56.47	0.91	-0.806	-0.797
15Ala	31.26	0.54	-0.761	-0.797
16Lys	31.38	0.44	-0.488	-0.463
17Glu	32.51	0.39	-0.424	-0.342
18Gly	18.17	0.82	-0.253	-0.362
19Asp	140.05	24.54	-0.871	-0.935
21Asn	170.30	14.60	-2.184	-2.092
27Glu	96.65	3.21	0.592	0.528
28Leu	143.83	9.98	1.175	1.010
30Leu	28.26	0.37	0.497	0.488
31Leu	24.76	0.38	0.427	0.370
32Leu	24.83	0.39	0.636	0.564
33Gln	12.89	0.22	0.440	0.417
34Thr	8.48	0.26	0.267	0.257
35Glu	5.53	0.31	0.239	0.198
36Phe	18.12	0.36	0.311	0.275
39Leu	10.21	0.22	0.400	0.368
40Leu	11.07	0.24	0.442	0.433
41Lys	6.62	0.17	0.374	0.339
46Leu	12.32	0.33	0.648	0.771
47Asp	25.18	0.31	0.713	0.892
48Glu	25.14	0.40	0.703	0.774
49Leu	54.25	0.85	1.182	1.334
50Phe	129.36	8.12	2.145	2.742
51Glu	278.28	92.13	1.762	2.272
69Leu	137.07	11.21	1.387	1.312
70Val	48.99	0.81	0.622	0.570
71Lys	22.42	0.30	0.470	0.442
72Lys	16.11	0.27	0.564	0.482
73Ile	9.42	0.21	0.423	0.424
74Ser	6.18	0.22	0.271	0.258
75Gln	2.50	0.11	0.195	0.179

**Table S13:** Data from experiment G (see Table S3)

	R2(PRE)	Error-R2(PRE)	PCS(H)	PCS(N)
1Lys	29.43	0.32	0.011	0.088
4Glu	37.86	0.37	0.977	1.085
5Glu	87.93	1.12	0.826	0.881
6Leu	93.56	1.47	0.920	0.993
7Lys	61.20	0.83	1.677	1.926
8Gly	38.00	0.60	1.450	1.587

9Ile	82.49	1.54	1.017	1.015
11Glu	81.25	1.56	2.329	2.272
12Lys	79.50	1.27	1.227	1.171
13Tyr	110.93	2.38	0.578	0.398
16Lys	72.90	1.12	0.073	0.062
17Glu	94.40	1.34	-0.486	-0.434
18Gly	101.41	3.75	-0.726	-0.806
31Leu	61.96	1.14	-2.142	-1.771
32Leu	76.44	1.14	-2.447	-2.138
33Gln	92.23	1.18	-1.730	-1.647
34Thr	92.72	1.92	-1.140	-1.057
35Glu	46.00	0.58	-0.955	-0.790
36Phe	64.18	0.95	-1.130	-1.059
39Leu	77.17	0.95	-1.306	-1.247
40Leu	62.61	0.60	-1.477	-1.456
41Lys	52.32	0.46	-1.229	-1.147
42Gly	18.32	1.12	-1.036	-1.002
43Gly	17.89	4.86	-0.937	-0.939
46Leu	69.72	1.23	-1.984	-2.381
47Asp	87.93	1.63	-1.992	-2.595
48Glu	92.02	5.71	-1.988	-2.340
70Val	107.41	5.35	-1.049	-1.537
71Lys	52.40	0.69	-0.900	-0.774
72Lys	89.25	2.32	-1.466	-1.413
73Ile	94.90	1.41	-1.137	-1.213
74Ser	34.00	0.38	-0.639	-0.482
75Gln	21.51	0.15	-0.452	-0.380

**Table S14:** Data from experiment H (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	4.65	0.30	0.159	0.191
2Ser	4.53	0.30	0.297	0.289
4Glu	2.48	0.26	0.531	0.599
5Glu	14.37	0.29	0.512	0.553
6Leu	9.44	0.29	0.633	0.673
7Lys	11.73	0.30	1.012	1.142
8Gly	8.75	0.37	0.912	1.004
9Ile	14.45	0.39	0.809	0.806
10Phe	22.65	0.39	1.259	1.251
11Glu	22.48	0.43	1.601	1.656
12Lys	14.18	0.34	0.969	0.939
13Tyr	24.37	0.48	0.742	0.599
14Ala	38.26	0.56	1.237	1.015
15Ala	21.08	0.36	0.924	0.886
16Lys	23.52	0.32	0.221	0.181
17Glu	25.14	0.41	-0.275	-0.293
18Gly	19.02	0.62	-0.597	-0.647
19Asp	129.56	16.47	-0.793	-0.244
21Asn	157.68	14.35	2.823	2.992
28Leu	229.84	99.68	-2.724	-2.068

29Lys	31.89	1.06	-1.999	-1.698
30Leu	12.90	0.41	-1.071	-0.928
31Leu	19.46	0.32	-0.717	-0.548
32Leu	20.74	0.35	-0.780	-0.608
33Gln	12.50	0.28	-0.611	-0.531
34Thr	6.13	0.38	-0.339	-0.297
35Glu	4.45	0.39	-0.202	-0.139
36Phe	8.98	0.41	-0.266	-0.175
38Ser	7.80	0.53	-0.241	-0.246
39Leu	6.09	0.27	-0.351	-0.332
40Leu	5.38	0.29	-0.454	-0.456
41Lys	2.97	0.31	-0.397	-0.383
42Gly	5.69	0.44	-0.363	-0.343
43Met	6.02	1.94	-0.348	-0.338
44Ser	10.92	0.61	-0.427	-0.443
45Thr	14.14	0.36	-0.546	-0.581
46Leu	12.39	0.44	-0.892	-1.029
47Asp	13.15	0.31	-1.021	-1.247
48Glu	15.18	0.37	-0.812	-0.888
49Leu	33.64	0.60	-1.298	-1.428
50Phe	96.81	8.10	-2.570	-3.107
69Leu	100.55	7.85	-0.330	-0.497
70Val	32.93	0.58	0.181	0.046
71Lys	17.02	0.31	-0.052	-0.043
72Lys	12.52	0.31	-0.264	-0.254
73Ile	12.57	0.32	-0.151	-0.188
74Ser	4.04	0.32	-0.054	-0.027
75Gln	5.11	0.19	-0.048	-0.033

**Table S15:** Data from experiment I (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	3.19	0.81	0.205	0.250
2Ser	4.28	0.99	0.420	0.407
4Glu	6.49	0.89	0.828	0.930
5Glu	19.11	1.20	0.737	0.779
6Leu	18.56	1.34	0.883	0.923
7Lys	13.13	1.17	1.441	1.611
8Gly	8.53	1.32	1.213	1.311
9Ile	14.19	1.65	0.940	0.895
10Phe	21.97	2.01	1.425	1.317
11Glu	25.44	2.22	1.874	1.820
12Lys	20.56	1.94	0.941	0.854
13Tyr	25.32	2.36	0.388	0.177
14Ala	41.85	3.99	0.735	0.328
16Lys	29.99	2.93	-0.496	-0.429
17Glu	23.63	2.06	-1.247	-1.118
18Gly	17.44	3.36	-1.410	-1.619
21Asn	117.54	53.37	2.831	3.416
31Leu	18.04	3.58	-1.651	-1.319
32Leu	21.72	1.76	-1.598	-1.307



33Gln	11.31	1.25	-1.169	-1.031
34Thr	11.19	1.55	-0.761	-0.674
35Glu	5.88	1.18	-0.557	-0.435
36Phe	8.15	1.29	-0.600	-0.479
38Ser	5.62	1.56	-0.448	-0.433
39Leu	7.93	0.94	-0.618	-0.589
40Leu	10.65	0.94	-0.755	-0.753
41Lys	1.56	1.00	-0.622	-0.584
42Gly	2.98	1.06	-0.543	-0.507
43Met	5.89	5.78	-0.491	-0.506
44Ser	7.12	1.91	-0.578	-0.589
45Thr	12.17	1.45	-0.698	-0.752
46Leu	8.72	1.78	-1.207	-1.388
47Asp	20.28	2.13	-1.219	-1.520
48Glu	29.25	3.15	-0.888	-0.952
70Val	26.34	1.61	0.281	0.092
71Lys	23.61	1.88	0.031	0.034
72Lys	14.02	1.66	-0.372	-0.365
73Ile	11.88	1.29	-0.260	-0.324
74Ser	5.70	0.91	-0.091	-0.044
75Gln	3.18	0.53	-0.072	-0.046

**Table S16:** Data from experiment J (see Table S3)

	R2 (PRE)	Error-R2 (PRE)	PCS (H)	PCS (N)
1Lys	5.78	0.31	0.028	0.101
2Ser	5.77	0.28	0.391	0.381
4Glu	5.38	0.26	0.982	1.103
5Glu	30.42	0.33	0.825	0.895
6Leu	33.07	0.39	0.925	0.982
7Lys	21.57	0.34	1.678	1.893
8Gly	9.61	0.36	1.450	1.598
9Ile	15.56	0.40	1.036	1.027
10Phe	35.05	0.47	1.565	1.470
11Glu	31.35	0.52	2.351	2.324
12Lys	25.96	0.39	1.246	1.195
13Tyr	34.77	0.52	0.620	0.437
14Ala	41.94	0.67	1.322	0.954
15Ala	24.84	0.41	1.123	1.098
16Lys	23.70	0.37	0.091	0.096
17Glu	24.52	0.40	-0.476	-0.425
18Gly	17.51	0.71	-0.723	-0.775
19Asp	97.29	6.96	-1.204	-0.660
31Leu	22.96	1.08	-2.087	-1.732
32Leu	20.43	0.53	-2.391	-2.054
33Gln	17.28	0.32	-1.691	-1.573
34Thr	21.76	0.54	-1.104	-1.008
35Glu	9.81	0.42	-0.928	-0.770
36Phe	9.35	0.42	-1.096	-0.977
38Ser	6.35	0.49	-0.888	-0.873
39Leu	13.78	0.28	-1.243	-1.219

40Leu	19.47	0.46	-1.420	-1.445
41Lys	7.15	0.46	-1.201	-1.116
42Gly	5.78	0.42	-1.054	-0.986
43Met	8.70	1.91	-0.989	-1.011
44Ser	14.85	0.62	-1.220	-1.230
45Thr	12.73	0.40	-1.454	-1.500
46Leu	8.74	0.48	-1.982	-2.348
47Asp	22.08	1.00	-1.991	-2.528
48Glu	13.28	4.75	-1.983	-2.255
70Val	28.79	2.61	-1.014	-1.347
71Lys	14.23	0.35	-0.864	-0.741
72Lys	23.29	2.25	-1.451	-1.320
73Ile	12.71	0.55	-1.109	-1.146
74Ser	6.88	0.31	-0.624	-0.507
75Gln	2.62	0.19	-0.447	-0.379

**Table S17:** Data from experiment K (see Table S3)

	R2(PRE)	Error-R2(PRE)	PCS(H)	PCS(N)
1Lys	10.74	0.71	-0.023	-0.073
2Ser	4.67	0.48	-0.245	-0.229
5Glu	8.24	0.54	-0.531	-0.589
7Lys	4.24	0.48	-1.109	-1.278
8Gly	6.19	0.52	-1.091	-1.234
9Ile	8.49	0.64	-0.912	-0.948
10Phe	3.90	0.58	-1.424	-1.449
11Glu	7.58	0.66	-2.109	-2.248
12Lys	7.99	1.02	-1.512	-1.595
13Tyr	12.03	1.07	-1.115	-1.005
14Ala	1.31	0.56	-1.928	-1.734
16Lys	5.97	0.80	-1.010	-0.938
17Glu	8.34	0.51	-0.681	-0.555
18Gly	4.39	0.84	-0.305	-0.480
19Asp	38.79	2.44	-1.800	-1.975
24Ser	91.87	10.49	1.365	2.965
25Lys	104.08	13.49	7.570	7.235
26Glu	26.99	1.18	3.467	3.399
27Glu	29.35	1.52	2.518	2.160
28Leu	34.31	1.79	3.719	2.941
29Lys	15.66	0.91	3.003	2.681
30Leu	7.04	0.61	1.530	1.408
31Leu	8.18	0.58	1.150	0.943
32Leu	11.39	0.66	1.475	1.234
33Gln	4.43	0.54	1.095	0.996
34Thr	2.00	0.69	0.647	0.587
35Glu	0.36	0.78	0.482	0.395
36Phe	9.36	0.72	0.613	0.589
38Ser	7.31	1.02	0.544	0.536
39Leu	10.54	0.59	0.779	0.734
40Leu	11.97	0.62	0.935	0.959
41Lys	4.21	0.56	0.806	0.787

42Gly	8.71	0.81	0.722	0.667
43Met	9.38	3.33	0.685	0.710
44Ser	4.95	1.06	0.861	0.887
45Thr	8.84	0.63	1.103	1.169
46Leu	8.05	0.85	1.662	1.934
47Asp	8.37	0.53	1.934	2.405
48Glu	7.47	0.55	1.719	1.951
49Leu	17.66	0.91	2.656	2.949
50Phe	29.44	1.75	5.102	6.266
51Glu	54.91	3.45	4.103	5.129
52Glu	63.85	4.15	3.332	2.944
53Leu	191.74	66.02	8.584	8.378
67Gln	51.10	12.04	-1.658	-0.685
68Val	47.12	3.26	1.329	1.143
69Leu	35.42	1.71	1.854	1.805
70Val	9.85	0.69	0.646	0.787
71Lys	4.16	0.54	0.563	0.485
72Lys	2.34	0.51	0.819	0.744
73Ile	3.92	0.48	0.635	0.641
74Ser	0.38	0.59	0.366	0.300
75Gln	0.39	0.31	0.253	0.225

**Table S18:** Data from experiment L (see Table S3)

	R1(PRE)	Error-R1(PRE)	PCS(H)	PCS(N)
6Leu	0.120	0.020	0.916	0.967
7Lys	0.158	0.020	1.661	1.883
8Gly	0.127	0.019	1.439	1.582
9Ile	0.120	0.021	1.018	0.999
11Glu	0.310	0.024	2.323	2.271
12Lys	0.249	0.022	1.228	1.179
13Tyr	0.310	0.024	0.601	0.407
15Ala	0.264	0.023	1.120	1.112
16Lys	0.194	0.021	0.096	0.087
17Glu	0.445	0.020	-0.450	-0.410
19Asp	1.615	0.284	-1.095	-0.603
21Asn	1.253	0.238	4.129	4.930
31Leu	0.172	0.039	-2.094	-1.737
32Leu	0.178	0.025	-2.395	-2.063
33Gln	0.096	0.020	-1.699	-1.577
34Thr	0.078	0.028	-1.114	-1.018
35Glu	0.031	0.018	-0.930	-0.785
36Phe	0.047	0.019	-1.095	-1.022
39Leu	0.032	0.019	-1.274	-1.253
47Asp	0.147	0.048	-1.979	-2.529
48Glu	0.351	0.155	-1.959	-2.266
70Val	0.523	0.091	-0.999	-1.438
71Lys	0.175	0.033	-0.869	-0.724
72Lys	0.130	0.065	-1.416	-1.327
73Ile	0.131	0.032	-1.100	-1.132

**Table S19:** Data from experiment M (see Table S3)

	R1(PRE)	Error-R1(PRE)	PCS(H)	PCS(N)
6Leu	0.015	0.017	-0.169	-0.190
7Lys	0.098	0.017	-0.361	-0.405
8Gly	0.054	0.018	-0.368	-0.425
9Ile	0.059	0.018	-0.288	-0.319
11Glu	0.214	0.020	-0.753	-0.838
12Lys	0.123	0.017	-0.495	-0.531
13Tyr	0.202	0.019	-0.416	-0.413
14Ala	0.447	0.023	-0.793	-0.779
15Ala	0.216	0.040	-0.745	-0.793
16Lys	0.212	0.021	-0.476	-0.459
17Glu	0.210	0.018	-0.416	-0.340
19Asp	0.973	0.186	-0.865	-0.916
21Asn	1.209	0.073	-2.148	-2.072
27Glu	0.727	0.037	0.596	0.525
28Leu	1.195	0.054	1.168	0.985
30Leu	0.205	0.019	0.494	0.469
31Leu	0.236	0.017	0.422	0.364
32Leu	0.238	0.017	0.628	0.536
33Gln	0.066	0.016	0.435	0.395
34Thr	0.051	0.017	0.265	0.237
35Glu	0.031	0.017	0.235	0.194
36Phe	0.061	0.017	0.309	0.260
39Leu	0.037	0.018	0.393	0.371
47Asp	0.149	0.021	0.714	0.872
48Glu	0.279	0.026	0.699	0.763
49Leu	0.476	0.028	1.169	1.308
50Phe	1.258	0.065	2.123	2.690
51Glu	2.222	0.250	1.745	2.222
69Leu	1.053	0.065	1.335	1.243
70Val	0.375	0.026	0.597	0.533
71Lys	0.051	0.024	0.451	0.410
72Lys	0.136	0.019	0.545	0.448
73Ile	0.063	0.017	0.409	0.397

**Table S20:** Data from experiment N (see Table S3)

	R1(PRE)	Error-R1(PRE)	PCS(H)	PCS(N)
6Leu	0.085	0.038	-0.170	-0.189
7Lys	0.161	0.043	-0.362	-0.407
8Gly	0.053	0.045	-0.370	-0.420
9Ile	0.087	0.043	-0.290	-0.322
11Glu	0.157	0.052	-0.753	-0.830
12Lys	0.062	0.042	-0.496	-0.524
13Tyr	0.261	0.053	-0.428	-0.401
14Ala	0.459	0.051	-0.795	-0.775
15Ala	0.261	0.133	-0.758	-0.791
17Glu	0.185	0.042	-0.415	-0.333

19Asp	0.935	0.309	-0.859	-0.908
21Asn	0.947	0.136	-2.132	-2.031
24Ser	3.034	1.216	-0.109	0.169
27Glu	0.583	0.079	0.600	0.453
28Leu	1.227	0.128	1.168	0.993
29Lys	0.514	0.054	1.022	0.965
30Leu	0.174	0.044	0.496	0.479
31Leu	0.229	0.039	0.422	0.364
32Leu	0.398	0.043	0.625	0.544
35Glu	0.088	0.045	0.235	0.198
39Leu	0.064	0.043	0.392	0.364
49Leu	0.433	0.056	1.163	1.296
50Phe	1.310	0.145	2.118	2.693
51Glu	2.036	0.349	1.746	2.233
68Val	2.132	0.498	1.153	0.958
69Leu	1.077	0.158	1.342	1.235
70Val	0.357	0.062	0.599	0.546
71Lys	0.194	0.058	0.454	0.414
72Lys	0.318	0.047	0.545	0.455

Bruker pulse program of the  $R_2(^1\text{H})$  pseudo-3D interleaved  $[^{15}\text{N}-^1\text{H}]$ -HSQC relaxation experiment (see Figure S1)

```
;hsqcNwgt2ps3dr2.ho
;avance-version (02/05/31)
;15N-HSQC with watergate(using selective 90 deg. pulses) and R2 relaxation
;block
;32-step phase cycle, NS=8 is OK
;with decoupling during acquisition
;using f3 - channel
;water suppression using 90sel-180-90sel Watergate
;T2 relaxation measured via pseudo 3d interleaved acquisition

;Pulse sequence originally from
;Iwahara, J., Tang, C., & Clore, G. M. (2007).
;Practical Aspects of  $^1\text{H}$  Transverse Paramagnetic Relaxation Enhancement
;Measurements on Macromolecules. Journal of Magnetic Resonance
;(San Diego, Calif. : 1997), 184(2), 185–195

;Modifications by
;H. W. Orton and G. Otting 2018

;$DIM=3D

#include <Avance.incl>
#include <Grad.incl>
#include <Delay.incl>

prosol relations=<triple>

"d0=3u"
"in0=inf2/2"

"p2=p1*2"
"p22=p21*2"

"d0=in0*0.5*cnst10-p21*0.63-p2*0.5"
"d16=200u"
"d11=30m"
"d26=1s/(cnst4*4)"
"DELTA=d26-p16-d16-p11-16u"
"DELTA1=d26-p16-d16-p11-16u+p1*0.63"
"DELTA3=d26-p35*0.5"
"DELTA4=d26-p19-d16-4u"
"DELTA5=d26-4u-8u-4u-p10-4u-p19-d16"
"d10=3u"

aqseq 312

1 30m ze
   d11 st0
2 6m do:f3
3 3m
4 d1 p13:f3 p12:f2 p11:f1
   "TAU=vd*0.25+4u"
   4u UNBLKGRAD
   (p21 ph0):f3
   p16:gp7
   1m p10:f1
   (p11:sp1 ph6):f1
   4u
```

```

8u p11:f1
4u
(p1 ph5)
4u
p19:gp4
d16
DELTA4
TAU
(p22 ph8):f3
TAU
(p2 ph9):f1
4u
8u p10:f1
4u
(p10:sp2 ph10)
4u
p19:gp4
d16
DELTA5
4u p11:f1
TAU
(p22 ph8):f3
TAU
(p1 ph1)
4u
p19:gp4
d16
(p21 ph4):f3
if "d0>p8*0.5"{
    "d10=d0-p8*0.5"
    d10
    (center (p8:sp13 ph0):f2 (p2 ph1):f1)
    d10
}
else{
d0
(p2 ph1)
d0
}
(p21 ph0):f3
4u
p19:gp4
d16
(p1 ph1)
8u
p16:gp2
d16 p10:f1
DELTA
(p11:sp1 ph2:r):f1
4u
4u p11:f1
(center (p2 ph0) (p22 ph0):f3 )
4u p10:f1
(p11:sp1 ph2:r):f1
4u
p16:gp2
d16
DELTA1
4u BLKGRAD
4u p116:f3

goscnp ph31 cpd3:f3

```

```

3m do:f3
3m st ivd
lo to 3 times nbl

3m ipp4 ipp5 ipp6 ipp8 ipp9 ipp10 ipp31
lo to 4 times ns

d1 mc #0 to 4
  F1QF()
  F2PH(calph(ph4, +90) & calph(ph8, +90) & exec(rppall), caldel(d0, +in0))
exit

ph0=0
ph1=1
ph2=2
ph3=3
ph4=0 2
ph5=0 0 2 2
ph6=2 2 0 0
ph8=0
ph9=1 1 1 1 3 3 3 3
ph10=3 3 3 3 1 1 1 1
ph31=0 2 2 0 0 2 2 0

;td1: number of delays in vd-list
;td2: number of experiments in F2
;nbl: must be set to same values as tdl
;p11 : f1 channel - power level for pulse (default)
;p13 : f3 channel - power level for pulse (default)
;p116: f3 channel - power level for CPD/BB decoupling
;p1 : f1 channel - 90 degree high power pulse
;p8 : f2 channel - carbon decoupling pulse
;p11 : 1000u - 90 degree selective water
;p10 : 2000u - 180 degree selective water
;p15: 1500u - gaussian water refocusing pulse
;spnam1: Sincl.1000 - shaped pulse 90 degree Watergate
;sp1: 45.8 dB for 1.5 ms 90 deg. Sincl.1000 pulse
;p16: 0.9m gradient pulse of Watergate
;p19: 0.5m gradient pulse
;p20: pfg for initial spin echo and inept
;p21: f3 channel - 90 degree high power pulse
;p22: f3 channel - 180 degree high power pulse
;p24: (240us) f3 channel - 180 degree decoupling train
;p10: 120 dB
;vd : variable delay, taken from vd-list
;NBL: = tdl
;d0 : incremented delay (2D) [3 usec]
;d1 : relaxation delay; 1-5 * T1
;d10: computed internally for C-decoupling
;d11: delay for disk I/O [30 msec]
;d16: delay for homospoil/gradient recovery
;cnst10: 1 for 90/-180, 2 for 180/-360 phase correction
;d26 : 1/(4J)NH (refocussing INEPT)
;in0: 1/(2 * SW(X)) = DW(X)
;nd0: 2
;NS: 4 * n
;DS: 16
;td1: number of experiments
;FnMODE: States-TPPI, TPPI, States or QSEC
;cpd3: decoupling according to sequence defined by cpdprg3
;pcpd3: f3 channel - 90 degree pulse for decoupling sequence

;for z-only gradients:

```



```

;gpz0: 15%
;gpz1: 10% (suppression of radiation damping)
;gpz2: 25% (Watergate)
;gpz3: 35%
;gpz4: 50%
;gpz7: 27% (kill initial N magnetisation)

;use gradient files:
;gpnam0: SINE.100
;gpnam1: SINE.100
;gpnam2: SINE.100
;gpnam3: SINE.100
;gpnam7: SINE.100

```

Bruker pulse program of the  $R_1(^1\text{H})$  pseudo-3D interleaved  $[^{15}\text{N}-^1\text{H}]$ -HSQC saturation–recovery relaxation experiment (see Figure 7 of main text)

```

;hsqcNwgtlps3d.ho
;avance-version (02/05/31)
;15N-HSQC with watergate(using selective 90 deg. pulses) with saturation
;recovery
;32-step phase cycle, NS=8 is OK
;with decoupling during acquisition
;using f3 - channel
;water suppression using 90sel-180-90sel Watergate
;t1 relaxation measured via pseudo 3d interleaved acquisition

;H. W. Orton and G. Otting

;$DIM=3D

#include <Avance.incl>
#include <Grad.incl>
#include <Delay.incl>

prosol relations=<triple>

;shaped pulse powers for 3.5 ms selective WET saturation
"p13=3500"
"spw2=plw1*pow((((0.25*(81.4/90.))/(p13*0.5889))/(0.25/p1)),2)" ;Sinc pulse power
"spw3=plw1*pow((((0.25*(101.4/90.))/(p13*0.5889))/(0.25/p1)),2)" ;Sinc pulse power
"spw4=plw1*pow((((0.25*(69.3/90.))/(p13*0.5889))/(0.25/p1)),2)" ;Sinc pulse power
"spw5=plw1*pow((((0.25*(161.1/90.))/(p13*0.5889))/(0.25/p1)),2)" ;Sinc pulse power

"spoal2=0.5"
"spoal3=0.5"
"spoal4=0.5"
"spoal5=0.5"

"spoffs2=0"
"spoffs3=0"
"spoffs4=0"
"spoffs5=0"

"d0=3u"

```

```

"in0=inf2/2"

"p2=p1*2"

"p22=p21*2"

"d0=in0*0.5*cnst10-p21*0.63-p2*0.5"
"d16=200u"
"d11=30m"
"d26=1s/(cnst4*4)"
"DELTA=d26-p16-d16-p11-16u"
"DELTA1=d26-p16-d16-p11-16u+p1*0.63"
"DELTA2=d26-p19-4u-d16"
"d10=3u"

```

aqseq 312

```

1 30m ze
   d11 st0
2 6m do:f3
3 3m
4 d1 pl3:f3 pl2:f2 pl1:f1
   "TAU=vd"
   (p21 ph0):f3
   8u
   4u UNBLKGRAD
   (p1*0.904 ph0)
   10u pl0:f1
   (p13:sp2 ph2):f1
   4u
   p16:gp21
   d16 pl1:f1
   (p1*1.127 ph1)
   10u pl0:f1
   (p13:sp3 ph3):f1
   4u
   p16:gp22
   d16 pl1:f1
   (p1*0.77 ph1)
   10u pl0:f1
   (p13:sp4 ph3):f1
   4u
   p16:gp23
   d16 pl1:f1
   (p1*1.789 ph1)
   10u pl0:f1
   (p13:sp5 ph3):f1
   4u
   p16:gp24
   d16 pl1:f1
   2u BLKGRAD
   TAU
   2u UNBLKGRAD
   (p1 ph5)
   4u
   p19:gp0
   d16
   DELTA2
   (center (p22 ph6):f3 (p2 ph1):f1)
   DELTA2
   4u
   p19:gp0
   d16

```

```

(p1 ph9)
4u
p19:gp4
d16
(p21 ph4):f3
if "d0>p8*0.5"{
  "d10=d0-p8*0.5"
  d10
  (center (p8:sp13 ph0):f2 (p2 ph8):f1)
  d10
}
else{
d0
(p2 ph8)
d0
}
(p21 ph11):f3
4u
p19:gp4
d16
(p1 ph10)
8u
p16:gp2
d16 p10:f1
DELTA
(p11:sp1 ph2:r):f1
4u
4u p11:f1
(center (p2 ph0) (p22 ph0):f3 )
4u p10:f1
(p11:sp1 ph2:r):f1
4u
p16:gp2
d16
DELTA1
4u BLKGRAD
4u p116:f3

goscnp ph31 cpd3:f3

3m do:f3
3m st ivd
lo to 3 times nbl

3m ipp4 ipp5 ipp8 ipp11 ipp9 ipp10 ipp31
lo to 4 times ns

d1 mc #0 to 4
  F1QF()
  F2PH(calph(ph4, +90) & calph(ph6, +90) & exec(rppall), caldel(d0, +in0))
exit

ph0=0
ph1=1
ph2=2
ph3=3

ph4=0 2
ph5=0 0 0 0 2 2 2 2
ph6=0
ph8=1 1 3 3
ph11=(4) {0}*16 {2}*16
ph9=(4) {1}*8 {3}*8

```

```

ph10=2 2 2 2 0 0 0 0
ph31=(4) {0 2 0 2 0 2 0 2 2 0 2 0 2 0 2 0}^2

;td1: number of delays in vd-list
;td2: number of experiments in F2

;p11 : f1 channel - power level for pulse (default)
;p13 : f3 channel - power level for pulse (default)
;p116: f3 channel - power level for CPD/BB decoupling
;p1 : f1 channel - 90 degree high power pulse
;p8 : f2 channel - carbon decoupling pulse
;p11 : 1500u - 90 degree
;spnam1: Sinc1.1000 - shaped pulse 90 degree Watergate
;sp1: Sinc1.1000 pulse
;p16: 0.9m gradient pulse of Watergate
;p19: 0.5m gradient pulse
;p21: f3 channel - 90 degree high power pulse
;p22: f3 channel - 180 degree high power pulse
;p24: (240us) f3 channel - 180 degree decoupling train
;p10: 120 dB

;vd : variable delay, taken from vd-list
;NBL: = td1

;d0 : incremented delay (2D) [3 usec]
;d1 : relaxation delay; 1-5 * T1
;d10: computed internally for C-decoupling
;d11: delay for disk I/O [30 msec]
;d12: delay for power switching [20 usec]
;d16: delay for homospoil/gradient recovery
;cnst10: 1 for 90/-180, 2 for 180/-360 phase correction
;d26 : 1/(4J)NH (refocussing INEPT)
;in0: 1/(2 * SW(X)) = DW(X)
;nd0: 2
;NS: 4 * n
;DS: 16
;td1: number of experiments
;FnMODE: States-TPPI, TPPI, States or QSEC
;cpd3: decoupling according to sequence defined by cpdprg3
;pcpd3: f3 channel - 90 degree pulse for decoupling sequence

;for z-only gradients:
;gpz0: 15%
;gpz1: 8% (suppression of radiation damping)
;gpz2: 25% (Watergate)
;gpz3: 35% (strong gradient)
;gpz4: 50%
;gpz21: 80% (WET gradient 1)
;gpz22: 40% (WET gradient 2)
;gpz23: 20% (WET gradient 3)
;gpz24: 10% (WET gradient 4)

;use gradient files:
;gpnam0: SMSQ10.100
;gpnam1: SMSQ10.100
;gpnam2: SMSQ10.100
;gpnam3: SMSQ10.100
;gpnam4: SMSQ10.100
;gpnam21: SMSQ10.100
;gpnam22: SMSQ10.100
;gpnam23: SMSQ10.100
;gpnam24: SMSQ10.100

```

Brucker pulse program for  $R_2(^1\text{H})$  pseudo-4D interleaved HNCO relaxation experiment (see Figure 6 of main text)

```
;hncogpwgps4dr2.ho
;avance-version (12/01/11)
;HNCO-R2(1H)
;3D sequence with psuedo R2 relaxation dimension
;  inverse correlation for triple resonance using multiple
;    inept transfer steps
;
;    F1(H, TAU(R2,t1)) -> F3(N) -> F2(C=O,t2) -> F3(N,t3) -> ; F1(H,t4)
;
;on/off resonance Ca and C=O pulses using shaped pulse
;phase sensitive (t1)
;using constant time in t2
;relaxation delays controlled by vdlist with pseudo-4D
;interleaved acquisition
;
;$DIM=4D

prosol relations=<triple>

#include <Avance.incl>
#include <Grad.incl>
#include <Delay.incl>

;;;;;;;;Gradient definitions;;;;;;;;
"p16=600u"
"p19=900u"
"d16=200u"

;;;;;;;;High power pulse lengths ;;;;
"p2=p1*2"
"p4=p3*2"
"p22=p21*2"
"d11=30m"

;;;;;;;;;; J-coupling parameters ;;;;;;;;;;
;d21: 1/(2J(NH) [5.5 msec]
;d23: 1/(4J(NCO) [12 msec]
;d26: 1/(4J'(NH) [2.3 msec]
"d21=5.5m"
"d23=12m"
"d26=2.3m"

;;;;;;;;;;;;;Shaped pulse calculations;;;;;;;;;;;;;

;H2O Sinc Excitation/De-excitation
;spnam5 : Sinc.1000
"p5=1m"
"spw5= plw1*pow((90/(0.5889*p5))/(90/p1),2)"
"spoff5=0"

;H2O Sinc Inversion/Flipback
;spnam6 : Sinc.1000
"p6=1m"
"spw6= plw1*pow((180/(0.5889*p6))/(90/p1),2)"
"spoff6=0"
;H2O Square Excitation/Flipback
"p7=60u"
"plw19=plw1*pow((90/(1.0*p7))/(90/p1),2)"
```

```

;Carbon offsets
"cnst21=(sfo2-bf2)*1000000/bf2"
;cnst21 : Carbon carrier ppm
"cnst22= cnst21 + (173.7-cnst21) "
;cnst22 : CO frequency ppm
"cnst23= cnst21 + (53.7 -cnst21) "
;cnst23 : CA frequency ppm

;CO Q3 Inversion 15N INEPT
;spnam11 : Q3.1000
"p11 = 3.41/(60.0*bf2/1000000)"
"spw11= plw2*pow((180/(0.1515*p11))/(90/p3),2)"
"spoff11= (cnst22-cnst21)*bf2/1000000"
"spoal11=0.5"
;CO Q5 Excitation
;spnam12 : Q5.1000
"p12 = 6.18/(60.0*bf2/1000000)"
"spw12= plw2*pow((90/(0.0545*p12))/(90/p3),2)"
"spoff12= (cnst22-cnst21)*bf2/1000000"
"spoal12=1"
;CA inversion with Bloch-Seigert zero phase on CO
;spnam13 : Squal00.400
"p13 = (180/360)*pow(3,0.5)/(abs(cnst23-cnst21)*bf2/1000000)"
"spw13= plw2*pow((180/p13)/(90/p3),2)"
"spoff13= (cnst23-cnst21)*bf2/1000000"
"spoal13=0.5"
;CO Q5 Flipback
;spnam14 : Q5tr.1000
"p14 = 6.18/(60.0*bf2/1000000)"
"spw14= plw2*pow((90/(0.0545*p14))/(90/p3),2)"
"spoff14= (cnst22-cnst21)*bf2/1000000"
"spoal14=0"
;CA Q3 Inversion
;spnam15 : Q3.1000
"p15 = 3.41/(100.0*bf2/1000000)"
"spw15= plw2*pow((180/(0.1515*p15))/(90/p3),2)"
"spoff15= (cnst23-cnst21)*bf2/1000000"
"spoal15=0.5"

;;;;;;;;;; Evolution increments ;;;;;;;;;;
"in0=inf2/2"
"in10=inf1/4"
"in29=inf1/4"
"in30=inf1/4"

;;;;;;;;;; CO evolution delay ;;;;;;;;;;
"d0=in0*0.5*cnst10-0.5*larger(p13,p22)"
;cnst10 : if 1: phase correction 90/-180

;;;;;;;;;; N constant-time evolution delays ;;;;;;;;;;
"d10=d23/2-p11/2"
"d29=d23/2-p15/2-p7-d21-4u"
"d30=d23/2-p15/2"

;;;;;;;;;; Reverse INEPT delays ;;;;;;;;;;
"DELTA1=d23-d21-p7"
"DELTA2=d26-p16-d16-p6-12u"

;;;;;;;;;; INEPT+relaxation delays;;;;;;;;;
"DELTA4=d26-p16-d16-4u"
"DELTA5=d26-4u-8u-4u-p6-4u-p16-d16"

```

```

define list<delay> R2Delay = <$VDLIST>

1 d11 ze
  d11 pl16:f3
2 d11
3 d11 do:f3
4 d11
5 d1 pl1:f1 pl3:f3
  "TAU = R2Delay*0.25 + 2u"
  4u UNBLKGRAD
  (p21 ph0):f3
  p19:gp7 ;Kill1 nitrogen magnetisation
  1m pl0:f1
  (p5:sp5 ph4):f1 ;Water selective excitation
  8u
  16u pl1:f1
  8u
  ;;;;;;;;; Start Relaxation Block ;;;;;;;;;
  (p1 ph5):f1
  4u
  p16:gp4
  d16
  DELTA4
  TAU
  (p22 ph0):f3
  TAU
  (p2 ph0):f1
  4u
  8u pl0:f1
  4u
  (p6:sp6 ph2)
  4u
  p16:gp4
  d16
  DELTA5
  4u pl1:f1
  TAU
  (p22 ph0):f3
  TAU
  (p1 ph1):f1
  ;;;;;;;;; End Relaxation Block ;;;;;;;;;

  4u pl0:f1
  (p5:sp5 ph3:r):f1
  4u
  p16:gp1
  d16

  (p21 ph0):f3
  d21 pl19:f1
  (p7 ph1):f1
  DELTA1 cpds1:f1 ph0
  (center (pl1:sp11 ph0):f2 (p22 ph0):f3 )
  d23
  (p21 ph0):f3

  ;;;;;;;;; Start CO evolution ;;;;;;;;;
  (p12:sp12 ph6):f2;
  d0
  (center (p13:sp13 ph0):f2 (p22 ph0):f3 );
  d0

```

```

(p14:sp14 ph0):f2;
;;;;;;;;; End CO evolution ;;;;;;;;;;

4u do:f1
(p7 ph3):f1
4u
p16:gp2
d16
(p7 ph1):f1
20u cpds1:f1 ph0

;;;;;;;;;; Start N evolution ;;;;;;;;;;
(p21 ph7):f3
d30
(p15:sp15 ph0):f2;
d30
(center (p11:sp11 ph0):f2 (p22 ph0):f3 );
d10
(p15:sp15 ph0):f2;
d29
4u do:f1
(p7 ph3):f1
;;;;;;;;; End N evolution ;;;;;;;;;;
d21
(p21 ph0):f3

p16:gp3
d16 pl0:f1
(p5:sp5 ph2):f1;
4u
4u pl1:f1

(p1 ph0)
4u
p16:gp4
d16
DELTA2 pl0:f1
(p5:sp5 ph2):f1;
4u
4u pl1:f1
(center (p2 ph0) (p22 ph0):f3 )
4u pl0:f1
(p5:sp5 ph2):f1;
4u
DELTA2
p16:gp4
d16 pl16:f3
4u BLKGRAD

go=3 ph31 cpd3:f3
3m do:f3 zd

d11 st R2Delay.inc
lo to 4 times nbl

d11 st0 R2Delay.res
d11 mc #0 to 5
  F3QF()
  F2PH(calph(ph6, +90), caldel(d0, +in0))
  F1PH(calph(ph7, +90), caldel(d10, +in10) & caldel(d29, +in29) &
caldel(d30, -in30))
exit

```



```

ph0=0
ph1=1
ph2=2
ph3=3
ph4=2 2 2 2 0 0 0 0
ph5=0 0 0 0 2 2 2 2
ph6=0 2
ph7=0 0 2 2
ph31=0 2 2 0 2 0 0 2

;p10 : 0W
;p11 : f1 channel - power level for pulse (default)
;p13 : f3 channel - power level for pulse (default)
;p116: f3 channel - power level for CPD/BB decoupling
;p119: f1 channel - power level for CPD/BB decoupling
;p1 : f1 channel - 90 degree high power pulse
;p2 : f1 channel - 180 degree high power pulse
;p11: f1 channel - 90 degree shaped pulse [2 msec]
;p13: f2 channel - 90 degree shaped pulse
;p16: homospoil/gradient pulse [1 msec]
;p21: f3 channel - 90 degree high power pulse
;p22: f3 channel - 180 degree high power pulse
;d0 : incremented delay (F1 in 3D) [3 usec]
;d1 : relaxation delay; 1-5 * T1
;d10: incremented delay (F2 in 3D) = d23/2-p14/2
;d11: delay for disk I/O [30 msec]
;d13: short delay [4 usec]
;d16: delay for homospoil/gradient recovery
;d29: incremented delay (F2 in 3D) = d23/2-p14/2-p26-d21-4u
;d30: decremented delay (F2 in 3D) = d23/2-p14/2
;inf1: 1/SW(CO) = 2 * DW(CO)
;inf2: 1/SW(N) = 2 * DW(N)
;in0: 1/(2 * SW(CO)) = DW(CO)
;nd0: 2
;in10: 1/(4 * SW(N)) = (1/2) DW(N)
;nd10: 4
;in29: = in10
;in30: = in10
;ns: 8 * n
;ds: >= 16
;td1: number of experiments in F1
;td2: number of experiments in F2 td2 max = 2 * d30 / in30
;FnMODE: States-TPPI (or TPPI) in F2
;FnMODE: States-TPPI (or TPPI) in F3
;cpds1: decoupling according to sequence defined by cpdprg1
;cpd3: decoupling according to sequence defined by cpdprg3
;pcpd1: f1 channel - 90 degree pulse for decoupling sequence
;pcpd3: f3 channel - 90 degree pulse for decoupling sequence

;use gradient files:
;gpnam1: SMSQ10.100
;gpnam2: SMSQ10.100
;gpnam3: SMSQ10.100
;gpnam4: SMSQ10.100
;gpnam5: SMSQ10.100

```

## References

- (1) Chazin, W. J.; Kördel, J.; Thulin, E.; Hofmann, T.; Drakenberg, T.; Forsen, S. Identification of an isoaspartyl linkage formed upon deamidation of bovine calbindin D<sub>9k</sub> and structural characterization by 2D proton NMR. *Biochemistry* **1989**, *28*, 8646–8653.
- (2) Svensson, L. A.; Thulin, E.; Forsén, S. Proline *cis-trans* isomers in calbindin D<sub>9k</sub> observed by X-ray crystallography. *J. Mol. Biol.* **1992**, *223*, 601–606.
- (3) Kördel, J.; Pearlman, D. A.; Chazin, W. J. Protein solution structure calculations in solution: solvated molecular dynamics refinement of calbindin D<sub>9k</sub>. *J. Biomol. NMR* **1997**, *10*, 231–243.
- (4) Neylon, C.; Brown, S. E.; Kralicek, A. V.; Miles, C. S.; Love, C. A.; Dixon, N. E. Interaction of the *Escherichia coli* replication terminator protein (Tus) with DNA: a model derived from DNA-binding studies of mutant proteins by surface plasmon resonance. *Biochemistry* **2000**, *39*, 11989–11999.
- (5) Allegrozzi, M.; Bertini, I.; Janik, M. B. L.; Lee, Y.-M.; Liu, G.; Luchinat, C. Lanthanide-induced pseudocontact shifts for solution structure refinements of macromolecules in shells up to 40 Å from the metal ion. *J. Am. Chem. Soc.* **2000**, *122*, 4154–4161.
- (6) Hitchman, A. J. W.; Kerr, M.-K.; Harrison, J. E. The purification of pig vitamin D-induced intestinal calcium binding protein. *Arch. Biochem. Biophys.* **1973**, *155*, 221–222.
- (7) Oktaviani, N. A.; Otten, R.; Dijkstra, K.; Scheek, R. M.; Thulin, E.; Akke, M.; Mulder, F. A. A. 100% complete assignment of non-labile <sup>1</sup>H, <sup>13</sup>C, and <sup>15</sup>N signals for calcium-loaded calbindin D<sub>9k</sub> P43G. *Biomol. NMR Assign.* **2011**, *5*, 79–84.
- (8) Schmitz, C.; Stanton-Cook, M. J.; Su, X.-C.; Otting, G.; Huber, T. Numbat: An interactive software tool for fitting Δχ-tensors to molecular coordinates using pseudocontact shifts. *J. Biomol. NMR* **2008**, *41*, 179–189.
- (9) Bleaney, B. Nuclear magnetic resonance shifts in solution due to lanthanide ions. *J. Magn. Reson.* **1972**, *8*, 91–100.
- (10) Iwahara, J.; Tang, C.; Marius Clore, G. Practical aspects of <sup>1</sup>H transverse paramagnetic relaxation enhancement measurements on macromolecules. *J. Magn. Reson.* **2007**, *184*, 185–195.

## **Appendix C**

# **Protein NMR Resonance Assignment without Spectral Analysis: 5D SOlid-State Automated Projection Spectroscopy (SO-APSY)**



Australian  
National  
University

## Statement of Contribution

This thesis is submitted as a Thesis by Compilation in accordance with:

[https://policies.anu.edu.au/ppl/document/ANUP\\_003405](https://policies.anu.edu.au/ppl/document/ANUP_003405)

I declare that the research presented in this Thesis represents original work that I carried out during my candidature at the Australian National University, except for contributions to multi-author papers incorporated in the Thesis where my contributions are specified in this Statement of Contribution.

**Title:** Protein NMR resonance assignment without spectral analysis: 5D SOLid-state Automated Projection Spectroscopy (SO-APSY)

**Authors:** Henry W. Orton, Jan Stanek, Tobias Schubeis, Dylan Foucaudeau, Claire Ollier, Adrian W. Draney, Tanguy Le Marchand, Diane Cala-De Paepe, Isabella C. Felli, Roberta Pierattelli, Sebastian Hiller, Wolfgang Bermel and Guido Pintacuda

**Publication Outlet:** Angewandte Chemie International Edition

**Current status of paper:** Published

**Contribution:** I programmed the 5D pulse sequence and helped to setup the experimental acquisition for two of the three samples. I processed the spectra and assigned the chemical shifts. I wrote the draft for the manuscript and made figures 2, 3 and 5

Henry William Orton

CANDIDATE

SIGNATURE

10/02/2020

DATE

### Endorsed

Guido Pintacuda

SENIOR AUTHOUR

SIGNATURE

11.2.20

DATE

Gottfried Otting

PRIMARY SUPERVISOR

SIGNATURE

10/02/2020

DATE

Luke Connell

DELEGATED AUTHORITY

SIGNATURE

11/02/2020

DATE

# Protein NMR Resonance Assignment without Spectral Analysis: 5D SOLid-State Automated Projection SpectroscopY (SO-APSY)

Henry W. Orton<sup>+</sup>, Jan Stanek<sup>+</sup>, Tobias Schubeis, Dylan Foucaudeau, Claire Ollier, Adrian W. Draney, Tanguy Le Marchand, Diane Cala-De Paepe, Isabella C. Felli, Roberta Pierattelli, Sebastian Hiller, Wolfgang Bermel, and Guido Pintacuda\*

**Abstract:** Narrow proton signals, high sensitivity, and efficient coherence transfers provided by fast magic-angle spinning at high magnetic fields make automated projection spectroscopy feasible for the solid-state NMR analysis of proteins. We present the first ultrahigh dimensional implementation of this approach, where 5D peak lists are reconstructed from a number of 2D projections for protein samples of different molecular sizes and aggregation states, which show limited dispersion of chemical shifts or inhomogeneous broadenings. The resulting datasets are particularly suitable to automated analysis and yield rapid and unbiased assignments of backbone resonances.

One of the major bottlenecks for the widespread use of NMR spectroscopy in structural biology is the difficulty to automate data acquisition and analysis of complex protein spectra. Specific information on individual nuclei is typically achieved by the combination of several 3D spectra, which correlate shifts of sequential or proximal <sup>1</sup>H, <sup>15</sup>N, and <sup>13</sup>C nuclei.<sup>[1]</sup> Protein spectra with higher (> 3) dimensionality may yield multiple nuclear correlations within fewer experiments

and with increased resolution, thereby simplifying resonance assignment and, thus, expanding the size limits of the proteins amenable to site-specific NMR studies.<sup>[2]</sup> The acquisition of such spectra, however, requires alternative approaches to sample the indirect time space, since regular sampling leads to impractical experimental durations and data matrices. Randomized non-uniform sampling (NUS)<sup>[3]</sup> is a sparse sampling technique proposed to solve the first problem. NUS aims to reconstruct a high-dimensional spectrum from data acquired at randomly chosen time points, but it requires particular expertise in processing and analysis. However, the issue of the size of the data matrix remains. Automated projection spectroscopy (APSY)<sup>[4]</sup> is an alternative tool which allows direct inference of a high-dimensional peak list from a number of lower order projection spectra (typically 2D). APSY is an elegant concept which allows spectral analysis to be by-passed and delivers an output that directly contains the positions of all the resonances. As such, it is, in principle, amenable to widespread application, even by inexperienced spectroscopists.

APSY in solution NMR spectroscopy has suffered from a major drawback: the sensitivity required to implement APSY can only be provided by globular proteins with molecular sizes smaller than about 20 kDa (fast tumbling)<sup>[5]</sup> or by intrinsically disordered proteins.<sup>[4b,6]</sup> Figure 1 illustrates the rapid loss of sensitivity with increasing molecular size for a 5D experiment on a protonated and highly deuterated molecule in solution. This limitation is removed in proton-detected solid-state NMR with fast magic-angle spinning (MAS). Narrow proton signals, high sensitivity, and efficient coherence transfers can be obtained at high magnetic fields, independent of the molecular mass, at 60 kHz MAS with deuterated proteins, reprotoated at the exchangeable sites,<sup>[7]</sup> and at 100 kHz MAS and above with fully protonated samples.<sup>[8]</sup>

As illustrated in Figure 1, the efficiency of hyperdimensional experiments with solids outperforms the solution case above 12 and 33 kDa for a protonated and highly deuterated protein, respectively. As demonstrated below, this makes five-dimensional correlations by SOLid-state APSY (SO-APSY) feasible on microcrystalline and fibrillar proteins, which in turn allows the expeditious automated assignment of their backbone resonances.

We have developed a 5D experiment (H)NCOCANH (Figure 2a) which transfers coherences from each amide proton to the amide proton of the previous residue. This unidirectional coherence pathway is ensured by two cross-polarization (CP) steps between the <sup>1</sup>H and <sup>15</sup>N nuclei, two

[\*] Dr. J. Stanek,<sup>[†]</sup> Dr. T. Schubeis, D. Foucaudeau, C. Ollier, Dr. A. W. Draney, Dr. T. Le Marchand, Dr. D. Cala-De Paepe, Dr. G. Pintacuda  
Centre de Résonance Magnétique Nucléaire à Très Hauts Champs (FRE 2034 CNRS, UCBL, ENS Lyon)  
Université de Lyon  
69100 Villeurbanne (France)  
E-mail: guido.pintacuda@ens-lyon.fr

H. W. Orton<sup>[†]</sup>  
Research School of Chemistry  
Australian National University  
Canberra, ACT 2601 (Australia)

Dr. J. Stanek<sup>[†]</sup>  
Faculty of Chemistry, University of Warsaw  
02089 Warsaw (Poland)

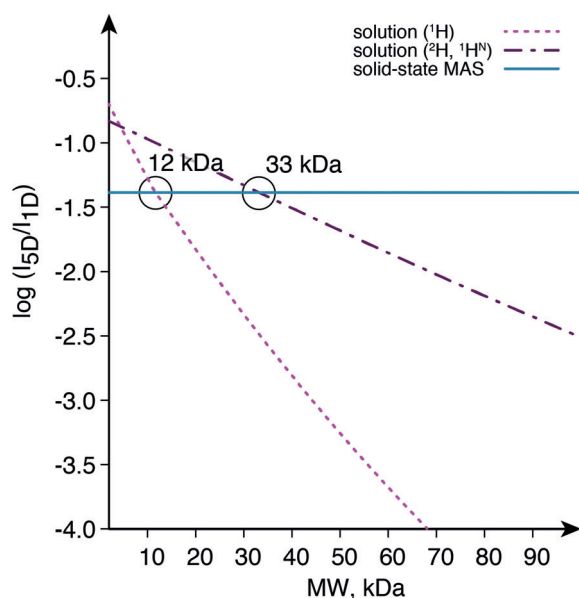
Prof. Dr. I. C. Felli, Prof. Dr. R. Pierattelli  
CERM and Department of Chemistry, University of Florence  
50019 Sesto Fiorentino (Italy)

Prof. Dr. S. Hiller  
Biozentrum, University of Basel  
4056 Basel (Switzerland)

Dr. W. Bermel  
Bruker BioSpin GmbH  
Silberstreifen, 76287 Rheinstetten (Germany)

[†] These authors contributed equally to this work.

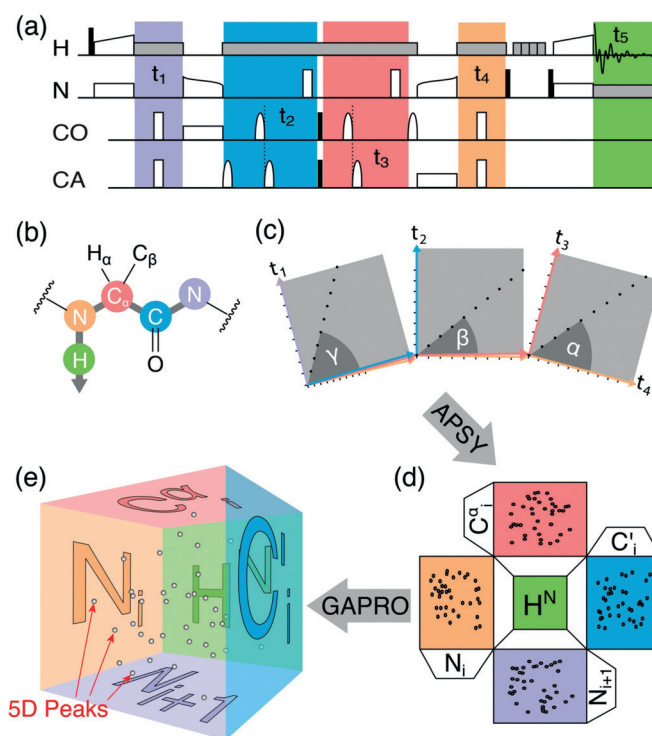
Supporting information and the ORCID identification numbers for some of the authors of this article can be found under:  
<https://doi.org/10.1002/anie.201912211>.



**Figure 1.** Sensitivity of hyperdimensional NMR spectroscopy plotted as a function of molecular weight in solution and in the solid state with respect to 1D  $^1\text{H}$  spectra. The curves show the calculated efficiency of a 5D experiment linking amide protons for a protonated molecule in solution (pink dotted curve), of the TROSY-based variant on a highly deuterated molecule in solution (brown dashed curve), and of the analogous experiment in the solid state at fast MAS (cyan solid curve). Calculations assumed scalar transfers in solution, with transverse relaxation modeled using Redfield theory, and a mix of scalar and dipolar transfers in solids (see the Supporting Information for details).

selective CP steps between bonded  $^{15}\text{N}$  and  $^{13}\text{C}\alpha/^{13}\text{C}'$  nuclei, and one  $^{13}\text{C}\alpha\text{--}^{13}\text{C}'$  scalar transfer. In contrast to previous implementations,<sup>[9]</sup> this experiment measures the chemical shift of all the nuclei ( $^{13}\text{C}\alpha$ ,  $^{13}\text{C}'$ ,  $^{15}\text{N}$ ) along the protein backbone. The experiment generates a 5D dataset where each peak correlates the chemical shift of five consecutive backbone nuclei (Figure 2b). The dataset can be acquired as a SO-APSY series by collecting a number of 2D projections, each characterized by projection angles  $\alpha$ ,  $\beta$ , and  $\gamma$  within the 4D indirect time space (Figure 2c–e).

The method was first established on a sample of uniformly  $^{15}\text{N}$ - and  $^{13}\text{C}$ -labeled and fully protonated microcrystalline GB1 protein (56 amino acids), packed in a 0.7 mm rotor and spun at 111 kHz on a 23.5 T NMR spectrometer ( $^1\text{H}$  frequency of 1 GHz). Under these conditions, the efficiency of the five-step transfer is ensured by long  $T_{1\rho}$  and  $T_2'$  coherence lifetimes, and the sensitivity amounts to about 20% with respect to a reference  $^1\text{H}\text{--}^{15}\text{N}$  correlation experiment. The SO-APSY series was acquired using a newly developed TopSpin module over about 1.5 days, at which point convergence was achieved with 24 projections (no additional 5D peaks were found by including the last projections). The first four projections, where the chemical shift of a single heteronucleus is evolved in the indirect dimension, are shown in Figure 3. In contrast to fully dimensional spectral data obtained with NUS, identifying resonances in APSY is straightforward and does not require prior knowledge.<sup>[10]</sup> The algorithm for Geometrical Analysis of PROjections (GAPRO)<sup>[4a]</sup> identifies robust peaks, as long as



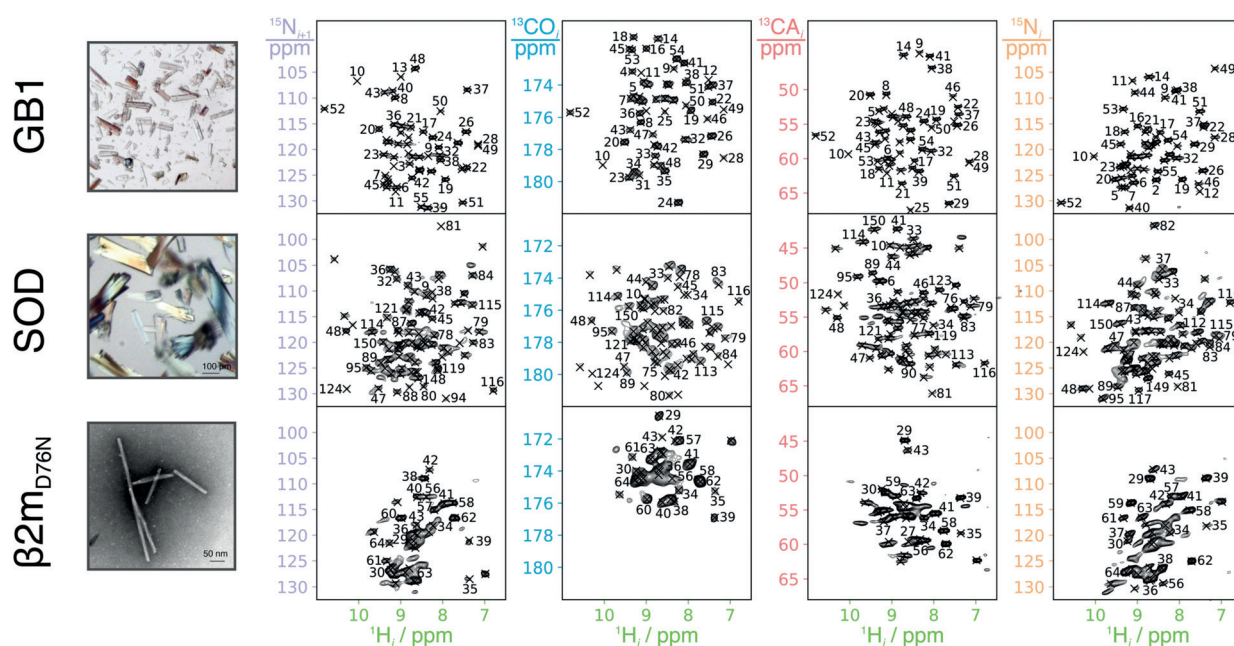
**Figure 2.** 5D ( $^1\text{H}$ )NCOCANH experiment and schematic representation of the acquisition and analysis of SO-APSY. a) Time diagram of the RF irradiation. High-power  $90^\circ$  and  $180^\circ$  pulses are represented by black and white rectangles, respectively, and selective pulses by bell shapes. Cross-polarization transfers are represented as open shapes on two parallel channels, and gray rectangles indicate heteronuclear decoupling. The five evolution periods of the chemical shift are highlighted as colored fields. Additional details can be found in the Supporting Information. b) Coherence transfer pathway. c) Time-domain sampling scheme for the SO-APSY experiment in the indirect 4D space, where each projection is characterized by three polar angles  $\alpha$ ,  $\beta$ , and  $\gamma$ . d) Frequency-domain orthogonal projections. e) Reconstruction of the peak positions in the 5D space by Geometrical Analysis of PROjections (GAPRO algorithm).<sup>[4a]</sup>

they are supported (resolved) in at least  $S_{\min}$  projections (here 7), and the on-the-fly analysis yielded a list with all 54 expected 5D peaks (Table S8).

The experiment was then carried out on two more challenging targets, a much larger microcrystalline protein (the 32 kDa Cu<sup>I</sup>-Zn loaded dimeric superoxide dismutase, SOD)<sup>[11]</sup> and a fibrillar aggregate (D76N  $\beta$ 2 microglobulin,  $\beta$ 2m<sub>D76N</sub>).<sup>[12]</sup> Both samples were uniformly  $^{13}\text{C}$ -,  $^{15}\text{N}$ -, and  $^2\text{H}$ -labeled (with 100% reprotonation of the labile amide protons) and packed in a 1.3 mm rotor spun at 60 kHz MAS on a 1 GHz spectrometer.

For SOD, which contains 153 residues, but several dynamic regions and proline residues, a maximum of 115 peaks can be expected in the 5D experiment. The SO-APSY series was accomplished in 32 projections over 3 days. Despite the significant degree of peak overlap in the first (orthogonal) projections (Figure 3, middle row), the tilted projections contributed a total of 95 peaks. For  $\beta$ 2m<sub>D76N</sub>, spectral overlap does not come from the protein size (only 62 residues S20–R81 are observed in solid samples), but from the limited dispersion of the chemical shifts and increased linewidths that





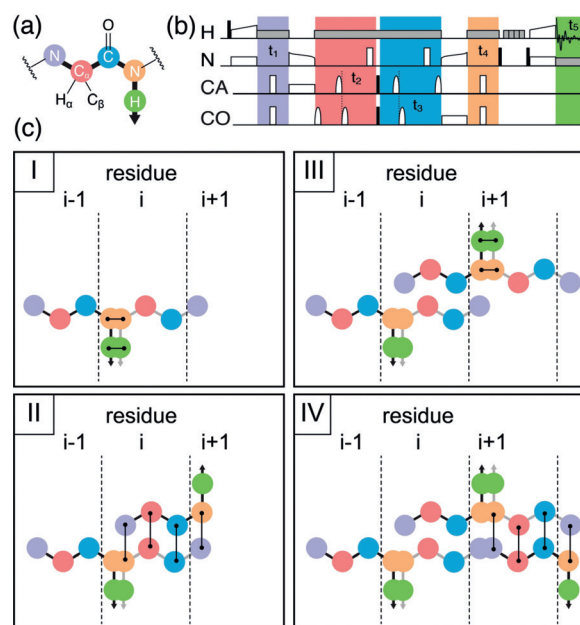
**Figure 3.** Orthogonal SO-APSY projections from 5D (H)NCOCANH experiment on microcrystalline GB1, SOD, and fibrils of  $\beta 2m_{D76N}$ . These projections exclusively evolve a single nucleus in the indirect dimension: from left to right, intra-residue ( $i$ )  $^{15}\text{N}$ ,  $^{13}\text{C}\alpha$ ,  $^{13}\text{C}'$ , and sequential ( $i+1$ )  $^{15}\text{N}$  with respect to the directly detected amide proton ( $i$ ) shifts. Crosses indicate experimental peaks identified by the GAPRO algorithm, and peak labels indicate residue assignments obtained from automated analysis (see below).

are characteristic of fibrillar samples. The SO-APSY series was acquired with 32 projections in about 2.5 days. GAPRO identified 46 correlations, thereby proving that consistent 5D peak coordinates can be determined even within regions of severe overlap in orthogonal 2D projections (Figure 3, bottom).

The 5D experiment has the particular advantage over conventional 3D triple-resonance spectra that a single peak encodes amide shifts from two neighboring residues. This alone allows sequential linking of amide resonances along the protein backbone by matching shared  $^{15}\text{N}$  frequencies. In cases where  $^{15}\text{N}$  resolution is limited or an amide proton is unobservable, the assignment of sequential residues can be accomplished with the acquisition of a second 5D experiment, (H)NCACONH. This performs a similar coherence transfer along the protein backbone, but in a reverse (i.e. forward) direction (Figure 4a), with comparable efficiency (Table S1). Such an experiment was indeed carried out as a SO-APSY series for the three samples and yielded an equivalent number of signals (Table S9). Peaks from both experiments overlay in the  $^{15}\text{N}(i)$ - $^{15}\text{N}(i+1)$  plane, and effectively connect consecutive  $^1\text{H}^{\text{N}}(i)$  and  $^1\text{H}^{\text{N}}(i+1)$  amide protons (two-way  $^{15}\text{N}$  linking).<sup>[9]</sup> A pair of forward and backward 5D data sets also correlate consecutive amide ( $i$  and  $i+1$ ) protons to pairs of  $^{13}\text{C}\alpha(i)$ - $^{13}\text{C}'(i)$  frequencies, effectively building quartets of shifts ( $^{15}\text{N}(i)$ ,  $^{13}\text{C}\alpha(i)$ ,  $^{13}\text{C}'(i)$ ,  $^{15}\text{N}(i+1)$ ) that unambiguously link amide protons at every second step of the assignment protocol (Figure 4c).

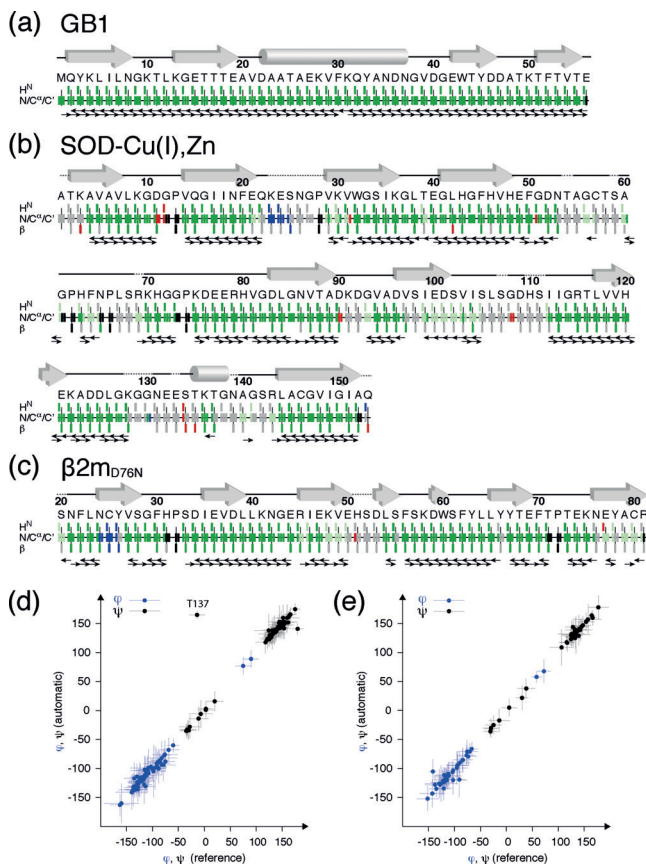
Five-dimensional peak lists are inconvenient for a conventional manual analysis, but their potential for sequential assignment can be optimally exploited with a powerful automated assignment algorithm. We employed the FLYA

software,<sup>[13]</sup> which is particularly flexible with respect to customized correlation schemes (see the Supporting Information for FLYA definitions, parameters, and input data). In



**Figure 4.** Sequential resonance assignment from a pair of 5D backward and forward experiments. a) Coherence transfer pathway of a forward (H)NCACONH experiment. b) Time diagram of RF irradiation, with the same conventions as in Figure 2a. c) Schematic representation of resonance assignment, obtained by matching either  $^{15}\text{N}$ ,  $^1\text{H}^{\text{N}}$  pairs (odd steps) or quartets of  $^{15}\text{N}(i)$ ,  $^{13}\text{C}\alpha(i)$ ,  $^{13}\text{C}'$ , and  $^{15}\text{N}(i+1)$  chemical shifts (even steps).

the case of GB1, a complete assignment of all backbone nuclei was obtained when two 5D peak lists from a pair of forward and backward SO-APSY experiments were provided. The schematic overview of the FLYA results, which compares the assigned peaks to a reference assignment,<sup>[8a]</sup> is shown in Figure 5 a.



**Figure 5.** Graphical summary of the automated assignments of microcrystalline GB1 (a), SOD (b), and fibrils of  $\beta 2m_{D76N}$  (c) by FLYA, superimposed with the amino acid sequence and a sketch of secondary structure elements. Assignments in agreement or incompatible with the reference are colored green and red, respectively. Assignments with low confidence levels in FLYA are colored light green or in gray, depending on whether they can or cannot be confirmed by manual inspection. Additional assignments found by FLYA are shown in dark blue. Black arrows represent all correlations observed in the two 5D experiments. Dotted lines connecting gray arrows ( $\beta$ -strands) and cylinders (helices) denote flexible regions where resonances are missing or sparse. d,e) Correlation of backbone dihedral angles  $\varphi$  and  $\psi$  predicted by TALOS+ based on the reference, and automatically obtained  $^1H^N$ ,  $^{15}N$ ,  $^{13}Ca$ ,  $^{13}C\beta$ , and  $^{13}C'$  chemical shifts for SOD (d) and  $\beta 2m_{D76N}$  (e).

In the case of SOD, the automated analysis by FLYA reveals 8 segments of robust (“strong”) assignments (Figure S3a), with 280 chemical shifts consistent with published data.<sup>[11]</sup> Assignments were not found in regions 23–27 and 129–140, known to be dynamic,<sup>[11]</sup> and signals were sparse in the regions 50–70 and 100–110 because of the presence of five proline residues. An almost complete assignment can be recovered for these regions (Figure 5b), if the 5D peak lists

are supplemented with information on residue type, as for example that contained in 3D  $C\beta$ -N-H correlations (in total 346 correct assignments). About 4 % of the chemical shifts were incorrectly assigned when compared to a reference assignment obtained by conventional proton-detected 3D spectra. This reflects an inherent uncertainty that arises during automated assignment, which must be considered when inferring firm conclusions on protein site-specific dynamics or interactions. FLYA analysis of 5D peak lists obtained for  $\beta 2m_{D76N}$  leads to three regions of reliable assignments at the fibril core (27–31, 34–45, 59–65), while many regions with reasonably complete observed correlations are assigned only tentatively (Figure S3b). We attribute this result to the significantly smaller distribution of chemical shifts in this sample and their decreased precision (increased line widths). Similar to SOD, this issue can be addressed by providing supplementary input for the automatic analysis. Notably, an additional unassigned 3D  $C\beta$ -N-H peak list extends the assignment to most of the fibril core, including regions 46–49, 56–58, 67–71, and 73–75 (Figure 5c), thereby giving rise to 166 correctly assigned chemical shifts and 5 misassignments (3 %).

Although a small number of incorrect assignments were identified within this study when compared to manually assigned 3D spectra, they do not form clusters spanning more than 1 or 2 residues and are frequently at the extremities of long tracts of assigned residues. In effect, these misassigned shifts do not affect the prediction of dihedral angles  $\varphi$  and  $\psi$ , which is the primary structural information encoded in  $^1H^N$ ,  $^{15}N$ ,  $^{13}Ca$ ,  $^{13}C\beta$ , and  $^{13}C'$  shifts. Indeed, when using the widely applied TALOS+ algorithm,<sup>[14]</sup> the dihedral angles obtained from the chemical shifts are virtually identical, with only 1 outlier ( $\psi$  for residue T137 of SOD) among 214 angles reliably predicted in both the reference and FLYA-derived data sets for SOD and  $\beta 2m_{D76N}$  (Figure 5d,e). Additionally, it has been shown that reliable structural modeling can be obtained with up to 10 % missing or incorrect proton chemical shifts provided that robust algorithms (such as CANDID<sup>[15]</sup>) are employed.<sup>[16]</sup> Therefore, we would expect 5D APSY of yet unassigned proteins in the solid state to offer appropriate data for subsequent modeling of structure and dynamics.

In summary, we have presented the first example of ultrahigh dimensional (> 3) automated projection spectroscopy in solid-state NMR spectroscopy (SO-APSY) and demonstrated its power with respect to the automatic assignment of resonances of samples of different molecular sizes and aggregation states. The approach was illustrated by the design of two new 5D pulse sequences that correlate backbone nuclei with amide protons. A module to acquire 2D projections and to reconstruct 5D peak lists was designed within the widespread software TopSpin, and is now readily accessible to the NMR community. SO-APSY tolerates high peak overlap and yields data suitable for automatic analysis, thus accelerating the identification of sequential correlations. We anticipate that in a context where even faster MAS rates and longer coherence lifetimes are available, SO-APSY techniques will become key for rapid and simple spectroscopic analysis, thus broadening the range of targets amenable to NMR analysis.



## Acknowledgements

The work was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC-2015-CoG GA 648974), by the CNRS (IR-RMN FR3050), and by the EU-project iNext (GA 653706). H.W.O. was supported by the Westpac Foundation with a Future Leaders Scholarship, and J.S. by the EC's REA with a MSCA fellowship (GA 661799). The project was co-financed by the Polish National Agency for Academic Exchange (contract No PPN/PPO/2018/1/00098).

## Conflict of interest

The authors declare no conflict of interest.

**Keywords:** automation · NMR spectroscopy · projection spectroscopy · proton detection · solid-state structures

**How to cite:** *Angew. Chem. Int. Ed.* **2020**, *59*, 2380–2384  
*Angew. Chem.* **2020**, *132*, 2400–2405

- [1] M. Sattler, J. Schleucher, C. Griesinger, *Prog. Nucl. Magn. Reson. Spectrosc.* **1999**, *34*, 93–158.
- [2] a) M. Nowakowski, S. Saxena, J. Stanek, S. Žerko, W. Koźmiński, *Prog. Nucl. Magn. Reson. Spectrosc.* **2015**, *90–91*, 49–73; b) W. T. Franks, K. D. Kloepper, B. J. Wylie, C. M. Rienstra, *J. Biomol. NMR* **2007**, *39*, 107–131; c) R. Linser, B. Bardiaux, V. Higman, U. Fink, B. Reif, *J. Am. Chem. Soc.* **2011**, *133*, 5905–5912; d) M. Huber, A. Böckmann, S. Hiller, B. H. Meier, *Phys. Chem. Chem. Phys.* **2012**, *14*, 5239–5246; e) B. J. Wylie, M. P. Bhate, A. E. McDermott, *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 185–190; f) S. Xiang, V. Chevelkov, S. Becker, A. Lange, *J. Biomol. NMR* **2014**, *60*, 85–90.
- [3] M. Mobli, J. C. Hoch, *Prog. Nucl. Magn. Reson. Spectrosc.* **2014**, *83*, 21–41.
- [4] a) S. Hiller, F. Fiorito, K. Wüthrich, G. Wider, *Proc. Natl. Acad. Sci. USA* **2005**, *102*, 10876–10881; b) S. Hiller, C. Wasmer, G. Wider, K. Wüthrich, *J. Am. Chem. Soc.* **2007**, *129*, 10823–10828.
- [5] A. D. Gossert, S. Hiller, C. Fernández, *J. Am. Chem. Soc.* **2011**, *133*, 210–213.
- [6] a) R. L. Narayanan, U. H. Dürr, S. Bibow, J. Biernat, E. Mandelkow, M. Zweckstetter, *J. Am. Chem. Soc.* **2010**, *132*, 11906–11907; b) M. G. Murrall, M. Schiavina, V. Sainati, W. Bermel, R. Pierattelli, I. C. Felli, *J. Biomol. NMR* **2018**, *70*, 167–175.
- [7] a) J. R. Lewandowski, J. N. Dumez, Ü. Akbey, S. Lange, L. Emsley, H. Oschkinat, *J. Phys. Chem. Lett.* **2011**, *2*, 2205–2211; b) M. J. Knight, A. L. Webber, A. J. Pell, P. Guerry, E. Barbet-Massin, I. Bertini, I. C. Felli, L. Gonnelli, R. Pierattelli, L. Emsley, A. Lesage, T. Herrmann, G. Pintacuda, *Angew. Chem. Int. Ed.* **2011**, *50*, 11697–11701; *Angew. Chem.* **2011**, *123*, 11901–11905; c) E. Barbet-Massin, A. J. Pell, J. S. Retel, L. B. Andreas, K. Jaudzems, W. T. Franks, A. J. Nieuwkoop, M. Hiller, V. Higman, P. Guerry, A. Bertarello, M. J. Knight, M. Felletti, T. Le Marchand, S. Kotelovica, I. Akopjana, K. Tars, M. Stoppini, V. Bellotti, M. Bolognesi, S. Ricagno, J. J. Chou, R. G. Griffin, H. Oschkinat, A. Lesage, L. Emsley, T. Herrmann, G. Pintacuda, *J. Am. Chem. Soc.* **2014**, *136*, 12489–12497.
- [8] a) L. B. Andreas, K. Jaudzems, J. Stanek, D. Lalli, A. Bertarello, T. Le Marchand, D. C. De Paepe, S. Kotelovica, I. Akopjana, B. Knott, S. Wegner, F. Engelke, A. Lesage, L. Emsley, K. Tars, T. Herrmann, G. Pintacuda, *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 9187–9192; b) J. Stanek, L. B. Andreas, K. Jaudzems, D. Cala, D. Lalli, A. Bertarello, T. Schubeis, I. Akopjana, S. Kotelovica, K. Tars, A. Pica, S. Leone, D. Picone, Z. Q. Xu, N. E. Dixon, D. Martinez, M. Berbon, N. El Mammeri, A. Noubhani, S. Saupe, B. Habenstein, A. Loquet, G. Pintacuda, *Angew. Chem. Int. Ed.* **2016**, *55*, 15504–15509; *Angew. Chem.* **2016**, *128*, 15730–15735; c) J. Struppe, C. M. Quinn, M. Lu, M. Wang, G. Hou, X. Lu, J. Kraus, L. B. Andreas, J. Stanek, D. Lalli, A. Lesage, G. Pintacuda, W. Maas, A. M. Gronenborn, T. Polenova, *Solid State Nucl. Magn. Reson.* **2017**, *87*, 117–125.
- [9] a) S. Xiang, K. Grohe, P. Rovó, S. K. Vasa, K. Giller, S. Becker, R. Linser, *J. Biomol. NMR* **2015**, *62*, 303–311; b) L. B. Andreas, J. Stanek, T. Le Marchand, A. Bertarello, D. Cala-De Paepe, D. Lalli, M. Krejčíková, C. Doyen, C. Öster, B. Knott, S. Wegner, F. Engelke, I. C. Felli, R. Pierattelli, N. E. Dixon, L. Emsley, T. Herrmann, G. Pintacuda, *J. Biomol. NMR* **2015**, *62*, 253–261.
- [10] A. Klein, S. K. Vasa, R. Linser, *J. Biomol. NMR* **2018**, *72*, 163–170.
- [11] M. J. Knight, A. J. Pell, I. Bertini, I. C. Felli, L. Gonnelli, R. Pierattelli, T. Herrmann, L. Emsley, G. Pintacuda, *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 11095–11100.
- [12] T. Le Marchand, M. de Rosa, N. Salvi, B. M. Sala, L. B. Andreas, E. Barbet-Massin, P. Sormanni, A. Barbiroli, R. Porcari, C. Sousa Mota, D. de Sanctis, M. Bolognesi, L. Emsley, V. Bellotti, M. Blackledge, C. Camilloni, G. Pintacuda, S. Ricagno, *Nat. Commun.* **2018**, *9*, 1658.
- [13] E. Schmidt, P. Güntert, *J. Am. Chem. Soc.* **2012**, *134*, 12817–12829.
- [14] Y. Shen, F. Delaglio, G. Cornilescu, A. Bax, *J. Biomol. NMR* **2009**, *44*, 213–223.
- [15] T. Herrmann, P. Güntert, K. Wüthrich, *J. Mol. Biol.* **2002**, *319*, 209–227.
- [16] P. Güntert, *Prog. Nucl. Magn. Reson. Spectrosc.* **2003**, *43*, 105–125.

Manuscript received: September 26, 2019

Accepted manuscript online: October 28, 2019

Version of record online: December 19, 2019

## Supporting Information

### **Protein NMR Resonance Assignment without Spectral Analysis: 5D SOLid-State Automated Projection SpectroscopY (SO-APSY)**

*Henry W. Orton<sup>+</sup>, Jan Stanek<sup>+</sup>, Tobias Schubeis, Dylan Foucaudeau, Claire Ollier,  
Adrian W. Draney, Tanguy Le Marchand, Diane Cala-De Paepe, Isabella C. Felli,  
Roberta Pierattelli, Sebastian Hiller, Wolfgang Bermel, and Guido Pintacuda\**

anie\_201912211\_sm\_miscellaneous\_information.pdf

## Contents:

A. NMR sample preparation.....	2
B. Pulse sequence description .....	2
C. NMR data acquisition details.....	4
D. Amide proton and heteronuclear refocussed coherence lifetimes.....	7
E. Performance of interresidue amide proton transfer experiments.....	7
F. Fourier data processing and GAPRO parameters.....	12
G. 5D peak lists for GB1, SOD and $\theta 2m_{D76N}$ .....	13
H. Details of FLYA calculations .....	20
I. FLYA results for SOD and $\theta 2m_{D76N}$ with additional C $\beta$ information .....	22
J. Pulse program code for Bruker spectrometers .....	24
K. References.....	34

## A. NMR sample preparation

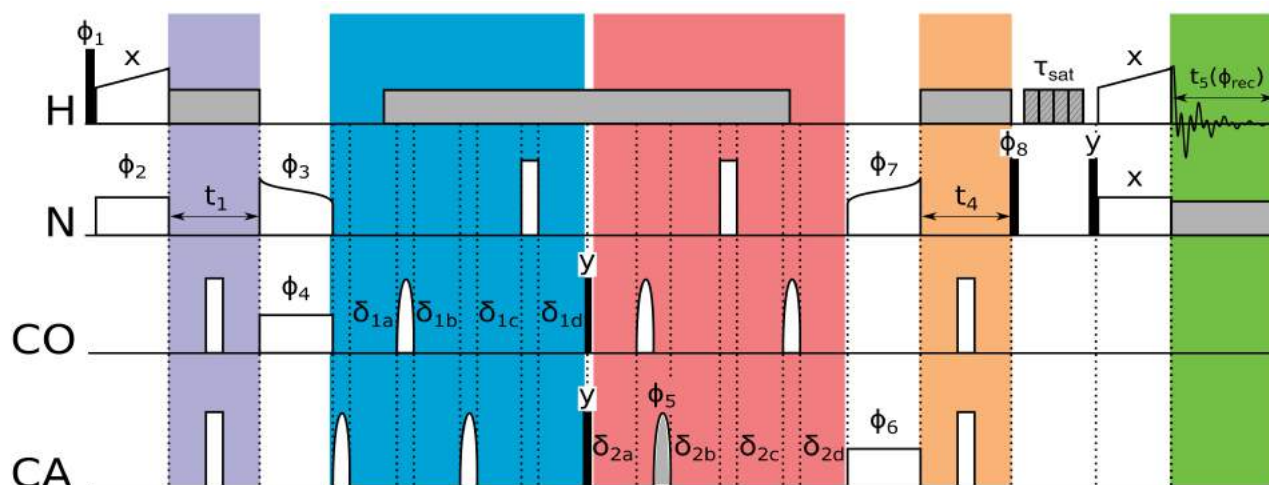
**GB1.** Uniformly  $^{13}\text{C}$ ,  $^{15}\text{N}$ -labelled GB1 was overexpressed, purified, dialysed against phosphate buffer and and precipitated to microcrystals in form A according to published protocol.<sup>1-2</sup> The crystals were packed into a 0.7 mm Bruker MAS rotor.

**Cu(I),Zn-SOD.** Uniformly  $^2\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ -labelled and completely back-exchanged human Cu<sup>I</sup>,Zn-loaded superoxide dismutase (SOD) was obtained as previously reported,<sup>3-4</sup> crystallised using the sitting drop method,<sup>5</sup> and packed into a 1.3 mm Bruker MAS rotor.

**$\beta 2\text{m}_{\text{D76N}}$ .** Expression of  $^2\text{H}$ ,  $^{13}\text{C}$ ,  $^{15}\text{N}$ -labelled  $\beta 2\text{m}_{\text{D76N}}$  was performed as described earlier, with purification steps, including refolding, performed in non-deuterated water to allow proton back-exchange at all amide sites.<sup>6</sup> A solution of purified (unlabeled)  $\beta 2\text{m}_{\text{D76N}}$  (100  $\mu\text{M}$   $\beta 2\text{m}$ , 25 mM sodium phosphate, pH 7.4) was then incubated at 37°C with vigorous shaking for 5 days, resulting in a mother generation of fibrils ( $G_0$ ).<sup>7</sup> A second generation  $G_1$  was obtained in the same conditions with the addition of 1.5% v/v of  $G_0$  and, in order to limit polymorphism, the same procedure was iterated for seven generations. Homogeneity of the samples was tested with transmission electron-microscopy and MAS NMR. The final sample ( $G_8$ ) was generated from triple-labelled  $\beta 2\text{m}_{\text{D76N}}$ .

Microcrystals (GB1 and SOD) and fibrils ( $\beta 2\text{m}_{\text{D76N}}$ ) were packed by ultracentrifugation at 165,000 $\times g$  at 12°C directly into the NMR rotor using a device provided by Giotto Biotech, similar to those described in literature.<sup>8-9</sup>

## B. Pulse sequence description



**Figure S1.** Pulse sequence for 5D (H)NCOCANH experiment. Heteronuclear cross-polarisation (CP) transfers are shown as wide open shapes simultaneously on two channels. RF strengths, durations and amplitude shapes used for CP transfers are summarised in Table S3. Heteronuclear decoupling is shown in grey. Swept-TPPM<sup>10-11</sup> or WALTZ-16<sup>12</sup>  $^1\text{H}$  decoupling with r.f. field of 10 kHz and pulse length of 25  $\mu\text{s}$  was used for GB1 and SOD/ $\beta 2\text{m}_{\text{D76N}}$ , respectively. WALTZ-16 with r.f. field of 10 kHz and pulse length of 25  $\mu\text{s}$  was used for  $^{15}\text{N}$  decoupling during acquisition. Open pulses represent a flip-angle of 180°, while solid pulses are 90°. Bell shapes on carbon channels represent C' or C $\alpha$  selective shaped pulses. All shaped pulses were Q3,<sup>13</sup> with the exception of the C $\alpha$  refocussing pulse (shown in grey), where a REBURP was used.<sup>14</sup> The shaped pulse duration and max. r.f. field strength were 144.3  $\mu\text{s}$  and 22.9 kHz ( $^{13}\text{C}'$ -selective Q3), 129.1  $\mu\text{s}$  and 25.6 kHz ( $^{13}\text{C}\alpha$  selective Q3), and 638.0  $\mu\text{s}$  and 9.82  $\mu\text{s}$  ( $^{13}\text{C}\alpha$  selective ReBURP). The MISSISSIPPI water suppression,<sup>15</sup> with duration  $\tau_{\text{sat}}$  of 150 (for GB1) or 100 ms (SOD,  $\beta 2\text{m}$ ) and r.f. strength of either 53 kHz (GB1), 34 kHz (SOD) or 39 kHz ( $\beta 2\text{m}$ ), is shown in hatched grey. The C $\alpha$ -C' coherence transfer delays are maximally exploited for the either constant-time or shared-time ("semi-constant time") chemical shift evolution of  $^{13}\text{C}'$

( $t_2$ ) and  $^{13}\text{C}\alpha$  ( $t_3$ ). The respective formulas for delays  $\delta_{1a}$ ,  $\delta_{1b}$ ,  $\delta_{1c}$ ,  $\delta_{1d}$  (or  $\delta_{2a}$ ,  $\delta_{2b}$ ,  $\delta_{2c}$ ,  $\delta_{2d}$ ) vary depending on whether  $t_2$  (or  $t_3$ ) exceeds the relaxation-optimised transfer delay  $2\delta_{\text{CA/CO}}^{\text{opt}} = \tan^{-1}(\pi J_{\text{T}_2}^{\text{CA/CO}} / (\pi J))$ , or the relaxation-independent transfer delay  $2\delta = (2^1 J_{\text{CA-CO}})^{-1}$ . They are determined as follows:

$$\delta_{1a} = \max(\delta_{\text{CO}}^{\text{opt}} + 2pw_{90}^{\text{N}} - \frac{1}{2} t_2, 0),$$

$$\delta_{1b} = \max(\frac{1}{2} t_2 - \delta, 0),$$

$$\delta_{1c} = \max(\delta_{\text{CO}}^{\text{opt}}, \min(\frac{1}{2} t_2, \delta), 0),$$

$$\delta_{1d} = \frac{1}{2} t_2,$$

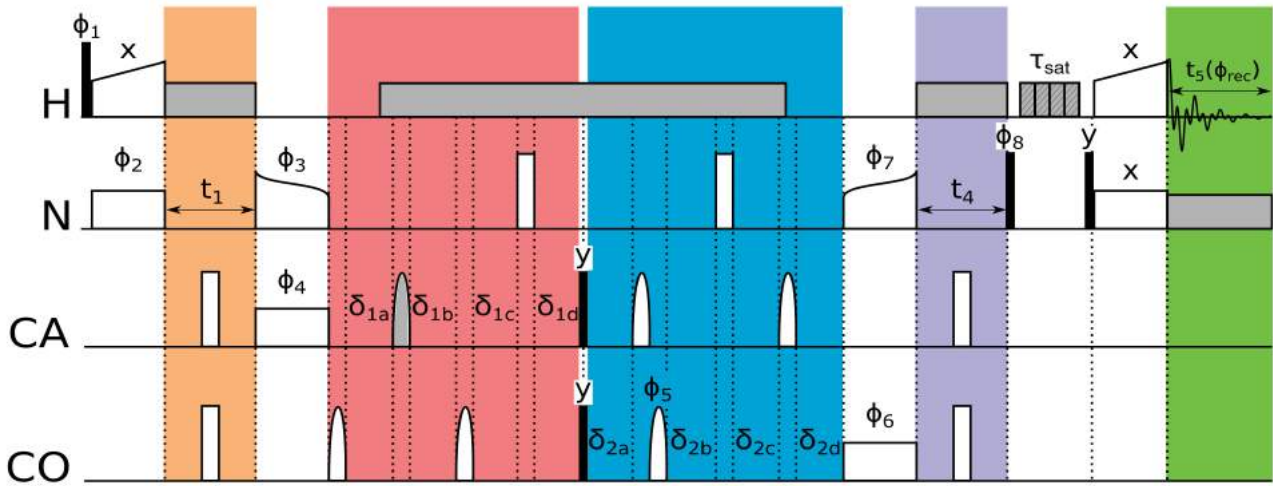
$$\delta_{2a} = \max(\delta_{\text{CA}}^{\text{opt}} + 2pw_{90}^{\text{N}} - \frac{1}{2} t_3, 0),$$

$$\delta_{2b} = \max(\delta_{\text{CA}}^{\text{opt}}, \frac{1}{2} t_3),$$

$$\delta_{2c} = \max(\frac{1}{2} t_3, \delta),$$

$$\delta_{2d} = \max(\frac{1}{2} t_3 - \delta_{\text{CA}}^{\text{opt}}, 0).$$

The coupling constant  $^1 J_{\text{CA-CO}}$  of 55 Hz was assumed, and the bulk  $^{13}\text{C}$  refocused coherence lifetimes are summarised in Table S5.  $^{15}\text{N}$  offset was set to 117.5 (for GB1), 115.6 (SOD) or 119.5 ( $\beta 2\text{M}_{\text{D76N}}$ ) ppm while  $^1\text{H}$  offset was set on resonance with  $\text{H}_2\text{O}$  line. The carbon carrier frequency was set to the centre of the  $\text{C}\alpha/\text{C}'$  spectrum at 114 ppm for the entire experiment. Shaped pulses and CP transfer offsets were set to -60 and +60 ppm for the  $\text{C}\alpha$  and  $\text{C}'$  channels respectively while the hard  $90^\circ$  pulse was applied at zero offset. Stepwise phase modulation was used to approximate the off-resonance effect of these pulses, and zero phase was enforced (aligned) at the end and at the beginning of the  $^{13}\text{C}$  pulse for  $^{15}\text{N} \rightarrow ^{13}\text{C}'$  and  $^{13}\text{C}\alpha \rightarrow ^{15}\text{N}$  CP transfers, respectively. A time-dependent phase increment was applied to carbon refocussing pulses to control the centre of the respective carbon dimension ( $\text{C}\alpha$  or  $\text{C}'$ ) and was calculated in degrees as  $180^\circ \Omega \cdot t$  where  $\Omega$  is the carbon offset from the carrier in Hz and  $t$  is the time increment. Re-equilibration delay of 1.0, 0.5 and 1.18 s was used for GB1, Cu(I),Zn-SOD and  $\beta 2\text{M}_{\text{D76N}}$ , respectively. The phases cycle was  $\phi_1 = \{y, -y\}$ ,  $\phi_2 = y$ ,  $\phi_3 = \{y, y, -y, -y\}$ ,  $\phi_4 = y$ ,  $\phi_5 = \{4(x), 4(-x)\}$ ,  $\phi_6 = y$ ,  $\phi_7 = x$ ,  $\phi_8 = \{8(y), 8(-y)\}$  and  $\phi_{\text{rec}} = \{x, -x, -x, x, -x, x, x, -x, -x, x, x, -x, x, -x, x, -x, x\}$ . Quadrature detection in the indirect dimensions was accomplished by incrementing  $\phi_2$  (in  $\omega_1$ ), decrementing  $\phi_4$  (in  $\omega_2$ ), decrementing  $\phi_6$  (in  $\omega_3$ ), or incrementing  $\phi_7$  (in  $\omega_4$ ), simultaneously with  $t_1$ ,  $t_2$ ,  $t_3$ , and  $t_4$  time incrementation.



**Figure S2.** Pulse sequence for 5D (H)NCACONH experiment. The description of Figure S1 remains valid with a few exceptions that follow: delays  $\delta_{1a}$  and  $\delta_{1c}$  are determined with the use of  $\delta_{\text{CA}}^{\text{opt}}$ , while delays  $\delta_{2a}$ ,  $\delta_{2b}$  and  $\delta_{2d}$  depend on  $\delta_{\text{CO}}^{\text{opt}}$ . The zero phase is enforced (aligned) at the end of  $^{13}\text{C}\alpha$ , and at the beginning of the  $^{13}\text{C}'$  pulse for the  $^{15}\text{N} \rightarrow ^{13}\text{C}\alpha$  and  $^{13}\text{C}' \rightarrow ^{15}\text{N}$  CP transfers, respectively.  $^{13}\text{C}\alpha$  and  $^{13}\text{C}'$  shaped pulses are exchanged to preserve their context, i.e. the  $^{13}\text{C}\alpha$  refocussing pulse (ReBURP) is performed between  $\delta_{1a}$  and  $\delta_{1b}$ , etc.

## C. NMR data acquisition details

Table S1: (H)NCACONH experiment acquisition parameters

Sample	Max. evolution time, ms in dimension					Spectral widths, ppm <sup>a</sup> in dimension					MAS rate, kHz	Scans	Efficiency <sup>e</sup>	Number of projec- tions	Time, h
	$\omega_1$ N	$\omega_2$ C $\alpha$	$\omega_3$ C'	$\omega_4$ N	$\omega_5$ H	$\omega_1$ N	$\omega_2$ C $\alpha$	$\omega_3$ C'	$\omega_4$ N	$\omega_5$ H					
GB1	18.1	10.0	13.9	18.1	20.5 <sup>b</sup>	30	25	12	30	100	110	32	0.21	24	40.5
SOD	13.1	9.1	9.9	13.1	24.8 <sup>c</sup>	37	28	12	37	100	60	80	0.09	32	72.0
$\beta$ 2M <sub>D76N</sub>	9.9	7.0	8.0	9.9	20.5 <sup>d</sup>	30	30	10	30	100	55	48	0.09	32	57.5

<sup>a</sup> at the static field used (23.5 T) 1 ppm <sup>1</sup>H = 1000.3 Hz, 1 ppm <sup>13</sup>C = 251.5 Hz, and 1 ppm <sup>15</sup>N = 101.4 Hz

<sup>b</sup> truncated to 10.0 ms during data processing

<sup>c</sup> truncated to 9.9 ms during data processing

<sup>d</sup> truncated to 7.2 ms during data processing

<sup>e</sup> the signal magnitude with respect to the first FID of (H)NH dipolar correlation experiment

Table S2: (H)NCOCANH experiment acquisition parameters

Sample	Max. evolution time, ms in dimension					Spectral widths, ppm <sup>a</sup> in dimension					MAS rate, kHz	Scans	Efficiency <sup>e</sup>	Number of projec- tions	Time, h
	$\omega_1$ N	$\omega_2$ C'	$\omega_3$ C $\alpha$	$\omega_4$ N	$\omega_5$ H	$\omega_1$ N	$\omega_2$ C'	$\omega_3$ C $\alpha$	$\omega_4$ N	$\omega_5$ H					
GB1	18.1	13.9	9.9	18.1	20.5 <sup>b</sup>	30	12	26	30	100	110	32	0.18	24	41.5
SOD	13.1	9.9	9.1	13.1	24.8 <sup>c</sup>	37	12	28	37	100	60	80	0.11	32	71.0
$\beta$ 2M <sub>D76N</sub>	9.9	8.0	7.0	9.9	20.5 <sup>d</sup>	30	10	30	30	55	55	48	0.09	32	58.0

<sup>a</sup> at the static field used (23.5 T) 1 ppm <sup>1</sup>H = 1000.3 Hz, 1 ppm <sup>13</sup>C = 251.5 Hz, and 1 ppm <sup>15</sup>N = 101.4 Hz

<sup>b</sup> truncated to 10.0 ms during data processing

<sup>c</sup> truncated to 9.9 ms during data processing

<sup>d</sup> truncated to 7.2 ms during data processing

<sup>e</sup> the signal magnitude with respect to the first FID of (H)NH dipolar correlation experiment

Table S3: Details of cross-polarization transfers

CP transfer no	Sample			
		GB1	SOD	$\beta$ 2M <sub>D76N</sub>
(H)NCACONH				
1 ( $^1\text{H}\rightarrow^{15}\text{N}$ )	contact time, ms	2.3	1.25	1.0
	$^1\text{H}$ r.f. max. field, kHz	149.1	103.5	95.7
	$^1\text{H}$ r.f. amplitude ramp <sup>a</sup>	+10%	+10%	+10%
	$^{15}\text{N}$ r.f. field (const), kHz	33.4	35.2	35.0
2 ( $^{15}\text{N}\rightarrow^{13}\text{C}\alpha$ )	contact time, ms	19.0	19.0	18.0
	$^{15}\text{N}$ r.f. max. field, kHz <sup>b</sup>	42.8	27.0	30.9
	$^{15}\text{N}$ r.f. average field, kHz <sup>b</sup>	37.2	23.5	26.8
	$^{15}\text{N}$ r.f. modulation depth, kHz <sup>b</sup>	5.6	3.5	4.0
	$^{15}\text{N}$ r.f. effective coupling, Hz <sup>b</sup>	1359	858	980
	$^{13}\text{C}\alpha$ r.f. field (const), kHz	70.0	43.5	25.0
3 ( $^{13}\text{C}'\rightarrow^{15}\text{N}$ )	contact time, ms	19.0	14.0	16.0
	$^{13}\text{C}'$ r.f. field (const), kHz	70.0	43.9	25.0
	$^{15}\text{N}$ r.f. max. field, kHz <sup>b</sup>	43.3	27.0	31.3
	$^{15}\text{N}$ r.f. average field, kHz <sup>b</sup>	37.7	23.5	27.2
	$^{15}\text{N}$ r.f. modulation depth, kHz <sup>b</sup>	-5.7	-3.5	4.1
	$^{15}\text{N}$ r.f. effective coupling, Hz <sup>b</sup>	1376	858	993
4 ( $^1\text{H}\rightarrow^{15}\text{N}$ )	contact time, ms	0.7	0.6	0.4
	$^{15}\text{N}$ r.f. field (const), kHz	33.4	35.2	35.0
	$^1\text{H}$ r.f. max. field, kHz	143.9	100.7	94.5
	$^1\text{H}$ r.f. amplitude ramp <sup>a</sup>	+10%	+10%	+10%
(H)NCOCANH				
1 ( $^1\text{H}\rightarrow^{15}\text{N}$ )	as for (H)NCACONH			
2 ( $^{15}\text{N}\rightarrow^{13}\text{C}'$ )	contact time, ms	19.0	14.0	16.0
	$^{15}\text{N}$ r.f. max. field, kHz <sup>b</sup>	42.8	27.0	31.1
	$^{15}\text{N}$ r.f. average field, kHz <sup>b</sup>	37.2	23.5	27.0

Table S3: Details of cross-polarization transfers (cont.)

CP transfer no		Sample		
		GB1	SOD	$\beta 2M_{D76N}$
	$^{15}\text{N}$ r.f. modulation depth, kHz <sup>b</sup>	5.6	3.5	4.1
	$^{15}\text{N}$ r.f. effective coupling, Hz <sup>b</sup>	1359	858	986
	$^{13}\text{C}$ r.f. field (const), kHz	70.0	44.4	25.0
3 ( $^{13}\text{C}\alpha \rightarrow ^{15}\text{N}$ )	contact time, ms	19.0	17.0	18.0
	$^{13}\text{C}\alpha$ r.f. field (const), kHz	70.0	44.3	25.0
	$^{15}\text{N}$ r.f. max. field, kHz <sup>b</sup>	43.3	27.0	30.9
	$^{15}\text{N}$ r.f. average field, kHz <sup>b</sup>	37.7	23.5	26.8
	$^{15}\text{N}$ r.f. modulation depth, kHz <sup>b</sup>	-5.7	-3.5	4.0
	$^{15}\text{N}$ r.f. effective coupling, Hz <sup>b</sup>	1376	858	980
4 ( $^1\text{H} \rightarrow ^{15}\text{N}$ )	as for (H)NCACONH			

<sup>a</sup> An x % ramp means a linear amplitude modulation from (1-x/100) to 1.0

<sup>b</sup> The tangential shapes (for adiabatic CP<sup>16</sup>) are defined by 4 empirical parameters: the average r.f. field  $\omega_{\text{avg}}$  (close to the Hartmann-Hahn matching condition), the effective dipolar coupling  $d^{\text{eff}}$  (averaged over crystal orientations and decreased due to internal motions), the modulation depth  $\Delta$ , and the contact time  $\tau$ , according to the following equation:

$$\omega(t) = \omega_{\text{avg}} + d^{\text{eff}} \tan \left( \frac{2}{\tau} \text{atan} \left( \frac{\Delta}{d_{\text{NC}}^{\text{eff}}} \right) \left( \frac{\tau}{2} - t \right) \right)$$

Table S4: Projection angles used in APSY series

No	$\alpha^a$	$\beta^a$	$\gamma^a$	Number of projections	Number of time points <sup>b</sup>
GB1: (H)NCACONH					
1	0.0	0.0	0.0	1	55x2
2	0.0	0.0	90.0	1	55x2
3	0.0	90.0	0.0	1	63x2
4	90.0	0.0	0.0	1	42x2
5-6	$\pm 37.6$	0.0	0.0	2	69x2x2
7-8	0.0	$\pm 29.0$	0.0	2	79x2x2
9-10	0.0	0.0	$\pm 45.0$	2	78x2x2
11-12	90.0	$\pm 35.8$	0.0	2	71x2x2
13-14	90.0	0.0	$\pm 52.4$	2	69x2x2
15-16	0.0	90.0	$\pm 61.0$	2	79x2x2
17-20	$\pm 37.6$	$\pm 23.7$	0.0	4	89x2x2x2
21-24	$\pm 37.6$	0.0	$\pm 38.4$	4	89x2x2x2
Total				24	3634
GB1: (H)NCOCANH					
1	0.0	0.0	0.0	1	55x2
2	0.0	0.0	90.0	1	55x2
3	0.0	90.0	0.0	1	42x2
4	90.0	0.0	0.0	1	75x2
5-6	$\pm 28.8$	0.0	0.0	2	80x2x2
7-8	0.0	$\pm 37.6$	0.0	2	69x2x2
9-10	0.0	0.0	$\pm 45.0$	2	78x2x2
11-12	90.0	$\pm 54.5$	0.0	2	72x2x2
13-14	90.0	0.0	$\pm 61.2$	2	80x2x2
15-16	0.0	90.0	$\pm 52.4$	2	69x2x2
17-20	$\pm 28.8$	$\pm 34.0$	0.0	4	90x2x2x2
21-24	$\pm 28.8$	0.0	$\pm 41.2$	4	96x2x2
Total				24	3350

**Table S4: Projection angles used in APSY series (cont.)**

No	$\alpha^a$	$\beta^a$	$\gamma^a$	Number of projections	Number of time points <sup>b</sup>
SOD: (H)NCACONH					
1	0.0	0.0	0.0	1	49x2
2	0.0	0.0	90.0	1	49x2
3	0.0	90.0	0.0	1	64x2
4	90.0	0.0	0.0	1	30x2
5-6	$\pm 37.3$	0.0	0.0	2	57x2x2
7-8	0.0	$\pm 34.8$	0.0	2	77x2x2
9-10	0.0	0.0	$\pm 45.0$	2	70x2x2
11-12	90.0	$\pm 42.4$	0.0	2	66x2x2
13-14	90.0	0.0	$\pm 52.7$	2	57x2x2
15-16	0.0	90.0	$\pm 55.2$	2	77x2x2
17-20	$\pm 37.3$	$\pm 29.0$	0.0	4	81x2x2x2
21-24	$\pm 37.3$	0.0	$\pm 38.5$	4	75x2x2x2
25-28	0.0	$\pm 34.8$	$\pm 39.4$	4	91x2x2x2
29-32	90.0	$\pm 42.4$	$\pm 44.1$	4	81x2x2x2
Total				32	4624
SOD: (H)NCOCANH					
1	0.0	0.0	0.0	1	49x2
2	0.0	0.0	90.0	1	49x2
3	0.0	90.0	0.0	1	30x2
4	90.0	0.0	0.0	1	64x2
5-6	$\pm 34.8$	0.0	0.0	2	77x2x2
7-8	0.0	$\pm 37.3$	0.0	2	57x2x2
9-10	0.0	0.0	$\pm 45.0$	2	70x2x2
11-12	90.0	$\pm 47.6$	0.0	2	66x2x2
13-14	90.0	0.0	$\pm 55.2$	2	77x2x2
15-16	0.0	90.0	$\pm 52.7$	2	57x2x2
17-20	$\pm 34.8$	$\pm 32.0$	0.0	4	81x2x2x2
21-24	$\pm 34.8$	0.0	$\pm 39.4$	4	91x2x2x2
25-28	0.0	$\pm 37.3$	$\pm 38.5$	4	75x2x2x2
29-32	90.0	$\pm 47.6$	$\pm 44.1$	4	81x2x2x2
Total				32	4624
$\beta 2M_{D76N}$ : (H)NCACONH					
1	0.0	0.0	0.0	1	30x2
2	0.0	0.0	90.0	1	30x2
3	0.0	90.0	0.0	1	53x2
4	90.0	0.0	0.0	1	20x2
5-6	$\pm 38.9$	0.0	0.0	2	36x2x2
7-8	0.0	$\pm 35.4$	0.0	2	55x2x2
9-10	0.0	0.0	$\pm 45.0$	2	43x2x2
11-12	90.0	$\pm 41.5$	0.0	2	50x2x2
13-14	90.0	0.0	$\pm 51.1$	2	36x2x2
15-16	0.0	90.0	$\pm 54.6$	2	55x2x2
17-20	$\pm 38.9$	$\pm 29.0$	0.0	4	57x2x2x2
21-24	$\pm 38.9$	0.0	$\pm 37.9$	4	47x2x2x2
25-28	0.0	$\pm 35.4$	$\pm 39.2$	4	62x2x2x2
29-32	90.0	$\pm 41.5$	$\pm 42.9$	4	57x2x2x2
Total				32	3150



**Table S4: Projection angles used in APSY series (cont.)**

No	$\alpha^a$	$\beta^a$	$\gamma^a$	Number of projections	Number of time points <sup>b</sup>
<b><math>\beta 2M_{D76N}</math>: (H)NCOCANH</b>					
1	0.0	0.0	0.0	1	30x2
2	0.0	0.0	90.0	1	30x2
3	0.0	90.0	0.0	1	20x2
4	90.0	0.0	0.0	1	53x2
5-6	$\pm 35.4$	0.0	0.0	2	55x2x2
7-8	0.0	$\pm 38.9$	0.0	2	36x2x2
9-10	0.0	0.0	$\pm 45.0$	2	43x2x2
11-12	90.0	$\pm 48.5$	0.0	2	50x2x2
13-14	90.0	0.0	$\pm 54.6$	2	55x2x2
15-16	0.0	90.0	$\pm 51.1$	2	36x2x2
17-20	$\pm 35.4$	$\pm 33.3$	0.0	4	57x2x2x2
21-24	$\pm 35.4$	0.0	$\pm 39.2$	4	62x2x2x2
25-28	0.0	$\pm 38.9$	$\pm 37.9$	4	47x2x2x2
29-32	90.0	$\pm 48.5$	$\pm 42.9$	4	57x2x2x2
Total				32	3150

<sup>a</sup>Time increments in the indirect dimensions are determined by angles  $\alpha$ ,  $\beta$  and  $\gamma$  in the following way:

$$\Delta_1 = (sw_1)^{-1} \sin(\gamma) \quad (1)$$

$$\Delta_2 = (sw_2)^{-1} \sin(\beta) \cos(\gamma) \quad (2)$$

$$\Delta_3 = (sw_3)^{-1} \sin(\alpha) \cos(\beta) \cos(\gamma) \quad (3)$$

$$\Delta_4 = (sw_4)^{-1} \cos(\alpha) \cos(\beta) \cos(\gamma) \quad (4)$$

<sup>b</sup>The number of time increments is followed by the number of quadratures ("x2") depending on the type of projection. Details on data handling that leads to actual projections are given in section E.

## D. Amide proton and heteronuclear refocussed coherence lifetimes

**Table S5: Bulk  $^1H^N$ ,  $^{15}N$ ,  $^{13}C\alpha$ , and  $^{13}C'$  refocussed coherence lifetimes under fast MAS and low-power decoupling**

Sample	MAS rate, kHz	Refocussed coherence lifetime, ms			
		$^{13}C'$	$^{13}C\alpha$	$^{15}N$	$^1H^N$
GB1	110	89 <sup>a</sup>	71 <sup>a</sup>	120 <sup>a,b</sup>	4.7
SOD	60	36 <sup>c</sup>	22 <sup>c</sup>	42 <sup>c,d</sup>	9.0 <sup>d</sup>
$\beta 2M_{D76N}$	55	36 <sup>c</sup>	23 <sup>c</sup>	30	4.7

<sup>a</sup> Under 10 kHz swept-TPPM decoupling<sup>10-11</sup>

<sup>b</sup> Reported by Andreas et al<sup>17</sup> for an equivalent sample of GB1, at identical MAS rate and  $B_0$  field

<sup>c</sup> Under 10 kHz WALTZ-16 decoupling<sup>12</sup>

<sup>d</sup> Reported by Knight et al<sup>3</sup> for an equivalent sample of SOD-Cu<sup>I</sup>,Zn at identical MAS rate and  $B_0$  field

## E. Performance of interresidue amide proton transfer experiments

In solution two equivalent experiments have been proposed to record interresidue amide proton and nitrogen correlations: 5D HN(CA)CONH<sup>18</sup> and 6D APSY HNCOCANH.<sup>19</sup> In the following discussion we will not consider two other experiments, 4D HN(CA)NH<sup>20</sup> and 3D TROSY-(H)N(CA)NH<sup>21</sup>, since their performance is in general inferior to the experiments passing through CO.

5D HN(CA)CONH provides two kinds of correlations: interresidue H(i-1)N(i-1)CO(i-1)N(i)H(i), and intraresidue H(i)N(i)CO(i)N(i)H(i). 6D HNCOCANH performs the same coherence transfer but in the opposite direction, yielding interresidue H(i)N(i)CO(i-1)CA(i-1)N(i-1)H(i-1) and an intraresidue peak H(i)N(i)CO(i-1)CA(i-1)N(i)H(i). Since both experiment provide basically the same information, we will focus here on the analysis of the latter experiment only.

We will calculate stepwise the transfer function  $f$  for the entire experiment, using the original naming convention of experiment steps ("s", "t", "u", "v", "w"): <sup>19</sup>

- 1)  $H^{(i)}_z \rightarrow 2H^{(i)}_z N^{(i)}_z$  (up to point "s"):

$$f_1 = \sin(\pi J_{NH} \tau_1) \exp(-\bar{R}_{2,H} \tau_1)$$

where  $\bar{R}_{2,H}$  is the average relaxation rate of the in-phase ( $H^\pm$ ) and anti-phase SQ coherences ( $2H^\pm N_z$ ). The optimal transfer is obtained for  $\tau_1 = (\pi J_{NH})^{-1} \text{atan}(\pi J_{NH} / \bar{R}_{2,H})$  and equals to:

$$f_1^{max} = \frac{\eta}{\sqrt{1+\eta^2}} \exp\left(-\frac{1}{\eta} \text{atan} \eta\right)$$

where  $\eta = \pi J_{NH} / \bar{R}_{2,H}$ .

- 2)  $2H^{(i)}_z N^{(i)}_z \rightarrow 2N^{(i)}_z C'^{(i-1)}_z$  (from point "s" to "t"):

$$f_2 = \sin(\pi {}^1J_{NC'} \tau_2) \sin(\pi J_{NH} \tau_{NH}) \exp(-R_{2,N} \tau_2)$$

where  $R_{2,N}$  is the relaxation rate of the in-phase SQ  $^{15}N$  coherence ( $N^\pm$ ). The optimal delay  $\tau_{NH} = (2J_{NH})^{-1}$  and  $\tau_2 = (\pi {}^1J_{NC'})^{-1} \text{atan}(\pi {}^1J_{NC'} / R_{2,N})$ , hence the optimal transfer is:

$$f_2^{max} = \frac{\zeta}{\sqrt{1+\zeta^2}} \exp\left(-\frac{1}{\zeta} \text{atan} \zeta\right)$$

where  $\zeta = \pi {}^1J_{NC'} / R_{2,N}$ .

- 3)  $2N^{(i)}_z C'^{(i-1)}_z \rightarrow 4N^{(i)}_z C'^{(i-1)}_z C^{\alpha(i-1)}_z$  (from point "t" to "u"):

$$f_3 = \sin(\pi {}^1J_{C\alpha C'} \tau_3) \exp(-R_{2,C'} \tau_3)$$

where  $R_{2,C'}$  is the relaxation rate of the in-phase SQ  $^{13}C'$  coherence. The optimal transfer is obtained for  $\tau_3 = (\pi {}^1J_{C\alpha C'})^{-1} \text{atan}(\pi {}^1J_{C\alpha C'} / R_{2,C'})$  and equals to:

$$f_3^{max} = \frac{\kappa}{\sqrt{1+\kappa^2}} \exp\left(-\frac{1}{\kappa} \text{atan} \kappa\right)$$

where  $\kappa = \pi {}^1J_{C\alpha C'} / R_{2,C'}$ .

- 4)  $4N^{(i)}_z C'^{(i-1)}_z C^{\alpha(i-1)}_z \rightarrow 2N^{(i)}_z C^{\alpha(i-1)}_z$  or  $2N^{(i)}_z C^{\alpha(i-1)}_z$  (from point "u" to "v"): the active evolution of both 2- and 1-bond  $N-C^\alpha$  couplings leads to splitting of the coherence pathway into the desired inter- and undesired intraresidue correlations ("back transfer"). Their transfer functions are as follows:

$$f_{4inter} = \sin(\pi {}^2J_{NC\alpha} \tau_4) \sin(\pi {}^1J_{NC\alpha} \tau_4) \cos(\pi {}^1J_{C\alpha C\beta} \tau_4) \sin(\pi {}^1J_{C\alpha C'} \tau_{C\alpha C'}) \exp(-R_{2,C\alpha} \tau_4)$$

$$f_{4intra} = \cos(\pi {}^2J_{NC\alpha} \tau_4) \cos(\pi {}^1J_{NC\alpha} \tau_4) \cos(\pi {}^1J_{C\alpha C\beta} \tau_4) \sin(\pi {}^1J_{C\alpha C'} \tau_{C\alpha C'}) \exp(-R_{2,C\alpha} \tau_4)$$

with the exception of  $i-1$  glycine residues where  $C^\alpha$  does not dephase due to  ${}^1J_{C\alpha C\beta}$ . The middle term  $\cos(\pi {}^1J_{C\alpha C\beta} \tau_4)$  can be omitted for all  $i-1$  residues except TSNDLFY, for which the evolution under  ${}^1J_{C\alpha C\beta}$  can be eliminated with a suitably selective  $C^\alpha$  refocusing pulse.  $R_{2,C\alpha}$  stands for relaxation rate of SQ  $C^\alpha$  coherence, and delay  $\tau_{C\alpha C'} = (2 {}^1J_{C\alpha C'})^{-1}$  can be exactly matched to maximize the transfer. The best choice of the delay  $\tau_4$  depends on whether the back transfer is suppressed (as in the original implementation of Fiorito et al) or not. In such a case,  $\tau_4 \approx 2(J_{\alpha\beta})^{-1}$  approximately maximizes the  $\sin(\pi {}^2J_{NC\alpha} \tau_4) \sin(\pi {}^1J_{NC\alpha} \tau_4)$  while suppressing the  $\cos(\pi {}^2J_{NC\alpha} \tau_4) \cos(\pi {}^1J_{NC\alpha} \tau_4)$  term, due to coincidence of coupling constant values. The best value of  $\tau_4$  can be found only numerically, e.g. by iterating the formula:

$$\tau_{4,i+1} = \frac{k}{J_{\alpha\beta}} + \frac{1}{\pi J_{\alpha\beta}} \text{atan}\left(\frac{1}{\pi J_{\alpha\beta}} \left(\frac{\pi {}^1J_{NC\alpha}}{\tan(\pi {}^1J_{NC\alpha} \tau_{4,i})} + \frac{\pi {}^2J_{NC\alpha}}{\tan(\pi {}^2J_{NC\alpha} \tau_{4,i})} - R_{2,C\alpha}\right)\right) \text{ with } k = 2.$$

For relaxation rates above  $R_{2,C\alpha} \approx 30$  Hz the optimum transfer of intraresidue coherences occurs at lower value of  $\tau_4 \approx (J_{\alpha\beta})^{-1}$ , and the precise value can be found with a numerical scheme as above but

with  $k = 1$ . At this condition the intraresidue coherences are not suppressed. For residues for which the evolution under  ${}^1J_{C\alpha C\beta}$  is absent or nearly suppressed, the approximate optimum value of  $f_{4inter}$  is found at delay  $\tau_4 = 2R_{2,C\alpha}\xi^{-1} \operatorname{atan} \xi$ , where  $\xi = \pi({}^2J_{NC\alpha} + {}^1J_{NC\alpha})/R_{2,C\alpha}$ , and the precise one can be found using an iterative scheme:

$$\tau_{4,i+1} = \frac{1}{\pi {}^1J_{NC\alpha}} \operatorname{atan} \left( \frac{\pi {}^2J_{NC\alpha} \tan(\pi {}^1J_{NC\alpha} \tau_{4,i})}{R_{2,C\alpha} \tan(\pi {}^2J_{NC\alpha} \tau_{4,i})} + \frac{\pi {}^1J_{NC\alpha}}{R_{2,C\alpha}} \right)$$

In practice, one rather chooses the best condition for residues TSNDLFY (affected by  ${}^1J_{C\alpha C\beta}$ ) since the best transfer conditions for other residues are significantly broader.

- 5)  $2N^{(i-1)}_z C^{\alpha(i-1)}_z \rightarrow 2H^{N(i-1)}_z N^{(i-1)}_z$  or  $2N^{(i)}_z C^{\alpha(i-1)}_z \rightarrow 2H^{N(i)}_z N^{(i)}_z$  (from point “v” to “w”):

$$f_{5inter} = \sin(\pi {}^1J_{NC\alpha} \tau_5) \cos(\pi {}^2J_{NC\alpha} \tau_5) \sin(\pi J_{NH} \tau_{NH}) \exp(-R_{2,N} \tau_5)$$

$$f_{5intra} = \sin(\pi {}^2J_{NC\alpha} \tau_5) \cos(\pi {}^1J_{NC\alpha} \tau_5) \sin(\pi J_{NH} \tau_{NH}) \exp(-R_{2,N} \tau_5)$$

Here again two- and one-bond N-C $\alpha$  couplings are in competition, and the optimal delay  $\tau_5$  cannot be derived in the analytic form. However, since the magnitude of these couplings is similar, one usually approximates  $\sin(\pi {}^1J_{NC\alpha} \tau_5) \cos(\pi {}^2J_{NC\alpha} \tau_5) \approx \frac{1}{2} \sin(\pi({}^2J_{NC\alpha} + {}^1J_{NC\alpha}) \tau_5)$ . Effectively, the optimal transfer is given by the formula:

$$f_5^{max} = \frac{\lambda}{2\sqrt{1+\lambda^2}} \exp\left(-\frac{1}{\lambda} \operatorname{atan} \lambda\right)$$

where  $\lambda = \pi({}^2J_{NC\alpha} + {}^1J_{NC\alpha})/R_{2,N}$  for both inter- and intraresidue correlations. The precise value of optimal delay for intraresidue transfer function can be found using a series:

$$\tau_{5,i+1} = \frac{1}{\pi {}^1J_{NC\alpha}} \operatorname{atan} \left( \frac{\pi {}^1J_{NC\alpha}}{R_{2,N} + \pi {}^2J_{NC\alpha} \tan(\pi {}^2J_{NC\alpha} \tau_{5,i})} \right)$$

- 6)  $2H^{N(i-1)}_z N^{(i-1)}_z \rightarrow H^{N(i-1)}_{x/y}$  or  $2H^{N(i)}_z N^{(i)}_z \rightarrow H^{N(i)}_{x/y}$  (from point “w” to acquisition): Fiorito et al implemented this transfer as a simple INEPT, however, it is fair to assume that the sensitivity-enhancement scheme (as used by Kazimierzczuk et al) can be used to optimize the transfer, at least for proteins with MW up to ~25 kDa. The transfer function is roughly approximated by an average of efficiency of two transfer pathways:

$$f_6 = \frac{\sqrt{2}}{2} (f_{MQ,SQ} + f_{SQ,z})$$

$$f_{MQ,SQ} = \exp(-R_{2MQ} \tau_{6a}) \exp(-\bar{R}_{2,H} \tau_{6b}) \sin(\pi J_{NH} \tau_{6b})$$

$$f_{SQ,z} = \exp(-\bar{R}_{2,N} \tau_{6a}) \exp(-R_{1,H} \tau_{6b}) \sin(\pi J_{NH} \tau_{6a})$$

where  $R_{2MQ}$  stands for relaxation rate of  $2H_N x$  term, i.e. a mixture of ZQ and DQ H-N coherences. The factor  $\sqrt{2}$  is not detectable in a 1D measurement, but was introduced here to reflect a higher efficiency of the multidimensional experiment.

For the TROSY version of the experiment (on a deuterated protein, protonated only at labile amide proton sites), the above considerations remain valid, but the relaxation rate of  ${}^{15}N^{\pm}H\beta$  (TROSY) coherence is used at steps 2 and 5. Also, ST2-ST2 TROSY detection block (analogous to step 6 above) is characterized by the following transfer function:

$$f_6^{TR} = \frac{\sqrt{2}}{2} \cdot \frac{1}{4} (f_{SQ,SQ} + f_{MQ,MQ} + f_{SQ,MQ} + f_{MQ,SQ})$$

Here again the factor  $\sqrt{2}$  accounts for the increased sensitivity in multidimensional experiment due to preservation of equivalent pathways after  ${}^{15}N$  evolution ( ${}^{15}N^{\pm}$ ), the factor  $\frac{1}{2}$  reflects the loss due to selection of TROSY coherence, and the factor  $\frac{1}{4}$  which averages four pathways necessary to transfer TROSY coherence, for which the efficiencies are as follows:

$$\begin{aligned}
f_{SQ,SQ} &= \exp(-\bar{R}_{2,N}\tau_{6a}) \exp(-\bar{R}_{2,H}\tau_{6b}) \\
f_{MQ,MQ} &= \exp(-R_{2MQ}\tau_{6a}) \exp(-R_{2MQ}\tau_{6b}) \\
f_{SQ,MQ} &= \exp(-\bar{R}_{2,N}\tau_{6a}) \exp(-R_{2MQ}\tau_{6b}) \\
f_{MQ,SQ} &= \exp(-R_{2MQ}\tau_{6a}) \exp(-\bar{R}_{2,H}\tau_{6b})
\end{aligned}$$

For simplicity, we assume here the relaxation unbalanced implementation of ST2ST2, where  $\tau_{6a} = \tau_{6b} = (2J_{NH})^{-1}$ .

The following values of scalar coupling were used:  $^1J_{NH} = 93$  Hz,  $^1J_{C\alpha N} = 11$  Hz,  $^2J_{C\alpha N} = 7$  Hz,  $^1J_{NC'} = 15$  Hz,  $^1J_{C\alpha C'} = 55$  Hz,  $^1J_{C\alpha C\beta} = 35$  Hz.

Additionally, we took into account the nonideal response of coherences to RF pulses. The implementation of Fiorito et al incorporates 10 refocussing, 13 inversion and 15 excitation or flip-back pulses. We assumed a realistic 0.99 performance of each  $\pi/2$  pulse, 0.98 for inversion, and 0.95 for refocussing pulses, giving rise to RF pulse factor,  $f_{RF}$ .

Overall, the experiment sensitivity in the liquid-state (LS) for the interresidue pathway is the following product:

$$f_{LS} = f_{RF} f_1^{max} f_2^{max} f_3^{max} f_4^{max} f_{inter}^{max} f_5^{max} f_6^{max}$$

We assumed (i) isotropic tumbling of a spherical rotor, and (ii) aqueous solution with viscosity at T=300 K, thus approximately a correlation time of  $\tau_C \approx q$  MW where  $q = 1$  ns / 2.6 kDa, and spectral density in a form:  $J(\omega) = \frac{\tau_C}{1+\omega^2\tau_C^2}$ . The transverse relaxation rates of coherences relevant for the current discussion were:

- a) amide proton SQ coherence relaxation rate,  $\bar{R}_{2,H}$ , with a contribution of dipolar interactions to nearby proton, nitrogen and carbon (and, optionally, deuterium) spins, and CSA of  $\Delta\sigma_H = -16$  ppm:

$$\bar{R}_{2,H} = \frac{1}{2} \left( R_2^{NH}(H) + R_2^{NH}(2HN_z) + R_1^{CSA}(N) \right) + R_2^{HH}(H) + R_2^{CH}(H) + R_2^{DH}(H) + R_2^{CSA}(H)$$

- b) amide nitrogen SQ coherence relaxation rate,  $\bar{R}_{2,N}$ , with a contribution of dipolar interactions only to nearby proton and carbon spins, and CSA of  $\Delta\sigma_N = 164.5$  ppm:

$$\bar{R}_{2,N} = \frac{1}{2} \left( R_2^{NH}(N) + R_2^{NH}(2NH_z) + R_1^{HH}(H) + R_1^{CSA}(H) \right) + R_2^{NC}(N) + R_2^{CSA}(N)$$

and, for purely in-phase ( $^1H$ -decoupled) SQ coherence:

$$R_{2,N} = R_2^{NH}(N) + R_2^{NC}(N) + R_2^{CSA}(N)$$

- c) carbonyl carbon SQ coherence relaxation rate,  $R_{2,C'}$ , with a contribution of dipolar interactions to nearby proton, nitrogen and carbon (and, optionally, deuterium) spins, and CSA of  $\Delta\sigma_C = -119.5$  ppm:

$$R_{2,C'} = R_2^{CH}(C') + R_2^{CN}(C') + R_2^{CC}(C') + R_2^{CD}(C') + R_2^{CSA}(C')$$

- d) alpha carbon SQ coherence relaxation rate,  $R_{2,C\alpha}$ , with a contribution of dipolar interactions to nearby proton, nitrogen and carbon (and, optionally, deuterium) spins, and CSA of  $\Delta\sigma_{CA} = 25$  ppm:

$$R_{2,C\alpha} = R_2^{C\alpha H}(C\alpha) + R_2^{C\alpha N}(C\alpha) + R_2^{C\alpha C}(C\alpha) + R_2^{CD}(C') + R_2^{CSA}(C\alpha)$$

- e) amide proton-nitrogen interconverted zero- and double quantum coherence relaxation rate, with a contribution of dipolar interactions to remote proton and carbon (and, optionally, deuterium) spins:

$$R_{2MQ} = \frac{1}{2} \left( R_2^{NH}(N^{\pm}H^{\mp}) + R_2^{NH}(N^{\pm}H^{\pm}) \right) + R_2^{HH}(H) + R_2^{DH}(H) + R_2^{CH}(H) + R_2^{CSA}(H) + R_2^{CSA}(N)$$

- f) amide-proton and amide-nitrogen TROSY coherence relaxation rates:

$$R_{2,H(N\beta)} = \bar{R}_{2,H} + \eta_{xy}^{NH,H}(H)$$

$$R_{2,N(H\beta)} = \bar{R}_{2,N} + \eta_{xy}^{NH,N}(N)$$

The respective terms were calculated according to well-known formulas for dipolar relaxation:<sup>22</sup>

$$\begin{aligned}
R_2^{NH}(H) &= \frac{d_{NH}^2}{20} (4J(0) + J(\omega_H - \omega_N) + 3J(\omega_H) + 6J(\omega_N) + 6J(\omega_H + \omega_N)) \\
R_2^{NH}(2HN_z) &= \frac{d_{NH}^2}{20} (4J(0) + J(\omega_H - \omega_N) + 3J(\omega_H) + 6J(\omega_H + \omega_N)) \\
R_2^{HH}(H) &= \frac{1}{20} \sum_{K \neq H} d_{HK}^2 (5J(0) + 9J(\omega_H) + 6J(2\omega_H)) \\
R_1^{HH}(H) &= \frac{1}{10} \sum_{K \neq H} d_{HK}^2 (J(0) + 3J(\omega_H) + 6J(2\omega_H)) \\
R_2^{CH}(H) &= \frac{1}{20} \sum_C d_{HC}^2 (4J(0) + J(\omega_H - \omega_C) + 3J(\omega_H) + 6J(\omega_C) + 6J(\omega_H + \omega_C)) \\
R_2^{DH}(H) &= \frac{1}{20} \sum_D d_{HD}^2 (4J(0) + J(\omega_H - \omega_D) + 3J(\omega_H) + 6J(\omega_D) + 6J(\omega_H + \omega_D)) \\
R_2^{NH}(N) &= \frac{1}{20} \left( d_{NH}^2 + \sum_K d_{NK}^2 \right) (4J(0) + J(\omega_H - \omega_N) + 3J(\omega_N) + 6J(\omega_H) + 6J(\omega_H + \omega_N)) \\
R_2^{NH}(2NH_z) &= R_2^{NH}(N) - \frac{d_{NH}^2}{20} 6J(\omega_H) \\
R_2^{NC}(N) &= \frac{1}{20} \sum_C d_{CN}^2 (4J(0) + J(\omega_C - \omega_N) + 3J(\omega_N) + 6J(\omega_C) + 6J(\omega_C + \omega_N)) \\
R_2^{NH}(N^\pm H^\mp) &= \frac{1}{20} (2J(\omega_H - \omega_N) + 3J(\omega_H) + 3J(\omega_N)) \\
R_2^{NH}(N^\pm H^\pm) &= \frac{1}{20} (12J(\omega_H + \omega_N) + 3J(\omega_H) + 3J(\omega_N)) \\
R_2^{CH}(C') &= \frac{1}{20} \left( d_{CH}^2 + \sum_K d_{CK}^2 \right) (4J + J(\omega_H - \omega_C) + 3J(\omega_C) + 6J(\omega_H) + 6J(\omega_H + \omega_C)) \\
R_2^{CN}(C') &= \frac{d_{NC}^2}{20} (4J(0) + J(\omega_C - \omega_N) + 3J(\omega_C) + 6J(\omega_N) + 6J(\omega_C + \omega_N)) \\
R_2^{CC}(C') &= \frac{1}{20} \sum_{C' \neq C} d_{CC'}^2 (5J(0) + 9J(\omega_C) + 6J(2\omega_C)) \\
R_2^{CD}(C') &= \frac{1}{20} \sum_D d_{CD}^2 (4J(0) + J(\omega_C - \omega_D) + 3J(\omega_C) + 6J(\omega_D) + 6J(\omega_C + \omega_D))
\end{aligned}$$

(the last four equations can readily be transformed for  $C\alpha$  by a substitution of  $C\alpha$  and  $C'$ ). In the above equations “H” refers exclusively to the amide proton, “K” stands for remote (aliphatic) protons, and dipolar constant for spins  $I$  and  $S$ ,  $d_{IS}$  is given by formula:

$$d_{IS} = \frac{\mu_0 \hbar \gamma_I \gamma_S}{4\pi r_{IS}^3}$$

For the dipolar proton-proton interactions, we assumed a distance cut-off of 5 Å, and included the following contacts, depending on the kind of secondary structure:  $d(^1H^N, ^1H^N i+1) = 4.3$  Å,  $d(^1H^N, ^1H^N i-1) = 4.3$  Å,  $d(^1H^N, ^1H^N \text{ cross-strand}) = 4.3$  Å,  $d(^1H^N, ^1H\alpha) = 2.9$  Å,  $d(^1H^N, ^1H\alpha i-1) = 2.2$  Å,  $d(^1H^N, ^1H\alpha \text{ cross-strand}) = 3.9$  Å,  $d(^1H^N, ^1H\beta_2) = 2.5$  Å,  $d(^1H^N, ^1H\beta_3) = 2.5$  Å in the  $\beta$ -sheets, and the following ones in  $\alpha$ -helices:  $d(^1H^N, ^1H^N i+1) = 2.8$  Å,  $d(^1H^N, ^1H^N i-1) = 2.8$  Å,  $d(^1H^N, ^1H^N i+2) = 4.2$  Å,  $d(^1H^N, ^1H^N i-2) = 4.2$  Å,  $d(^1H^N, ^1H^N i+3) = 4.8$  Å,  $d(^1H^N, ^1H^N i-3) = 4.8$  Å,  $d(^1H^N, ^1H\alpha) = 2.9$  Å,  $d(^1H^N, ^1H\alpha i+1) = 3.5$  Å,  $d(^1H^N, ^1H\alpha i-1) = 3.5$  Å,  $d(^1H^N, ^1H\alpha i+2) = 4.4$  Å,  $d(^1H^N, ^1H\alpha i-2) = 4.4$  Å,  $d(^1H^N, ^1H\alpha i+3) = 4.4$  Å,  $d(^1H^N, ^1H\alpha i-3) = 4.4$  Å,  $d(^1H^N, ^1H\alpha i+4) = 4.4$  Å,  $d(^1H^N, ^1H\alpha i-4) = 4.4$  Å,  $d(^1H^N, ^1H\beta_2) = 2.5$  Å,  $d(^1H^N, ^1H\beta_3) = 2.5$  Å,  $d(^1H^N, ^1H\beta_2 i+1) = 3.0$  Å,  $d(^1H^N, ^1H\beta_3 i+1) = 3.0$  Å. For heteronuclei, we took into account the following spatial proximities:  $d(^{15}N, ^1H^N) = 1.02$  Å,  $d(^{13}C\alpha, ^1H\alpha) = 1.09$  Å,  $d(^{13}C\alpha, ^1H^N i) = 2.1$  Å,  $d(^{13}C\alpha, ^1H^N i+1) = 2.5$  Å,  $d(^{13}C\alpha, ^1H\beta_2) = 2.1$  Å,  $d(^{13}C\alpha, ^1H\beta_3) = 2.1$  Å,  $d(^{15}N, ^{13}C\alpha) = 1.47$  Å,  $d(^{15}N, ^{13}C') = 1.32$  Å,  $d(^{13}C', ^{13}C\alpha) = 1.54$  Å,  $d(^{13}C\beta, ^{13}C\alpha) = 1.54$  Å,  $d(^{13}C', ^1H^N) = 2.8$  Å,  $d(^{13}C', ^1H\alpha) = 2.2$  Å, and additionally, in beta-sheets:  $d(^{13}C', ^1H^N i+1) = 2.0$  Å,  $d(^{13}C', ^1H^N \text{ cross-strand}) = 3.0$  Å,  $d(^{13}C', ^1H\alpha i+1) = 2.5$  Å,  $d(^{13}C', ^1H\alpha i-1) = 4.6$  Å,  $d(^{13}C', ^1H\alpha \text{ cross-strand}) = 3.0$  Å. In the case of a deuterated protein, a 100% replacement of side-chain protons (“K” indices) to deuterons was assumed. The relaxation rates were averaged equally between  $\alpha$ -helices and  $\beta$ -sheets.

The contributions originating from CSA mechanism are as follows:

$$R_2^{CSA}(I) = \frac{1}{45} (\Delta\sigma_I \omega_I)^2 (4J(0) + 3J(\omega_I))$$

$$R_1^{CSA}(I) = \frac{2}{15} (\Delta\sigma_I \omega_I)^2 J(\omega_I)$$

where  $I = H, N, C\alpha$ , or  $C'$ , and  $\Delta\sigma_I$  stands for the difference of axial and perpendicular principal components of the axially symmetric chemical shift tensor of spin  $I$ .

The interference terms crucial for the favorable relaxation of amide H-N TROSY coherences are:

$$\eta_{xy}^{NH,H}(H) = -(3 \cos(\Theta)^2 - 1) \frac{d_{NH}}{2\sqrt{5}} \cdot \frac{\Delta\sigma_H \omega_H}{5\sqrt{3}} (4J(0) + 3J(\omega_H))$$

$$\eta_{xy}^{NH,N}(N) = -(3 \cos(\Theta)^2 - 1) \frac{d_{NH}}{2\sqrt{5}} \cdot \frac{\Delta\sigma_N \omega_N}{5\sqrt{3}} (4J(0) + 3J(\omega_N))$$

where  $\Theta = 17^\circ$  is the assumed angle between the N-H vector and the principal axis of CSA (H or N) tensor.

We neglected here the effects of chemical exchange (with solvent protons) and internal dynamics.

In solids coherence transfer efficiencies do not exhibit any dependence on molecular weight. We assumed the following performance of four dipolar-based (CP) transfers, according to values reported in the literature for microcrystalline proteins at fast MAS frequencies (90-100 kHz)<sup>17, 23</sup>:

$$\begin{aligned} f_1(CP: H \rightarrow N) &= 0.9 \\ f_2(CP: N \rightarrow C') &= 0.57 \\ f_4(CP: C\alpha \rightarrow N) &= 0.39 \\ f_5(CP: N \rightarrow H) &= 0.58 \end{aligned}$$

The efficiency of in-phase  $^{13}C'$  to in-phase  $^{13}C\alpha$  transfer,  $f_3$ , which is here implemented as INEPT, depends on the  $^{13}C'$  and  $^{13}C\alpha$  coherence lifetimes. As a representative case we assumed *Acinetobacter phage* coat protein 205 for which the following values were reported in the literature,<sup>17, 24</sup> as a function of the spinning rate and level of deuteration:

Labelling	MAS rate, kHz	Refocussed coherence lifetime, ms	
		$^{13}C'$	$^{13}C\alpha$
U- $^1H$ , $^2H$ , $^{13}C$ , $^{15}N$	60	40	25
U- $^1H$ , $^{13}C$ , $^{15}N$	60	26	8.6
U- $^1H$ , $^{13}C$ , $^{15}N$	100	27	22

The performance obtained for a deuterated protein spun at 60 kHz, and a fully protonated protein at 100 kHz MAS was virtually identical, and was thus reported in Figure 1 simply as “MAS NMR”. Similarly to the solution NMR case, we accounted for the limited performance of RF pulses (5 excitation/flip-back, 2 refocussing and 6 relevant inversion pulses). The total transfer function for the solid-state NMR case was calculated as:

$$f_{ssNMR} = f_{RF} f_1 f_2 f_3 f_4 f_5$$

## F. Fourier data processing and GAPRO parameters

The orthogonal projections are Fourier transformed as the conventional data (details can be found in Table S6). The skewed projections are acquired with a total number of  $2^{K+1}$  quadrature components where  $K$  is the number of projection angles that are neither 0 nor 90 degrees (“non-orthogonal”). Such data is treated in the following way: the components are collected in the  $2^K$  groups of two, Fourier transformed and stored separately in the interim locations. Subsequently, their linear  $2^K$  combinations (either sums or differences) are calculated and stored as final projections. These projections differ by the sign of the (scaled) chemical

shift offsets of the co-evolved shifts, which is taken into account by GAPRO. Data splitting, regrouping and final linear combinations are performed by Topspin AU programs transparently for the user.

**Table S6: Fourier processing parameters of the 2D projections**

Sample <sup>a</sup>	Direct dimension		Indirect dimension	
	N <sup>b</sup>	Window <sup>c</sup>	N <sup>b</sup>	Window <sup>c</sup>
GB1	8192	QSINE 3	1024	QSINE 3
SOD	8192	QSINE 3	1024	QSINE 3
β2M <sub>D76N</sub>	8192	QSINE 4	1024	QSINE 3

<sup>a</sup> Identical processing parameters were used for forward and backward 5D experiments

<sup>b</sup> Number of points in the real part of the Fourier Transformed signal

<sup>c</sup> The window function in the time domain: QSINE $n$  – a squared, shifted sine bell function  $w(t) = \sin(\pi(1 - 1/n)(t/t_{\max})) + \pi/n)^2$  (where  $n = 3$  and  $4$  for QSINE3 and QSINE4, respectively)

**Table S7: GAPRO peak picking and geometric analysis parameters**

Sample <sup>a</sup>	$S_{\min 1,2}$ <sup>b</sup>	$\Delta v_{\min}$ <sup>c</sup> Hz	$r_{\min}$ <sup>d</sup> Hz	S/N <sup>e</sup>	<sup>1</sup> H range, ppm
GB1	7	50	40	6	[5.6, 11.7]
SOD	7	100	80	6	[5.4, 10.9]
β2M <sub>D76N</sub>	6	160	60	6	[6.9, 12.4]

<sup>a</sup> Identical parameters were used for forward and backward 5D experiments

<sup>b</sup> Minimal support = the number of projections confirming a 5D peak coordinates

<sup>c</sup> Peak matching tolerance in the directly detected dimension

<sup>d</sup> Peak matching tolerance in the indirectly detected dimensions

<sup>e</sup> Minimum signal-to-noise ratio of a peak

## G. 5D peak lists for GB1, SOD and β2m<sub>D76N</sub>

Tables S8-13 show the FLYA output peak list with automated assignment encoded in the last 6 columns. In the FLYA input files these columns contain zeros. They were directly copied from APSY experiment data folders, and completed with a respective header to allow the proper interpretation of columns by FLYA.

**Table S8: Peak list from (H)NCACONH APSY experiment on GB1**

# Number of dimensions 5

#FORMAT xeasy5D

#INAME 1 M

#INAME 2 A

#INAME 3 O

#INAME 4 N

#INAME 5 H

#SPECTRUM hNCACONH M A O N H

1	118.574	57.264	179.310	121.493	9.169	1 U	5.500E+00	0.000E+00	a 0 N.35	CA.35	C.35	N.36	H.36 #MAP	35
2	119.040	53.939	178.949	104.311	7.173	1 U	5.500E+00	0.000E+00	a 0 N.48	CA.48	C.48	N.49	H.49 #MAP	48
3	121.473	56.001	175.749	115.192	7.437	1 U	5.500E+00	0.000E+00	a 0 N.36	CA.36	C.36	N.37	H.37 #MAP	36
4	117.660	60.452	178.483	118.937	7.652	1 U	5.500E+00	0.000E+00	a 0 N.28	CA.28	C.28	N.29	H.29 #MAP	28
5	123.477	54.600	179.685	121.003	8.253	1 U	5.500E+00	0.000E+00	a 0 N.23	CA.23	C.23	N.24	H.24 #MAP	23
6	108.461	46.905	173.811	122.099	8.368	1 U	5.500E+00	0.000E+00	a 0 N.38	CA.38	C.38	N.39	H.39 #MAP	38
7	118.196	58.673	172.346	124.291	8.529	1 U	5.500E+00	0.000E+00	a 0 N.54	CA.54	C.54	N.55	H.55 #MAP	54
8	122.768	56.172	179.354	118.584	8.577	1 U	5.500E+00	0.000E+00	a 0 N.34	CA.34	C.34	N.35	H.35 #MAP	34
9	118.979	66.575	178.264	118.734	8.829	1 U	5.500E+00	0.000E+00	a 0 N.29	CA.29	C.29	N.30	H.30 #MAP	29
10	121.408	53.933	173.975	115.465	9.013	1 U	5.500E+00	0.000E+00	a 0 N.15	CA.15	C.15	N.16	H.16 #MAP	15
11	125.888	54.342	175.545	125.852	9.531	1 U	5.500E+00	0.000E+00	a 0 N.19	CA.19	C.19	N.20	H.20 #MAP	19
12	109.971	44.673	172.976	121.323	10.048	1 U	5.500E+00	0.000E+00	a 0 N.9	CA.9	C.9	N.10	H.10 #MAP	9
13	112.569	62.586	174.085	130.370	10.832	1 U	5.500E+00	0.000E+00	a 0 N.51	CA.51	C.51	N.52	H.52 #MAP	51
14	115.161	53.624	173.987	108.497	8.066	1 U	5.500E+00	0.000E+00	a 0 N.37	CA.37	C.37	N.38	H.38 #MAP	37
15	116.599	59.232	177.694	117.657	7.169	1 U	5.500E+00	0.000E+00	a 0 N.27	CA.27	C.27	N.28	H.28 #MAP	27
16	122.084	61.888	174.888	131.415	9.208	1 U	5.500E+00	0.000E+00	a 0 N.39	CA.39	C.39	N.40	H.40 #MAP	39
17	126.485	59.966	174.990	127.365	9.385	1 U	5.500E+00	0.000E+00	a 0 N.6	CA.6	C.6	N.7	H.7 #MAP	6
18	108.949	61.003	173.763	118.966	9.422	1 U	5.500E+00	0.000E+00	a 0 N.44	CA.44	C.44	N.45	H.45 #MAP	44
19	125.899	50.765	177.515	116.084	8.773	1 U	5.500E+00	0.000E+00	a 0 N.20	CA.20	C.20	N.21	H.21 #MAP	20
20	116.573	61.471	170.984	125.880	7.959	1 U	5.500E+00	0.000E+00	a 0 N.18	CA.18	C.18	N.19	H.19 #MAP	18

21	130.374	56.658	175.668	112.147	9.354 1 U	5.500E+00	0.000E+00	a 0 N.52	CA.52	C.52	N.53	H.53 #MAP	52
22	116.009	63.673	174.863	115.664	7.445 1 U	5.500E+00	0.000E+00	a 0 N.21	CA.21	C.21	N.22	H.22 #MAP	21
23	115.437	60.245	171.702	116.726	8.484 1 U	5.500E+00	0.000E+00	a 0 N.16	CA.16	C.16	N.17	H.17 #MAP	16
24	108.687	45.185	172.589	119.624	8.748 1 U	5.500E+00	0.000E+00	a 0 N.41	CA.41	C.41	N.42	H.42 #MAP	41
25	105.936	45.002	171.106	121.414	8.997 1 U	5.500E+00	0.000E+00	a 0 N.14	CA.14	C.14	N.15	H.15 #MAP	14
26	118.974	57.937	171.691	126.731	7.565 1 U	5.500E+00	0.000E+00	a 0 N.45	CA.45	C.45	N.46	H.46 #MAP	45
27	121.160	61.713	178.379	122.794	9.303 1 U	5.500E+00	0.000E+00	a 0 N.33	CA.33	C.33	N.34	H.34 #MAP	33
28	124.266	61.396	173.906	131.230	8.035 1 U	5.500E+00	0.000E+00	a 0 N.55	CA.55	C.55	N.56	H.56 #MAP	55
29	124.148	55.188	177.116	116.578	8.818 1 U	5.500E+00	0.000E+00	a 0 N.26	CA.26	C.26	N.27	H.27 #MAP	26
30	131.420	52.750	174.924	108.675	8.113 1 U	5.500E+00	0.000E+00	a 0 N.40	CA.40	C.40	N.41	H.41 #MAP	40
31	121.032	54.580	181.278	117.642	8.580 1 U	5.500E+00	0.000E+00	a 0 N.24	CA.24	C.24	N.25	H.25 #MAP	24
32	125.105	50.706	176.241	109.934	8.357 1 U	5.500E+00	0.000E+00	a 0 N.8	CA.8	C.8	N.9	H.9 #MAP	8
33	125.531	57.672	176.730	108.979	9.074 1 U	5.500E+00	0.000E+00	a 0 N.43	CA.43	C.43	N.44	H.44 #MAP	43
34	120.840	60.241	179.570	121.672	8.092 1 U	5.500E+00	0.000E+00	a 0 N.31	CA.31	C.31	N.32	H.32 #MAP	31
35	112.134	60.377	171.802	118.211	8.290 1 U	5.500E+00	0.000E+00	a 0 N.53	CA.53	C.53	N.54	H.54 #MAP	53
36	116.713	60.330	173.994	116.567	9.319 1 U	5.500E+00	0.000E+00	a 0 N.17	CA.17	C.17	N.18	H.18 #MAP	17
37	125.937	55.919	174.813	123.961	9.169 1 U	5.500E+00	0.000E+00	a 0 N.2	CA.2	C.2	N.3	H.3 #MAP	2
38	119.579	55.230	177.803	125.531	9.399 1 U	5.500E+00	0.000E+00	a 0 N.42	CA.42	C.42	N.43	H.43 #MAP	42
39	121.652	58.950	177.347	121.160	8.806 1 U	5.500E+00	0.000E+00	a 0 N.32	CA.32	C.32	N.33	H.33 #MAP	32
40	127.352	53.045	174.644	126.487	9.111 1 U	5.500E+00	0.000E+00	a 0 N.5	CA.5	C.5	N.6	H.6 #MAP	5
41	119.334	55.465	175.223	112.582	7.543 1 U	5.500E+00	0.000E+00	a 0 N.50	CA.50	C.50	N.51	H.51 #MAP	50
42	117.710	67.475	175.553	124.172	7.458 1 U	5.500E+00	0.000E+00	a 0 N.25	CA.25	C.25	N.26	H.26 #MAP	25
43	126.746	51.011	176.061	123.898	8.866 1 U	5.500E+00	0.000E+00	a 0 N.46	CA.46	C.46	N.47	H.47 #MAP	46
44	123.835	54.465	176.991	119.081	8.675 1 U	5.500E+00	0.000E+00	a 0 N.47	CA.47	C.47	N.48	H.48 #MAP	47
45	115.620	52.533	175.006	123.497	9.416 1 U	5.500E+00	0.000E+00	a 0 N.22	CA.22	C.22	N.23	H.23 #MAP	22
46	106.735	62.099	173.207	128.187	7.533 1 U	5.500E+00	0.000E+00	a 0 N.11	CA.11	C.11	N.12	H.12 #MAP	11
47	123.248	54.947	173.113	127.352	9.317 1 U	5.500E+00	0.000E+00	a 0 N.4	CA.4	C.4	N.5	H.5 #MAP	4
48	121.283	59.304	178.889	106.695	9.132 1 U	5.500E+00	0.000E+00	a 0 N.10	CA.10	C.10	N.11	H.11 #MAP	10
49	123.682	53.445	175.579	105.951	8.740 1 U	5.500E+00	0.000E+00	a 0 N.13	CA.13	C.13	N.14	H.14 #MAP	13
50	127.383	54.826	174.829	125.140	9.151 1 U	5.500E+00	0.000E+00	a 0 N.7	CA.7	C.7	N.8	H.8 #MAP	7
51	124.006	57.150	174.726	123.213	9.358 1 U	5.500E+00	0.000E+00	a 0 N.3	CA.3	C.3	N.4	H.4 #MAP	3
52	104.291	60.567	175.472	119.337	8.077 1 U	5.500E+00	0.000E+00	a 0 N.49	CA.49	C.49	N.50	H.50 #MAP	49
53	118.361	54.506	171.226	125.974	8.571 1 U	5.500E+00	0.000E+00	a 0 N.1	CA.1	C.1	N.2	H.2 #MAP	1
54	128.201	54.612	173.630	123.726	9.004 1 U	5.500E+00	0.000E+00	a 0 N.12	CA.12	C.12	N.13	H.13 #MAP	12

**Table S9: Peak list from (H)NCOCANH APSY experiment on GB1**

# Number of dimensions 5

#FORMAT xeasy5D

#INAME 1 N

#INAME 2 O

#INAME 3 A

#INAME 4 M

#INAME 5 J

#SPECTRUM hNCOCANH N O A M J

1	121.460	179.344	57.305	118.587	8.561 1 U	5.500E+00	0.000E+00	a 0 N.36	C.35	CA.35	N.35	H.35 #MAP	34
2	116.007	177.541	50.845	125.856	9.520 1 U	5.500E+00	0.000E+00	a 0 N.21	C.20	CA.20	N.20	H.20 #MAP	19
3	127.336	175.029	60.017	126.488	9.095 1 U	5.500E+00	0.000E+00	a 0 N.7	C.6	CA.6	N.6	H.6 #MAP	5
4	118.689	178.300	66.607	118.981	7.644 1 U	5.500E+00	0.000E+00	a 0 N.30	C.29	CA.29	N.29	H.29 #MAP	28
5	125.860	175.590	54.415	125.897	7.944 1 U	5.500E+00	0.000E+00	a 0 N.20	C.19	CA.19	N.19	H.19 #MAP	18
6	119.590	172.631	45.242	108.664	8.094 1 U	5.500E+00	0.000E+00	a 0 N.42	C.41	CA.41	N.41	H.41 #MAP	40
7	117.645	181.319	54.644	121.012	8.238 1 U	5.500E+00	0.000E+00	a 0 N.25	C.24	CA.24	N.24	H.24 #MAP	23
8	124.251	172.390	58.744	118.227	8.277 1 U	5.500E+00	0.000E+00	a 0 N.55	C.54	CA.54	N.54	H.54 #MAP	53
9	121.395	171.123	45.071	105.957	8.722 1 U	5.500E+00	0.000E+00	a 0 N.15	C.14	CA.14	N.14	H.14 #MAP	13
10	115.445	173.990	53.993	121.416	8.978 1 U	5.500E+00	0.000E+00	a 0 N.16	C.15	CA.15	N.15	H.15 #MAP	14
11	109.929	176.299	50.765	125.148	9.134 1 U	5.500E+00	0.000E+00	a 0 N.9	C.8	CA.8	N.8	H.8 #MAP	7
12	112.087	175.694	56.680	130.373	10.815 1 U	5.500E+00	0.000E+00	a 0 N.53	C.52	CA.52	N.52	H.52 #MAP	51
13	108.467	174.022	53.680	115.180	7.421 1 U	5.500E+00	0.000E+00	a 0 N.38	C.37	CA.37	N.37	H.37 #MAP	36
14	115.152	175.768	56.053	121.517	9.153 1 U	5.500E+00	0.000E+00	a 0 N.37	C.36	CA.36	N.36	H.36 #MAP	35
15	118.988	178.531	60.523	117.664	7.157 1 U	5.500E+00	0.000E+00	a 0 N.29	C.28	CA.28	N.28	H.28 #MAP	27
16	122.784	178.421	61.768	121.137	8.787 1 U	5.500E+00	0.000E+00	a 0 N.34	C.33	CA.33	N.33	H.33 #MAP	32
17	118.555	179.391	56.226	122.813	9.291 1 U	5.500E+00	0.000E+00	a 0 N.35	C.34	CA.34	N.34	H.34 #MAP	33
18	121.282	172.988	44.753	109.948	8.345 1 U	5.500E+00	0.000E+00	a 0 N.10	C.9	CA.9	N.9	H.9 #MAP	8
19	116.573	174.001	60.398	116.729	8.467 1 U	5.500E+00	0.000E+00	a 0 N.18	C.17	CA.17	N.17	H.17 #MAP	16
20	121.151	177.380	58.996	121.683	8.067 1 U	5.500E+00	0.000E+00	a 0 N.33	C.32	CA.32	N.32	H.32 #MAP	31
21	104.302	179.014	53.991	119.101	8.659 1 U	5.500E+00	0.000E+00	a 0 N.49	C.48	CA.48	N.48	H.48 #MAP	47
22	116.575	177.170	55.240	124.160	7.446 1 U	5.500E+00	0.000E+00	a 0 N.27	C.26	CA.26	N.26	H.26 #MAP	25
23	123.952	174.841	55.989	125.943	8.554 1 U	5.500E+00	0.000E+00	a 0 N.3	C.2	CA.2	N.2	H.2 #MAP	1
24	125.098	174.845	54.846	127.362	9.369 1 U	5.500E+00	0.000E+00	a 0 N.8	C.7	CA.7	N.7	H.7 #MAP	6
25	130.342	174.119	62.639	112.603	7.524 1 U	5.500E+00	0.000E+00	a 0 N.52	C.51	CA.51	N.51	H.51 #MAP	50
26	131.401	174.937	61.942	122.119	8.358 1 U	5.500E+00	0.000E+00	a 0 N.40	C.39	CA.39	N.39	H.39 #MAP	38
27	124.165	175.633	67.528	117.624	8.556 1 U	5.500E+00	0.000E+00	a 0 N.26	C.25	CA.25	N.25	H.25 #MAP	24
28	125.519	177.821	55.309	119.629	8.735 1 U	5.500E+00	0.000E+00	a 0 N.43	C.42	CA.42	N.42	H.42 #MAP	41



29	126.743	171.746	57.969	118.956	9.404	1 U	5.500E+00	0.000E+00	a 0 N.46	C.45	CA.45	N.45	H.45 #MAP	44
30	112.539	175.255	55.514	119.331	8.053	1 U	5.500E+00	0.000E+00	a 0 N.51	C.50	CA.50	N.50	H.50 #MAP	49
31	118.155	171.828	60.407	112.142	9.340	1 U	5.500E+00	0.000E+00	a 0 N.54	C.53	CA.53	N.53	H.53 #MAP	52
32	123.469	175.049	52.596	115.632	7.426	1 U	5.500E+00	0.000E+00	a 0 N.23	C.22	CA.22	N.22	H.22 #MAP	21
33	118.917	173.793	61.051	108.987	9.061	1 U	5.500E+00	0.000E+00	a 0 N.45	C.44	CA.44	N.44	H.44 #MAP	43
34	106.723	178.968	59.344	121.352	10.038	1 U	5.500E+00	0.000E+00	a 0 N.11	C.10	CA.10	N.10	H.10 #MAP	9
35	116.705	171.736	60.291	115.467	9.001	1 U	5.500E+00	0.000E+00	a 0 N.17	C.16	CA.16	N.16	H.16 #MAP	15
36	125.815	171.022	61.530	116.576	9.306	1 U	5.500E+00	0.000E+00	a 0 N.19	C.18	CA.18	N.18	H.18 #MAP	17
37	131.181	173.940	61.446	124.299	8.515	1 U	5.500E+00	0.000E+00	a 0 N.56	C.55	CA.55	N.55	H.55 #MAP	54
38	117.643	177.711	59.252	116.594	8.812	1 U	5.500E+00	0.000E+00	a 0 N.28	C.27	CA.27	N.27	H.27 #MAP	26
39	108.664	174.926	52.818	131.398	9.193	1 U	5.500E+00	0.000E+00	a 0 N.41	C.40	CA.40	N.40	H.40 #MAP	39
40	115.631	174.869	63.711	116.100	8.766	1 U	5.500E+00	0.000E+00	a 0 N.22	C.21	CA.21	N.21	H.21 #MAP	20
41	119.017	177.050	54.537	123.892	8.848	1 U	5.500E+00	0.000E+00	a 0 N.48	C.47	CA.47	N.47	H.47 #MAP	46
42	128.220	173.242	62.123	106.683	9.120	1 U	5.500E+00	0.000E+00	a 0 N.12	C.11	CA.11	N.11	H.11 #MAP	10
43	123.842	176.099	51.062	126.753	7.544	1 U	5.500E+00	0.000E+00	a 0 N.47	C.46	CA.46	N.46	H.46 #MAP	45
44	127.331	173.163	55.028	123.346	9.335	1 U	5.500E+00	0.000E+00	a 0 N.5	C.4	CA.4	N.4	H.4 #MAP	3
45	108.945	176.782	57.736	125.547	9.383	1 U	5.500E+00	0.000E+00	a 0 N.44	C.43	CA.43	N.43	H.43 #MAP	42
46	120.986	179.744	54.644	123.491	9.404	1 U	5.500E+00	0.000E+00	a 0 N.24	C.23	CA.23	N.23	H.23 #MAP	22
47	119.279	175.507	60.585	104.318	7.157	1 U	5.500E+00	0.000E+00	a 0 N.50	C.49	CA.49	N.49	H.49 #MAP	48
48	105.907	175.596	53.495	123.715	8.999	1 U	5.500E+00	0.000E+00	a 0 N.14	C.13	CA.13	N.13	H.13 #MAP	12
49	122.079	173.851	46.954	108.497	8.054	1 U	5.500E+00	0.000E+00	a 0 N.39	C.38	CA.38	N.38	H.38 #MAP	37
50	126.430	174.699	53.080	127.415	9.303	1 U	5.500E+00	0.000E+00	a 0 N.6	C.5	CA.5	N.5	H.5 #MAP	4
51	120.771	178.950	57.623	118.780	8.816	1 U	5.500E+00	0.000E+00	a 0 N.31	C.30	CA.30	N.30	H.30 #MAP	29
52	123.243	174.728	57.139	124.012	9.164	1 U	5.500E+00	0.000E+00	a 0 N.4	C.3	CA.3	N.3	H.3 #MAP	2
53	121.471	179.604	60.293	120.888	9.163	1 U	5.500E+00	0.000E+00	a 0 N.32	C.31	CA.31	N.31	H.31 #MAP	30
54	123.640	173.681	54.587	128.156	7.515	1 U	5.500E+00	0.000E+00	a 0 N.13	C.12	CA.12	N.12	H.12 #MAP	11

**Table S10: Peak list from (H)NCACONH APSY experiment on Cu<sup>I</sup>,Zn-SOD**

# Number of dimensions 5

#FORMAT xeasy5D

#INAME 1 M

#INAME 2 A

#INAME 3 O

#INAME 4 N

#INAME 5 H

#SPECTRUM hNCACONH M A O N H

1	125.420	60.900	173.510	120.950	7.993	1 U	5.500000E+00	0.000000E+00	e 0 N.118	CA.118	C.118	N.119	H.119 #MAP	108
2	118.370	55.420	173.210	112.490	7.168	1 U	5.500000E+00	0.000000E+00	e 0 N.70	CA.70	C.70	N.71	H.71 #MAP	62
3	123.770	61.150	175.450	126.570	8.907	1 U	5.500000E+00	0.000000E+00	e 0 N.7	CA.7	C.7	N.8	H.8 #MAP	7
3	123.770	61.150	175.450	126.570	8.907	1 U	5.500000E+00	0.000000E+00	e 0 N.91	CA.91	C.91	N.92	H.92 #MAP	81
4	106.730	45.530	171.130	124.390	8.451	1 U	5.500000E+00	0.000000E+00	e 0 N.147	CA.147	C.147	N.148	H.148 #MAP	137
5	97.700	45.410	173.630	121.320	7.145	1 U	5.500000E+00	0.000000E+00	e 0 N.82	CA.82	C.82	N.83	H.83 #MAP	72
5	97.700	45.410	173.630	121.320	7.145	1 U	5.500000E+00	0.000000E+00	e 0 N.138	CA.138	C.138	N.139	H.139 #MAP	128
6	118.430	54.340	174.330	112.660	6.665	1 U	5.500000E+00	0.000000E+00	e 0 N.115	CA.115	C.115	N.116	H.116 #MAP	105
7	112.830	44.670	172.790	118.440	7.433	1 U	5.500000E+00	0.000000E+00	e 0 N.114	CA.114	C.114	N.115	H.115 #MAP	104
8	117.130	60.190	174.530	121.530	7.578	1 U	5.500000E+00	0.000000E+00	e 0 N.112	CA.112	C.112	N.113	H.113 #MAP	102
9	115.470	53.380	174.720	121.120	7.744	1 U	5.500000E+00	0.000000E+00	e 0 N.122	CA.122	C.122	N.123	H.123 #MAP	112
10	126.540	55.730	172.120	115.820	8.130	1 U	5.500000E+00	0.000000E+00	e 0 N.45	CA.45	C.45	N.46	H.46 #MAP	41
11	124.370	62.360	175.350	127.430	8.618	1 U	5.500000E+00	0.000000E+00	e 0 N.148	CA.148	C.148	N.149	H.149 #MAP	138
12	118.660	60.950	176.650	131.350	9.690	1 U	5.500000E+00	0.000000E+00	e 0 N.94	CA.94	C.94	N.95	H.95 #MAP	84
13	120.860	61.450	177.150	129.450	10.173	1 U	5.500000E+00	0.000000E+00	e 0 N.47	CA.47	C.47	N.48	H.48 #MAP	43
14	126.010	50.290	174.940	123.770	9.100	1 U	5.500000E+00	0.000000E+00	e 0 N.6	CA.6	C.6	N.7	H.7 #MAP	6
15	121.030	51.600	176.360	122.320	10.152	1 U	5.500000E+00	0.000000E+00	e 0 N.123	CA.123	C.123	N.124	H.124 #MAP	113
16	128.970	66.620	176.670	97.730	8.463	1 U	5.500000E+00	0.000000E+00	e 0 N.81	CA.81	C.81	N.82	H.82 #MAP	71
17	121.640	59.630	171.090	125.990	9.177	1 U	5.500000E+00	0.000000E+00	e 0 N.5	CA.5	C.5	N.6	H.6 #MAP	5
18	123.250	50.270	175.430	121.630	9.558	1 U	5.500000E+00	0.000000E+00	e 0 N.4	CA.4	C.4	N.5	H.5 #MAP	4
19	113.740	42.780	173.270	120.570	8.282	1 U	5.500000E+00	0.000000E+00	e 0 N.41	CA.41	C.41	N.42	H.42 #MAP	37
19	113.740	42.780	173.270	120.570	8.282	1 U	5.500000E+00	0.000000E+00	e 0 N.56	CA.56	C.56	N.57	H.57 #MAP	52
20	127.430	61.910	175.140	116.740	9.290	1 U	5.500000E+00	0.000000E+00	e 0 N.149	CA.149	C.149	N.150	H.150 #MAP	139
21	126.030	57.180	176.170	113.770	8.749	1 U	5.500000E+00	0.000000E+00	e 0 N.40	CA.40	C.40	N.41	H.41 #MAP	36
21	126.030	57.180	176.170	113.770	8.749	1 U	5.500000E+00	0.000000E+00	e 0 N.55	CA.55	C.55	N.56	H.56 #MAP	51
22	118.940	53.940	175.340	118.380	8.329	1 U	5.500000E+00	0.000000E+00	e 0 N.79	CA.79	C.79	N.80	H.80 #MAP	69
23	123.040	55.980	177.540	114.410	7.362	1 U	5.500000E+00	0.000000E+00	e 0 N.144	CA.144	C.144	N.145	H.145 #MAP	134
24	115.800	52.020	175.680	120.910	9.402	1 U	5.500000E+00	0.000000E+00	e 0 N.46	CA.46	C.46	N.47	H.47 #MAP	42
25	119.630	54.820	171.540	119.010	6.990	1 U	5.500000E+00	0.000000E+00	e 0 N.78	CA.78	C.78	N.79	H.79 #MAP	68
26	129.590	53.180	174.320	125.460	8.931	1 U	5.500000E+00	0.000000E+00	e 0 N.117	CA.117	C.117	N.118	H.118 #MAP	107
27	111.200	46.460	172.910	118.690	7.798	1 U	5.500000E+00	0.000000E+00	e 0 N.93	CA.93	C.93	N.94	H.94 #MAP	83
28	107.240	46.250	170.900	120.330	7.987	1 U	5.500000E+00	0.000000E+00	e 0 N.16	CA.16	C.16	N.17	H.17 #MAP	14
28	107.240	46.250	170.900	120.330	7.987	1 U	5.500000E+00	0.000000E+00	e 0 N.141	CA.141	C.141	N.142	H.142 #MAP	131
29	116.740	42.840	173.390	120.350	8.959	1 U	5.500000E+00	0.000000E+00	e 0 N.150	CA.150	C.150	N.151	H.151 #MAP	140
30	111.240	45.120	172.940	121.130	8.671	1 U	5.500000E+00	0.000000E+00	e 0 N.10	CA.10	C.10	N.11	H.11 #MAP	10
31	112.620	62.240	173.080	129.790	8.823	1 U	5.500000E+00	0.000000E+00	e 0 N.116	CA.116	C.116	N.117	H.117 #MAP	106

32	123.020	53.570	174.540	107.340	8.351	1 U	5.500000E+00	0.000000E+00	e 0	N.15	CA.15	C.15	N.16	H.16 #MAP	13
33	109.330	46.790	171.840	126.590	8.119	1 U	5.500000E+00	0.000000E+00	e 0	N.44	CA.44	C.44	N.45	H.45 #MAP	40
34	121.200	55.640	172.030	120.460	7.158	1 U	5.500000E+00	0.000000E+00	e 0	N.83	CA.83	C.83	N.84	H.84 #MAP	73
34	121.200	55.640	172.030	120.460	7.158	1 U	5.500000E+00	0.000000E+00	e 0	N.139	CA.139	C.139	N.140	H.140 #MAP	129
35	126.160	56.290	172.150	123.590	8.735	1 U	5.500000E+00	0.000000E+00	e 0	N.18	CA.18	C.18	N.19	H.19 #MAP	16
36	120.240	62.480	176.010	122.970	7.947	1 U	5.500000E+00	0.000000E+00	e 0	N.14	CA.14	C.14	N.15	H.15 #MAP	12
37	121.450	61.000	176.810	112.820	9.557	1 U	5.500000E+00	0.000000E+00	e 0	N.113	CA.113	C.113	N.114	H.114 #MAP	103
38	108.020	44.220	171.630	114.430	7.882	1 U	5.500000E+00	0.000000E+00	e 0	N.33	CA.33	C.33	N.34	H.34 #MAP	29
39	113.540	58.650	174.830	118.130	8.673	1 U	5.500000E+00	0.000000E+00	e 0	N.87	CA.87	C.87	N.88	H.88 #MAP	77
40	114.690	53.910	174.750	109.360	8.868	1 U	5.500000E+00	0.000000E+00	e 0	N.43	CA.43	C.43	N.44	H.44 #MAP	39
41	110.900	53.920	173.310	106.700	8.035	1 U	5.500000E+00	0.000000E+00	e 0	N.146	CA.146	C.146	N.147	H.147 #MAP	136
42	112.360	55.360	174.670	113.100	7.216	1 U	5.500000E+00	0.000000E+00	e 0	N.71	CA.71	C.71	N.72	H.72 #MAP	63
43	114.350	50.910	175.020	110.990	8.995	1 U	5.500000E+00	0.000000E+00	e 0	N.145	CA.145	C.145	N.146	H.146 #MAP	135
44	118.080	61.440	173.590	129.130	9.341	1 U	5.500000E+00	0.000000E+00	e 0	N.88	CA.88	C.88	N.89	H.89 #MAP	78
45	121.320	60.420	175.540	127.630	9.203	1 U	5.500000E+00	0.000000E+00	e 0	N.29	CA.29	C.29	N.30	H.30 #MAP	25
46	119.390	45.600	171.360	115.200	5.836	1 U	5.500000E+00	0.000000E+00	e 0	N.51	CA.51	C.51	N.52	H.52 #MAP	47
47	129.130	49.190	177.460	124.460	8.391	1 U	5.500000E+00	0.000000E+00	e 0	N.89	CA.89	C.89	N.90	H.90 #MAP	79
48	131.360	49.640	174.970	125.260	8.506	1 U	5.500000E+00	0.000000E+00	e 0	N.95	CA.95	C.95	N.96	H.96 #MAP	85
49	125.570	56.270	175.970	121.820	9.193	1 U	5.500000E+00	0.000000E+00	e 0	N.23	CA.23	C.23	N.24	H.24 #MAP	21
49	125.570	56.270	175.970	121.820	9.193	1 U	5.500000E+00	0.000000E+00	e 0	N.120	CA.120	C.120	N.121	H.121 #MAP	110
50	120.430	63.120	175.390	130.130	8.251	1 U	5.500000E+00	0.000000E+00	e 0	N.151	CA.151	C.151	N.152	H.152 #MAP	141
51	126.490	53.920	175.670	120.930	8.269	1 U	5.500000E+00	0.000000E+00	e 0	N.8	CA.8	C.8	N.9	H.9 #MAP	8
52	116.860	54.550	177.180	104.090	8.678	1 U	5.500000E+00	0.000000E+00	e 0	N.126	CA.126	C.126	N.127	H.127 #MAP	116
53	113.080	44.020	171.830	105.470	8.745	1 U	5.500000E+00	0.000000E+00	e 0	N.72	CA.72	C.72	N.73	H.73 #MAP	64
54	120.380	61.060	174.300	126.210	8.697	1 U	5.500000E+00	0.000000E+00	e 0	N.17	CA.17	C.17	N.18	H.18 #MAP	15
54	120.380	61.060	174.300	126.210	8.697	1 U	5.500000E+00	0.000000E+00	e 0	N.142	CA.142	C.142	N.143	H.143 #MAP	132
55	106.120	45.160	174.530	120.720	8.125	1 U	5.500000E+00	0.000000E+00	e 0	N.37	CA.37	C.37	N.38	H.38 #MAP	33
56	115.000	52.740	176.670	105.320	8.285	1 U	5.500000E+00	0.000000E+00	e 0	-	-	-	-	-	-
57	111.810	62.860	172.440	107.930	6.935	1 U	5.500000E+00	0.000000E+00	e 0	N.134	CA.134	C.134	N.135	H.135 #MAP	124
58	118.250	60.430	174.920	110.470	8.569	1 U	5.500000E+00	0.000000E+00	e 0	N.49	CA.49	C.49	N.50	H.50 #MAP	45
59	120.360	53.370	176.550	118.280	8.664	1 U	5.500000E+00	0.000000E+00	e 0	N.57	CA.57	C.57	N.58	H.58 #MAP	53
60	120.530	58.180	177.230	119.640	8.078	1 U	5.500000E+00	0.000000E+00	e 0	N.77	CA.77	C.77	N.78	H.78 #MAP	67
61	114.210	54.890	176.360	121.590	9.541	1 U	5.500000E+00	0.000000E+00	e 0	N.20	CA.20	C.20	N.21	H.21 #MAP	18
62	119.710	52.860	176.960	101.720	7.732	1 U	5.500000E+00	0.000000E+00	e 0	N.60	CA.60	C.60	N.61	H.61 #MAP	56
63	110.440	55.550	178.360	119.370	10.212	1 U	5.500000E+00	0.000000E+00	e 0	N.50	CA.50	C.50	N.51	H.51 #MAP	46
64	115.100	54.700	176.790	120.630	7.455	1 U	5.500000E+00	0.000000E+00	e 0	N.75	CA.75	C.75	N.76	H.76 #MAP	65
65	129.370	53.830	178.340	117.020	10.449	1 U	5.500000E+00	0.000000E+00	e 0	N.125	CA.125	C.125	N.126	H.126 #MAP	115
66	123.520	51.850	172.170	114.120	8.372	1 U	5.500000E+00	0.000000E+00	e 0	N.19	CA.19	C.19	N.20	H.20 #MAP	17
67	121.570	55.700	173.490	130.350	9.283	1 U	5.500000E+00	0.000000E+00	e 0	N.21	CA.21	C.21	N.22	H.22 #MAP	19
68	120.890	54.720	175.760	111.250	8.884	1 U	5.500000E+00	0.000000E+00	e 0	N.9	CA.9	C.9	N.10	H.10 #MAP	9
69	120.970	57.970	174.290	125.540	8.943	1 U	5.500000E+00	0.000000E+00	e 0	N.119	CA.119	C.119	N.120	H.120 #MAP	109
70	125.910	54.430	175.310	127.530	9.288	1 U	5.500000E+00	0.000000E+00	e 0	-	-	-	-	-	-
71	114.290	56.720	172.800	123.040	8.521	1 U	5.500000E+00	0.000000E+00	e 0	N.34	CA.34	C.34	N.35	H.35 #MAP	30
72	120.680	55.220	177.310	114.730	8.691	1 U	5.500000E+00	0.000000E+00	e 0	N.42	CA.42	C.42	N.43	H.43 #MAP	38
73	123.280	59.640	173.050	124.100	9.123	1 U	5.500000E+00	0.000000E+00	e 0	N.35	CA.35	C.35	N.36	H.36 #MAP	31
74	108.250	45.690	174.640	118.370	8.902	1 U	5.500000E+00	0.000000E+00	e 0	N.130	CA.130	C.130	N.131	H.131 #MAP	120
75	115.600	54.830	175.150	121.690	9.387	1 U	5.500000E+00	0.000000E+00	e 0	N.63	CA.63	C.63	N.64	H.64 #MAP	57
76	104.240	45.410	176.860	118.430	7.154	1 U	5.500000E+00	0.000000E+00	e 0	N.127	CA.127	C.127	N.128	H.128 #MAP	117
77	125.460	55.950	173.410	108.060	8.374	1 U	5.500000E+00	0.000000E+00	e 0	N.32	CA.32	C.32	N.33	H.33 #MAP	28
78	121.390	54.840	174.650	122.990	8.383	1 U	5.500000E+00	0.000000E+00	e 0	-	-	-	-	-	-
79	126.780	59.840	174.320	124.470	8.627	1 U	5.500000E+00	0.000000E+00	e 0	N.1	CA.1	C.1	N.2	H.2 #MAP	1
80	124.560	54.580	174.570	123.430	9.131	1 U	5.500000E+00	0.000000E+00	e 0	N.90	CA.90	C.90	N.91	H.91 #MAP	80
81	124.020	54.020	175.440	106.070	8.303	1 U	5.500000E+00	0.000000E+00	e 0	N.36	CA.36	C.36	N.37	H.37 #MAP	32
82	126.020	60.030	175.410	125.820	8.955	1 U	5.500000E+00	0.000000E+00	e 0	N.31	CA.31	C.31	N.32	H.32 #MAP	27
83	121.250	53.070	177.200	110.220	8.470	1 U	5.500000E+00	0.000000E+00	e 0	N.11	CA.11	C.11	N.12	H.12 #MAP	11
84	117.640	52.110	177.150	113.640	8.949	1 U	5.500000E+00	0.000000E+00	e 0	N.86	CA.86	C.86	N.87	H.87 #MAP	76
85	119.000	53.970	175.380	110.720	8.087	1 U	5.500000E+00	0.000000E+00	e 0	N.26	CA.26	C.26	N.27	H.27 #MAP	24
86	119.550	54.820	171.410	119.100	6.881	1 U	5.500000E+00	0.000000E+00	e 0	-	-	-	-	-	-
87	123.250	64.280	178.340	111.990	7.983	1 U	5.500000E+00	0.000000E+00	e 0	N.103	CA.103	C.103	N.104	H.104 #MAP	93
87	123.250	64.280	178.340	111.990	7.983	1 U	5.500000E+00	0.000000E+00	e 0	N.106	CA.106	C.106	N.107	H.107 #MAP	96
88	107.170	45.760	171.010	117.790	8.069	1 U	5.500000E+00	0.000000E+00	e 0	N.85	CA.85	C.85	N.86	H.86 #MAP	75
89	115.180	53.770	171.690	111.700	8.137	1 U	5.500000E+00	0.000000E+00	e 0	-	-	-	-	-	-
90	113.460	57.360	177.220	117.240	7.834	1 U	5.500000E+00	0.000000E+00	e 0	N.98	CA.98	C.98	N.99	H.99 #MAP	88
91	120.440	53.300	176.370	107.350	8.264	1 U	5.500000E+00	0.000000E+00	e 0	N.84	CA.84	C.84	N.85	H.85 #MAP	74
91	120.440	53.300	176.370	107.350	8.264	1 U	5.500000E+00	0.000000E+00	e 0	N.140	CA.140	C.140	N.141	H.141 #MAP	130
92	108.600	51.410	179.660	108.290	7.254	1 U	5.500000E+00	0.000000E+00	e 0	N.129	CA.129	C.129	N.130	H.130 #MAP	119

**Table S11: Peak list from (H)NCOCANH APSY experiment on Cu<sup>I</sup>,Zn-SOD**

# Number of dimensions 5  
#FORMAT xeasy5D

#INAME 1 N  
#INAME 2 O  
#INAME 3 A  
#INAME 4 M  
#INAME 5 J

#SPECTRUM hNCOCANH N O A M J

1	126.510	175.480	61.120	123.780	9.094 1 U	5.500000E+00	0.000000E+00 e 0 N.8	C.7	CA.7	N.7	H.7 #MAP	6
1	126.510	175.480	61.120	123.780	9.094 1 U	5.500000E+00	0.000000E+00 e 0 N.92	C.91	CA.91	N.91	H.91 #MAP	80
2	121.010	173.520	60.840	125.450	8.930 1 U	5.500000E+00	0.000000E+00 e 0 N.119	C.118	CA.118	N.118	H.118 #MAP	107
3	110.940	175.040	50.920	114.440	7.359 1 U	5.500000E+00	0.000000E+00 e 0 N.146	C.145	CA.145	N.145	H.145 #MAP	134
4	122.930	176.020	62.430	120.230	7.338 1 U	5.500000E+00	0.000000E+00 e 0 N.15	C.14	CA.14	N.14	H.14 #MAP	11
5	124.380	171.120	45.490	106.700	8.032 1 U	5.500000E+00	0.000000E+00 e 0 N.148	C.147	CA.147	N.147	H.147 #MAP	136
6	120.470	172.050	55.600	121.260	7.143 1 U	5.500000E+00	0.000000E+00 e 0 N.84	C.83	CA.83	N.83	H.83 #MAP	72
6	120.470	172.050	55.600	121.260	7.143 1 U	5.500000E+00	0.000000E+00 e 0 N.140	C.139	CA.139	N.139	H.139 #MAP	128
7	129.800	173.100	62.230	112.690	6.664 1 U	5.500000E+00	0.000000E+00 e 0 N.117	C.116	CA.116	N.116	H.116 #MAP	105
8	118.350	175.400	53.960	119.000	6.989 1 U	5.500000E+00	0.000000E+00 e 0 N.80	C.79	CA.79	N.79	H.79 #MAP	68
9	122.220	176.370	51.610	121.110	7.743 1 U	5.500000E+00	0.000000E+00 e 0 N.124	C.123	CA.123	N.123	H.123 #MAP	112
10	125.870	174.380	61.090	120.640	7.986 1 U	5.500000E+00	0.000000E+00 e 0 N.18	C.17	CA.17	N.17	H.17 #MAP	14
10	125.870	174.380	61.090	120.640	7.986 1 U	5.500000E+00	0.000000E+00 e 0 N.143	C.142	CA.142	N.142	H.142 #MAP	131
11	121.280	173.690	45.420	97.750	8.458 1 U	5.500000E+00	0.000000E+00 e 0 N.83	C.82	CA.82	N.82	H.82 #MAP	71
11	121.280	173.690	45.420	97.750	8.458 1 U	5.500000E+00	0.000000E+00 e 0 N.139	C.138	CA.138	N.138	H.138 #MAP	127
12	116.760	175.170	61.880	127.460	8.613 1 U	5.500000E+00	0.000000E+00 e 0 N.150	C.149	CA.149	N.149	H.149 #MAP	138
13	120.560	173.310	42.780	113.790	8.748 1 U	5.500000E+00	0.000000E+00 e 0 N.42	C.41	CA.41	N.41	H.41 #MAP	36
13	120.560	173.310	42.780	113.790	8.748 1 U	5.500000E+00	0.000000E+00 e 0 N.57	C.56	CA.56	N.56	H.56 #MAP	51
14	118.420	172.800	44.640	112.850	9.551 1 U	5.500000E+00	0.000000E+00 e 0 N.115	C.114	CA.114	N.114	H.114 #MAP	103
15	125.310	174.970	49.670	131.410	9.686 1 U	5.500000E+00	0.000000E+00 e 0 N.96	C.95	CA.95	N.95	H.95 #MAP	84
16	118.270	174.310	55.630	129.470	10.170 1 U	5.500000E+00	0.000000E+00 e 0 N.49	C.48	CA.48	N.48	H.48 #MAP	43
17	107.330	176.510	53.280	120.480	7.158 1 U	5.500000E+00	0.000000E+00 e 0 N.85	C.84	CA.84	N.84	H.84 #MAP	73
17	107.330	176.510	53.280	120.480	7.158 1 U	5.500000E+00	0.000000E+00 e 0 N.141	C.140	CA.140	N.140	H.140 #MAP	129
18	120.890	175.720	52.020	115.850	8.127 1 U	5.500000E+00	0.000000E+00 e 0 N.47	C.46	CA.46	N.46	H.46 #MAP	41
19	104.220	177.190	54.500	117.010	10.445 1 U	5.500000E+00	0.000000E+00 e 0 N.127	C.126	CA.126	N.126	H.126 #MAP	115
20	112.860	176.870	60.910	121.550	7.578 1 U	5.500000E+00	0.000000E+00 e 0 N.114	C.113	CA.113	N.113	H.113 #MAP	102
21	125.960	171.120	59.640	121.740	9.570 1 U	5.500000E+00	0.000000E+00 e 0 N.6	C.5	CA.5	N.5	H.5 #MAP	4
22	124.380	177.550	49.160	129.120	9.334 1 U	5.500000E+00	0.000000E+00 e 0 N.90	C.89	CA.89	N.89	H.89 #MAP	78
23	120.390	170.920	46.250	107.310	8.324 1 U	5.500000E+00	0.000000E+00 e 0 N.17	C.16	CA.16	N.16	H.16 #MAP	13
23	120.390	170.920	46.250	107.310	8.324 1 U	5.500000E+00	0.000000E+00 e 0 N.142	C.141	CA.141	N.141	H.141 #MAP	130
24	123.600	172.200	56.260	126.200	8.688 1 U	5.500000E+00	0.000000E+00 e 0 N.19	C.18	CA.18	N.18	H.18 #MAP	15
25	107.230	174.520	53.510	123.000	7.944 1 U	5.500000E+00	0.000000E+00 e 0 N.16	C.15	CA.15	N.15	H.15 #MAP	12
26	118.040	174.820	58.620	113.660	8.946 1 U	5.500000E+00	0.000000E+00 e 0 N.88	C.87	CA.87	N.87	H.87 #MAP	76
27	126.560	171.870	46.770	109.380	8.873 1 U	5.500000E+00	0.000000E+00 e 0 N.45	C.44	CA.44	N.44	H.44 #MAP	39
28	127.400	175.400	62.350	124.430	8.452 1 U	5.500000E+00	0.000000E+00 e 0 N.149	C.148	CA.148	N.148	H.148 #MAP	137
29	123.770	174.960	50.270	126.070	9.170 1 U	5.500000E+00	0.000000E+00 e 0 N.7	C.6	CA.6	N.6	H.6 #MAP	5
30	115.260	171.450	45.590	119.470	10.209 1 U	5.500000E+00	0.000000E+00 e 0 N.52	C.51	CA.51	N.51	H.51 #MAP	46
31	114.790	177.380	55.070	120.570	8.282 1 U	5.500000E+00	0.000000E+00 e 0 N.43	C.42	CA.42	N.42	H.42 #MAP	37
32	109.340	174.790	53.860	114.750	8.698 1 U	5.500000E+00	0.000000E+00 e 0 N.44	C.43	CA.43	N.43	H.43 #MAP	38
33	106.190	175.520	54.030	124.190	9.115 1 U	5.500000E+00	0.000000E+00 e 0 N.37	C.36	CA.36	N.36	H.36 #MAP	31
34	120.320	173.410	42.860	116.790	9.282 1 U	5.500000E+00	0.000000E+00 e 0 N.151	C.150	CA.150	N.150	H.150 #MAP	139
35	105.480	171.830	44.030	113.130	7.216 1 U	5.500000E+00	0.000000E+00 e 0 N.73	C.72	CA.72	N.72	H.72 #MAP	63
36	112.690	174.370	54.340	118.480	7.433 1 U	5.500000E+00	0.000000E+00 e 0 N.116	C.115	CA.115	N.115	H.115 #MAP	104
37	129.400	177.150	61.440	120.890	9.395 1 U	5.500000E+00	0.000000E+00 e 0 N.48	C.47	CA.47	N.47	H.47 #MAP	42
38	129.000	178.960	53.250	118.370	8.320 1 U	5.500000E+00	0.000000E+00 e 0 N.81	C.80	CA.80	N.80	H.80 #MAP	69
39	121.810	175.970	56.220	125.520	8.933 1 U	5.500000E+00	0.000000E+00 e 0 N.24	C.23	CA.23	N.23	H.23 #MAP	20
39	121.810	175.970	56.220	125.520	8.933 1 U	5.500000E+00	0.000000E+00 e 0 N.121	C.120	CA.120	N.120	H.120 #MAP	109
40	130.170	173.580	55.670	121.550	9.528 1 U	5.500000E+00	0.000000E+00 e 0 N.22	C.21	CA.21	N.21	H.21 #MAP	18
41	115.860	172.190	55.680	126.570	8.114 1 U	5.500000E+00	0.000000E+00 e 0 N.46	C.45	CA.45	N.45	H.45 #MAP	40
42	131.310	176.690	60.920	118.760	7.788 1 U	5.500000E+00	0.000000E+00 e 0 N.95	C.94	CA.94	N.94	H.94 #MAP	83
43	101.810	177.010	52.850	119.690	6.915 1 U	5.500000E+00	0.000000E+00 e 0 N.61	C.60	CA.60	N.60	H.60 #MAP	55
44	113.100	174.710	55.370	112.370	7.162 1 U	5.500000E+00	0.000000E+00 e 0 N.72	C.71	CA.71	N.71	H.71 #MAP	62
45	119.070	171.560	54.780	119.780	8.067 1 U	5.500000E+00	0.000000E+00 e 0 N.79	C.78	CA.78	N.78	H.78 #MAP	67
46	114.540	171.630	44.210	108.130	8.352 1 U	5.500000E+00	0.000000E+00 e 0 N.34	C.33	CA.33	N.33	H.33 #MAP	28
47	110.510	174.880	60.380	118.260	8.531 1 U	5.500000E+00	0.000000E+00 e 0 N.50	C.49	CA.49	N.49	H.49 #MAP	44
47	110.510	174.880	60.380	118.260	8.531 1 U	5.500000E+00	0.000000E+00 e 0 N.59	C.58	CA.58	N.58	H.58 #MAP	53
48	127.270	175.020	56.470	121.790	9.374 1 U	5.500000E+00	0.000000E+00 e 0 N.65	C.64	CA.64	N.64	H.64 #MAP	57
49	118.080	174.770	45.530	108.030	7.260 1 U	5.500000E+00	0.000000E+00 e 0 N.131	C.130	CA.130	N.130	H.130 #MAP	119
50	115.570	175.440	58.730	121.880	9.191 1 U	5.500000E+00	0.000000E+00 e 0 N.25	C.24	CA.24	N.24	H.24 #MAP	21
50	115.570	175.440	58.730	121.880	9.191 1 U	5.500000E+00	0.000000E+00 e 0 N.122	C.121	CA.121	N.121	H.121 #MAP	110
51	112.440	173.230	55.430	118.270	8.675 1 U	5.500000E+00	0.000000E+00 e 0 N.71	C.70	CA.70	N.70	H.70 #MAP	61
51	112.440	173.230	55.430	118.270	8.675 1 U	5.500000E+00	0.000000E+00 e 0 N.111	C.110	CA.110	N.110	H.110 #MAP	99
52	113.930	176.200	57.100	125.990	8.721 1 U	5.500000E+00	0.000000E+00 e 0 N.41	C.40	CA.40	N.40	H.40 #MAP	35
53	123.200	172.670	56.730	114.530	7.878 1 U	5.500000E+00	0.000000E+00 e 0 N.35	C.34	CA.34	N.34	H.34 #MAP	29
54	108.060	173.450	55.910	125.460	8.964 1 U	5.500000E+00	0.000000E+00 e 0 N.33	C.32	CA.32	N.32	H.32 #MAP	27
55	121.490	176.430	54.900	114.230	8.392 1 U	5.500000E+00	0.000000E+00 e 0 N.21	C.20	CA.20	N.20	H.20 #MAP	17
56	120.650	174.610	45.140	106.080	8.269 1 U	5.500000E+00	0.000000E+00 e 0 N.38	C.37	CA.37	N.37	H.37 #MAP	32

57	118.510	176.860	45.430	104.150	8.670	1 U	5.500000E+00	0.000000E+00	e 0 N.128	C.127	CA.127	N.127	H.127 #MAP	116
58	111.860	178.890	64.270	123.510	8.106	1 U	5.500000E+00	0.000000E+00	e 0 N.134	C.133	CA.133	N.133	H.133 #MAP	122
59	120.900	175.670	53.900	126.590	8.892	1 U	5.500000E+00	0.000000E+00	e 0 N.9	C.8	CA.8	N.8	H.8 #MAP	7
60	125.730	175.410	55.510	127.500	9.202	1 U	5.500000E+00	0.000000E+00	e 0 N.31	C.30	CA.30	N.30	H.30 #MAP	25
61	126.200	175.180	54.900	124.420	8.610	1 U	5.500000E+00	0.000000E+00	e 0 N.3	C.2	CA.2	N.2	H.2 #MAP	1
61	126.200	175.180	54.900	124.420	8.610	1 U	5.500000E+00	0.000000E+00	e 0 N.69	C.68	CA.68	N.68	H.68 #MAP	59
62	111.290	175.770	54.720	120.880	8.252	1 U	5.500000E+00	0.000000E+00	e 0 N.10	C.9	CA.9	N.9	H.9 #MAP	8
63	114.710	177.730	55.880	123.110	8.421	1 U	5.500000E+00	0.000000E+00	e 0 N.145	C.144	CA.144	N.144	H.144 #MAP	133
64	121.530	174.550	60.190	117.100	7.843	1 U	5.500000E+00	0.000000E+00	e 0 N.100	C.99	CA.99	N.99	H.99 #MAP	88
64	121.530	174.550	60.190	117.100	7.843	1 U	5.500000E+00	0.000000E+00	e 0 N.113	C.112	CA.112	N.112	H.112 #MAP	101
65	114.170	172.140	51.870	123.640	8.733	1 U	5.500000E+00	0.000000E+00	e 0 N.20	C.19	CA.19	N.19	H.19 #MAP	16
66	118.570	174.590	54.130	126.190	9.300	1 U	5.500000E+00	0.000000E+00	e 0 N.70	C.69	CA.69	N.69	H.69 #MAP	60
67	128.120	176.890	51.790	115.320	5.827	1 U	5.500000E+00	0.000000E+00	e 0 N.53	C.52	CA.52	N.52	H.52 #MAP	47
68	129.130	173.700	61.470	118.100	8.657	1 U	5.500000E+00	0.000000E+00	e 0 N.89	C.88	CA.88	N.88	H.88 #MAP	77
69	121.700	175.390	50.300	123.040	9.144	1 U	5.500000E+00	0.000000E+00	e 0 N.5	C.4	CA.4	N.4	H.4 #MAP	3
70	125.440	175.580	60.040	126.010	9.239	1 U	5.500000E+00	0.000000E+00	e 0 N.32	C.31	CA.31	N.31	H.31 #MAP	26
71	130.100	175.450	63.130	120.400	8.950	1 U	5.500000E+00	0.000000E+00	e 0 N.152	C.151	CA.151	N.151	H.151 #MAP	140
72	108.090	172.670	62.750	111.850	7.961	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
73	125.420	174.420	53.200	129.800	8.835	1 U	5.500000E+00	0.000000E+00	e 0 N.118	C.117	CA.117	N.117	H.117 #MAP	106
74	121.050	174.700	53.410	115.530	8.476	1 U	5.500000E+00	0.000000E+00	e 0 N.123	C.122	CA.122	N.122	H.122 #MAP	111
75	124.270	174.500	59.750	125.980	9.424	1 U	5.500000E+00	0.000000E+00	e 0 N.68	C.67	CA.67	N.67	H.67 #MAP	58
76	120.540	175.910	54.100	120.610	7.462	1 U	5.500000E+00	0.000000E+00	e 0 N.77	C.76	CA.76	N.76	H.76 #MAP	65
77	106.650	173.310	53.840	110.970	9.004	1 U	5.500000E+00	0.000000E+00	e 0 N.147	C.146	CA.146	N.146	H.146 #MAP	135
78	119.710	177.320	58.150	120.730	8.323	1 U	5.500000E+00	0.000000E+00	e 0 N.78	C.77	CA.77	N.77	H.77 #MAP	66
79	125.480	174.360	57.940	121.000	7.990	1 U	5.500000E+00	0.000000E+00	e 0 N.120	C.119	CA.119	N.119	H.119 #MAP	108
80	127.530	175.570	60.430	121.290	9.015	1 U	5.500000E+00	0.000000E+00	e 0 N.30	C.29	CA.29	N.29	H.29 #MAP	24
81	124.280	172.980	59.660	123.080	8.520	1 U	5.500000E+00	0.000000E+00	e 0 N.36	C.35	CA.35	N.35	H.35 #MAP	30
82	117.030	178.350	53.830	129.420	10.009	1 U	5.500000E+00	0.000000E+00	e 0 N.126	C.125	CA.125	N.125	H.125 #MAP	114
83	118.710	172.970	46.470	111.140	8.432	1 U	5.500000E+00	0.000000E+00	e 0 N.94	C.93	CA.93	N.93	H.93 #MAP	82
84	121.230	172.960	45.120	111.340	8.909	1 U	5.500000E+00	0.000000E+00	e 0 N.11	C.10	CA.10	N.10	H.10 #MAP	9
85	120.420	176.930	54.700	115.210	8.600	1 U	5.500000E+00	0.000000E+00	e 0 N.76	C.75	CA.75	N.75	H.75 #MAP	64
86	126.160	176.240	61.020	110.610	8.460	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
87	105.910	177.150	58.280	122.790	7.711	1 U	5.500000E+00	0.000000E+00	e 0 N.137	C.136	CA.136	N.136	H.136 #MAP	125
88	126.110	175.690	53.630	125.510	8.510	1 U	5.500000E+00	0.000000E+00	e 0 N.97	C.96	CA.96	N.96	H.96 #MAP	85
89	123.150	172.890	57.990	118.500	8.901	1 U	5.500000E+00	0.000000E+00	e 0 N.103	C.102	CA.102	N.102	H.102 #MAP	91
89	123.150	172.890	57.990	118.500	8.901	1 U	5.500000E+00	0.000000E+00	e 0 N.106	C.105	CA.105	N.105	H.105 #MAP	94
90	97.840	176.650	66.610	129.020	7.904	1 U	5.500000E+00	0.000000E+00	e 0 N.82	C.81	CA.81	N.81	H.81 #MAP	70
91	110.660	177.070	52.870	120.540	8.116	1 U	5.500000E+00	0.000000E+00	e 0 N.39	C.38	CA.38	N.38	H.38 #MAP	33
92	118.580	178.360	53.770	118.170	8.910	1 U	5.500000E+00	0.000000E+00	e 0 N.132	C.131	CA.131	N.131	H.131 #MAP	120
93	123.390	176.520	61.520	125.890	8.647	1 U	5.500000E+00	0.000000E+00	e 0 N.4	C.3	CA.3	N.3	H.3 #MAP	2
94	121.720	175.140	54.700	115.540	7.551	1 U	5.500000E+00	0.000000E+00	e 0 N.64	C.63	CA.63	N.63	H.63 #MAP	56
95	129.460	177.590	52.190	122.240	10.151	1 U	5.500000E+00	0.000000E+00	e 0 N.125	C.124	CA.124	N.124	H.124 #MAP	113

**Table S12: Peak list from (H)NCACONH APSY experiment on  $\beta$ 2m<sub>D76N</sub>**

# Number of dimensions 5

#FORMAT xeasy5D

#INAME 1 M

#INAME 2 A

#INAME 3 O

#INAME 4 N

#INAME 5 H

#SPECTRUM hNCACONH M A O N H

1	126.300	53.200	175.740	108.970	7.345	1 U	5.500000E+00	0.000000E+00	e 0 N.38	CA.38	C.38	N.39	H.39 #MAP	17
2	116.730	55.310	173.160	125.050	7.687	1 U	5.500000E+00	0.000000E+00	e 0 N.61	CA.61	C.61	N.62	H.62 #MAP	40
3	121.140	55.820	175.870	112.460	7.932	1 U	5.500000E+00	0.000000E+00	e 0 N.40	CA.40	C.40	N.41	H.41 #MAP	19
4	109.010	44.790	170.590	121.210	9.202	1 U	5.500000E+00	0.000000E+00	e 0 N.29	CA.29	C.29	N.30	H.30 #MAP	10
5	129.230	59.430	174.500	112.510	8.173	1 U	5.500000E+00	0.000000E+00	e 0 N.56	CA.56	C.56	N.57	H.57 #MAP	35
6	130.400	59.600	174.330	113.600	6.961	1 U	5.500000E+00	0.000000E+00	e 0 N.67	CA.67	C.67	N.68	H.68 #MAP	46
7	113.880	52.430	172.190	107.270	8.599	1 U	5.500000E+00	0.000000E+00	e 0 N.42	CA.42	C.42	N.43	H.43 #MAP	21
8	115.140	57.960	174.540	113.750	9.121	1 U	5.500000E+00	0.000000E+00	e 0 N.58	CA.58	C.58	N.59	H.59 #MAP	37
9	119.760	56.580	174.420	126.330	8.399	1 U	5.500000E+00	0.000000E+00	e 0 N.37	CA.37	C.37	N.38	H.38 #MAP	16
10	108.970	53.170	176.860	121.260	8.571	1 U	5.500000E+00	0.000000E+00	e 0 N.24	CA.24	C.24	N.25	H.25 #MAP	5
10	108.970	53.170	176.860	121.260	8.571	1 U	5.500000E+00	0.000000E+00	e 0 N.39	CA.39	C.39	N.40	H.40 #MAP	18
11	112.530	59.440	172.120	115.160	7.736	1 U	5.500000E+00	0.000000E+00	e 0 N.57	CA.57	C.57	N.58	H.58 #MAP	36
12	113.730	52.330	174.480	127.650	8.950	1 U	5.500000E+00	0.000000E+00	e 0 N.59	CA.59	C.59	N.60	H.60 #MAP	38
13	116.650	52.960	173.230	127.290	9.252	1 U	5.500000E+00	0.000000E+00	e 0 N.63	CA.63	C.63	N.64	H.64 #MAP	42
14	112.510	55.450	173.570	113.970	8.291	1 U	5.500000E+00	0.000000E+00	e 0 N.41	CA.41	C.41	N.42	H.42 #MAP	20
15	127.590	54.980	172.860	130.980	9.677	1 U	5.500000E+00	0.000000E+00	e 0 N.69	CA.69	C.69	N.70	H.70 #MAP	48
16	125.020	59.880	174.680	116.590	8.865	1 U	5.500000E+00	0.000000E+00	e 0 N.62	CA.62	C.62	N.63	H.63 #MAP	41
17	128.570	54.510	174.030	119.830	9.166	1 U	5.500000E+00	0.000000E+00	e 0 N.36	CA.36	C.36	N.37	H.37 #MAP	15
18	126.500	54.000	174.250	122.310	8.629	1 U	5.500000E+00	0.000000E+00	e 0 N.47	CA.47	C.47	N.48	H.48 #MAP	26
19	127.070	50.830	177.050	116.720	8.057	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-

20	127.640	55.920	175.790	116.740	9.312	1 U	5.500000E+00	0.000000E+00	e 0 N.60	CA.60	C.60	N.61	H.61 #MAP	39
21	121.410	54.550	173.570	125.640	9.416	1 U	5.500000E+00	0.000000E+00	e 0 N.25	CA.25	C.25	N.26	H.26 #MAP	6
22	122.120	54.190	174.430	128.700	8.665	1 U	5.500000E+00	0.000000E+00	e 0 N.48	CA.48	C.48	N.49	H.49 #MAP	27
23	121.660	54.320	173.140	113.840	7.232	1 U	5.500000E+00	0.000000E+00	e 0 N.22	CA.22	C.22	N.23	H.23 #MAP	3
24	126.660	55.000	174.010	129.510	9.585	1 U	5.500000E+00	0.000000E+00	e 0 N.74	CA.74	C.74	N.75	H.75 #MAP	51
25	127.400	55.080	174.620	121.630	8.966	1 U	5.500000E+00	0.000000E+00	e 0 N.64	CA.64	C.64	N.65	H.65 #MAP	43
26	128.610	61.580	173.880	129.030	9.307	1 U	5.500000E+00	0.000000E+00	e 0 N.49	CA.49	C.49	N.50	H.50 #MAP	28
27	107.150	46.240	171.890	118.030	8.311	1 U	5.500000E+00	0.000000E+00	e 0 N.43	CA.43	C.43	N.44	H.44 #MAP	22
28	118.240	58.480	175.260	128.610	8.649	1 U	5.500000E+00	0.000000E+00	e 0 N.35	CA.35	C.35	N.36	H.36 #MAP	14
29	122.300	51.390	174.280	121.910	8.270	1 U	5.500000E+00	0.000000E+00	e 0 N.21	CA.21	C.21	N.22	H.22 #MAP	2
30	121.230	51.900	174.020	126.960	8.949	1 U	5.500000E+00	0.000000E+00	e 0 N.30	CA.30	C.30	N.31	H.31 #MAP	11
31	119.610	56.320	175.270	118.060	7.333	1 U	5.500000E+00	0.000000E+00	e 0 N.34	CA.34	C.34	N.35	H.35 #MAP	13
31	119.610	56.320	175.270	118.060	7.333	1 U	5.500000E+00	0.000000E+00	e 0 N.52	CA.52	C.52	N.53	H.53 #MAP	31
32	117.890	62.400	173.060	126.530	9.103	1 U	5.500000E+00	0.000000E+00	e 0 N.73	CA.73	C.73	N.74	H.74 #MAP	50
33	123.350	55.740	175.900	117.520	8.580	1 U	5.500000E+00	0.000000E+00	e 0 N.54	CA.54	C.54	N.55	H.55 #MAP	33
34	114.690	56.280	173.620	119.330	8.193	1 U	5.500000E+00	0.000000E+00	e 0 N.33	CA.33	C.33	N.34	H.34 #MAP	12
35	114.020	54.470	174.880	109.150	7.998	1 U	5.500000E+00	0.000000E+00	e 0 N.23	CA.23	C.23	N.24	H.24 #MAP	4
36	131.130	56.720	172.630	119.510	9.126	1 U	5.500000E+00	0.000000E+00	e 0 N.70	CA.70	C.70	N.71	H.71 #MAP	49
37	113.490	62.280	172.120	127.550	8.664	1 U	5.500000E+00	0.000000E+00	e 0 N.68	CA.68	C.68	N.69	H.69 #MAP	47
38	121.620	55.800	171.940	121.600	8.620	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
39	119.560	54.440	173.850	124.400	9.103	1 U	5.500000E+00	0.000000E+00	e 0 N.45	CA.45	C.45	N.46	H.46 #MAP	24
40	119.920	54.330	175.310	109.030	8.680	1 U	5.500000E+00	0.000000E+00	e 0 N.28	CA.28	C.28	N.29	H.29 #MAP	9
41	124.350	59.930	174.560	126.260	8.710	1 U	5.500000E+00	0.000000E+00	e 0 N.46	CA.46	C.46	N.47	H.47 #MAP	25
42	124.230	55.770	171.780	124.610	9.049	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
43	117.920	55.450	175.690	119.230	8.471	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
44	126.640	59.480	172.730	119.380	8.545	1 U	5.500000E+00	0.000000E+00	e 0 N.51	CA.51	C.51	N.52	H.52 #MAP	30
45	127.870	57.380	172.950	123.400	8.691	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-

**Table S13: Peak list from (H)NCOCANH APSY experiment on  $\beta 2m_{D76N}$**

# Number of dimensions 5

#FORMAT xeasy5D

#INAME 1 N

#INAME 2 O

#INAME 3 A

#INAME 4 M

#INAME 5 J

#SPECTRUM hNCOCANH N O A M J

1	121.066	176.934	53.197	108.954	7.337	1 U	5.500000E+00	0.000000E+00	e 0 N.40	C.39	CA.39	N.39	H.39 #MAP	18
2	116.630	174.682	59.862	125.089	7.680	1 U	5.500000E+00	0.000000E+00	e 0 N.63	C.62	CA.62	N.62	H.62 #MAP	41
3	113.819	173.586	55.462	112.428	7.930	1 U	5.500000E+00	0.000000E+00	e 0 N.42	C.41	CA.41	N.41	H.41 #MAP	20
4	108.942	175.787	53.217	126.371	8.388	1 U	5.500000E+00	0.000000E+00	e 0 N.39	C.38	CA.38	N.38	H.38 #MAP	17
5	127.630	174.509	52.390	113.723	9.125	1 U	5.500000E+00	0.000000E+00	e 0 N.60	C.59	CA.59	N.59	H.59 #MAP	38
6	121.123	170.598	44.830	109.061	8.663	1 U	5.500000E+00	0.000000E+00	e 0 N.30	C.29	CA.29	N.29	H.29 #MAP	10
7	116.667	175.804	55.933	127.642	8.951	1 U	5.500000E+00	0.000000E+00	e 0 N.61	C.60	CA.60	N.60	H.60 #MAP	39
8	119.730	174.037	54.543	128.545	8.642	1 U	5.500000E+00	0.000000E+00	e 0 N.37	C.36	CA.36	N.36	H.36 #MAP	15
9	127.511	172.130	62.243	113.506	6.953	1 U	5.500000E+00	0.000000E+00	e 0 N.69	C.68	CA.68	N.68	H.68 #MAP	47
10	113.698	174.564	57.977	115.146	7.732	1 U	5.500000E+00	0.000000E+00	e 0 N.59	C.58	CA.58	N.58	H.58 #MAP	37
11	115.099	172.099	59.469	112.468	8.170	1 U	5.500000E+00	0.000000E+00	e 0 N.58	C.57	CA.57	N.57	H.57 #MAP	36
12	121.600	174.572	55.107	127.237	9.238	1 U	5.500000E+00	0.000000E+00	e 0 N.65	C.64	CA.64	N.64	H.64 #MAP	43
13	119.366	175.493	53.836	129.452	9.604	1 U	5.500000E+00	0.000000E+00	e 0 N.76	C.75	CA.75	N.75	H.75 #MAP	52
14	107.248	172.149	52.457	113.956	8.288	1 U	5.500000E+00	0.000000E+00	e 0 N.43	C.42	CA.42	N.42	H.42 #MAP	21
15	125.061	173.156	55.308	116.697	9.297	1 U	5.500000E+00	0.000000E+00	e 0 N.62	C.61	CA.61	N.61	H.61 #MAP	40
16	112.587	175.984	55.879	121.389	8.579	1 U	5.500000E+00	0.000000E+00	e 0 N.41	C.40	CA.40	N.40	H.40 #MAP	19
17	126.312	174.449	56.605	119.823	9.160	1 U	5.500000E+00	0.000000E+00	e 0 N.38	C.37	CA.37	N.37	H.37 #MAP	16
18	127.210	173.291	52.944	116.616	8.865	1 U	5.500000E+00	0.000000E+00	e 0 N.64	C.63	CA.63	N.63	H.63 #MAP	42
19	112.551	174.491	59.425	129.279	8.355	1 U	5.500000E+00	0.000000E+00	e 0 N.57	C.56	CA.56	N.56	H.56 #MAP	35
20	118.159	175.237	56.094	119.454	8.203	1 U	5.500000E+00	0.000000E+00	e 0 N.35	C.34	CA.34	N.34	H.34 #MAP	13
20	118.159	175.237	56.094	119.454	8.203	1 U	5.500000E+00	0.000000E+00	e 0 N.77	C.76	CA.76	N.76	H.76 #MAP	53
21	128.825	173.961	61.553	128.806	8.671	1 U	5.500000E+00	0.000000E+00	e 0 N.50	C.49	CA.49	N.49	H.49 #MAP	28
22	128.506	175.293	58.332	118.227	7.327	1 U	5.500000E+00	0.000000E+00	e 0 N.36	C.35	CA.35	N.35	H.35 #MAP	14
23	126.466	173.320	62.406	117.963	8.762	1 U	5.500000E+00	0.000000E+00	e 0 N.74	C.73	CA.73	N.73	H.73 #MAP	50
24	119.721	172.693	56.925	131.013	9.690	1 U	5.500000E+00	0.000000E+00	e 0 N.71	C.70	CA.70	N.70	H.70 #MAP	49
25	119.392	175.926	59.191	118.095	8.296	1 U	5.500000E+00	0.000000E+00	e 0 N.45	C.44	CA.44	N.44	H.44 #MAP	23
25	119.392	175.926	59.191	118.095	8.296	1 U	5.500000E+00	0.000000E+00	e 0 N.78	C.77	CA.77	N.77	H.77 #MAP	54
26	129.317	174.112	55.003	126.598	9.091	1 U	5.500000E+00	0.000000E+00	e 0 N.75	C.74	CA.74	N.74	H.74 #MAP	51
27	122.455	172.874	55.741	119.471	8.620	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
28	125.877	174.739	59.997	124.384	9.085	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
29	109.275	174.748	54.424	113.720	7.228	1 U	5.500000E+00	0.000000E+00	e 0 N.24	C.23	CA.23	N.23	H.23 #MAP	4
30	122.107	174.276	53.994	126.323	8.677	1 U	5.500000E+00	0.000000E+00	e 0 N.48	C.47	CA.47	N.47	H.47 #MAP	26
31	116.865	173.791	60.499	116.552	8.076	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-
32	117.984	171.924	46.239	107.223	8.590	1 U	5.500000E+00	0.000000E+00	e 0 N.44	C.43	CA.43	N.43	H.43 #MAP	22
33	125.373	174.474	52.763	121.628	8.977	1 U	5.500000E+00	0.000000E+00	e 0 N.66	C.65	CA.65	N.65	H.65 #MAP	44

34	119.580	172.951	59.300	127.316	8.437	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-	-	-	-	-	-	-
35	117.163	175.847	55.645	123.239	9.374	1 U	5.500000E+00	0.000000E+00	e 0 N.55	C.54	CA.54	N.54	H.54 #MAP	33					
36	125.699	173.522	54.594	121.320	8.505	1 U	5.500000E+00	0.000000E+00	e 0 N.26	C.25	CA.25	N.25	H.25 #MAP	6					
37	129.312	173.085	54.420	129.249	9.330	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-						
38	130.965	176.483	56.511	117.602	8.641	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-						
39	128.977	172.754	56.128	121.767	8.630	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-						
40	113.584	174.475	59.566	130.345	9.043	1 U	5.500000E+00	0.000000E+00	e 0 N.68	C.67	CA.67	N.67	H.67 #MAP	46					
41	126.749	174.003	51.915	121.179	9.209	1 U	5.500000E+00	0.000000E+00	e 0 N.31	C.30	CA.30	N.30	H.30 #MAP	11					
42	124.521	177.566	56.578	126.654	8.662	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-						
43	119.970	172.978	59.557	127.501	8.573	1 U	5.500000E+00	0.000000E+00	e 0 N.28	C.27	CA.27	N.27	H.27 #MAP	8					
44	113.881	173.171	54.373	121.446	8.243	1 U	5.500000E+00	0.000000E+00	e 0 N.23	C.22	CA.22	N.22	H.22 #MAP	3					
45	139.727	176.887	70.693	119.311	6.969	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-						
46	117.741	180.406	54.265	104.333	7.095	1 U	5.500000E+00	0.000000E+00	e 0 -	-	-	-	-						

## H. Details of FLYA calculations

FLYA module implemented in CYANA 3.98 software package was used. The experiments were defined in an auxiliary CYANA library as follows:

```

SPECTRUM hNCACONH M A O N H
0.980 H_AMI M:N_AMI A:C_ALI O:C_BYL N:N_AMI H:H_AMI

SPECTRUM hNCOCANH N O A M J
0.980 H_AMI N:N_AMI O:C_BYL A:C_ALI M:N_AMI (C_BYL) J:H_AMI

SPECTRUM ssCBcaNH HN N C
0.980 HN:H_AMI N:N_AMI (C_BYL) C_ALI C:C_ALI

```

The above definitions discard proline residues as a start of magnetization pathway. In the case of  $\beta 2M_{D76N}$  the amino acid sequence provided to FLYA was restricted to the structured fibril core (residues 20-81). Slightly modified definitions were used in this case to allow the magnetisation to originate from and be detected on (a non-N-terminal) residue S20 amide protons:

```

SPECTRUM ssCBcaNH HN N C
0.980 HN:H_AMI N:N_AMI C_ALI C:C_ALI

SPECTRUM hNCOCANH N O A M J
0.980 H_AMI N:N_AMI O:C_BYL A:C_ALI M:N_AMI J:H_AMI

```

In the case of GB1, FLYA results were compared to the reference resonance assignment reported by Andreas et al, and deposited in BMRB under accession number 30088.<sup>17</sup> For Cu<sup>I</sup>,Zn-SOD, we used an adapted resonance assignment obtained by Knight et al<sup>3</sup> (BMRB entry 18509). The shifts of <sup>13</sup>C' were recalibrated by +2.65 ppm to account for the difference in <sup>13</sup>C reference between DSS (assumed by FLYA) and TMS (as deposited in BMRB), while the shifts of <sup>13</sup>Ca were additionally increased by 0.66 ppm to roughly compensate <sup>2</sup>H-induced shifts. The reference resonance assignment of Cu<sup>I</sup>,Zn-SOD was then updated and completed based on four 3D experiments acquired on the same sample as the 5D datasets: (H)CANH, (H)(CO)CA(CO)NH, (H)CONH, (H)CO(CA)NH. The reference shifts for  $\beta 2M_{D76N}$  were obtained independently based on a set of six <sup>1</sup>H<sup>N</sup>-detected experiments: (H)CANH, (H)(CO)CA(CO)NH, (H)CONH, (H)CO(CA)NH, (H)(CA)CB(CA)NH and (H)(CA)CB(CA)(CO)NH.

**Table S14: FLYA parameters<sup>a</sup>**

Sample	Assignment tolerance			Tolerance for comparison with the reference		Population size
	$T_C$ , ppm ( <sup>13</sup> Ca, <sup>13</sup> C')	$T_N$ , ppm ( <sup>15</sup> N)	$T_H$ , ppm ( <sup>1</sup> H <sup>N</sup> )	$T_{CN}^{REF}$ , ppm	$T_H^{REF}$ , ppm	
GB1	0.25	0.35	0.10	0.25	0.10	50
SOD	0.20	0.40	0.10	0.40	0.10	1000
β2M <sub>D76N</sub>	0.20	0.40	0.05	0.40	0.10	200

<sup>a</sup> The parameter naming convention (highlighted in *blue*) as used in the following FLYA batch script (\*cya):  
structurepeaks:=

assignpeaks:=hNCAACONH,hNCOACANH

# Assignment tolerances

assigncs\_accH:=  $T_H$

assigncs\_accC:=  $T_C$

assigncs\_accN:=  $T_N$

# Tolerances for comparison with reference

tolerance:=  $T_H^{ref}$ ,  $T_H^{ref}$ ,  $T_{CN}^{ref}$

shiftassign\_population:= $N_{popul}$

shiftassign\_iterations:=15000

analyzeassign\_group := BB: N H CA C

randomseed := 3771

command select\_atoms

atom select "N H CA C"

end

read lib [apsy\\_experiment\\_definitions.lib](#) append

read seq [sequence\\_file.seq](#)

flya runs=20 assignpeaks=\$assignpeaks structure= shiftreference=[reference\\_shift\\_file.prot](#)

**Table S15: FLYA assignment summary**

Protein	Nucleus	Number of "strong" assignments				Max
		correct <sup>a</sup>	incompatible <sup>a</sup>	additional <sup>a</sup>	all	
GB1	<sup>1</sup> H <sup>N</sup>	55	0	0	55	55 <sup>b</sup>
	<sup>15</sup> N	55	0	1	56	56 <sup>c</sup>
	<sup>13</sup> Ca	55	0	0	55	56
	<sup>13</sup> C'	55	0	0	55	56
	all	220	0	1	221	223
SOD	<sup>1</sup> H <sup>N</sup>	65	7	3	75	127 <sup>b, e</sup>
	<sup>15</sup> N	74	3	0	77	127 <sup>c, e</sup>
	<sup>13</sup> Ca	71	6	1	78	133 <sup>e</sup>
	<sup>13</sup> C'	70	6	1	77	133 <sup>e</sup>
	all	280	22	5	307	520 <sup>e</sup>
β2M <sub>D76N</sub>	<sup>1</sup> H <sup>N</sup>	20	1	0	21	57 <sup>b, f</sup>
	<sup>15</sup> N	21	0	0	21	57 <sup>c, f</sup>
	<sup>13</sup> Ca	20	1	0	21	59 <sup>f</sup>
	<sup>13</sup> C'	26	3	0	29	59 <sup>f</sup>
	all	87	5	0	92	232 <sup>f</sup>

<sup>a</sup> With respect to the reference assignment

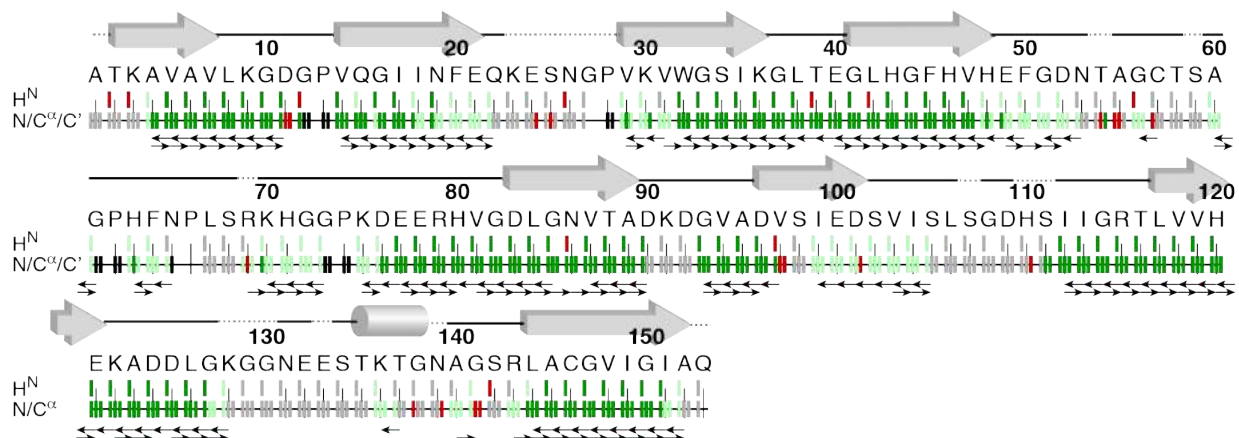
<sup>b</sup> The number of amide protons, i.e. excluding proline residues and N-terminus

<sup>c</sup> excluding proline <sup>15</sup>N atoms

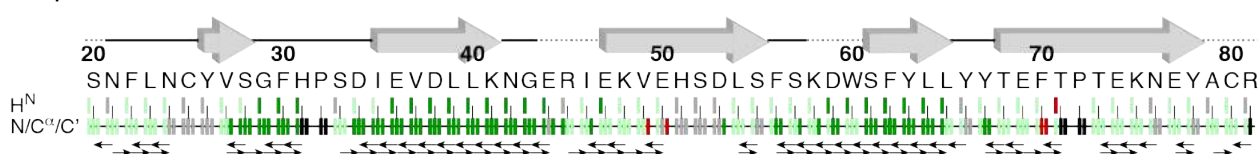
<sup>e</sup> discounting dynamic residues 1-3, 23-27, 54, 59, 69, 107, 110-111, 128-130, 133, 139, 153

<sup>f</sup> within the fibril core (residues 20-81), discounting dynamic residues 25, 51-52

### (a) SOD-Cu(I),Zn



### (b) $\beta 2m_{D76N}$



**Figure S3.** Graphical summary of FLYA automated resonance assignment of (a) Cu<sup>I</sup>,Zn-SOD and (b)  $\beta 2m_{D76N}$  based on the pairs of 5D data sets (i.e. forward and backward correlations). The same conventions for secondary structure elements and color-coding of resonances as for the main text Figure 5 are used. Black arrows denote the observed 5D correlations and define the upper limit for (manual) resonance assignment based on the available spectral data.

## I. FLYA results for SOD and $\beta 2m_{D76N}$ with additional C $\beta$ information

The calculations were performed with exactly same parameters as specified in the Table S15, with the following 2 exceptions:

1) An unassigned intraresidue  $^{13}\text{C}\beta$  ( $\delta$ ) –  $^{15}\text{N}$  ( $\delta$ ) –  $^1\text{H}^{\text{N}}$  ( $\delta$ ) peak list was provided

assignpeaks:=hNCACONH,hNCOCANH,hcaCBcaNH

In the case of Cu<sup>I</sup>,Zn-SOD, a synthetic peak list with 111 entries was constructed from published C $\beta$  shifts of Cu<sup>II</sup>,Zn-SOD,<sup>5</sup> and reference  $^{15}\text{N}$  and  $^1\text{H}^{\text{N}}$  shifts. In the case of  $\beta 2m_{D76N}$  an experimental, unfiltered peak list from a 3D (H)(CA)CB(CA)NH experiment (62 peaks) was used.

2) The respective lines in the FLYA batch script were adapted to allow C $\beta$  shifts in the spin system

```
analyzeassign_group := BB: N H CA CB C
command select_atoms
atom select "N H CA CB C"
end
```



**Table S16:** FLYA assignment summary (the calculations with additional C $\beta$  information)

Protein	Nucleus	Number of “strong” assignments				Max
		correct <sup>a</sup>	incompatible <sup>a</sup>	additional <sup>a</sup>	all	
SOD	<sup>1</sup> H <sup>N</sup>	90	3	3	96	127 <sup>b, e</sup>
	<sup>15</sup> N	89	3	3	95	127 <sup>c, e</sup>
	<sup>13</sup> C $\alpha$	84	3	2	89	133 <sup>e</sup>
	<sup>13</sup> C'	83	4	3	90	133 <sup>e</sup>
	<sup>13</sup> C $\beta$	71	6	1	78	107 <sup>e</sup>
	all- <sup>13</sup> C $\beta$	346	13	11	370	520 <sup>e</sup>
	all	417	19	12	448	637 <sup>e</sup>
$\beta$ 2M <sub>D76N</sub>	<sup>1</sup> H <sup>N</sup>	43	2	1	46	57 <sup>b, f</sup>
	<sup>15</sup> N	45	1	2	48	57 <sup>c, f</sup>
	<sup>13</sup> C $\alpha$	39	0	2	41	59 <sup>f</sup>
	<sup>13</sup> C'	39	2	0	41	59 <sup>f</sup>
	<sup>13</sup> C $\beta$	41	0	2	43	55 <sup>f</sup>
	all- <sup>13</sup> C $\beta$	166	5	5	176	232 <sup>f</sup>
	all	207	5	7	219	287 <sup>f</sup>

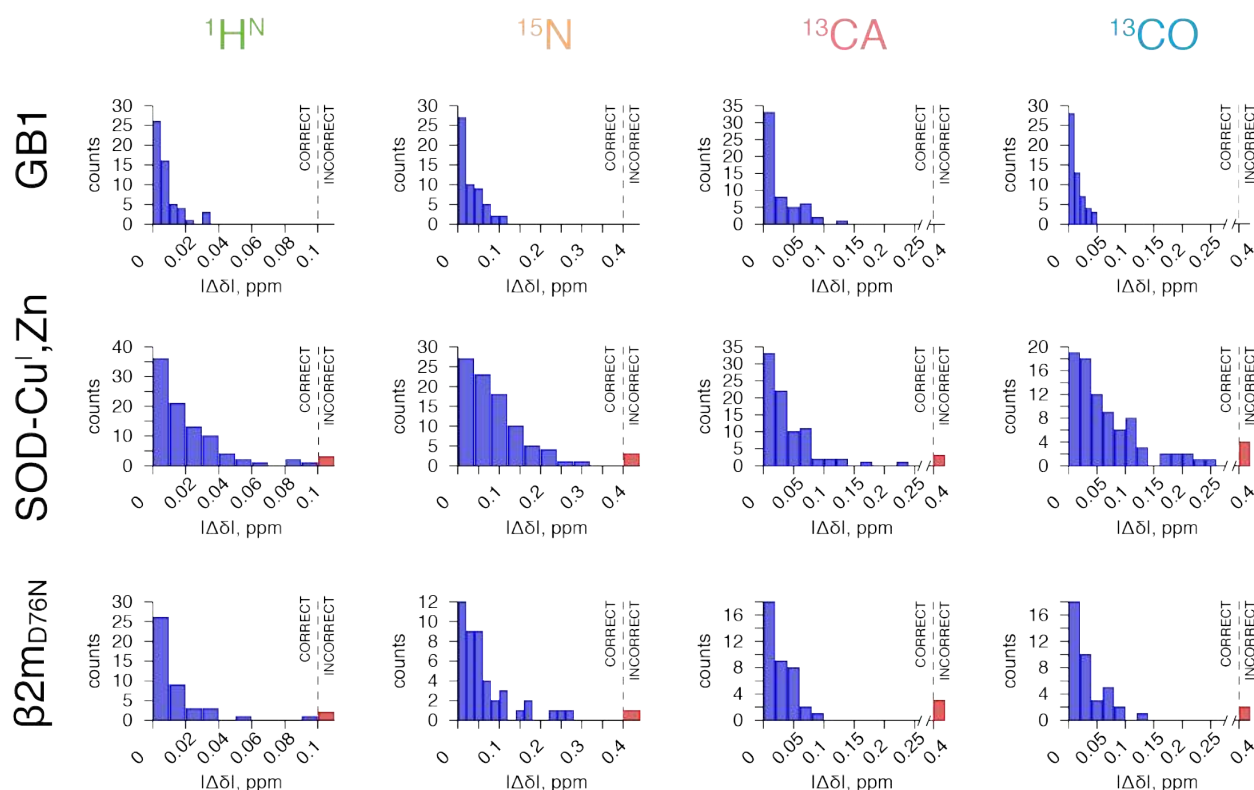
<sup>a</sup> With respect to the reference assignment

<sup>b</sup> The number of amide protons, i.e. excluding proline residues and N-terminus

<sup>c</sup> excluding proline <sup>15</sup>N atoms

<sup>e</sup> discounting dynamic residues 1-3, 23-27, 54, 59, 69, 107, 110-111, 128-130, 133, 139, 153

<sup>f</sup> within the fibril core (residues 20-81), discounting dynamic residues 25, 51-52



**Figure S4.** The histograms of consensus chemical shifts obtained by FLYA in function of their deviations from the reference shifts. The thresholds for classification of a shift as correct or incorrect are denoted by vertical dashed lines in each plot.

## J. Pulse program code for Bruker spectrometers

### Pulse sequence 1. (H)NCACONH

```
;hNCACONH5D.crmn
;5D pulse sequence for protein backbone assignment
;Forward transfer amide-detection
;with 15N and 13C decoupling
;written by Henry Orton and Jan Stanek

;Avance II+ version
;$COMMENT=5D hNCACONH
;$CLASS=Solids
;$DIM=5D
;$TYPE=cross polarisation
;$OWNER=CRMN

;p1 : 1H 90 pulse @ plw1
;p11 : 1H power for 90 pulse
;p7 : 15N 90 pulse @ plw7
;p17 : 15N power for 90 pulse
;p3 : 13C 90 pulse @ plw3
;p111 : 13C power for 90 pulse

;p15 : contact time for H->N CP
;sp0 : 1H power for H->N CP
;p12 : 15N power for H->N CP

;p17 : contact time for N->H CP
;sp10 : 1H power for N->H CP
;p120 : 15N power for N->H CP

;spnam10 : Ramp100.90
;spoffs10 : 0

;p16 : contact time for N->CO CP
;p26 : contact time for CA->N CP

;spnam0 : Ramp90.100
;spoffs0 : 0

;p26 : N -> Co CP
;p27 : Ca -> N CP
;p18 : Co selective pi
;p19 : Ca selective pi

;cpdprg1 : 1H decoupling (sltppm_40pTr41 for 1Hprot, waltz16_pl12 for 2Hprot)
;cpdprg2 : 15N decoupling (waltz16_pl16)
;cpdprg3 : 13C decoupling (waltz16_pl17)

;p12 : 1H decoupling power for (waltz16)
;p18 : 1H decoupling for (cw_pl18)
;p13 : 1H decoupling (sltppm)
;p16 : 15N decoupling power (waltz16)
;p17 : 13C decoupling power (waltz16)

;pcpd1 : 25u (waltz16 10 kHz) - 33.33 (sltppm 15kHz)
;pcpd2 : 25u (waltz16 10 kHz)
;pcpd3 : 25u (waltz16 10 kHz)
```

```

;d1 : recycle delay
;d31 : N evolution (first)
;d32 : CA evolution
;d33 : CO evolution
;d34 : N evolution (second)

;$COMMENT=basic cp experiment, arbitrary contact and decoupling schemes
;$CLASS=Solids
;$DIM=1D
;$TYPE=cross polarisation
;$SUBTYPE=simple 1D
;$OWNER=Bruker

prosol relations=<solids_cp>
#include <Avancesolids.incl>

;cnst9 : J(CA-CO) scalar coupling constant (55 Hz)
;cnst10 : estimated bulk T2 of CA [sec]
;cnst11 : estimated bulk T2 of CO [sec]
;cnst21 : CO frequency effective offset, about 173.3 ppm (ppm)
;cnst22 : CA frequency effective offset, about 53.3 ppm (ppm)

"plw16=plw7*(pow(p7/25,2))" ; 15N waltz 10kHz decoupling power level

;13C frequencies for center frequency calculation
"cnst20=(sfo3-bf3)*1000000/bf3" ;center frequency of 113.7 ppm

"plw11=plw3"

"spoal8=0.0"
"spoffs8=bf3*((cnst21 - cnst20)/1000000)" ;CO frequency

"spoal9=1.0"
"spoffs9=bf3*((cnst22 - cnst20)/1000000)" ;CA frequency

;CO selective Q3 pulse parameters (refocussing)
"p18=3.448/(95.0*bf3/1000000)" ;95 ppm bandwidth (safe)
"spw18=plw3*pow((0.5/(p18*0.1515))/(0.25/p3),2)" ;Q3 power level
"spoal18=0.5" ;default value
"spoff18=bf3*((cnst21 - cnst20)/1000000)" ;CO frequency

;CA selective Q3 pulse parameters (inversion)
"p19=3.41/(105.0*bf3/1000000)" ;105 ppm bandwidth (safe)
"spw19=plw3*pow((0.5/(p19*0.1515))/(0.25/p3),2)" ;Q3 power level
"spoal19=0.5" ;default value
"spoff19=bf3*((cnst22 - cnst20)/1000000)" ;CA frequency

;CO selective Q3 pulse parameters (inversion)
"p12=3.41/(95.0*bf3/1000000)" ;95 ppm bandwidth (safe)
"spw12=plw3*pow((0.5/(p12*0.1515))/(0.25/p3),2)" ;Q3 power level
"spoal12=0.5" ;default value
"spoff12=bf3*((cnst21 - cnst20)/1000000)" ;CO frequency

;CA selective REBURP pulse parameters (refocussing)
"p11=4.814/(30*bf3/1000000)" ;30 ppm bandwidth
"spw11=plw3*pow((0.5/(p11*0.0798))/(0.25/p3),2)" ;ReBURP power level
"spoal11=0.5" ;default value

```

```

"spoff11=bf3*((cnst22 - cnst20)/1000000)" ;CA frequency

;Carbon scalar coupling transfer delays
"cnst9=55.0" ;J(CA-CO)=55
"d2=2*p7" ;15N 180 delay
"d3=0.25/cnst9" ;1/4J(CA-CO)
"d28=0.5*atan(cnst11*PI*cnst9)/(PI*cnst9)" ;1/4J(CA-CO) relaxation CO optimised delay
"d29=0.5*atan(cnst10*PI*cnst9)/(PI*cnst9)" ;1/4J(CA-CO) relaxation CA optimised delay

;increments for apsy
"in31=inf1"
"in32=inf2"
"in33=inf3"
"in34=inf4"

;define additional phase increments for off resonance evolution
"cnst25=360*(cnst22-cnst20)*bf3/2" ; an intermediate variable (necessary) FOR phase 4
"cnst26=360*(cnst21-cnst20)*bf3/2" ; an intermediate variable (necessary) FOR phase 4

;delays for apsy
"d31=0.0"
"d32=0.0"
"d33=0.0"
"d34=0.0"

define delay DELTA1a
define delay DELTA2a
define delay DELTA3a
define delay DELTA4a

define delay DELTA1b
define delay DELTA2b
define delay DELTA3b
define delay DELTA4b

1 ze
2 d1 do:f2 do:f3

#include <p15_prot.incl>
#include <aq_prot.incl>

"DELTA1a= larger(d29+d2-d32*0.5, 0)"
"DELTA2a= larger(0, d32*0.5-d3)"
"DELTA3a= larger(-larger(-d32*0.5, -d3), d29)"
"DELTA4a= d32*0.5"

"DELTA1b= larger(d28+d2-d33*0.5, 0)"
"DELTA2b= larger(d28, d33*0.5)"
"DELTA3b= -larger(-d33*0.5, -d3)"
"DELTA4b= larger(0, d33*0.5-d3)"

"cnst23=cnst25*d32" ; CA
"cnst24=cnst26*d33" ; CO

1u ip7+cnst23 ; TPPI-like phase increment for offset evolution of CA
1u ip11+cnst24 ; TPPI-like phase increment for offset evolution of CO

1m fq=cnst20(bf ppm):f3 ;reset the 13C offset to CO/CA

```

```

(p1 pl1 ph1):f1
(p15 pl2 ph2):f2 (p15:sp0 ph3):f1      ;H -> N CP

1u cpds1:f1

;N t1 evolution
if "p3*2 > d31" {
  (center (d31) (p3*2 ph0 pl11):f3)
} else {
  d31
}

1u do:f1

(p26:sp9 ph4):f3 (p26:sp3 ph5):f2      ;N -> CA CP
2u cpds1:f1

;;CA-CO transfer

(p12:sp12 ph6):f3      ;CO Q3 (inversion)
6u
DELTA1a
(p11:sp11 ph7):f3      ;CA REBURP (refocusing)
4u
DELTA2a
(p12:sp12 ph6):f3      ;CO Q3 (inversion)
DELTA3a
(p7*2 ph0 pl7):f2      ;N 180
4u
DELTA4a

(p3 ph8 pl11):f3      ;C hard 90

9u
DELTA1b
(p19:sp19 ph10):f3      ;CA Q3 (inversion)
4u
(p18:sp18 ph11):f3
      ;CO Q3 (refocusing)
4u
DELTA2b
(p7*2 ph0 pl7):f2      ;N 180
4u
DELTA3b
(p19:sp19 ph10):f3      ;CA Q3 (inversion)
4u
DELTA4b
1u do:f1
(p16:sp8 ph12):f3 (p16:sp2 ph13):f2      ;CO -> N CP

;;end CA-CO transfer

1u cpds1:f1

;N t4 evolution
if "p3*2 > d34" {
  (center (d34) (p3*2 ph0 pl11):f3)
}

```

```

} else {
  d34
}

1u do:f1

;;water suppression starts
(p7 pl7 ph14):f2

1u pl12:f1
(d30*0.25 cw ph20):f1
(d30*0.25 cw ph21):f1
(d30*0.25 cw ph22):f1
(d30*0.25 cw ph23):f1
1u do:f1

(p7 pl7 ph15):f2
;;water suppression ends

(p17 pl20 ph16):f2 (p17:sp10 ph17):f1 ;N -> H CP

1u cpds2:f2
go=2 ph31
1m do:f2

10m mc #0 to 2
F1PH(calph(ph2, +90), caldel(d31, +in31))
F2PH(calph(ph4, -90), caldel(d32, +in32)) ; CA
F3PH(calph(ph12, -90), caldel(d33, +in33)) ; CO
F4PH(calph(ph13, +90), caldel(d34, +in34))

HaltAcqu, 1m
exit

ph0 = 0
ph1 = 1 3
ph2 = 1
ph3 = 0
ph4 = 1
ph5 = 1 1 3 3
ph6 = 0
ph7 = 0
ph8 = 1
ph10 = 1
ph11 = {0}*4 {1}*4
ph12 = 1
ph13 = 0
ph14 = {1}*8 {3}*8
ph15 = 1
ph16 = 0
ph17 = 0
ph20 = 0
ph21 = 1
ph22 = 2
ph23 = 3

ph31 = 0 2 2 0 2 0 0 2
      2 0 0 2 0 2 2 0

```

## Pulse sequence 2. (H)NCOCANH

```
;hNCOCANH5D.crmn
;5D pulse sequence for protein backbone assignment
;Reverse transfer amide detection
;with 15N and 13C decoupling
;written by Henry Orton and Jan Stanek

;Avance II+ version
;$COMMENT=5D hNCOCANH
;$CLASS=Solids
;$DIM=5D
;$TYPE=cross polarisation
;$OWNER=CRMN

;p1 : 1H 90 pulse @ plw1
;pl1 : 1H power for 90 pulse
;p7 : 15N 90 pulse @ plw7
;pl7 : 15N power for 90 pulse
;p3 : 13C 90 pulse @ plw3
;pl11 : 13C power for 90 pulse

;p15 : contact time for H->N CP
;sp0 : 1H power for H->N CP
;pl2 : 15N power for H->N CP

;p17 : contact time for N->H CP
;sp10 : 1H power for N->H CP
;pl20 : 15N power for N->H CP

;spnam10 : Ramp100.90
;spoffs10 : 0

;p16 : contact time for N->CO CP
;p26 : contact time for CA->N CP

;spnam0 : Ramp90.100
;spoffs0 : 0

;p18 : Co selective pi
;p19 : Ca selective pi

;cpdprg1 : 1H decoupling (sltpm_40pTr41 for 1Hprot, waltz16_pl12 for 2Hprot)
;cpdprg2 : 15N decoupling (waltz16_pl16)
;cpdprg3 : 13C decoupling (waltz16_pl17)

;pl12 : 1H decoupling power for (waltz16)
;pl13: 1H decoupling (sltpm)
;pl16: 15N decoupling power (waltz16)
;pl17: 13C decoupling power (waltz16)

;pcpd1 : 25u (waltz16 10 kHz) - 33.33 (sltpm 15kHz)
;pcpd2 : 25u (waltz16 10 kHz)
;pcpd3 : 25u (waltz16 10 kHz)

;d1 : recycle delay
;d31 : N evolution (first)
;d32 : Co evolution
;d33 : Ca evolution
```

```

;d34 : N evolution (second)

;$COMMENT=basic cp experiment, arbitrary contact and decoupling schemes
;$CLASS=Solids
;$DIM=1D
;$TYPE=cross polarisation
;$SUBTYPE=simple 1D
;$OWNER=Bruker

prosol relations=<solids_cp>
#include <Avancesolids.incl>

;cnst9 : J(CA-CO) scalar coupling constant (55 Hz)
;cnst10 : estimated bulk T2 of CA [sec]
;cnst11 : estimated bulk T2 of CO [sec]
;cnst21 : CO frequency effective offset, about 173.3 ppm (ppm)
;cnst22 : CA frequency effective offset, about 53.3 ppm (ppm)

"spoal0=0.5"
"spoff0=0.0"
"spoal10=0.5"
"spoff10=0.0"

"spoal2=0.5"
"spoff2=0.0"
"spoal3=0.5"
"spoff3=0.0"
"plw16=plw7*(pow(p7/25,2))" ; 15N waltz 10kHz decoupling power level

;13C frequencies for center frequency calculation
"cnst20=(sfo3-bf3)*1000000/bf3" ;center frequency of 113.7 ppm

"plw11=plw3"

"spoal8=1.0"
"spoffs8=bf3*((cnst21 - cnst20)/1000000)" ;CO frequency

"spoal9=0.0"
"spoffs9=bf3*((cnst22 - cnst20)/1000000)" ;CA frequency

;CO selective Q3 pulse parameters (refocussing)
"p18=3.448/(95.0*bf3/1000000)" ;95 ppm bandwidth (safe)
"spw18=plw3*pow((0.5/(p18*0.1515))/(0.25/p3),2)" ;Q3 power level
"spoal18=0.5" ;default value
"spoff18=bf3*((cnst21 - cnst20)/1000000)" ;CO frequency

;CA selective Q3 pulse parameters (inversion)
"p19=3.41/(105.0*bf3/1000000)" ;105 ppm bandwidth (safe)
"spw19=plw3*pow((0.5/(p19*0.1515))/(0.25/p3),2)" ;Q3 power level
"spoal19=0.5" ;default value
"spoff19=bf3*((cnst22 - cnst20)/1000000)" ;CA frequency

;CO selective Q3 pulse parameters (inversion)
"p12=3.41/(95.0*bf3/1000000)" ;95 ppm bandwidth (safe)
"spw12=plw3*pow((0.5/(p12*0.1515))/(0.25/p3),2)" ;Q3 power level
"spoal12=0.5" ;default value
"spoff12=bf3*((cnst21 - cnst20)/1000000)" ;CO frequency

```



```

;CA selective REBURP pulse parameters (refocussing)
"p11=4.814/(30*bf3/1000000)" ;30 ppm bandwidth
"spw11=plw3*pow((0.5/(p11*0.0798))/(0.25/p3),2)" ;ReBURP power level
"spoal11=0.5" ;default value
"spoff11=bf3*((cnst22 - cnst20)/1000000)" ;CA frequency

;Carbon scalar coupling delays
"cnst9=55.0" ;J(CA-CO)=55
"d2=2*p7" ;15N 180 delay
"d3=0.25/cnst9" ;1/4J(CA-CO)
"d28=0.5*atan(cnst11*PI*cnst9)/(PI*cnst9)" ;1/4J(CA-CO) relaxation CO optimised delay
"d29=0.5*atan(cnst10*PI*cnst9)/(PI*cnst9)" ;1/4J(CA-CO) relaxation CA optimised delay

;increments for apsy
"in31=inf1"
"in32=inf2"
"in33=inf3"
"in34=inf4"

;define additional phase increments for off resonance evolution
"cnst25=360*(cnst21-cnst20)*bf3/2" ; an intermediate variable (necessary) FOR phase 4
"cnst26=360*(cnst22-cnst20)*bf3/2" ; an intermediate variable (necessary) FOR phase 4

;delays for apsy
"d31=0.0"
"d32=0.0"
"d33=0.0"
"d34=0.0"

define delay DELTA1a
define delay DELTA2a
define delay DELTA3a
define delay DELTA4a

define delay DELTA1b
define delay DELTA2b
define delay DELTA3b
define delay DELTA4b

1 ze
2 d1 do:f2 do:f3

#include <p15_prot.incl>
#include <aq_prot.incl>

"DELTA1a= larger(d28+d2-d32*0.5, 0)"
"DELTA2a= larger(0, d32*0.5-d3)"
"DELTA3a= larger(-larger(-d32*0.5, -d3), d28)"
"DELTA4a= d32*0.5"

"DELTA1b= larger(d29+d2-d33*0.5, 0)"
"DELTA2b= larger(d29, d33*0.5)"
"DELTA3b= -larger(-d33*0.5, -d3)"
"DELTA4b= larger(0, d33*0.5-d3)"

"cnst23=cnst25*d32" ; CO
"cnst24=cnst26*d33" ; CA
1u ip7+cnst23 ; TPPI-like phase increment for offset evolution of CO

```

1u ip11+cnst24 ; TPPI-like phase increment for offset evolution of CA

1m fq=cnst20(bf ppm):f3 ;reset the <sup>13</sup>C offset to CO

(p1 pl1 ph1):f1  
(p15 pl2 ph2):f2 (p15:sp0 ph3):f1 ;H -> N CP

1u cpds1:f1  
;N t1 evolution  
if "p3\*2 > d31" {  
 (center (d31) (p3\*2 ph0 pl11):f3)  
} else {  
 d31  
}

1u do:f1

(p16:sp8 ph4):f3 (p16:sp2 ph5):f2 ;N -> CO CP  
2u cpds1:f1

;;CO-CA transfer

(p19:sp19 ph6):f3 ;CA Q3 (inversion)  
6u  
DELTA1a  
(p18:sp18 ph7):f3 ;CO Q3 (refocusing)  
4u  
DELTA2a  
(p19:sp19 ph6):f3 ;CA Q3 (inversion)  
DELTA3a  
(p7\*2 ph0 pl7):f2 ;N 180  
4u  
DELTA4a

(p3 ph8 pl11):f3 ;C hard 90

9u  
DELTA1b  
(p12:sp12 ph10):f3 ;CO Q3 (inversion)  
4u  
(p11:sp11 ph11):f3 ;CA REBURP (refocusing)  
4u  
DELTA2b  
(p7\*2 ph0 pl7):f2 ;N 180  
4u  
DELTA3b  
(p12:sp12 ph10):f3 ;CO Q3 (inversion)  
4u  
DELTA4b  
1u do:f1  
(p26:sp9 ph12):f3 (p26:sp3 ph13):f2 ;CA -> N CP

;;end CO-CA transfer

1u cpds1:f1

;N t4 evolution  
if "p3\*2 > d34" {

```

(center (d34) (p3*2 ph0 pl11):f3)
} else {
  d34
}
1u do:f1

;;water suppression starts
(p7 pl7 ph14):f2,

1u pl12:f1
(d30*0.25 cw ph20):f1
(d30*0.25 cw ph21):f1
(d30*0.25 cw ph22):f1
(d30*0.25 cw ph23):f1
1u do:f1

(p7 pl7 ph15):f2
;;water suppression ends

(p17 pl20 ph16):f2 (p17:sp10 ph17):f1 ;N -> H CP

1u cpds2:f2
go=2 ph31
1m do:f2

10m mc #0 to 2
F1PH(calph(ph2, +90), caldel(d31, +in31))
F2PH(calph(ph4, -90), caldel(d32, +in32)) ; CO
F3PH(calph(ph12, -90), caldel(d33, +in33)) ; CA
F4PH(calph(ph13, +90), caldel(d34, +in34))

HaltAcqu, 1m
exit

ph0 = 0
ph1 = 1 3
ph2 = 1
ph3 = 0
ph4 = 1
ph5 = 1 1 3 3
ph6 = 0
ph7 = 0 ;{0}*4 {1}*4
ph8 = 1
ph10 = 1
ph11 = {0}*4 {1}*4
ph12 = 1
ph13 = 0
ph14 = {1}*8 {3}*8
ph15 = 1
ph16 = 0
ph17 = 0
ph20 = 0
ph21 = 1
ph22 = 2
ph23 = 3

ph31 = 0 2 2 0 2 0 0 2
      2 0 0 2 0 2 2 0

```

## K. References

- (1) Franks, W. T.; Zhou, D. H.; Wylie, B. J.; Money, B. G.; Graesser, D. T.; Frericks, H. L.; Sahota, G.; Rienstra, C. M. Magic-angle spinning solid-state NMR spectroscopy of the beta 1 immunoglobulin binding domain of protein G (GB1):  $^{15}\text{N}$  and  $^{13}\text{C}$  chemical shift assignments and conformational analysis. *J. Am. Chem. Soc.* **2005**, *127*, 12291-12305.
- (2) Schmidt, H. L. F.; Sperling, L. J.; Gao, Y. G.; Wylie, B. J.; Boettcher, J. M.; Wilson, S. R.; Rienstra, C. M. Crystal polymorphism of protein GB1 examined by solid-state NMR spectroscopy and X-Ray diffraction. *J. Phys. Chem. B* **2007**, *111*, 14362-14369.
- (3) Knight, M. J.; Webber, A. L.; Pell, A. J.; Guerry, P.; Barbet-Massin, E.; Bertini, I.; Felli, I. C.; Gonnelli, L.; Pierattelli, R.; Emsley, L.; Lesage, A.; Herrmann, T.; Pintacuda, G. Fast resonance assignment and fold determination of human superoxide dismutase by high-resolution proton-detected solid-state MAS NMR spectroscopy. *Angew. Chem. Int. Ed. Engl.* **2011**, *50*, 11697-11701.
- (4) Knight, M. J.; Pell, A. J.; Bertini, I.; Felli, I. C.; Gonnelli, L.; Pierattelli, R.; Herrmann, T.; Emsley, L.; Pintacuda, G. Structure and backbone dynamics of a microcrystalline metalloprotein by solid-state NMR. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 11095-11100.
- (5) Pintacuda, G.; Giraud, N.; Pierattelli, R.; Böckmann, A.; Bertini, I.; Emsley, L. Solid-state NMR spectroscopy of a paramagnetic protein: Assignment and study of human dimeric oxidized CuII-ZnII superoxide dismutase (SOD). *Angew. Chem. Int. Ed. Engl.* **2007**, *46*, 1079-1082.
- (6) Esposito, G.; Michelutti, R.; Verdone, G.; Viglino, P.; Hernandez, H.; Robinson, C. V.; Amoresano, A.; Dal Piaz, F.; Monti, M.; Pucci, P.; Mangione, P.; Stoppini, M.; Merlini, G.; Ferri, G.; Bellotti, V. Removal of the N-terminal hexapeptide from human beta2-microglobulin facilitates protein aggregation and fibril formation. *Protein Sci.* **2000**, *9*, 831-45.
- (7) Valleix, S.; Gillmore, J. D.; Bridoux, F.; Mangione, P. P.; Dogan, A.; Nedelec, B.; Boimard, M.; Touchard, G.; Goujon, J. M.; Lacombe, C.; Lozeron, P.; Adams, D.; Lacroix, C.; Maisonobe, T.; Plante-Bordeneuve, V.; Vrana, J. A.; Theis, J. D.; Giorgetti, S.; Porcari, R.; Ricagno, S.; Bolognesi, M.; Stoppini, M.; Delpech, M.; Pepys, M. B.; Hawkins, P. N.; Bellotti, V. Hereditary systemic amyloidosis due to Asp76Asn variant beta2-microglobulin. *N. Engl. J. Med.* **2012**, *366*, 2276-83.
- (8) Bertini, I.; Luchinat, C.; Parigi, G.; Ravera, E.; Reif, B.; Turano, P. Solid-state NMR of proteins sedimented by ultracentrifugation. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 10396-10399.
- (9) Böckmann, A.; Gardiennet, C.; Verel, R.; Hunkeler, A.; Loquet, A.; Pintacuda, G.; Emsley, L.; Meier, B. H.; Lesage, A. Characterization of different water pools in solid-state NMR protein samples. *J. Biomol. NMR* **2009**, *45*, 319-27.
- (10) Bennett, A. E.; Rienstra, C. M.; Auger, M.; Lakshmi, K. V.; Griffin, R. G. Heteronuclear decoupling in rotating solids. *J. Chem. Phys.* **1995**, *103*, 6951-6958.
- (11) Lewandowski, J. R.; Sein, J.; Sass, H. J.; Grzesiek, S.; Blackledge, M.; Emsley, L. Measurement of site-specific  $^{13}\text{C}$  spin-lattice relaxation in a crystalline protein. *J. Am. Chem. Soc.* **2010**, *132*, 8252-8254.
- (12) Shaka, A. J.; Keeler, J.; Frenkiel, T.; Freeman, R. An Improved Sequence for Broad-Band Decoupling - WALTZ-16. *J. Magn. Reson.* **1983**, *52*, 335-338.
- (13) Emsley, L.; Bodenhausen, G. Optimization of Shaped Selective Pulses for NMR Using a Quaternion Description of Their Overall Propagators. *J. Magn. Reson.* **1992**, *97*, 135-148.
- (14) Geen, H.; Freeman, R. Band-Selective Radiofrequency Pulses. *J. Magn. Reson.* **1991**, *93*, 93-141.
- (15) Zhou, D. H.; Rienstra, C. M. High-performance solvent suppression for proton detected solid-state NMR. *J. Magn. Reson.* **2008**, *192*, 167-72.
- (16) Hediger, S.; Meier, B. H.; Kurur, N. D.; Bodenhausen, G.; Ernst, R. R. NMR cross polarization by adiabatic passage through the Hartmann—Hahn condition (APHH). *Chem. Phys. Lett.* **1994**, *223*, 283-288.
- (17) Andreas, L. B.; Jaudzems, K.; Stanek, J.; Lalli, D.; Bertarello, A.; Le Marchand, T.; De Paepe, D. C.; Kotlovica, S.; Akopjana, I.; Knott, B.; Wegner, S.; Engelke, F.; Lesage, A.; Emsley, L.; Tars, K.; Herrmann, T.; Pintacuda, G. Structure of fully protonated proteins by proton-detected magic-angle spinning NMR. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 9187-9192.
- (18) Kazimierczuk, K.; Zawadzka-Kazimierczuk, A.; Koźmiński, W. Non-uniform frequency domain for optimal exploitation of non-uniform sampling. *J. Magn. Reson.* **2010**, *205*, 286-92.
- (19) Fiorito, F.; Hiller, S.; Wider, G.; Wüthrich, K. Automated resonance assignment of proteins: 6D APSY-NMR. *J. Biomol. NMR* **2006**, *35*, 27-37.
- (20) Zawadzka-Kazimierczuk, A.; Kazimierczuk, K.; Koźmiński, W. A set of 4D NMR experiments of enhanced resolution for easy resonance assignment in proteins. *J. Magn. Reson.* **2010**, *202*, 109-16.
- (21) Frueh, D. P.; Sun, Z. Y.; Vosburg, D. A.; Walsh, C. T.; Hoch, J. C.; Wagner, G. Non-uniformly sampled double-TROSY hNcaNH experiments for NMR sequential assignments of large proteins. *J. Am. Chem. Soc.* **2006**, *128*, 5757-63.
- (22) Peng, J. W.; Wagner, G. Mapping of spectral density functions using heteronuclear NMR relaxation measurements. *J. Magn. Reson. (1969)* **1992**, *98*, 308-332.

- (23) Penzel, S.; Smith, A. A.; Agarwal, V.; Hunkeler, A.; Org, M. L.; Samoson, A.; Böckmann, A.; Ernst, M.; Meier, B. H. Protein resonance assignment at MAS frequencies approaching 100 kHz: a quantitative comparison of J-coupling and dipolar-coupling-based transfer methods. *J. Biomol. NMR* **2015**, *63*, 165-86.
- (24) Barbet-Massin, E.; Pell, A. J.; Jaudzems, K.; Franks, W. T.; Retel, J. S.; Kotelovica, S.; Akopjana, I.; Tars, K.; Emsley, L.; Oschkinat, H.; Lesage, A.; Pintacuda, G. Out-and-back  $^{13}\text{C}$ - $^{13}\text{C}$  scalar transfers in protein resonance assignment by proton-detected solid-state NMR under ultra-fast MAS. *J. Biomol. NMR* **2013**, *56*, 379-86.