Development of ^{205}Tl NMR Methods for the Direct Study of Monovalent Metal Ions and Ligands in Nucleic Acids

A dissertation
Presented to the Faculty of the Graduate School
Of
Yale University
In Candidacy for the Degree of
Doctor of Philosophy

By
Michelle Lynn Gill

Dissertation Directors: Scott A. Strobel and J. Patrick Loria
May 2006

Abstract

Development of ^{205}Tl NMR Methods for the
Direct Study of Monovalent Metal Ions and Ligands in Nucleic Acids

Michelle Lynn Gill
Yale University
May 2006

The requirement of monovalent cations for structure and function transcends all classes of biological macromolecules. In catalytic RNAs, a structural and/or catalytic requirement for monovalent ions has been identified in the ribosome, group I, and group II introns. Other RNAs, such as the hammerhead, hairpin, and VS ribozymes, can perform catalysis the presence of only high concentrations of monovalent cations. Despite their importance, few techniques exist for the direct, solution study of these cations. Toward this goal, I have used $^{205}Tl^{+}$, a K^{+} surrogate which is readily detectable by solution NMR, to study the binding of monovalent cations to a G-quadruplex, $d(G_4T_4G_4)_2$. The NMR and crystal structures of the Tl^{+}-form of $d(G_4T_4G_4)_2$ have been determined to assess the ability of Tl^{+} to mimic K^{+} in a nucleic acid setting. Direct detection ^{205}Tl NMR studies have been used to characterize the binding of $^{205}Tl^{+}$ to $d(G_4T_4G_4)_2$ and provide evidence for a previously undetected mode of monovalent binding. I have also developed a ^{1}H–^{205}Tl spin-echo difference experiment which was used to detect ^{1}H–^{205}Tl scalar couplings and assign two of the experimentally observed ^{205}Tl resonances to monovalent binding sites in $d(G_4T_4G_4)_2$. These results comprise the first ^{1}H–^{205}Tl scalar couplings observed in a biological system and the first ^{205}Tl heteronuclear experiment reported. Preliminary ^{205}Tl NMR studies in RNA systems are also discussed.

Development of ^{205}Tl NMR Methods for the Direct Study of Monovalent Metal Ions and Ligands in Nucleic Acids

A dissertation
Presented to the Faculty of the Graduate School
Of
Yale University
In Candidacy for the Degree of
Doctor of Philosophy

By
Michelle Lynn Gill

Dissertation Directors: Scott A. Strobel and J. Patrick Loria
May 2006

Acknowledgements

There are many individuals to whom I am grateful for their support and guidance throughout my graduate career. First, I would like to thank my advisors, Scott Strobel and Pat Loria. Their thoughtful insights, patience, and willingness to undertake a project that is new to both of their labs have made the past five years challenging and fulfilling. I would also like to thank Anna Pyle, the third member of my research committee, for her suggestions and keen ability to keep me focused on important scientific questions.

I am grateful to many other people for scientific discussion, advice, friendship, and well-timed distractions. Jesse Cochrane oversaw me when I was a rotation student in the Strobel lab and I have repeatedly come to her with problems, both scientific and personal, throughout my time in the lab. Ashley Hesslein, Anne Kosek, Amy White, K. Mark Parnell, Josh Weinger, Rachel Anderson, and Lara Szewczak taught me many experimental fundamentals in addition to making me feel at home when I first joined the Strobel lab. Jim Kempf, Evgueni Kovriguine, Roger Cole, and Dagny Ulrich welcomed me into the Loria lab when I became a joint student. In my later years, I have also enjoyed working with a number of graduate students and postdoctoral associates in the Strobel and Loria labs: Mary Stahley, Emmanuel Pfund, Peter Adams, TJ, Yan Wang, Nicolas Carrasco, Ethan Butler, James and Sarah Lipchock, Christina Ragain, Ian Suydam, and Dave Hiller.

I would like to thank Professor Kurt Zilm for the use of his high band amplifier and for helpful scientific discussions. Dr. Xiaoling Wu, Dr. Eric Paulson, Dr. Ben Bangerter, and Dr. Corey Morcombe each provided a wealth of technical assistance and

advice when I needed experimental help. Professor Melanie Cocco was instrumental in setting up the initial direct detection thallium experiment in the Department of Chemistry at Yale University. I am grateful to Professor Samuel Butcher and Jared Davis at the University of Wisconsin for providing GAAA tetraloop-tetraloop receptor samples and to Professor Thomas Nowak and Dr. Jarislav Zajicek at the University of Notre Dame for access to their lower field NMR instrument. I wish to thank Professor Victor Batista, Dr. Jose Gascon, and Christina Ragain for assistance with the magnetic susceptibility calculations. The Tl^+ diffraction data were collected at beamline X25 at the Brookaven NSLS, and I wish to thank the staff there for their assistance. I thank the CSB staff, Dr. Michael Strickler, Dr. Jimin Wang, Paul Pepin, Art Perlo, and Dave Keller, for technical assistance with the x-ray diffraction equipment and data processing. I was supported for three years of my graduate career by a National Science Foundation graduate fellowship.

Finally, I wish to thank my parents, Dr. Allen and Jeanne Sippel, my extended parents, Bob and Cheryl Gill, and my husband, Rob Gill, for their support through the many challenges I have encountered these past years. I could not have completed this journey without them.

Table of Contents

Figures and Tables

List of Figures

List of Tables

1 Introduction

1.1 Solution study of metal ions in biological macromolecules

Solution NMR plays a prominent role in structural biology because it enables the conformation of biomacromolecules to be studied without the required formation of single crystals. The power of this technique for determining macromolecular structures relies on establishing short range, interproton distances [1], torsion angles [2], and bond vector orientations [3]. These types of measurements are also useful in defining intermolecular interactions between macromolecules and small molecule ligands [4].

An important area of investigation that has not been as amenable to characterization by NMR is the study of interactions between inorganic cations and macromolecules. The difficulty in characterizing cation binding in solution arises primarily from the lack of metal ions with properties well suited for solution NMR. For the study of divalent cation binding sites by NMR, cadmium (^{113}Cd), a spin-½ nucleus has been used as a surrogate for the biologically essential metals Ca^{2+}, Zn^{2+}, and Fe^{2+} [5-8]. To a lesser extent, mercury (^{199}Hg) NMR has also been used to study divalent cation sites [7, 9, 10]. These studies have enabled characterization of the interaction between divalent metal ions and their protein partners [11-16]. In addition, ^{1}H–M^{2+} heteronuclear experiments have been used to determine metal-ligand identity, binding site conformation, and for the characterization of novel structural motifs in rubredoxin, metallothionein, superoxide dismutase, and the transcription factors GAL4 and LAC9 [7-9, 12-17].

1.2 Monovalent cations in biological systems

Monovalent cations are also essential for cellular function. Every major class of biomacromolecules including proteins [18, 19], nucleic acids [20-24], phospholipids [25-29], and carbohydrates [30-33] has a structural and/or functional requirement for monovalent cations. In nucleic acids, monovalent metal ions are known to play structural and/or catalytic roles within a number of catalytic RNAs [22, 23, 34-36].

In the ribosome, the requirement of K^+ for peptidyl transferase activity was noted over 30 years ago [37] and verified by later biochemical studies [38-40]. The identification of a K^+ within the peptidyl transferase center in the crystal structure of the *Haloarcula marismortui* 50 S ribosomal subunit provides a suggestion, though not conclusive evidence, of where this catalytically important monovalent may be located (Figure 1-1A) [36]. A second K^+ has also been identified within the highly conserved L11 binding portion of the 23 S rRNA (Figure 1-1B) [24, 41]. In group I introns from *Tetrahymena thermophila* and *Azoarcus sp. BH72* (Figure 1-1C), K^+ ions are utilized by the tetraloop-tetraloop receptor (TL-TR) tertiary motif [22, 23, 35, 42]. The ai5γ group II intron from *Saccharomyces cerevisiae* mitochondria also contains a TL-TR and requires high concentrations of K^+ for catalytic activity [34, 43-45]. Finally, the hammerhead, hairpin, and VS ribozymes have shown to be catalytically active in the presence of only high concentrations of monovalent cations [46, 47]. In general, the requirement for monovalent cations cannot be circumvented by the use of high concentrations of divalent ions, such as Mg^{2+}. Notable exceptions to this rule are isolated RNA systems containing TL-TR motifs, which have been shown to also bind divalent cations including Mg^{2+}, Mn^{2+}, and Ca^{2+} [48-52].

The apparent specificity of these binding sites for monovalent cations implies a functional role warranting further investigation. Thus far, the most effective technique for high-resolution study of monovalent cations has been x-ray crystallography, often involving soaks of heavy metal derivatives [22, 24, 53, 54]. While many metal binding sites have been characterized in this way, a spectroscopic technique that eliminates the need for formation of single crystals would expand the experimental arsenal with which monovalent cation sites could be studied.

1.3 Existing methods for the spectroscopic study of monovalent cations

The alkali metals tend to have nuclear spins (I) greater than $\frac{1}{2}$ (Na^+ $I=\frac{3}{2}$, K^+ $I=\frac{3}{2}$, Rb^+ $I=\frac{3}{2}$, $\frac{5}{2}$), making their study by solution NMR particularly challenging. Quadrupolar relaxation and magnetic relaxation dispersion (MRD) have been used to detect sequence specific binding of monovalent cations to DNA duplexes [55-57] and have provided information about Na^+ binding and exchange within G-quadruplexes [58-60]. However, both of these techniques provide information that is of relatively limited resolution and sensitivity.

Chemical shift mapping has also been successfully employed to study both monovalent and divalent cation binding to nucleic acids [61-63]. The interpretation of any observed spectral changes can be complicated by large scale conformational rearrangements which occur upon metal binding and by ambiguity in the number of metal binding sites. Finally, none of the techniques listed above allow the direct study of monovalent cations which is necessary for the determination of their binding sites.

A solution NMR method for probing monovalent cation sites using $^{15}NH_4^+$ as a K^+ replacement has also been developed. This technique allows direct observation of

bound ammonium ions and determination of their potential ligands, independent of whether the $^{15}NH_4^+$ is in fast or slow exchange with the biomacromolecules. It has been used successfully to study the binding of $^{15}NH_4^+$ to the minor groove of DNA duplexes and in the assignment of three monovalent binding sites in the G-quadruplex, $d(G_4T_4G_4)_2$ [60, 64-66]. While NH_4^+ has been shown to have an ionic radius and hydration energy similar to that of K^+, the presence of four tetrahedrally bonded hydrogen atoms certainly differentiates it from alkali metals. Further, the experiments are most easily performed at low pH ranges. Therefore, the use of established $^{15}NH_4^+$ NMR methods may prove difficult in some cases.

Accordingly, a set of NMR methods is desirable that enables the direct observation of bound monovalent metals, allows localization of their binding site(s), and permits differentiation between monovalent cations at distinct binding sites. Ideally, this technique should also be relatively insensitive to solution conditions, such as pH. A spin-½ nucleus well-suited for developing methods which meet these criteria is thallium ($^{205}Tl^+$), a monovalent metal whose strong anomalous signal and ability to replace K^+ have made it useful in x-ray crystallography and biochemistry [21, 22, 39, 40, 53, 54, 67].

1.4 Use of Tl^+ as a K^+ surrogate in biochemistry and solution NMR

Thallium's propensity to substitute for K^+ at its binding sites is due largely to the similar chemical properties of the two metals. The atomic radius of Tl^+ (1.40 Å) closely matches that of K^+ (1.33 Å) [68]. The two cations also have similar hydration energies—77.6 kcal/mol for Tl^+ and 76.4 kcal/mol for K^+ [53, 69]. The hydration energy, in particular, is a critical determinant of ion specificity [62, 70]. Further, Tl^+ and K^+ both form similar bond lengths (2.4–2.7 Å) and can support irregular coordination geometries [22, 71]. Tl^+ has been shown to support high levels of activity in many enzymes,

including the ribosome, aldehyde dehydrogenase, and adenosine triphosphatase [39, 40, 72-76].

Thallium is also particularly amenable to study by NMR. The isotope ^{205}Tl, present at 70.5% natural abundance, is a spin-½ nucleus with a high gyromagnetic ratio, γ=15.589 X 10^7 T^{-1}•s^{-1} [77]. These qualities make it the third most receptive nucleus to NMR. The ^{205}Tl chemical shift, scalar coupling constant, and spin-lattice relaxation rate (R_1) are extremely sensitive to the chemical environment [78, 79]. There is also a second thallium isotope, ^{203}Tl, which is a spin-½ nucleus with a slightly lower gyromagnetic ratio, γ=15.436 X 10^7 T^{-1}•s^{-1}, and natural abundance (29.5 %) [77].

There is a limited precedent for using direct detect ^{205}Tl NMR to study monovalent binding sites in proteins, membrane channels, and antibiotics, including pyruvate kinase, adenosine triphosphitase, gramicidin-A, valinomycin, nonactin, monactin, and dinactin [67, 72, 76, 79-83] but as of yet, only preliminary studies have been performed on nucleic acid systems [60, 73]. To our knowledge, there have been no reports of heteronuclear experiments involving ^{205}Tl.

1.5 G-quadruplex as a model system for ^{205}Tl NMR

The development of ^{205}Tl NMR in nucleic acids requires the use of a well-characterized system, which is capable of binding monovalent cations. A biologically-relevant example that meets these criteria is the G-quadruplex. These four-stranded structures are formed from DNA or RNA sequences containing tandem G-rich repeats. G-quadruplexes are characterized by consecutive stacks of four planar, hydrogen-bonded guanine nucleotides, called G-quartets (Figure 1-2A). There is variation in the number of G-quartets contained within a single quadruplex (see Figure 1-2B for an example) and in the number of DNA or RNA strands which comprise the quadruplex. If less than four

distinct oligonucleotide strands are involved, the strands fold back upon themselves leaving loop regions which are not involved in G-quartet formation. There is a considerable amount of variability in the location and conformation of these loop structures. Sequences shown to form G-quadruplexes *in vitro* have been identified in the telomeres of various organisms including humans [84-87], in immunoglobulin switch regions [88], and in several gene promoters [89, 90]. Because of their association with telomerase, G-quadruplexes have been targeted for cancer therapy [91-94] and have also been proposed to inhibit HIV integrase [95-97].

The arrangement of guanine bases within the G-quartets creates a negatively charged channel which has been shown to bind monovalent cations, including Na^+, K^+, NH_4^+, Cs^+, and Rb^+ [58, 98-101]. The identity of the monovalent cation present has been shown to significantly affect both the stability and structure of the G-quadruplex. In general, the ability of different monovalent species to stabilize G-quadruplex structures follows the Eisenman series V ($K^+ > Rb^+ > Na^+ > Cs^+ > Li^+$) [102], where K^+ is the most stabilizing and Li^+ is the least stabilizing, in some cases even inhibiting G-quadruplex formation [60, 84, 100, 103-105]. There have only been limited studies of the ability of Tl^+ to stabilize G-quadruplexes [60, 73, 106], but preliminary results indicate it does so at least as well as K^+.

The identity of the bound monovalent cation appears to be related to G-quadruplex structural polymorphism. The DNA sequence $d(T_2G_4)_4$, derived from the *Tetrahymena* telomere sequence, undergoes a transition from an intramolecular to a multi-stranded G-quadruplex structure when the solution cation is changed from Na^+ to K^+ [105]. This transition is characterized by an increase of over 25°C in the melting

temperature of the base pairs. A second example involves the 22 nucleotide sequence from human telomeres, d[AGGG(TTAGGG)$_3$], which forms an intramolecular G-quadruplex. In the solution structure of the Na^+-form, the G-quadruplex is antiparallel and contains one diagonal and two lateral d(TTA) loops (Figure 1-3A) [87]. The crystal structure of the K^+-form, reported by Neidle and coworkers [86] is instead a parallel stranded G-quadruplex (Figure 1-3B). In the K^+-form, all three d(TTA) loops traverse the sides of the G-quadruplex, forming a propeller-like structure. The explanation for these ion-dependent structural transitions is largely unknown and further illustrates the need to develop a technique for the direct solution study of monovalent ions.

We have used the 7.6 kDa G-quadruplex, d($G_4T_4G_4$)$_2$, from the telomeric sequence of the ciliate *Oxytricha nova* as the model system for development of ^{205}Tl NMR in nucleic acids. The G-quadruplex formed by d($G_4T_4G_4$)$_2$ is a homodimeric, antiparallel structure with four consecutively stacked G-quartets and diagonal loops at either end (Figure 1-4A). The formation of this fold results in a rotational symmetry axis that bisects the region between the two inner G-quartet planes (Figure 1-4B), greatly simplifying the observed NMR spectra.

Like other G-quadruplexes, d($G_4T_4G_4$)$_2$ is stabilized by the binding of monovalent cations (Na^+, K^+, and NH_4^+) [62, 65, 107-112]. However, the stoichiometry and location of the binding sites appear to be ion-dependent. Neidle and coworkers have shown that five K^+ are coordinated by d($G_4T_4G_4$)$_2$, three between successive G-quartet planes and two in the thymine loops (Figure 1-4C, blue) [111]. Solution studies of the NH_4^+-form by Feigon and coworkers have provided evidence for three NH_4^+ ions per G-quadruplex (none in the thymine loops) bound in a manner similar to the K^+ coordination (Figure

1-4B, black and blue) [65, 107]. A crystal structure by Horvath and Schultz of the Na^+-form of $d(G_4T_4G_4)_2$ bound to the *Oxytricha nova* telomeric protein shows a slightly different mode of Na^+ binding [112]. The G-quadruplex coordinates four Na^+ ions, each within a single G-quartet plane (Figure 1-4D, red).

In chapter 2, the solution structure of the Tl^+-form of $d(G_4T_4G_4)_2$ is presented. The topology of the structure is described and compared to previously reported solution structures of other cation forms of $d(G_4T_4G_4)_2$. Similarities to the K^+-form are discussed in light of the ability of Tl^+ to mimic K^+ in this nucleic acid system.

^{205}Tl NMR studies of $d(G_4T_4G_4)_2$ are detailed in chapter 3. The ability of Tl^+ to stabilize $d(G_4T_4G_4)_2$ is determined by temperature dependence studies. The nature of each ^{205}Tl resonance is explored, including the effects of titrations with other monovalent cations and the determination of bound lifetimes. This information is used to describe the nature of each bound ^{205}Tl resonance.

In chapter 4, the crystal structure of the Tl^+-form of $d(G_4T_4G_4)_2$ is described. This structure was solved to determine the location of Tl^+ binding sites within the G-quadruplex. The coordination of Tl^+ and K^+ to $d(G_4T_4G_4)_2$ are compared. An interpretation of the differences between the solution and crystal structures of the Tl^+-form of $d(G_4T_4G_4)_2$ is also provided.

In chapter 5, a ^1H–^{205}Tl spin echo difference experiment, developed to detect ^1H–^{205}Tl scalar couplings, is presented. The results of this experiment are used to assign ^{205}Tl resonances to binding sites within the G-quadruplex. Possible mechanisms for the observed scalar couplings are discussed.

Preliminary ^{205}Tl NMR studies of monovalent cation binding to two RNA systems are described in chapter 6. These studies illustrate important limitations in the suitability of a system for study by ^{205}Tl NMR and are a starting point for future endeavors in this field.

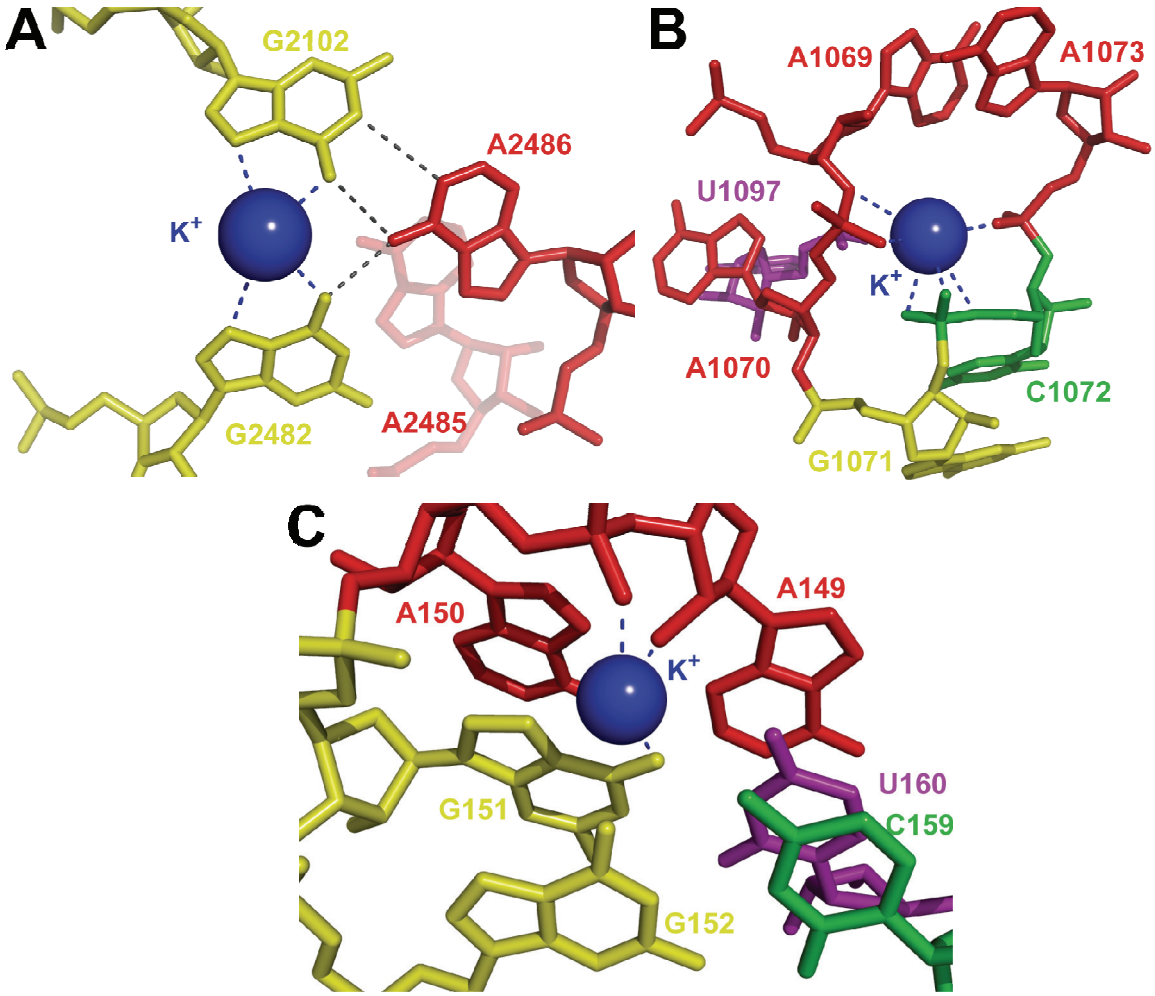


Figure 1-1. Examples of K^+ ions identified in catalytic RNAs.

A. The peptidyl transferase center (PTC) of the *Haloarcula marismortui* 50S ribosomal subunit contains a bound K^+ (PDB 1FFK) [36, 113]. In each picture, adenosines are red, cytosines are green, guanosines are yellow and uracils are color purple. K^+ ions are shown in blue.

B. A K^+ ion identified in the L11-associated portion of the 23 S rRNA from *Escherichia coli* (PDB 1HC8) [24, 41].

C. A K^+ ion coordinated by the tetraloop-tetraloop receptor of the *Azoarcus sp. BH72* group I intron (PDB 1ZZN) [23, 35, 114].

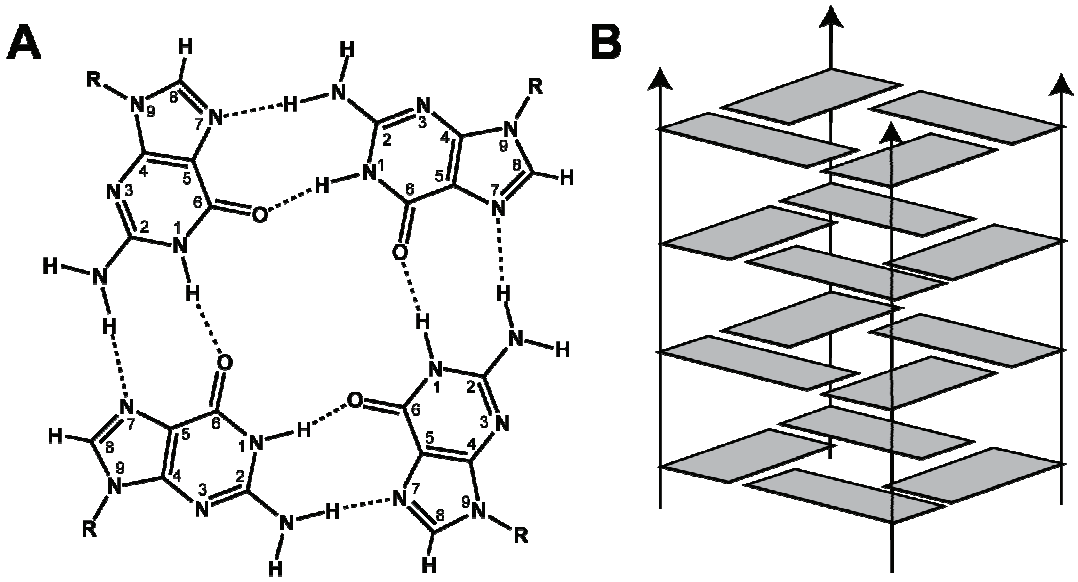


Figure 1-2. Schematic of G-quartet and an example G-quadruplex.

A. The structure of a G-quartet formed by four guanosine nucleosides. The base atoms have been numbered for convenience.

B. A G-quadruplex formed from four separate DNA strands and containing four G-quartets. The DNA strands in this G-quadruplex are all parallel. Figure adapted from [73].

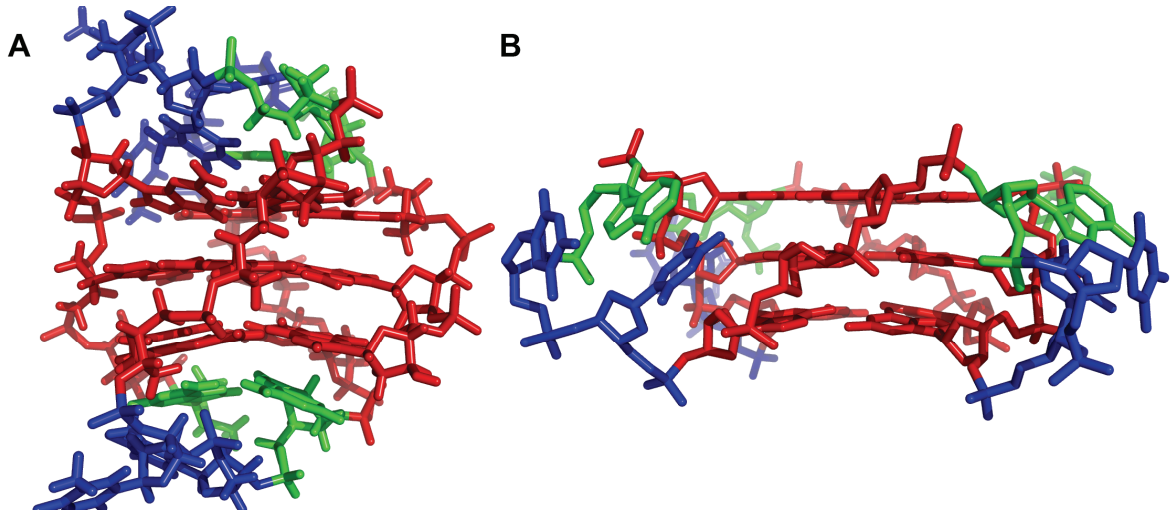


Figure 1-3. Ion-dependent structural variation in the Na^+- and K^+-forms of the human telomere sequence, d[AGGG(TTAGGG)$_3$].

A. The Na^+-form contains all three d(TTA) loops at either end of the G-quadruplex (PDB 143D) [87]. Guanines are shown in red, thymines in blue, and adenosines in green.

B. The K^+-form is characterized by d(TTA) loops which span three of the sides of the G-quadruplex (PDB 1KF1) [86].

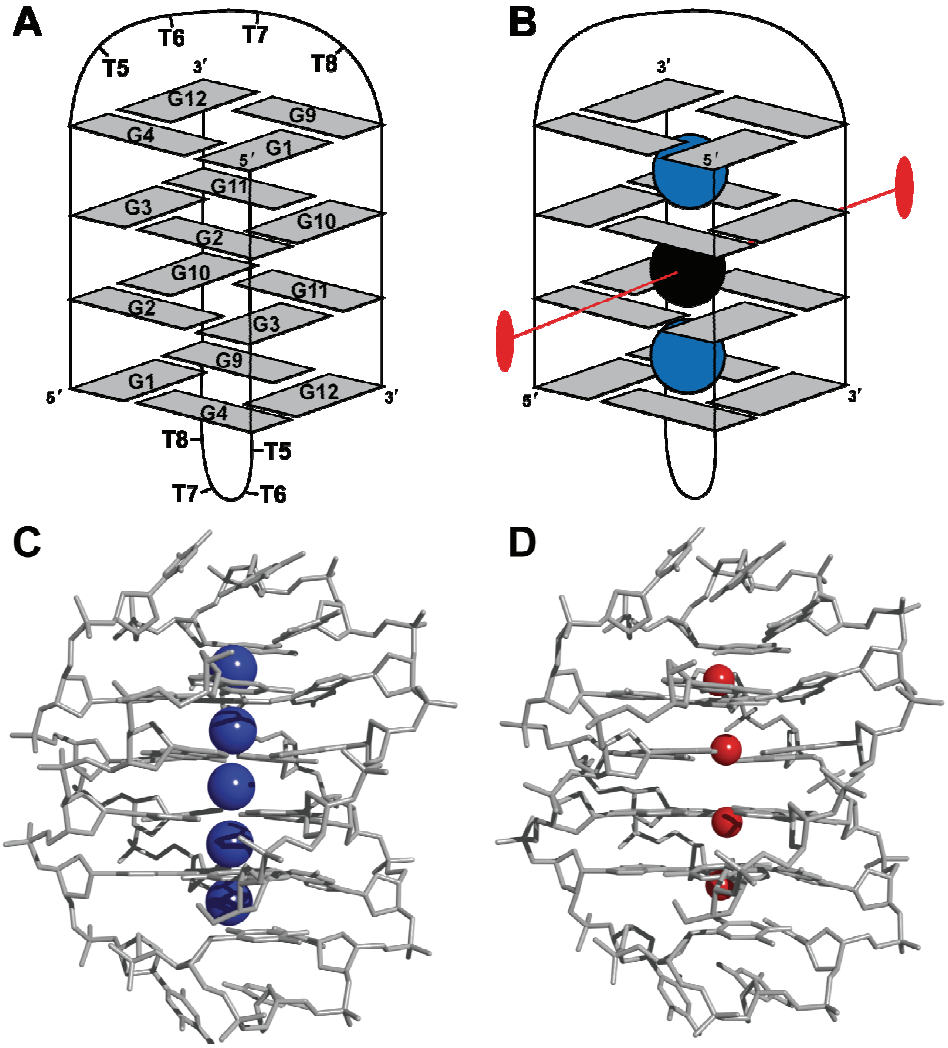


Figure 1-4. The G-quadruplex, $d(G_4T_4G_4)_2$, formed by the *Oxytricha nova* telomeric sequence.

A. $d(G_4T_4G_4)_2$ is a homodimer which contains four G-quartets. The diagonal loops are composed of thymine nucleotides.

B. The G-quadruplex (shown here with three bound metal ions colored black or blue) contains a rotational symmetry plane (red) which bisects the middle cation. The other cations are colored blue to emphasize their equivalence.

C. The K^+-form of $d(G_4T_4G_4)_2$ contains five K^+ ions (blue), three sandwiched between successive G-quartet planes and two in the loops [111]. NH_4^+ has also been shown to bind between G-quartet planes; however, no evidence exists for its binding to loops [65, 107].

D. A crystal structure of the Na^+-form of $d(G_4T_4G_4)_2$ contains four Na^+ ions. The smaller radius of Na^+ (compared to K^+) allows it to bind within a single G-quartet plane.

2 Solution Structure of the Tl^+-form of $d(G_4T_4G_4)_2$

2.1 Introduction

Structural heterogeneity, often related to the identity of the associated monovalent cation, is commonly reported in G-quadruplexes [86, 87, 115]. The G-quadruplex derived from *Oxytrica nova*, $d(G_4T_4G_4)_2$, is no exception to this observation. When studies of the Tl^+-form of $d(G_4T_4G_4)_2$ were initiated, four atomic-resolution structures were available—an NMR structure of the Na^+-form by Feigon and coworkers [108-110] and two structures of the K^+-form [107, 116]. The first of the two K^+ structures was a crystal structure reported by Rich and coworkers and the second was a solution structure also determined by Feigon and coworkers. A fourth structure, solved by x-ray crystallography, reported the Na^+-form of $d(G_4T_4G_4)_2$ in complex with the *Oxytricha nova* telomere binding protein [112]. The G-quadruplex in this structure is similar to the solution structure of the Na^+-form and, in the interest of simplicity, is not discussed further here.

The topology of the K^+-form in the crystal structure is very different from the solution structures of both the K^+- and Na^+-forms of $d(G_4T_4G_4)_2$. In the two solution structures, the G-quadruplex formed is an antiparallel homodimer with diagonal thymine loops at either end (Figure 2-1A). However, in the crystal structure of the K^+-form, the thymine loops span one side of a G-quartet, forming an edge looped structure (Figure 2-1B). Based on these structures, it was unclear whether the structural variation was related to crystallization conditions or the identity of monovalent cation present.

Despite having similar overall topologies, there are local variations in the solution structures of the Na^+- and K^+-forms. Specifically, the conformation of the two thymine

loops relative to the neighboring G-quartet is very different [107]. In the K^+-form, T8 stacks above and parallel to the proximal G-quartet (Figure 2-2). In the Na^+-form, however, T8 is more extended into solution, almost perpendicular to the G-quartet (Figure 2-2). The explanation for this structural heterogeneity is not precisely known, but Feigon and coworkers have suggested that the smaller ionic radius of Na^+ allows it to interact with the thymine loops while bound within the G-quadruplex channel [107].

Given the varying degrees of structural heterogeneity reported, we felt determining the solution structure of the Tl^+-form of $d(G_4T_4G_4)_2$ was a crucial step in accurately assessing to what degree Tl^+ can mimic K^+ within the context of a G-quadruplex. After efforts to determine the structure of the Tl^+-form were underway, Neidle and coworkers [111] reported a second crystal structure of the K^+-form which contained a diagonally looped topology, much like the two solution structures of $d(G_4T_4G_4)_2$. So, it seemed the K^+-form of $d(G_4T_4G_4)_2$ did not exhibit a different global conformation within a crystal lattice. However, the conformation of T8 within the thymine loops is extended into solution much like in the Na^+-form (Figure 2-2), adding an additional layer of complexity to the interpretation of structural variations in $d(G_4T_4G_4)_2$. Thus, establishing the isomorphic nature of K^+ and Tl^+ in nucleic acids remained an important goal.

The solution structure of the Tl^+-form of $d(G_4T_4G_4)_2$ and a comparison of it to the solution structures of the K^+- and Na^+-forms is reported here. The structure supports the ability of Tl^+ to specifically mimic K^+ in G-quadruplexes. These results have been published [117] and are described in more detail herein.

2.2 Materials and Methods

2.2.1 Materials and abbreviations.

Chemically synthesized DNA oligonucleotides, d(GGGGTTTTGGGG), were purchased from the W. M. Keck Facility, Yale University. Sep-Pak C_{18} desalting cartridges were from Waters, USA. Centricon 3 kDa NMWL centrifugal filter units were purchased from Millipore, USA. Acetonitrile, sodium phosphate monobasic (NaH_2PO_4), thallium nitrate ($TlNO_3$), and phosphoric acid (H_3PO_4) were from Sigma-Aldrich, USA. Ethylenediaminetetraacetic acid (EDTA-d_{12}), deuterium oxide (D_2O), and acetic acid-d_4 (CD_3COOD) were purchased from Cambridge Isotope Laboratories, USA.

2.2.2 G-quadruplex formation.

DNA oligonucleotides d(GGGGTTTTGGGG) were desalted using Sep-Pak C_{18} cartridges and eluted with 40% (v/v) acetonitrile/water. After lyophilization, the DNA was dissolved to a final concentration of ~500 μM in 50 mM NaH_2PO_4. G-quadruplex formation was facilitated by heating the solution to 363 K for 15 minutes followed by slow cooling to 277 K. The solution was concentrated to about 2.5 mM quadruplex (~5 mM DNA strand) by centrifugation and the buffer exchanged to 50 mM $TlNO_3$, 100 μM EDTA-d_{12} and either 10% or 100% D_2O. In some cases, G-quadruplex formation was also performed directly in the 50 mM $TlNO_3$ solution. The use of Na^+-containing solutions in the initial annealing step was done to reduce the amount of Tl^+ used. Similar ^1H and ^{205}Tl spectra were produced for G-quadruplexes formed in both manners. The sample pH was adjusted to 6.5 with H_2PO_4 or CD_3COOD.

2.2.3 NMR spectroscopy.

NMR experiments were performed on a Varian Unity Plus (14.1 T) spectrometer and a Unity Inova (11.75 T) spectrometer equipped with a cryogenically cooled probe. A

^1H–^1H NOESY [118-120] (with mixing times, τ_m = 80–350 ms), DQF-COSY, [118-120] and HOHAHA (using the DIPSI-2rc mixing sequence with a 10 kHz field and 125 ms mixing time) [118-123] spectra with WATERGATE [118] water suppression and States–TPPI [124, 125] frequency discrimination were collected using 2408 t_2 x 330 t_1 points, and a spectral width of 8000–9400 Hz in both dimensions. For each t_1 increment, 32–64 scans were acquired. The temperature was regulated at 298 K. Exchangeable protons were assigned from a NOESY collected at 278 K in 10% D_2O. Backbone assignments were aided with a ^{31}P–^1H CT-COSY [126] acquired with 2500 t_2 x 74 t_1 points with respective spectral widths of 2500 Hz and 1500 Hz. A total of 264 scans were acquired for each t_1 increment.

Distance calibration of NOE peak intensities was performed by semi-quantitative methods using crosspeaks corresponding to fixed distances (G/T H2′–H2″, G H8–H1, and T H6–methyl) assuming an r^{-6} relationship between peak intensity and distance. Experimentally determined constraints are listed in Appendix 1. After using characteristic NOE crosspeaks described by Smith and Feigon [109, 110] to verify that the topology of the G-quadruplex is a diagonally looped homodimer, the solution structure of the K^+-form of $d(G_4T_4G_4)_2$ (PDB 1K4X) [107] was used to assign crosspeaks residing near the symmetry axis. NOE constraints were duplicated for the symmetrically equivalent pair of protons in the G-quadruplex dimer.

2.2.4 Structure determination.

The solution structure of $d(G_4T_4G_4)_2$ was determined using CNS [4]. These calculations incorporated NOE derived distance constraints, known dihedral angles, hydrogen-bonds, and non-crystallographic symmetry constraints. Initial structures were

generated using distance geometry with simulated annealing. The structure refinement proceeded by rounds of *ab initio* simulated annealing involving high temperature torsion dynamics (2000 K for 60 ps with 15 fs time steps) followed by a Cartesian cooling stage (1000 K for 5 ps with 5 fs time steps) and restrained molecular dynamics (10 cycles of 200 steps each). The CNS script used for refinement and structural constraints are listed in Appendix 1. For each round of refinement, 100 structures were generated with the 10 lowest energy structures being averaged and used as a starting structure in the next round of calculations. Constraints with the highest number of violations were removed or loosened before proceeding with further rounds.

2.3 Results

A considerable amount of qualitative information about the global conformation of the Tl^+-form of $d(G_4T_4G_4)_2$ can be determined from the NMR data. The presence of imino (H1–H1) (Figure 2-3A) and strong aromatic (G H8–G H8) crosspeaks (Figure 2-3B) confirmed the existence of guanine–guanine pairing and are hallmarks of G-quadruplex structures [110]. Additionally, protons located in the thymine loops have faster transverse relaxation times and a reduced number of NOE crosspeaks (data not shown), both of which are consistent with a higher degree of conformation exchange in this region relative to the guanines. Further evidence of dynamics is provided by the reduced number of NOE crosspeaks to G12 which is located at the end of the G-quadruplex.

The aromatic–H1′ region contains information about glycosidic bond orientation as well as providing the basis for initial resonance assignments. The presence of "square" connectivities (G_i H8–G_i H1′–G_{i+1} H8–G_{i+1} H1′) rather than the canonical aromatic–

anomeric pattern is characteristic of a repeating *syn–anti* base conformation [107, 109, 110]. The connectivity from an H1′ proton of an *anti* base to a subsequent H8 proton of a *syn* base is too far (~8 Å) to observe an NOE crosspeak. Thus, the four *syn–anti* G connectivities could immediately be identified (Figure 2-4A). The observation of continuous H6–H1′ NOE crosspeaks for all T residues indicates that all thymines are in the *anti* conformation, as is energetically favored for pyrimidines. The crosspeaks observed between G4 H1′–T5 H6 helped to further reduce ambiguity in assignment possibilities. Non-sequential NOE crosspeaks in other regions (data not shown) and the high degree of consistency with published ^{1}H chemical shifts of the K^+- and Na^+-forms were used to complete the assignment of this region.

Assignment of the remainder of the sugar spin system was completed using data from the DQF-COSY (Figure 2-5) and HOHAHA (TOCSY) spectra (not shown). Qualitative determination of the sugar pucker (C2′- or C3′-*endo*) was made using H1′–H2′ COSY crosspeaks (Figure 2-5A). The C2'-*endo* conformation is characterized by a large $^3J_{H1'–H2'}$ coupling constant and a small $^3J_{H3'–H4'}$ coupling constant, while the opposite is true for C3'-*endo* sugar puckers [127, 128]. With the exception of T6, all sugar puckers were constrained to C2'-*endo*. The absence of a T6 H1′–H2′ COSY peak is consistent with a C3′–*endo* sugar pucker (Figure 2-5A). However, this absence could also be due to partial cancellation of the antiphase peak with the nearby H1′–H2″ peak (Figure 2-5B). Accordingly, the sugar pucker for T6 was left unconstrained during refinement.

With the exception of G12 H4′ and several amine protons (H21/H22), all resonances were assigned (Table 2-1). The resonances for G10 H2′ and H2″ are believed to be overlapped due to the repeated absence of a second ^{1}H resonance and the greatly

increased intensity of G10 crosspeaks to this sugar proton (NOESY, COSY, and TOCSY). The total number of non-redundant NOE distance restraints per DNA strand was 395. Of these, 192 occurred between different bases, 38 were long range (greater than one base apart), and 56 involved exchangeable protons (Table 2-1).

Refinement of the Tl^+-form of $d(G_4T_4G_4)_2$ in CNS proceeded by *ab initio* simulated annealing using NOE derived distance constraints. The final 10 lowest energy conformers (Figure 2-6) had no distance violations > 0.5 Å and no dihedral angle violations > 5º (Table 2-1) and were deposited in the PDB under accession number 2AKG. The all-atom RMSD of these ten structures was 0.76 ± 0.16 Å (Table 2-1).

The Tl^+-form of $d(G_4T_4G_4)_2$ is an antiparallel G-quadruplex with diagonally looped thymines at either end (Figure 2-6). The G-quadruplex has one narrow groove (11.7 Å wide, from C5′ to C5′), two intermediate grooves (14.1 Å), and a single, wide groove (17.3 Å). In the presence of Tl^+, T8 in the loop region stacks just above and parallel to the plane of the nearby G-quartet, composed of G4 and G9 from the identical strand and G1 and G12 from the opposing DNA strand (Figure 2-7A). NOE crosspeaks between the T7/T8 methyl group and the imino (H1) protons of G1, G4, and G12 define the loop conformation (Figure 2-7B).

2.4 Discussion

The diagonally looped conformation of the Tl^+-form of $d(G_4T_4G_4)_2$ is supported by the presence of aromatic–aromatic NOE crosspeaks between two (one *syn* and one *anti*) non-sequential guanines, as described by Smith and Feigon [110]. Crosspeaks meeting these criteria are observed between G1–G10, G2–G11, G3–G12, and G3–G10. By reversing the polarity of one strand, there are two possible ways to form a diagonally

looped G-quadruplex [110]; however, the presence of a G2–G11 NOE crosspeak is consistent only with the structure reported here.

The agreement between the Tl^+ and K^+ solution structures is excellent, with the two mean structures having an RMSD of 1.17 ± 0.13 Å. Chemical shift patterns in the H1 (imino), H8/H6 (aromatic), and H1' regions of the Tl^+-form are more similar to those reported for the K^+ NMR structure than the Na^+ NMR structure (Figure 2-4) [107]. The aromatic–H1' pattern of connectivities are nearly identical to the K^+ version (Figure 2-4A–C). Additionally, the thymine loop regions of the K^+ and Tl^+ structures also share a high degree of similarity with T8 stacking above G12 of the proximal G-quartet (Figure 2-8) in both structures. The loop region of the Tl^+-form of $d(G_4T_4G_4)_2$ also resembles the reported NMR structure of the NH_4^+-form [107]. In the solution structure of the Na^+-form of $d(G_4T_4G_4)_2$, T8 is pointed outwards in a more extended conformation (Figure 2-8) [107]. The occurrence of conformational exchange in the loop region likely causes some variation from the time average observed in the solution structure.

2.5 Conclusions

In solution, $d(G_4T_4G_4)_2$ forms a diagonally looped G-quadruplex in the presence of Tl^+. The similarity of this structure to the K^+-form in solution extends even to variable regions, such as the thymine loops. Establishing these similarities in a model system is an important first step in the development of ^{205}Tl NMR methods in nucleic acids.

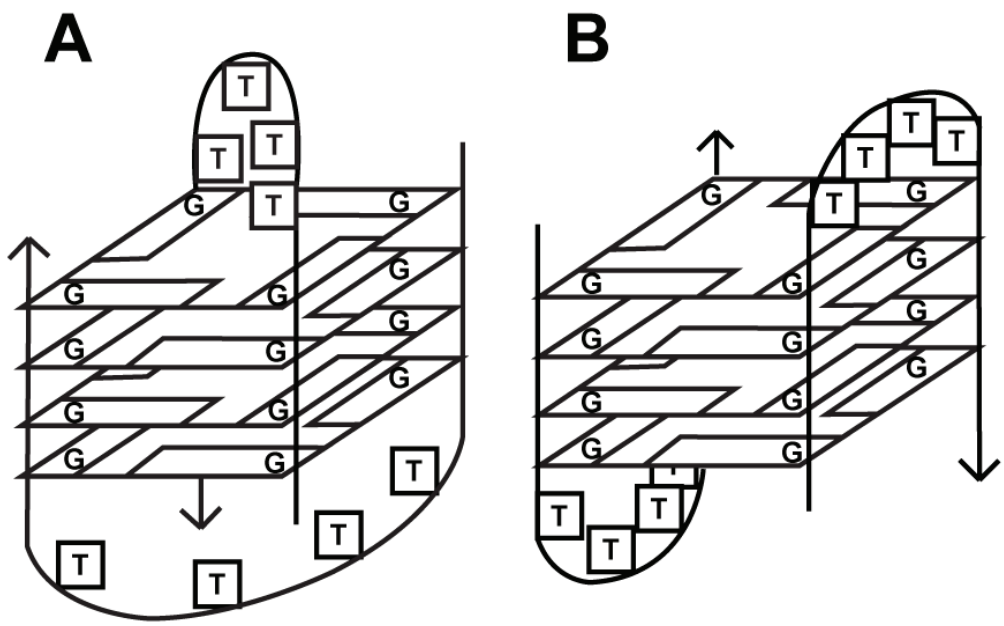


Figure 2-1. Different topologies observed in the G-quadruplex formed from the *Oxytricha nova* telomere sequence, $d(G_4T_4G_4)_2$.

A. The solution structures of the Na^+- and K^+-forms contain diagonal d(TTTT) loops [107, 110].

B. A crystal structure of the K^+-form displaying an edge looped topology [116].

Figure adapted from [85].

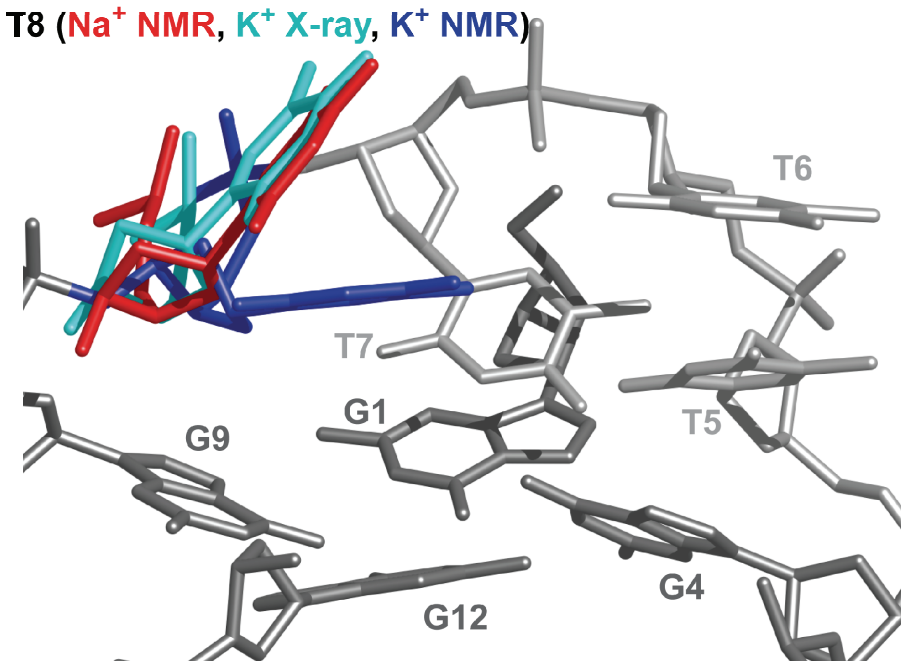


Figure 2-2. Conformational heterogeneity observed in the thymine loops of $d(G_4T_4G_4)_2$.

In the Na^+ NMR [109] and K^+ x-ray structures [111] (red and cyan), T8 is extended into solution. In the K^+ NMR structure [107], T8 (blue) is parallel to the adjacent G-quartet. For simplicity, only T8 is shown from the Na^+ NMR and K^+ x-ray structures.

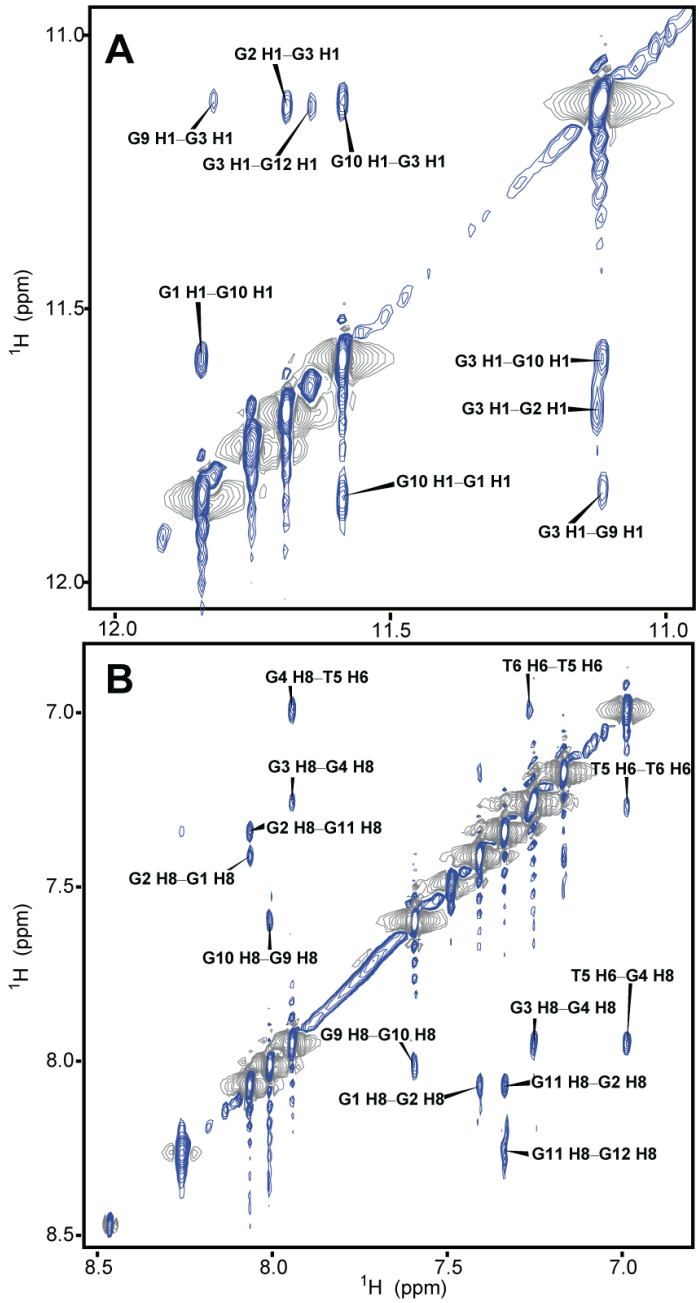


Figure 2-3. ^{1}H–^{1}H NOESY (τ_{mix}=350 ms) of the Tl^+-form of $d(G_4T_4G_4)_2$.
The sample contained 2.5 mM DNA, 50 mM $TlNO_3$, 0.1 mM EDTA-d_{12}, and 10% D_2O. The pH was adjusted to 6.5.

A. Imino–imino NOE crosspeaks are shown. Slowly exchanging imino protons are characteristic of G-quadruplex structures.

B. Aromatic–aromatic NOE crosspeaks are shown. H8–H8 crosspeaks between non-sequential *syn* and *anti* G bases (G1–G10, G2–G11, G3–G12, and G3–G10) confirmed the presence of a diagonally looped G-quadruplex structure [110].

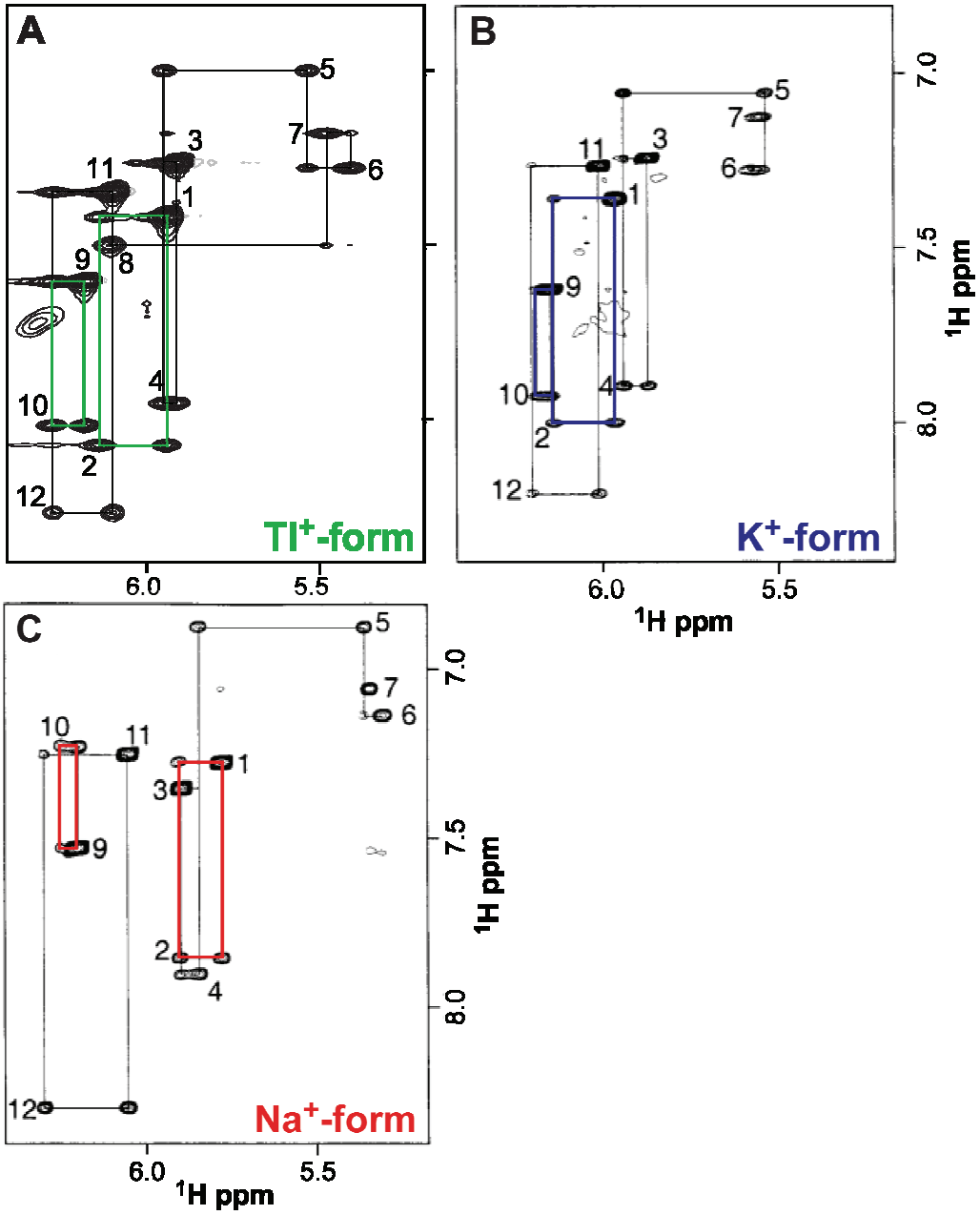


Figure 2-4. Comparison of the H8–H1′ connectivities observed for the solution structures of the Tl^+-, K^+-, and Na^+-forms of $d(G_4T_4G_4)_2$.
The Na^+ and K^+ structures are described in references [107-110].

A. The Tl^+-form of $d(G_4T_4G_4)_2$.

B. The K^+-form of $d(G_4T_4G_4)_2$. Figure adapted from Schultze *et. al.* [107].

C. The Na^+-form of $d(G_4T_4G_4)_2$. Figure adapted from Schultze *et. al.* [107].

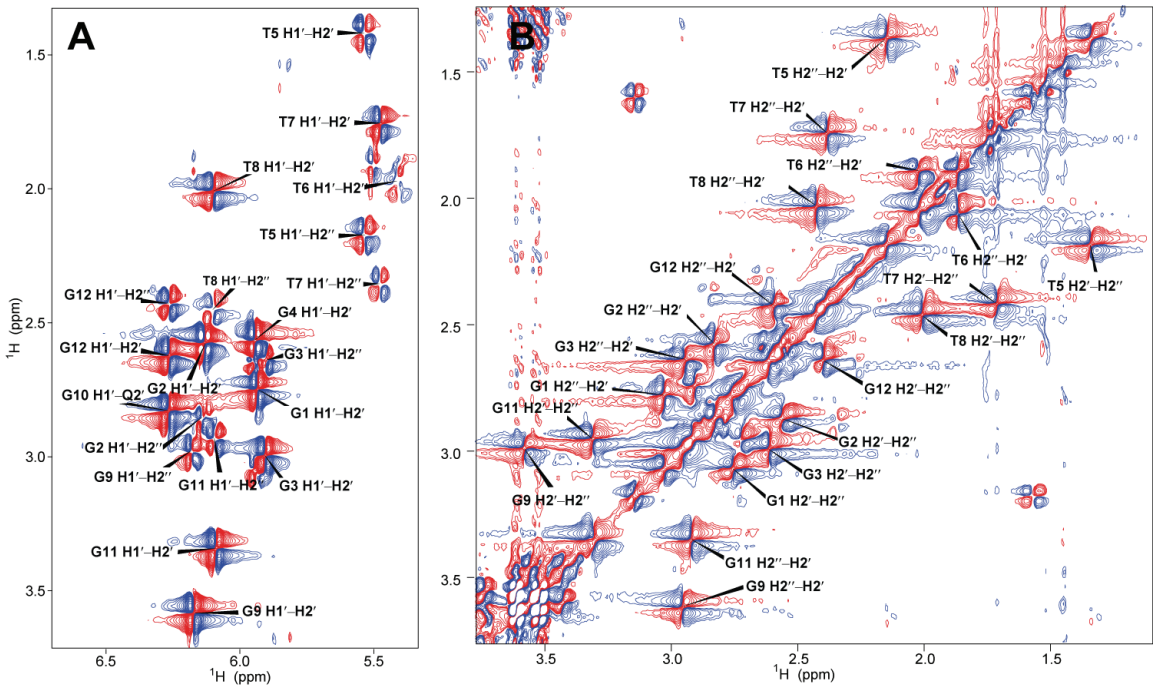


Figure 2-5. ^{1}H–^{1}H DQF-COSY used for the assignment of sugar resonances.
The sample contained 2.5 mM DNA, 50 mM $TlNO_3$, 0.1 mM EDTA-d_{12}, and 10% D_2O. The pH was adjusted to 6.5.

A. The H1′– H2′/ H2″ region.

B. The H2′– H2″ region.

A

	H8/H6	H1	H21	H22	Methyl
G1	7.43 ± 0.02	11.85 ± 0.01	6.31 ± 0.00	10.36 ± 0.00	-
G2	8.08 ± 0.02	11.61 ± 0.01	*na*	*na*	-
G3	7.25 ± 0.01	11.07 ± 0.01	6.24	8.33 ± 0.00	-
G4	7.95 ± 0.02	11.75 ± 0.00	*na*	*na*	-
T5	6.99 ± 0.02	-	-	-	1.77 ± 0.02
T6	7.27 ± 0.02	-	-	-	1.54 ± 0.02
T7	7.17 ± 0.02	-	-	-	1.46 ± 0.01
T8	7.53 ± 0.02	-	-	-	1.75 ± 0.00
G9	7.60 ± 0.02	11.83 ± 0.01	*na*	9.17 ± 0.01	-
G10	8.02 ± 0.02	11.53 ± 0.01	6.40 ± 0.00	9.37 ± 0.00	-
G11	7.34 ± 0.01	11.01 ± 0.01	6.70 ± 0.00	8.22 ± 0.00	-
G12	8.29 ± 0.02	11.62	*na*	*na*	-

B

	H1′	H2′	H2″	H3′	H4′	H5′/H5″		P
G1	5.93 ± 0.02	2.76 ± 0.02	3.05 ± 0.02	4.97 ± 0.01	3.87 ± 0.01	4.36 ± 0.01	3.97 ± 0.01	-
G2	6.12 ± 0.01	2.57 ± 0.02	2.85 ± 0.01	5.09 ± 0.01	4.07 ± 0.01	4.30 ± 0.02	4.43 ± 0.01	-0.97
G3	5.91 ± 0.02	2.97 ± 0.02	2.63 ± 0.02	5.04 ± 0.01	4.07 ± 0.01	4.16 ± 0.02	3.70 ± 0.01	-1.89
G4	5.94 ± 0.01	2.57 ± 0.02	2.65 ± 0.02	4.99 ± 0.01	4.16 ± 0.01	3.76 ± 0.01	3.82 ± 0.01	-0.97
T5	5.51 ± 0.01	1.36 ± 0.02	2.17 ± 0.02	4.54 ± 0.01	3.91 ± 0.01	4.34 ± 0.02	4.07 ± 0.01	-1.65 ± 0.01
T6	5.36 ± 0.01	2.04 ± 0.02	1.90 ± 0.05	4.55 ± 0.01	3.41 ± 0.02	3.48 ± 0.05	3.56 ± 0.05	-1.31 ± 0.00
T7	5.44 ± 0.02	1.72 ± 0.02	2.39 ± 0.01	4.47 ± 0.06	3.39 ± 0.01	3.96 ± 0.02	4.03 ± 0.01	-1.14
T8	6.15 ± 0.02	2.02 ± 0.01	2.44 ± 0.01	4.44 ± 0.01	2.74 ± 0.02	2.93 ± 0.02	4.69 ± 0.00	-2.64 ± 0.00
G9	6.18 ± 0.02	3.59 ± 0.02	2.98 ± 0.01	4.93 ± 0.01	4.08 ± 0.02	4.54 ± 0.00	4.65 ± 0.01	-1.48
G10	6.27 ± 0.02	2.82 ± 0.02		5.11 ± 0.01	4.11 ± 0.02	4.17 ± 0.01	4.32 ± 0.01	-0.97
G11	6.10 ± 0.02	3.33 ± 0.01	2.94 ± 0.01	5.10 ± 0.01	4.23 ± 0.02	4.50 ± 0.04	4.30 ± 0.01	-1.97 ± 0.00
G12	6.26 ± 0.02	2.62 ± 0.01	2.41 ± 0.02	4.30 ± 0.01	*na*	4.54	4.22 ± 0.01	-0.97

Table 2-1. Chemical shifts of the Tl^+-form of $d(G_4T_4G_4)_2$.
Proton resonances not assigned are denoted by '*na*'. Chemical shifts for exchange labile protons (H1, H21, and H22) were made using a sample containing 50 mM $TlNO_3$, 0.1 mM EDTA-d_{12}, and 10% D_2O at 5°C. All other shift assignments were made using a sample containing 100% D_2O at 25°C. Stereospecific assignments were not made for H5′/ H5″ protons. Assignment error is not listed when the redundancy is one.

A. The aromatic ^1H chemical shifts.

B. The sugar ^1H and ^{31}P chemical shifts.

NOE Restraints	
Total	395
Intraresidue	241
Interresidue	154
Long-range	38
Exchangeable	56
Hydrogen bond	32
Dihedral	26

Average	
NOE (> 0.5 Å)	0 ± 0
Dihedrals (> 5º)	0 ± 0

Ensemble RMSD (Å)	
All atoms	0.76 ± 0.16

Table 2-2. Structure statistics for the Tl^+-form of $d(G_4T_4G_4)_2$.

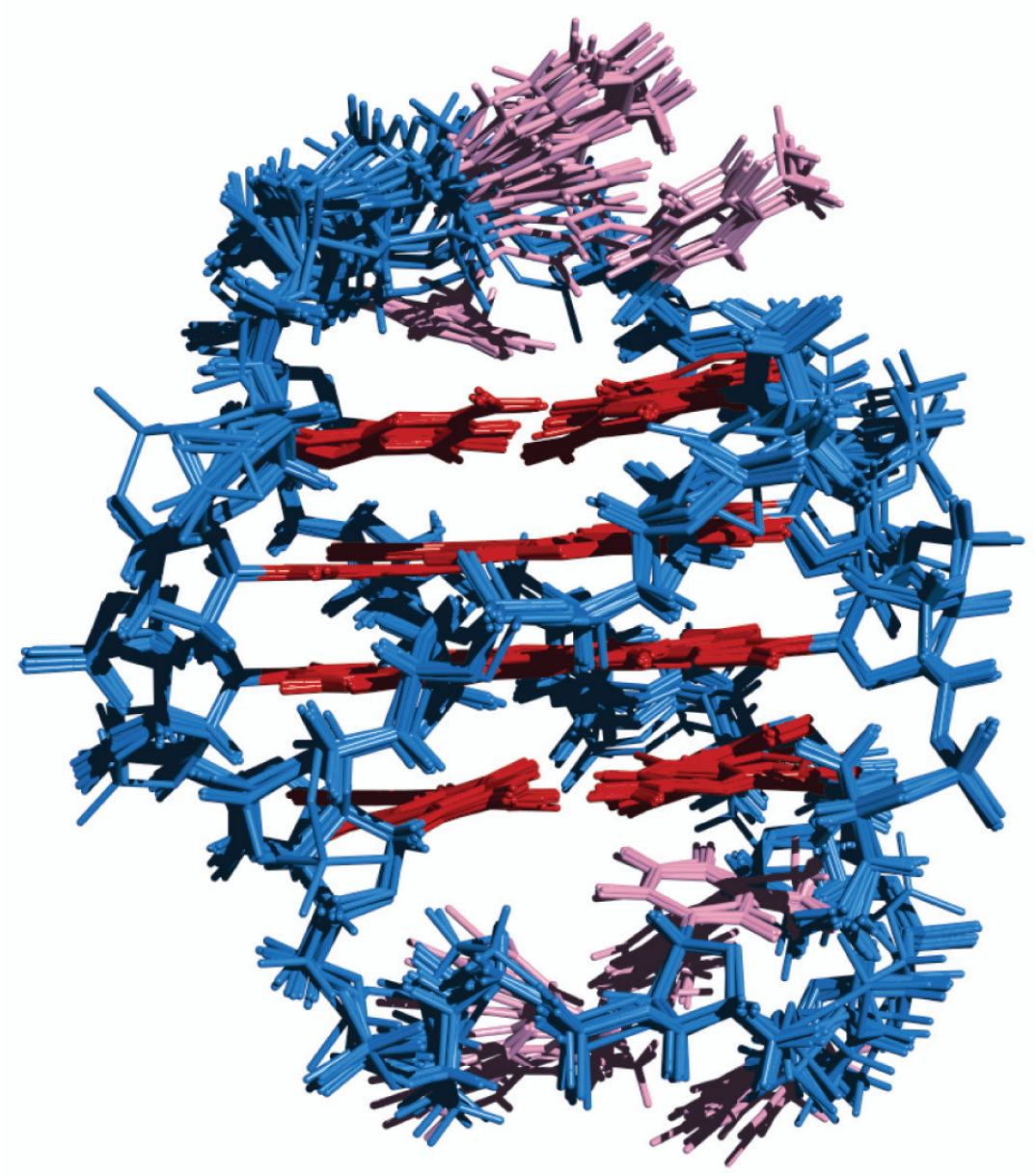


Figure 2-6. Solution NMR structure of the Tl^+-form of $d(G_4T_4G_4)_2$ (PDB 2AKG).
The ten lowest energy conformers are shown with the guanine nucleotides in red, thymine nucleotides in purple, and the deoxyribose-phosphate backbone in blue.

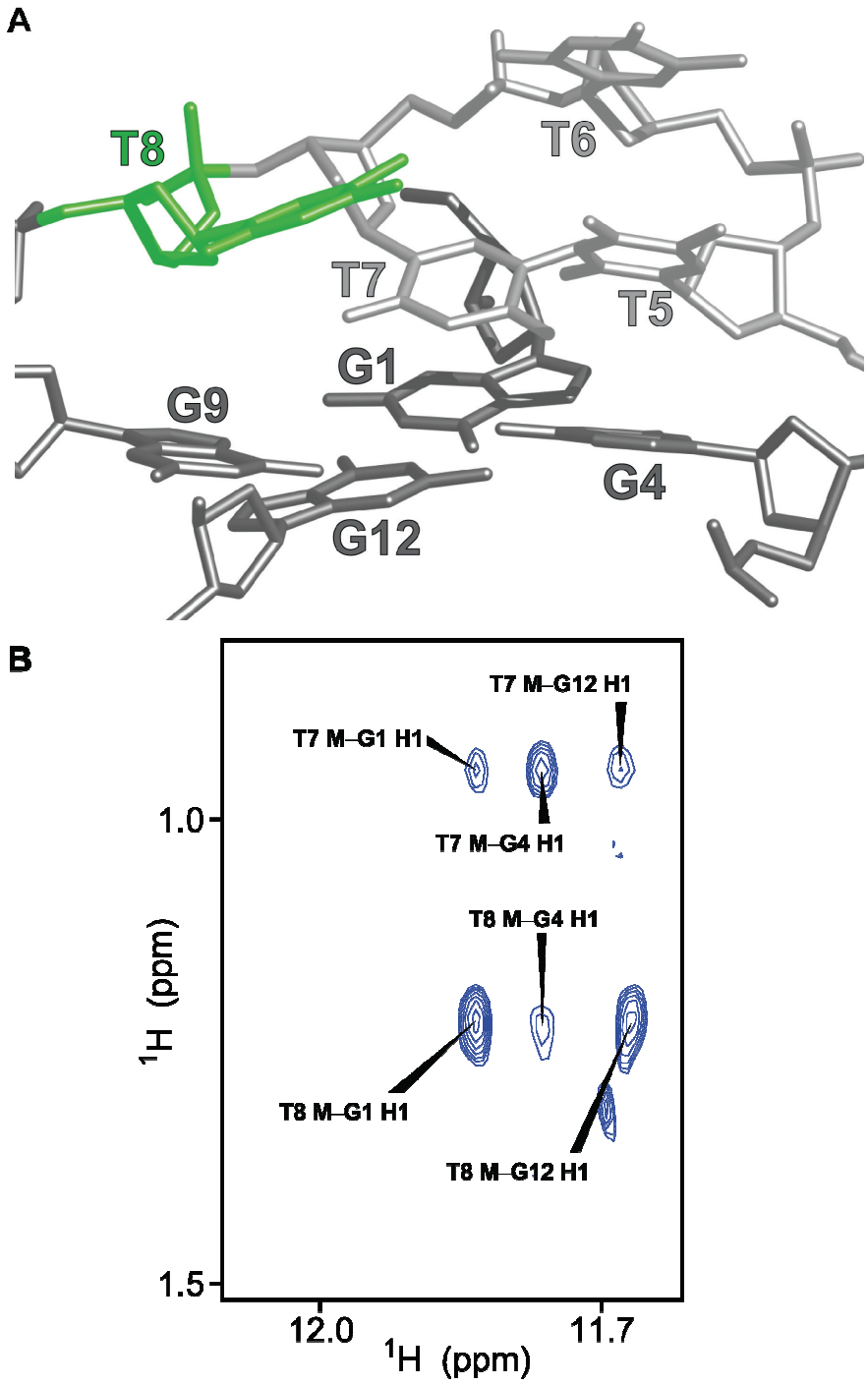


Figure 2-7. The conformation of the thymine loops in the Tl^+-form of $d(G_4T_4G_4)_2$.

A. T8 (green) stacks above G12 of the nearby G-quartet. G4 and G9 are from the same DNA strand as T8, while G1 and G12 are from the opposing DNA strand.

B. The loop conformation is supported by the presence of NOE crosspeaks from T8/T7 methyl protons to imino protons of the nearby G-quartet.

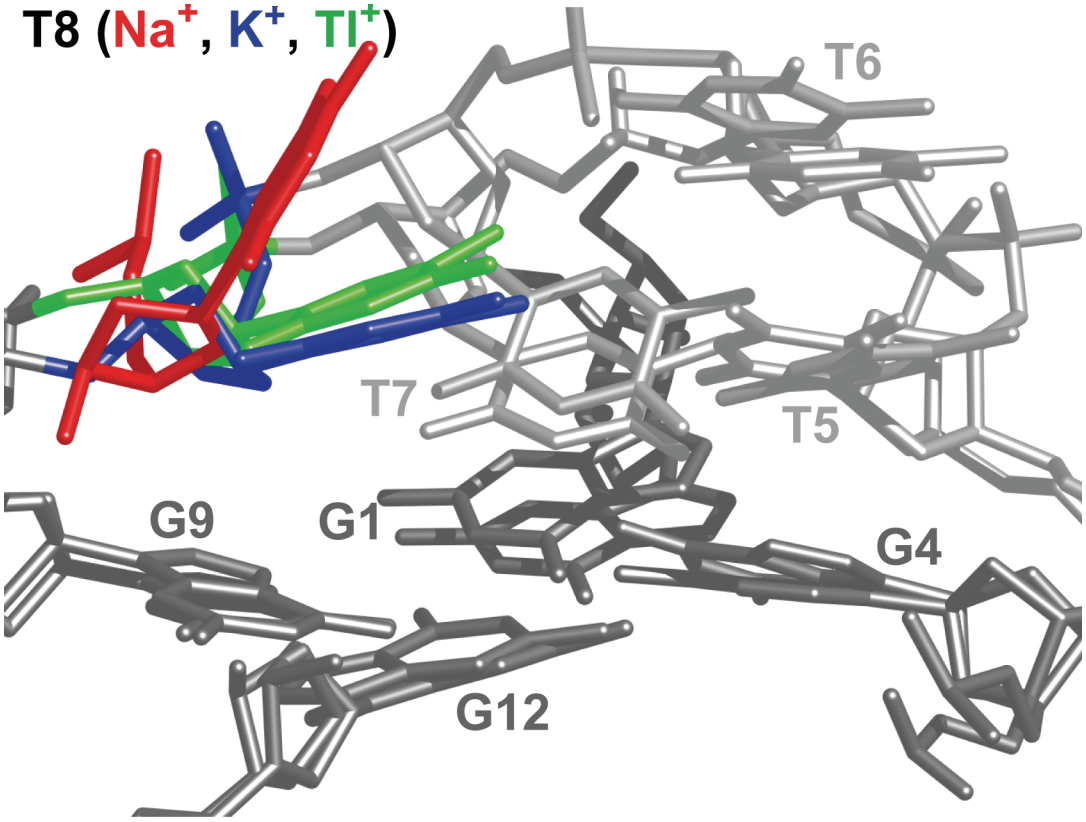


Figure 2-8. Comparison of loop conformations in the three solution structures (Na^+, K^+, and Tl^+) of $d(G_4T_4G_4)_2$.
T8 for the Na^+-form (red) is extended into solution, while in the K^+- and Tl^+-forms (blue and green) it stacks above the G-quartet formed by G1, G4, G9, and G12. For simplicity, T8 is the only base shown from the Na^+-form. The PDB IDs are 156D, 1K4X, and 2AKG for the Na^+-, K^+-, and Tl^+-forms.

3 Solution Studies of $^{205}Tl^{+}$ binding to $d(G_4T_4G_4)_2$

3.1 Introduction

Direct detection of ^{205}Tl nuclei is an appropriate first step in the development of solution ^{205}Tl NMR in nucleic acids. From these basic experiments, a substantial amount of information can be determined about both cation binding and the biological system itself. Solution NMR spectra of spin-½ metal surrogates (including $^{205}Tl^{+}$ and the divalent surrogate, $^{113}Cd^{2+}$) bound to proteins, ion channels, and antibiotics have revealed the number and specificity of cation binding sites, binding affinity, bound lifetime, activation energy of binding, and, in some cases, ligand type [11, 67, 73, 79, 80, 83, 129, 130].

Despite the wealth of information available, direct detection studies of ^{205}Tl binding to nucleic acids have been limited [60, 73]. The first of these two studies involved the G-quadruplex, $d(T_2G_4T_2)_4$. The reported ^{205}Tl direct detection spectrum contained a cluster of three small peaks about ~125 ppm downfield of a single, intense peak. No additional experimental-based information about peak assignment was reported. A second study utilized two systems, a DNA duplex formed by the self-complementary sequence d(GCAAAICTTTGC) and another G-quadruplex, $d(G_4T_4G_4)_2$ [60]. ^{205}Tl spectra of the DNA duplex demonstrated the feasibility of studying systems in which the bound cation is in fast exchange. One dimensional ^{1}H NMR spectra were used to show that Tl^{+} could displace Na^{+} in $d(G_4T_4G_4)_2$, causing chemical shift changes consistent with the conversion from the Na^{+}-form to a K^{+}-like form.

In order to fully establish ^{205}Tl NMR as a technique for the study of monovalent cation binding to nucleic acids, we wanted to further characterize the binding of Tl^{+} to

$d(G_4T_4G_4)_2$. The temperature dependence, specificity, bound lifetime, and general assignment of Tl^{+} ions coordinated by $d(G_4T_4G_4)_2$ have been determined by ^{205}Tl direct detection experiments. Portions of these results have been reported in a manuscript [117] and are described in greater detail here.

3.2 Materials and Methods

3.2.1 Materials and abbreviations.

Chemically synthesized DNA oligonucleotides, d(GGGGTTTTGGGG), d(GGGGBrdUTTTGGGG), d(GGGGTBrdUTTGGGG), d(GGGGTTBrdUTGGGG), and d(GGGGTTTBrdUGGGG) were purchased from the W. M. Keck Facility, Yale University. (BrdU denotes 5-bromo-2′-deoxy uracil.) Sep-Pak C_{18} desalting cartridges were from Waters, USA. Centricon 3 kDa NMWL centrifugal filter units were purchased from Millipore, USA. Acetonitrile, sodium phosphate monobasic (NaH_2PO_4), thallium nitrate ($TlNO_3$), potassium phosphate dibasic (K_2HPO_4), cesium nitrate ($CsNO_3$), phosphoric acid (H_3PO_4), and tetramethyl ammonium nitrate ($TMA\text{-}NO_3$) were from Sigma-Aldrich, USA. Ethylenediaminetetraacetic acid ($EDTA\text{-}d_{12}$), deuterium oxide (D_2O), and acetic acid-d_4 (CD_3COOD) were purchased from Cambridge Isotope Laboratories, USA.

3.2.2 Formation of G-quadruplexes.

The DNA oligonucleotides, d(GGGGTTTTGGGG), were desalted using Sep-Pak C_{18} cartridges and eluted with 40% (v/v) acetonitrile/water. After lyophilization, the DNA was dissolved to a final concentration of ~500 μM in 50 mM NaH_2PO_4. G-quadruplex formation was facilitated by heating the solution to 363 K for 15 minutes followed by slow cooling to 277 K. The solution was concentrated to about 2.5 mM quadruplex (~5 mM DNA strand) by centrifugation and the buffer exchanged to 50 mM

$TlNO_3$, 100 μM EDTA-d_{12} and 10% D_2O. The sample used for the K^+ titration contained only 25 mM $TlNO_3$. Samples containing bromine modifications were protected from light during preparation. In some cases, G-quadruplex formation was also performed directly in the 50 mM $TlNO_3$ solution. The use of Na^+-containing solutions in the initial annealing step was done to reduce the amount of Tl^+ used. Similar ^1H and ^{205}Tl spectra were produced for G-quadruplexes formed in both manners. The sample pH was adjusted to 6.5 with H_2PO_4 or CD_3COOD.

Monovalent cation titrations using Cs^+, TMA^+, and K^+ were performed by adding small amounts of concentrated solutions (2.5–3.0 M) of $CsNO_3$, TMA-NO_3, or K_2HPO_4 to the G-quadruplex sample. The total change in DNA concentration due to dilution was not more than 10%. NMR peak areas were corrected for this small amount of sample dilution.

3.2.3 NMR spectroscopy.

All ^{205}Tl NMR experiments performed at 11.75 T (288 MHz ^{205}Tl) used a Varian Inova wide bore spectrometer. Direct detection ^{205}Tl NMR experiments were performed with a Nalorac direct detection dual broadband probe using a pulse-acquire experiment with 90,000–100,000 acquisitions, a 60 kHz spectral width, and a 400–500 ms recycle delay. All ^{205}Tl spectra were externally referenced to samples containing identical concentrations of $TlNO_3$, EDTA, D_2O, and any additional ions present (Cs^+, TMA^+, or K^+). Further details of the experimental setup are provided in Appendix 2. ^1H NMR spectra were acquired on the same instrument using either the second broadband channel of the same Nalorac probe or a Nalorac indirect detection broadband probe. The ^{205}Tl NMR spectrum collected at 7.0 T (173 MHz ^{205}Tl) using a Varian Unity spectrometer

was performed in the Department of Chemistry and Biochemistry at the University of Notre Dame in a similar manner. All samples were incubated at the appropriate temperature for 60 min before experimental acquisition.

The exchange rate of $^{205}Tl^{+}$ coordinated by the G-quadruplex with the bulk solution was determined using magnetization transfer experiments [131-133]. The free ^{205}Tl resonance was saturated with a low power selective pulse for variable times (0, 50, 100, 200, 300, 400, and 500 ms). The effect of this saturation pulse on the bound ^{205}Tl resonances was measured with a $\pi/2$ pulse immediately following the saturation period. For each saturation time, a reference spectrum was acquired in which the frequency of the saturating pulse was placed upfield of the free ^{205}Tl resonance. A total of 25,000 interleaved acquisitions were collected for each saturation time with a 1 second recycle delay.

All NMR data were processed in NMRPipe [134]. Peak intensities were analyzed using NMRPipe and in-house written MATLAB (Mathworks, Inc.) code. The magnetization transfer data were analyzed using nonlinear least-squares regression, which was performed in MATLAB using the Levenberg-Marquardt method [135-137].

3.3 Results

The NMR observation of $^{205}Tl^{+}$ ions bound to G-quadruplexes was achieved by direct detection using a pulse-acquire sequence. The one-dimensional ^{205}Tl spectrum of $d(G_4T_4G_4)_2$ shows an intense peak located at approximately 5–10 ppm and a cluster of four (numbered 1–4), broader peaks centered at about 115 ppm (Figure 3-1). The linewidths (full width at half height) of these four peaks are 2.7, 1.9, 1.5, and 1.5 kHz for resonances 1–4, respectively. Small distortions of the lorentzian lineshape indicate that

Peak 1 may be inhomogeneously broadened. The linewidth of the upfield peak is approximately 440 Hz, which is considerably broader than that observed in the absence of any G-quadruplex (~100 Hz) (data not shown). The approximate areas of the downfield peaks are 2:1:1 for peaks 2–4, with peak 1 integrating to 1.5–2.0.

To assess the ability of Tl^{+} to stabilize the G-quadruplex and further examine the nature of these peaks, we examined the temperature dependence of each of the ^{205}Tl peaks. The linewidth of the ^{205}Tl resonances is extremely sensitive to temperature. The downfield peaks broaden progressively with increasing temperature while simultaneously shifting upfield (Figure 3-2). Peaks 1 and 4 appear to be somewhat more temperature sensitive as their areas, relative to peaks 2 and 3, are considerably reduced at 333 K. At lower (283 K) and higher (> 323 K) temperatures, peak 3 appears to be split by about 800 Hz. At 338 K, no downfield resonances remain. The simultaneous disappearance of the imino proton peaks confirms the melting of the G-quadruplex at this temperature (Figure 3-2). These results indicate that the binding of Tl^{+} stabilizes $d(G_4T_4G_4)_2$ to a similar degree as that provided by other monovalent metals (Na^{+} and NH_4^{+}) and differ somewhat from the results of Basu *et. al*. [73], where Tl^{+} is reported to bind more tightly to the G-quadruplex $d(TG_4T)_4$ than K^{+}. However, the melting temperature of the K^{+}-form of $d(G_4T_4G_4)_2$ would have to be measured to directly confirm this conjecture.

The linewidth of the upfield resonance decreases from 283 K to 313 K and then increases again to almost 1 kHz at 333 K before then decreasing sharply at 338 K (Table 3-1). The behavior at low temperatures is consistent with the predicted longer correlation time (τ_c) of the G-quadruplex. As the temperature is increased to 333 K, intermediate

exchange likely influences the linewidth, with the melting of the complex at 338 K causing the final decrease.

The position and intensity of the downfield peaks are resistant to excess concentrations of TMA^{+} or Cs^{+} (Figure 3-3). However, the upfield Tl^{+} resonance is shifted 3–5 ppm upfield from when TMA^{+} or Cs^{+} are added, respectively (Figure 3-4A). In addition the linewidth of this upfield ^{205}Tl resonance narrows from 443 Hz to 356 Hz and 413 Hz to 386 Hz upon addition of TMA^{+} or Cs^{+} (Figure 3-4B).

Unlike TMA^{+} and Cs^{+}, the addition of K^{+} has considerable effects on both the downfield and upfield ^{205}Tl resonances. The initial four resonances decrease in intensity and new resonances appear upfield of peaks 1–4 as the concentration of K^{+} is increased (Figure 3-5). Though specific assignment of these additional resonances cannot be made directly, they most likely correspond to G-quadruplexes where one or more K^{+} ions have been substituted for Tl^{+}. At 300 mM K^{+} (6X Tl^{+} concentration) all downfield ^{205}Tl resonances have disappeared. The conversion of the Tl^{+}-form to the K^{+}-form is supported by the ^{1}H spectra of the imino protons. The chemical shifts of these protons do not change upon further addition of K^{+} (500 mM), indicating that the conversion is complete (data not shown). The small peak at ~68 ppm is likely due to the binding of residual Na^{+} used in sample preparation. The upfield ^{205}Tl resonance shifts from 6.3 ppm to 3.5 ppm as the K^{+} concentration is increased to 300 mM.

The lifetime of $^{205}Tl^{+}$ at each of the binding sites defined by the four downfield ^{205}Tl resonances was determined as described by Forsén and Hoffman [131, 132], except that nonlinear least-squares regression was used for parameter fitting. Briefly, the decay

of the downfield ^{205}Tl resonances (site A) upon saturation of the bulk ^{205}Tl resonance (site B) for time (t) is given by

$$M_z^A = M_0^A[\frac{\tau_{1A}}{\tau_A}e^{(-t/\tau_{1A})} + \frac{\tau_{1A}}{T_{1A}}] \tag{3-1}$$

where M_0^A is the initial magnetization at site A when $t = 0$ and M_z^A is the magnetization as $t \rightarrow \infty$. T_{1A} is the spin-lattice relaxation time at site A and τ_A is the bound lifetime at site A. T_{1A} and τ_A are related by the equation

$$\frac{1}{\tau_{1A}} = \frac{1}{\tau_A} + \frac{1}{T_{1A}} \tag{3-2}$$

Combining equations (3-1) and (3-2) yields,

$$\frac{M_z^A}{M_0^A} = [(1 - \frac{\tau_{1A}}{T_{1A}})e^{(-t/\tau_{1A})} + \frac{\tau_{1A}}{T_{1A}}] \tag{3-3}$$

The values of $\frac{\tau_{1A}}{T_{1A}}$ and τ_{1A} were determined by nonlinear least-squares regression with equation (3-3). The results are shown for each of the downfield ^{205}Tl peaks as a function of saturation time of the free ^{205}Tl resonance (Figure 3-6 and Table 3-2). From these values, the spin-lattice relaxation time (T_{1A}) and the bound lifetime (τ_A) were determined for the $^{205}Tl^{+}$ ion bound to each respective site on the G-quadruplex (Table 3-2). The bound lifetimes range from 80 ± 10 ms for peak 1 to 155 ± 65 ms for peak 4. The errors in the fits are based on a 95% confidence interval.

To determine the effect of thymine substitutions on the chemical shifts of the downfield ^{205}Tl resonances, G-quadruplexes were made which have a 5-bromo-2′-deoxyuracil (BrdU) substituted for the T5, T6, T7, or T8. BrdU is identical to thymine

except that the C5 pyrimidine methyl group has been changed to a bromine. In each case, the formation of a G-quadruplex was verified by the presence of imino protons in a 1D ^{1}H NMR spectrum before acquiring the ^{205}Tl NMR spectrum (Figure 3-7). Based on the intensities of the ^{1}H spectrum, G-quadruplexes can form with BrdU substitutions at all four positions. The intensities of the imino protons in the T8→BrdU substituted G-quadruplex are slightly lower than the other three indicating that this alteration may reduce G-quadruplex formation somewhat (Figure 3-7). The ^{205}Tl NMR spectrum of the T5→BrdU G-quadruplex is very different from the other three, with possibly peak 2 and either peak 3 or peak 4 remaining. $^{205}Tl^{+}$ may bind less preferentially to the T8→BrdU substitution, although this could also be related to the overall reduced level of G-quadruplex formation (Figure 3-7).

Interpretation of the ^{205}Tl spectra of the BrdU substituted G-quadruplexes based on the predicted location of the thymine methyl groups is not straightforward. In the solution structure of the Tl^{+}-form of $d(G_4T_4G_4)_2$, the BrdU substitution would be predicted to have the greatest effect at T8 with lesser or no effects observed at T5, T6, and T7 (Figure 3-8). While the reduced level of T8→BrdU G-quadruplex formation could be consistent with this expectation, no significant perturbation of ^{205}Tl chemical shifts is observed (Figure 3-7). Further, the absence of almost any ^{205}Tl resonances in the spectrum of the T5→BrdU G-quadruplex is not expected based on the location of the T5 methyl group. One possible explanation is that the bromine in the T5→BrdU G-quadruplex has some alternate interaction with Tl^{+}, possibly related to the ability of bromine to interact with positively charged ions, which causes the Tl^{+} ions to enter the intermediate exchange regime. However, based on the spectra of the imino protons, this

interaction would have to disrupt Tl^+ binding without inhibiting G-quadruplex formation. Structural studies of the BrdU substituted G-quadruplexes would be required to conclusively demonstrate the presence of an alternate bromine–Tl^+ interaction. Temperature dependence studies could also determine if the bromine–Tl^+ interaction affects G-quadruplex stability.

3.4 Discussion

The ^{205}Tl direct detection experiments have confirmed that Tl^+ can be observed bound to $d(G_4T_4G_4)_2$ and that at least some of the bound Tl^+ ions are in slow exchange with the bulk solution (Figure 3-1). This is consistent with the results reported by Strobel and coworkers [73]. Further, the unfolding temperature (338 K) (Figure 3-2) indicates that the Tl^+ stabilizes $d(G_4T_4G_4)_2$ at least as well as K^+, NH_4^+, and Na^+, in whose presence the reported unfolding temperatures are 333 K, 328 K and between 308–338 K, respectively [59, 65, 138]. Finally, the bound lifetimes of the downfield peaks, ranging from 80–150 ms, are similar to that determined for $^{15}NH_4^+$ in $d(G_4T_4G_4)_2$ (250 ms) by lineshape analysis [65].

The ability of K^+ to compete for all binding sites represented by the downfield ^{205}Tl peaks (Figure 3-5) indicates that none of them are specific to Tl^+. The requirement of a stoichiometric excess of K^+ to abolish all downfield ^{205}Tl resonances is consistent with previous studies reporting Tl^+ to have a higher affinity than K^+ for monovalent binding sites [39, 40, 73].

The observation of four downfield peaks in the ^{205}Tl direct detection experiment was not expected (Figure 3-1). Based on solution studies of the NH_4^+ form of $d(G_4T_4G_4)_2$ [65, 107], it was anticipated that three $^{205}Tl^+$ cations would bind $d(G_4T_4G_4)_2$, each sandwiched between two adjacent G-quartet planes. Since the outer two binding sites are

related by a single rotational symmetry plane (Figure 1-4B), the anticipated scenario would result in the observation of two downfield ^{205}Tl peaks with relative areas of 2:1 (outer:inner). If Tl^{+} were also binding to the thymine loops as in the crystal structure of the K^{+}-form [111], three peaks would be expected with relative areas of 2:2:1 (loops:outer:inner). (The later assumption assumes complete metal occupancy within the loops, as is reported for K^{+} in the crystal structure [111].) However, neither of these two scenarios was observed. Thus, we have considered three possible explanations for the observed number of downfield ^{205}Tl peaks: (1) the occurrence of ^{205}Tl–^{205}Tl scalar couplings leading to peak splittings; (2) the existence of high-affinity, $^{205}Tl^{+}$-specific binding sites within the G-quadruplex grooves; and (3) the occurrence of conformational exchange, which falls into the slow exchange regime on the ^{205}Tl chemical shift timescale.

The areas of the downfield ^{205}Tl peaks do not correspond to any predicted canonical pattern of ^{205}Tl–^{205}Tl peak splittings, even when multiple couplings and peak overlap are considered. However, the binding of metal cations to $d(G_4T_4G_4)_2$ is predicted to result in the close juxtaposition of at least three $^{205}Tl^{+}$ cations. For this reason, the field dependence of the downfield ^{205}Tl peaks was examined by repeating the direct detection experiment at 7.0 T, 173 MHz ^{205}Tl (data not shown). The resulting ^{205}Tl spectrum has a considerably reduced signal-to-noise level compared to spectra acquired at 11.7 T. However, to the extent that we can determine, the relative peak positions are not consistent with those expected from scalar couplings at this static field strength.

It is possible that some of the downfield peaks result from $^{205}Tl^{+}$ binding to one or more site(s) within the four G-quadruplex grooves. There is a precedent for monovalent

cation-specific association in the major and minor grooves of B-form DNA [56, 66, 139]. One such example involved preferential binding of Tl^+ to the major and minor grooves of G-tract regions [139]. Interestingly, no K^+ cations were assigned to G-quadruplex grooves within the crystal structure of the K^+-form of $d(G_4T_4G_4)_2$ [111]. Furthermore, when included in a predominately Na^+-containing crystallization solution, Tl^+ was not reported to bind within the grooves of the parallel-stranded G-quadruplex, $d(TG_4T)_4$, despite its ability to compete effectively for binding sites between the G-quartet planes [106]. It also seems unlikely that nonspecific associations, such as those that would be expected from groove binding, would remain at temperatures as high as 333 K (*vide supra*). This conclusion is further supported by the invariability of peaks 1–4 to concentrations of Cs^+ or TMA^+, which are added to 6-fold excess of the Tl^+ concentration. Cs^+, in particular, has been shown to have a much higher affinity than K^+ for binding to the G-quadruplex grooves [100]; thus Cs^+ would be expected to displace $^{205}Tl^+$ from any backbone sites and result in a decrease in intensity for that ^{205}Tl peak. This effect is not observed. Thus, if Tl^+ does bind in the G-quadruplex grooves, it does so with a very high affinity and in a manner that has thus far precluded its detection by NMR studies of $d(G_4T_4G_4)_2$ and crystallography of other G-quadruplex systems. However, further questions about the location of the Tl^+ binding sites specifically in $d(G_4T_4G_4)_2$ are best resolved by crystallography.

The occurrence of conformational exchange is another possible explanation for the observed ^{205}Tl resonances. This may initially seem inconsistent with the observation of only one set of ^1H resonances (Figure 2-3); however the large ^{205}Tl chemical shift range affords a much larger limit on the slow exchange regime. Assuming the smallest

separation between any two of the downfield ^{205}Tl resonances is approximately 20 ppm (Figure 1-2), $\Delta\omega \approx 36{,}000\ s^{-1}$ at 11.7 T. For two resonances differentiated by conformational exchange to lie within a similar limit on the ^{1}H timescale, they would have to be separated by over 11.5 ppm at this same field strength. If a generous chemical shift difference of 2 ppm is considered for the ^{1}H chemical shifts, $\Delta\omega \approx 6{,}000\ s^{-1}$ which is still 1/6 of the limit on the ^{205}Tl timescale. Thus, it is highly possible that populations undergoing conformational exchange could lie within the slow exchange regime on the ^{205}Tl timescale and the fast exchange regime on the ^{1}H timescale. The potential to study a greater range of conformers under the slow exchange regime illustrates another advantage of using ^{205}Tl direct detection methods for the study of monovalent cation binding sites.

If conformational exchange is the explanation for the number of observed ^{205}Tl resonances, then the possible sources of this exchange come into question. The two most likely possibilities are the movement of Tl^{+} ions through the G-quadruplex or the G-quadruplex itself. It does not seem that the exchange of Tl^{+} through the G-quadruplex channel is the cause because the relatively long bound lifetime observed for each of the peaks would almost certainly place the resulting conformers in slow exchange on both the ^{205}Tl and ^{1}H timescales (Table 3-1). This leaves the G-quadruplex as the most likely candidate for conformational exchange. Indeed, the thymine loops do seem to be dynamic, as demonstrated by their elevated transverse relaxation and reduced NOE crosspeak intensities (see Chapter 2).

3.5 Conclusions

Direct detection ^{205}Tl NMR can be used to observe the binding of $^{205}Tl^{+}$ to $d(G_4T_4G_4)_2$, as previously described [60, 73]. Based on the melting temperature of the Tl^{+}-form of $d(G_4T_4G_4)_2$, the stabilization afforded by Tl^{+} binding is similar to that provided by Na^{+} and NH_4^{+}. The bound lifetime and resistance of the downfield ^{205}Tl resonances to high concentrations of Cs^{+} and TMA^{+} indicates that they are specifically bound by $d(G_4T_4G_4)_2$. Further, the sensitivity of these ^{205}Tl peaks to high concentrations of K^{+} implies that none of these sites are due to adventitious Tl^{+} binding.

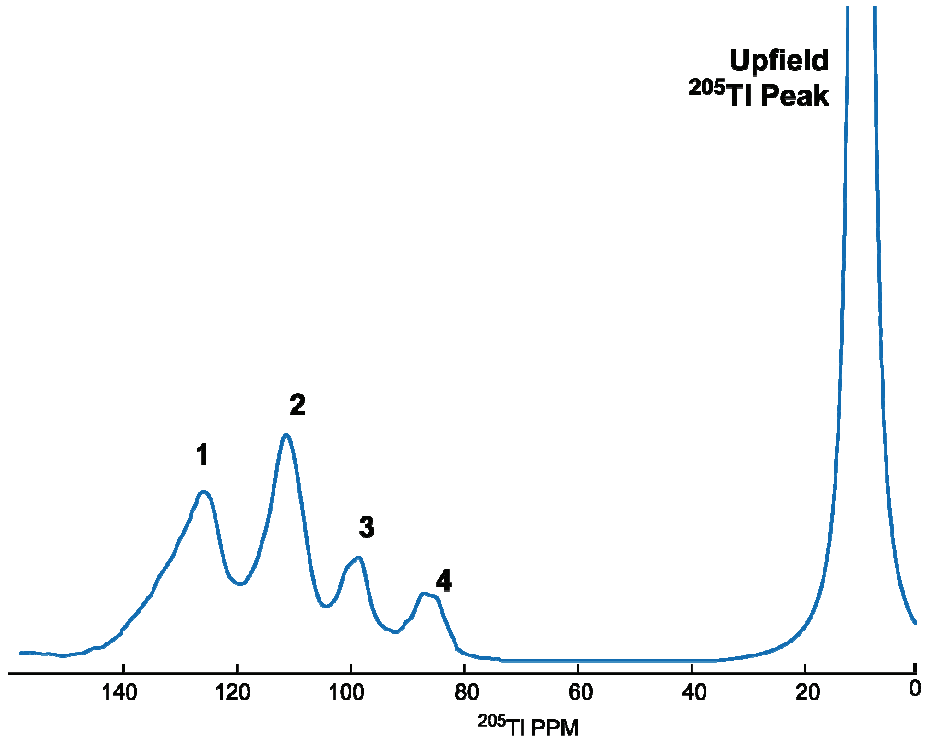


Figure 3-1. ^{205}Tl NMR spectrum of the Tl^{+}-form of $d(G_4T_4G_4)_2$.
The sample contained 2.5 mM G-quadruplex, 50 mM $TlNO_3$ 0.1 mM EDTA-d_{12}, and 10% D_2O at pH 6.5 and was regulated at 298 K. The downfield peaks are numbered 1–4.

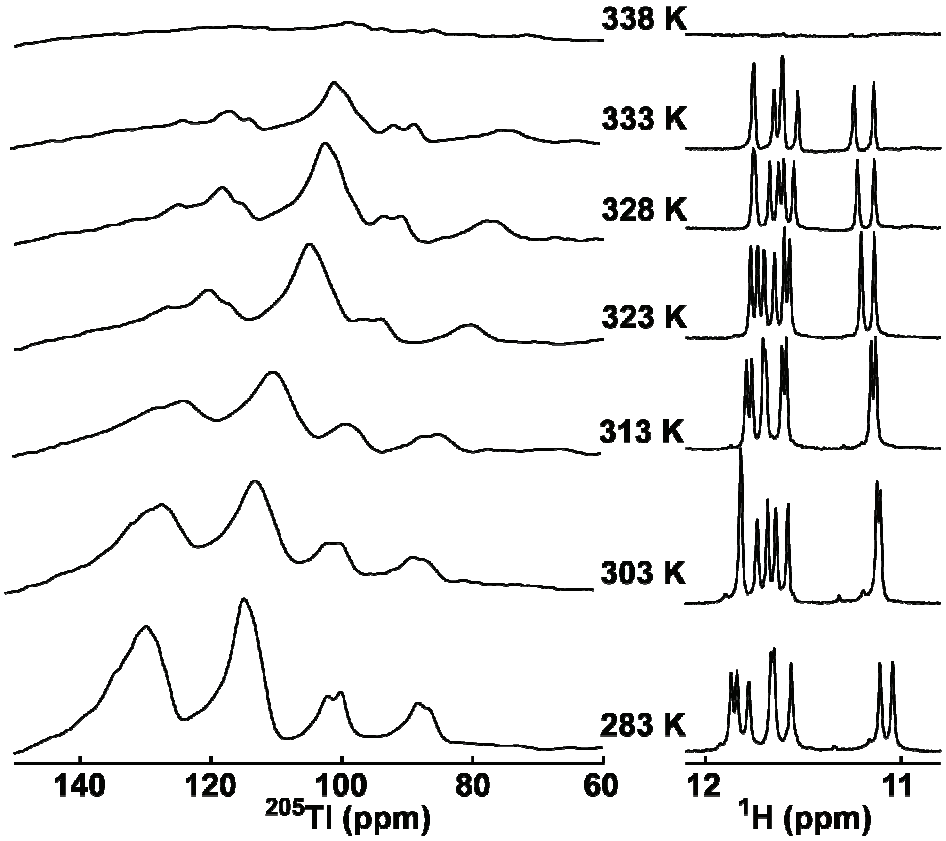


Figure 3-2. Temperature dependence of the downfield ^{205}Tl resonances.
The sample contained 2.5 mM G-quadruplex, 50 mM $TlNO_3$ 0.1 mM EDTA-d_{12}, and 10% D_2O at pH 6.5. The downfield region of the ^{205}Tl spectrum is shown on the left and the ^{1}H imino region is shown on the right. The sample was incubated for 60 minutes at the appropriate temperature before data acquisition.

Temperature (K)	Linewidth (Hz)
283	541
303	425
313	294
323	319
328	578
333	921
338	489

Table 3-1. Effect of temperature on the linewidth of the upfield ^{205}Tl resonance.

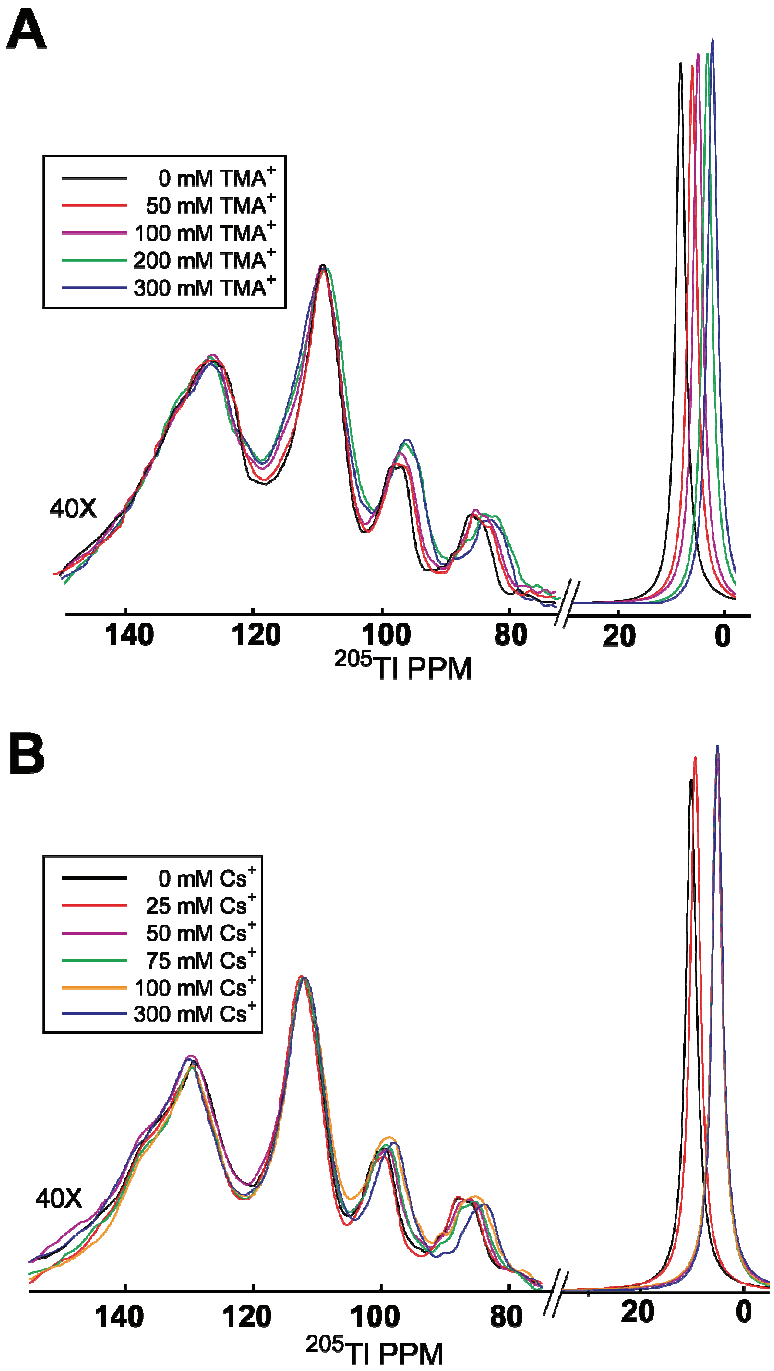


Figure 3-3. Ion titrations of TMA^{+} and Cs^{+} into $d(G_4T_4G_4)_2$.
The sample contained 2.5 mM G-quadruplex, 50 mM $TlNO_3$ 0.1 mM EDTA-d_{12}, and 10% D_2O at pH 6.5 and was regulated at 298 K. The downfield regions is vertically expanded 40X relative to the upfield region.

A. Titration of tetramethyl ammonium nitrate ($TMA\text{-}NO_3$).

B. Titration of cesium nitrate ($CsNO_3$).

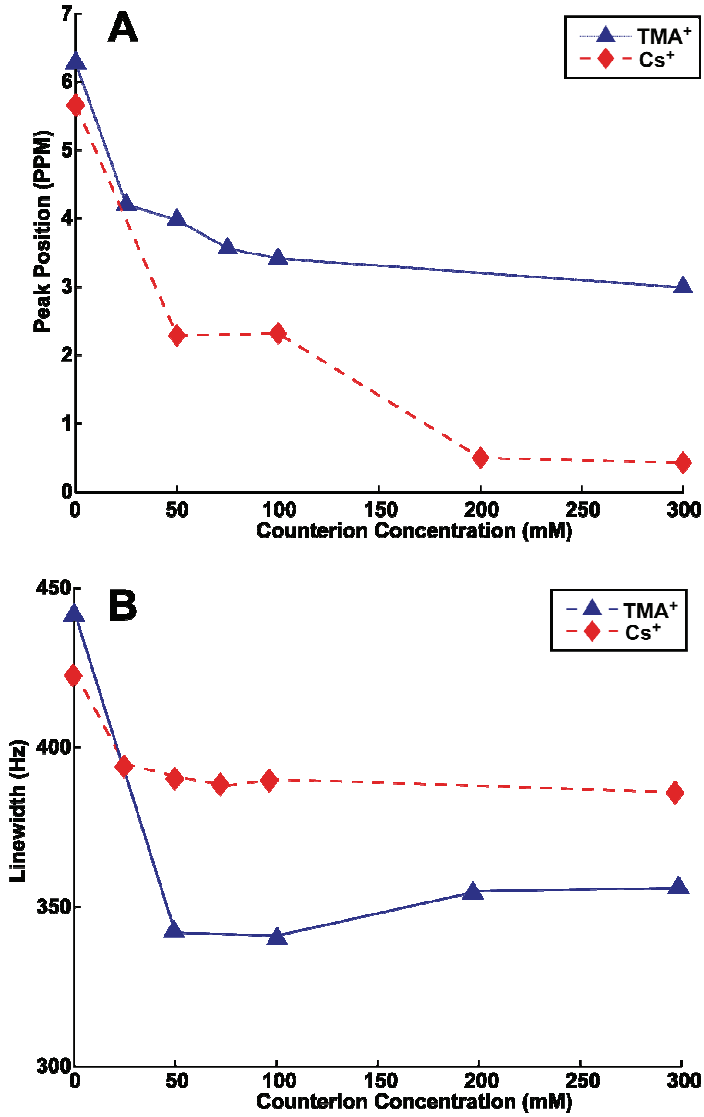


Figure 3-4. Effect of TMA^{+} and Cs^{+} of position and linewidth of upfield ^{205}Tl peak.

A. Position (^{205}Tl ppm) of the upfield peak as either TMA^{+} (blue) or Cs^{+} (red) is added.

B. Linewidth (Hz) of the upfield peak as either TMA^{+} (blue) or Cs^{+} (red) is added.

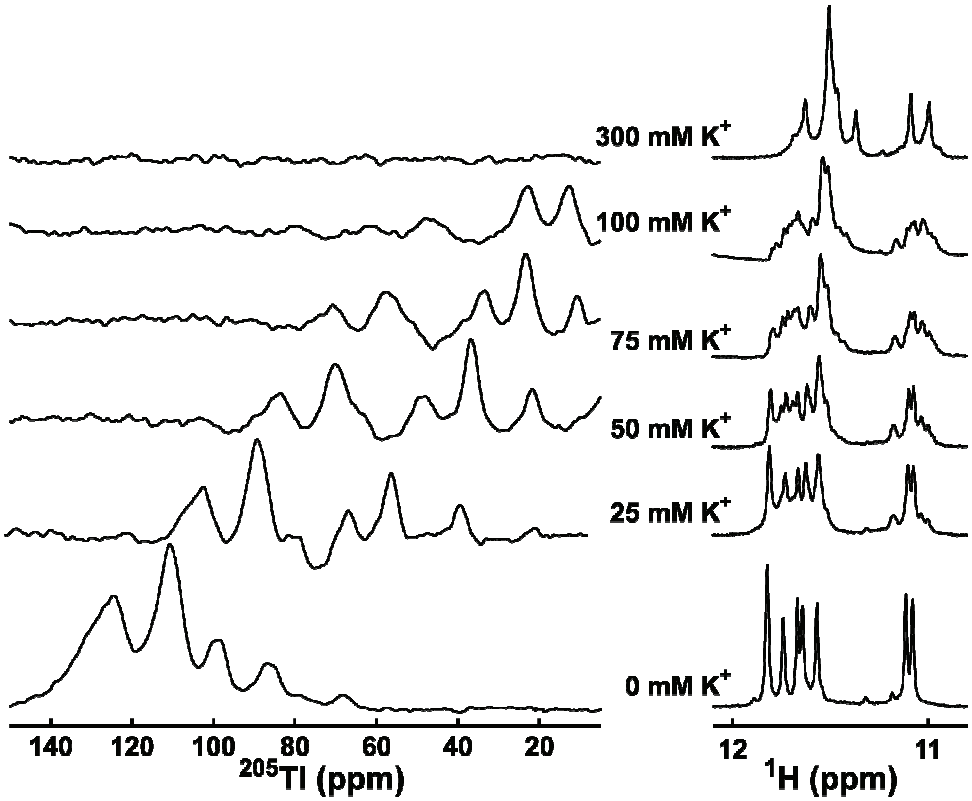


Figure 3-5. Titration of K^+ into $d(G_4T_4G_4)_2$.
The sample contained 2.5 mM G-quadruplex, 50 mM $TlNO_3$ 0.1 mM EDTA-d_{12}, and 10% D_2O at pH 6.5 and was regulated at 298 K. The downfield region of the ^{205}Tl spectrum is shown on the left and the ^1H imino region is shown on the right.

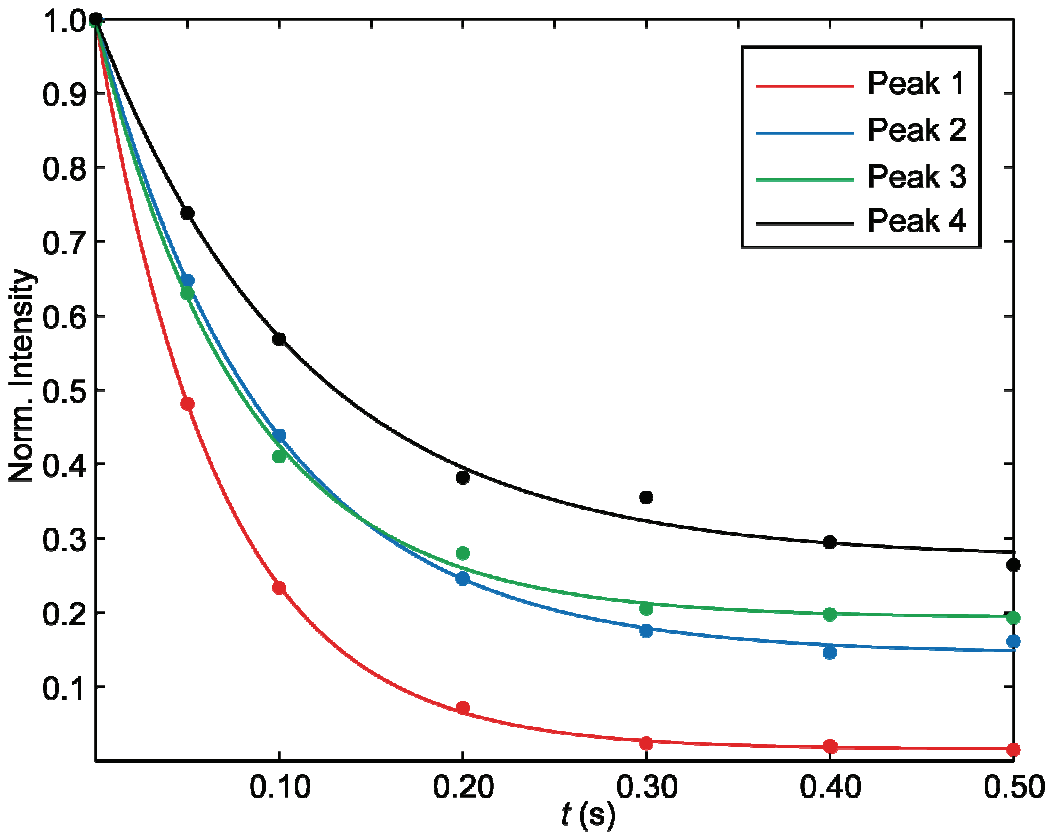


Figure 3-6. The decay of the intensity of the downfield ^{205}Tl peaks upon saturation of the free ^{205}Tl resonance, for time (*t*, seconds).
Solid lines indicate results of nonlinear least squares fitting to equation (3-3) with R_2 = 0.9999, 0.9995, 0.9987, and 0.9965 for peaks 1–4, respectively.

	Peak 1	**Peak 2**	**Peak 3**	**Peak 4**
$\frac{\tau_{1A}}{T_{1A}}$	0.16 ± 0.01	0.14 ± 0.02	0.20 ± 0.03	0.27 ± 0.05
τ_{1A} **(s)**	0.07 ± 0.01	0.09 ± 0.01	0.08 ± 0.01	0.11 ± 0.03
T_{1A} **(s)**	0.41 ± 0.02	0.64 ± 0.10	0.41 ± 0.09	0.42 ± 0.14
τ_{A} **(s)**	0.08 ± 0.01	0.11 ± 0.01	0.10 ± 0.02	0.15 ± 0.06

Table 3-2. Lifetimes (τ_A) of bound $^{205}Tl^{+}$ ions.

$\frac{\tau_{1A}}{T_{1A}}$ and τ_{1A} were determined by nonlinear least squares regression. τ_A is calculated using equation (3-2).

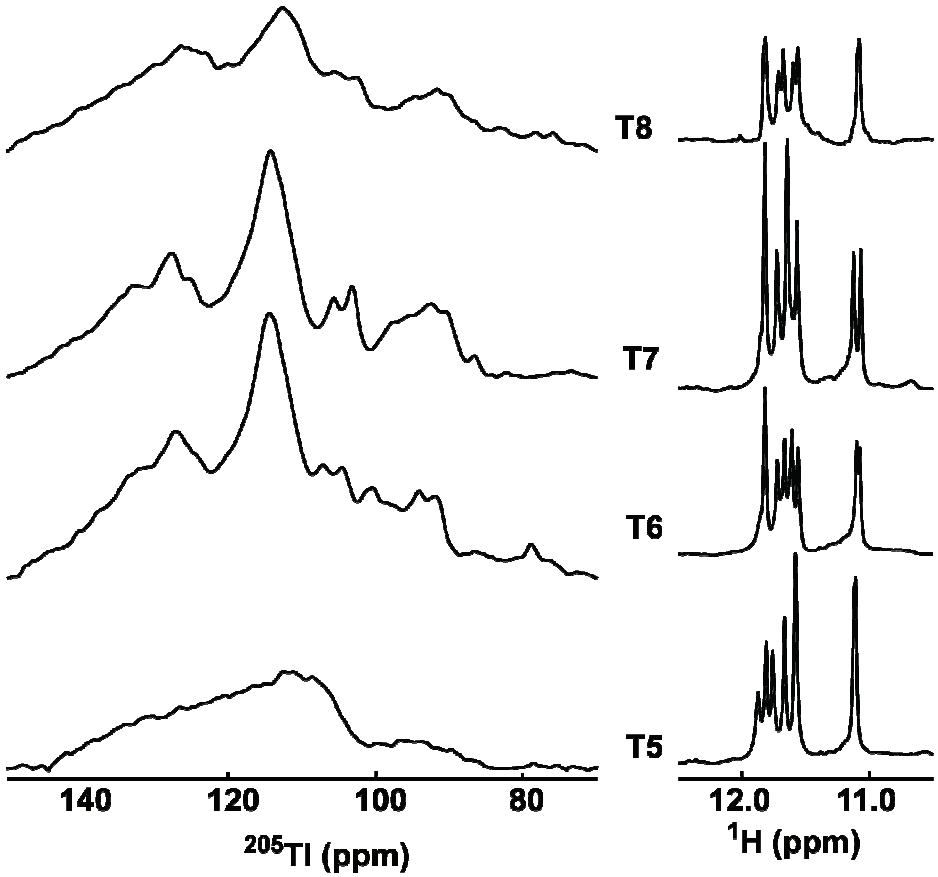


Figure 3-7. Effect of 5-bromo-2′-deoxyuracil (BrdU) substitution for T5, T6, T7, or T8. The downfield region of the ^{205}Tl spectrum is shown on the left and the ^{1}H imino region is shown on the right.

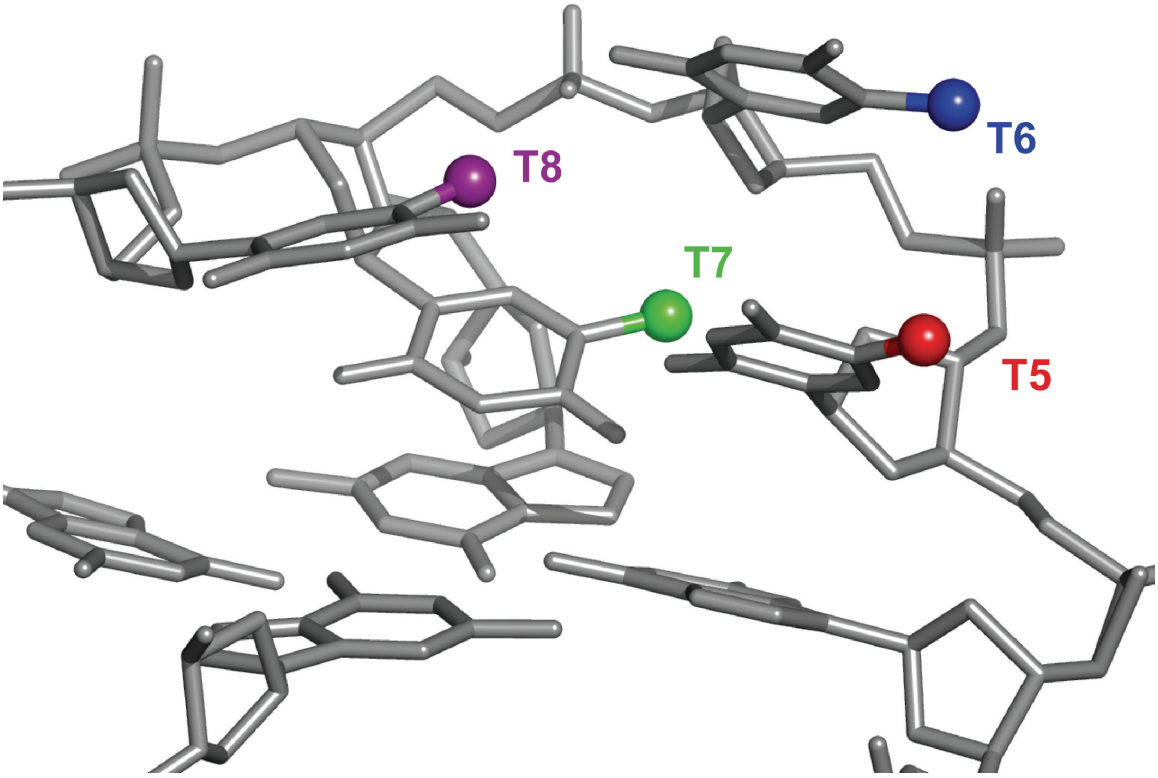


Figure 3-8. Approximate position of BrdU substitution in each of the thymines. The position is based on the solution structure of the Tl^{+}-form of $d(G_4T_4G_4)_2$ (PDB 2AKG). T5, T6, T7, and T8 are colored red, blue, green, and purple, respectively.

4 Crystallization of the Tl^+-form of $d(G_4T_4G_4)_2$

4.1 Introduction

The use of thallium's anomalous signal for crystallographic phase determination is particularly well-established in nucleic acids [22-24, 35, 106, 114, 139, 140]. However, the crystal structure of the K^+-form was solved using a combination of phosphorothioate and 5-bromo-2′-deoxyuracil substitutions. Thus, there is no crystallographic evidence of how many Tl^+ ions bind to $d(G_4T_4G_4)_2$.

To help reconcile resonances observed in the ^{205}Tl direct detection spectrum (Figure 3-1) and conflicting reports of other monovalent binding (K^+ and NH_4^+) [65, 111] to $d(G_4T_4G_4)_2$ with actual Tl^+ binding sites, we decided to crystallize the Tl^+-form of $d(G_4T_4G_4)_2$. Information from this crystal structure will provide strong evidence for the existence of any Tl^+ binding sites in the thymine loops and/or G-quadruplex grooves. Knowledge of these binding sites will be crucial in the site specific assignment of each of the ^{205}Tl resonances.

4.2 Materials and Methods

4.2.1 Materials and abbreviations.

Chemically synthesized DNA oligonucleotides, d(GGGGTTTTGGGG), were purchased from the W. M. Keck Facility, Yale University. Sep-Pak C_{18} desalting cartridges were from Waters, USA. Acetonitrile, thallium acetate (TlAc), potassium acetate (KAc), magnesium acetate ($MgAc_2$), spermine, potassium hydroxide (KOH), cacodylic acid, and 2-methyl-2,4-pentanediol (MPD) were from Sigma-Aldrich, USA.

4.2.2 Crystallization conditions.

DNA oligonucleotides d(GGGGTTTTGGGG) were desalted using Sep-Pak C_{18} cartridges and eluted with 40% (v/v) acetonitrile/water. After lyophilization, the DNA was dissolved to a final concentration of 10 mM in H_2O.

The crystallization solution is listed in Table 4-2. The crystallization buffer, 50 mM potassium cacodylate, was made by adjusting the pH of cacodylic acid to 6.5 using KOH and then diluting the solution to the appropriate concentration. The final concentration of K^+ in the buffer was approximately 85 mM.

G-quadruplex formation was facilitated by heating d(GGGGTTTTGGGG), KAc, potassium cacodylate, and H_2O to 358 K for 15 minutes followed by slow cooling to 277 K. The DNA concentration during annealing was ~2.14 mM. After annealing, appropriate volumes of 10X stocks of $MgAc_2$, spermine, and MPD were added to the crystallization solution, making the final DNA concentration 1.5 mM. The solutions were then centrifuged at 14,000 X *g* for 30 minutes to remove any precipitate. The well solution contained only 35% (v/v) MPD.

Crystals were grown using the hanging drop method (2 μL drops) at 18°C and appeared after 6–8 weeks as clear, and rod-like crystals. Stabilization was performed by soaking the crystals in solutions containing 60% MPD and 50 mM TlAc for two hours prior to freezing in liquid nitrogen. Attempts were also made at crystallizing $d(G_4T_4G_4)_2$ when TlAc was partially (50%) or completely substituted for KAc in the crystallization solution. However, no crystals were produced from these efforts.

4.2.3 Structure determination.

Data were collected on beamline X25 at the National Synchrotron Light Source at 0.9780 Å wavelength and diffracted to 1.55 Å. The data were processed using the HKL

2000 package [141]. Experimental phases were determined using molecular replacement with the orthorhombic crystal structure determined by Neidle and coworkers (PDB 1JRN) [111]. The space group ($P2_12_12_1$) and unit cell dimensions were similar to the published structure (Table 4-1).

Refinement was performed using Refmac5 [142-145] in the CCP4 program suite [146]. Several cycles of rigid body refinement were followed by restrained refinement using TLS parameters, resulting in an *R*-factor of 24.9%. Finally, restrained refinement was performed using anisotropic B-factors, further reducing the *R*-factor to 22.6%.

Thallium binding sites were determined after molecular replacement based on the presence of large, unoccupied peaks in both the $2F_o - F_c$ and anomalous maps. The thallium ions were included in refinement because their absence prevented the *R*-factor from dropping below ~45%. This is not surprising given the large number of thallium ions (10) per asymmetric unit. Coot was used for water assignment and viewing of all density maps [147].

4.3 Results

The crystal structure of the Tl^+-form of $d(G_4T_4G_4)_2$ was solved to 1.55 Å by molecular replacement using the crystal structure of the K^+-form (Figure 4-1) [111]. Like the K^+-form, the Tl^+-form was solved in the space group $P2_12_12_1$ with two G-quadruplexes in the asymmetric unit (Table 4-1 and Figure 4-2A). The unit cell dimensions of the Tl^+- and K^+-forms are also similar, differing by only 0.3–3.2%. The RMSD for the two molecules located within the asymmetric unit is 0.24 Å.

The topology of the Tl^+ crystal form is similar to the Tl^+ solution structure and both K^+ solution and crystal structures [107, 111]. The G-quadruplex is a dimer containing diagonal thymine loops at either end (Figure 4-2B). The guanine bases are in

an alternating *syn–anti* conformation and all thymines are *anti*. The thymine loops facilitate intermolecular packing both within the asymmetric unit and crystal lattice by forming a pair of pseudo two-fold related hydrogen bonds (Figure 4-2C). The average RMSD of the Tl^+ and K^+ crystal forms are 0.259 Å. The high degree of similarity between these two crystal structures is a likely explanation for the limited effect that soaks in high concentrations of thallium (≤ 50 mM) have on resolution.

The RMSD to the Tl^+ solution structure is 2.16 Å; however the G-quartets have an RMSD of only 1.25 Å. This indicates that loop region is the source of the greatest amount of variability among the x-ray and solution structures (Figure 4-3). In the crystal structure of the Tl^+-form, T8 is extended into solution, much like in the crystal structure of the K^+-form. This is in contrast to the Tl^+ (and K^+) solution structures where T8 stacks above the neighboring G-quartet plane. The effects of crystal packing could explain the differences observed in solution and crystal forms (Figure 4-2C).

The assignment of metal binding sites was facilitated by the presence of strong anomalous peaks (> 5.7 σ) (Figure 4-4). Each G-quadruplex contains five bound Tl^+ ions, three interdigitated between G-quartet planes and one in each of the two loops. The average spacing between each Tl^+ ion is 3.60 Å. The relative positions of these five metals are very similar to those found in the K^+ crystal structure, where the average metal–metal spacing is reported to be 3.38 Å [111]. The Tl^+ ions located between two successive G-quartet planes are coordinated by eight oxygens (one O6 from each of the surrounding guanines). These coordination distances range from 2.53–3.26 Å, which is similar to those observed in the K^+ structure (2.63–3.07 Å) [111]. The Tl^+ ions bound to the loops are coordinated by four guanine O6 carbonyls from the outer G-quartet plane,

two thymine carbonyls (T5 and T7 O2). The absence of any other regions of anomalous density, including from the region surrounding the phosphate backbone, indicates that only five ordered Tl^+ binding sites exist.

The average B-factors for the loop-associated metals (38.6 $Å^2$) are higher than the other thallium binding sites (28.3 $Å^2$). Additionally, the anomalous density in this region is less spherical (Figure 4-4). Accordingly, an attempt was made at refining the thallium ions located within the loops to partial occupancy. Reducing the thallium occupancy by as little as 10%, while resulting in lower B-factors, also produced large positive peaks in the $2F_o - F_c$ map. Accordingly, all thallium occupancies were left at 100%.

A total of 44 waters were assigned to the asymmetric unit. The assignments were made in regions of unexplained density (> 1.0 σ) in the $2F_o - F_c$ maps which did not have any anomalous density. The number of assigned waters is considerably less than the number reported for the crystal structure of the K^+-form (230) [111]. This is likely related to the use of differing methods for assignment of water peaks.

One of the assigned waters in the Tl^+ structure is located within the thymine loops, 3.87 Å from the loop-associated Tl^+ ion. A second region of density within the thymine loop region was also initially assigned as a water; however, this assignment could not be confirmed because the density is not resolved from the nearby Tl^+ density. Accordingly, this water assignment was deleted. The presence of one (and possibly two) waters within coordination distance of the loop-associated Tl^+ is consistent with the report that two waters participate in the coordination of the K^+ ions bound within the thymine loops in the crystal structure of the K^+-form of $d(G_4T_4G_4)_2$ [111].

4.4 Discussion

The crystal structure of the Tl^+-form of $d(G_4T_4G_4)_2$ demonstrates that the G-quadruplex is capable of binding five Tl^+ ions, three within the G-quartet planes and two in the thymine loops. The ability of such a large number of ions to replace K^+ without disrupting the crystal lattice further verifies the isomorphous nature of Tl^+ and K^+.

The observation of five bound Tl^+ ions, while in agreement with the K^+ crystal structure, is not in agreement with solution studies which indicate that only three NH_4^+ ions are coordinated by $d(G_4T_4G_4)_2$ [65]. One possible explanation which reconciles these data is that the thymine loops do not adopt a conformation in solution which can accommodate the binding of any monovalent cations.

Because the loops mediate crystal packing and are the only region which differ significantly between the solution and crystal structures of both K^+- and Tl^+-forms, it is possible to conclude that the crystal conformation does not exist in solution and that the metal bound by the loops is a crystallographic artifact. However, the association of this conformation with crystal packing does not rule out its existence in solution. Based on their relaxation rates in solution, the thymine residues appear to be undergoing conformational exchange (see Chapter 2). Further evidence of their dynamic nature is provided by the slightly elevated B-factors observed in the loops relative to the G-quartets (Table 4-2). Accordingly, it is reasonable to conclude that the loop conformation observed in the crystal structure is one of several which exist in solution. The role of conformational exchange will be further discussed in light of ^{205}Tl NMR data in Chapter 5.

4.5 Conclusions

The crystal structure of the Tl^+-form of $d(G_4T_4G_4)_2$ has been solved using molecular replacement. There are five Tl^+ binding sites per G-quadruplex, three within the G-quadruplex channel and two in the thymine loops. These sites are identical to those observed in the K^+-form of $d(G_4T_4G_4)_2$ [111]. No additional ordered Tl^+ binding sites exist on the G-quadruplex.

Components	
$MgAc_2$ (mM)	10.0
KAc (mM)	40.0
Spermine (mM)	3.5
DNA (mM)	1.5
MPD (%, v/v)	5.0
MPD (%, v/v) in well	35.0
Soaks and Cryostabilization	
TlAc (mM)	50.0
MPD (%, v/v)	60.0

Table 4-1. Crystallization conditions for the Tl^+-form of $d(G_4T_4G_4)_2$.

Crystallographic Data	
Space group	$P2_12_12_1$
Cell dimensions a, b, c (Å)	27.375, 48.210, 96.198
α, β, γ (°)	90.000, 90.000, 90.000
Wavelength (Å)	0.979
Resolution range (Å)	43.11–1.55 (1.61–1.55)
Maximum resolution (Å)	1.44
Completeness (%)	95.0* (98.1)
Mosacity	1.252
R_{merge} on I	0.187 (> 1)
Net$I/\sigma(I)$	6.43 (22.7)
No. unique reflections	9380
R-factor (%)	22.6
R_{free} (%)	24.8
F.O.M. (%)	80.6
RMS bond distance (Å)	0.010
RMS bond angles (°)	1.903
RMS chiral (°)	0.069
No. DNA strands/asymmetric unit	4
No. Tl^+ ions	10
Average B-factor(Å^2)	
G-quartets	27.68
Loops	31.12
Tl^+ ions	32.44

Table 4-2. Crystallographic data for the Tl^+-form of $d(G_4T_4G_4)_2$.
*The overall completeness (95.0%) is reduced considerably by the lowest resolution shell (50.00–3.34 Å) which has a completeness of 79.9%. Without this shell, the overall completeness is 96.7%.

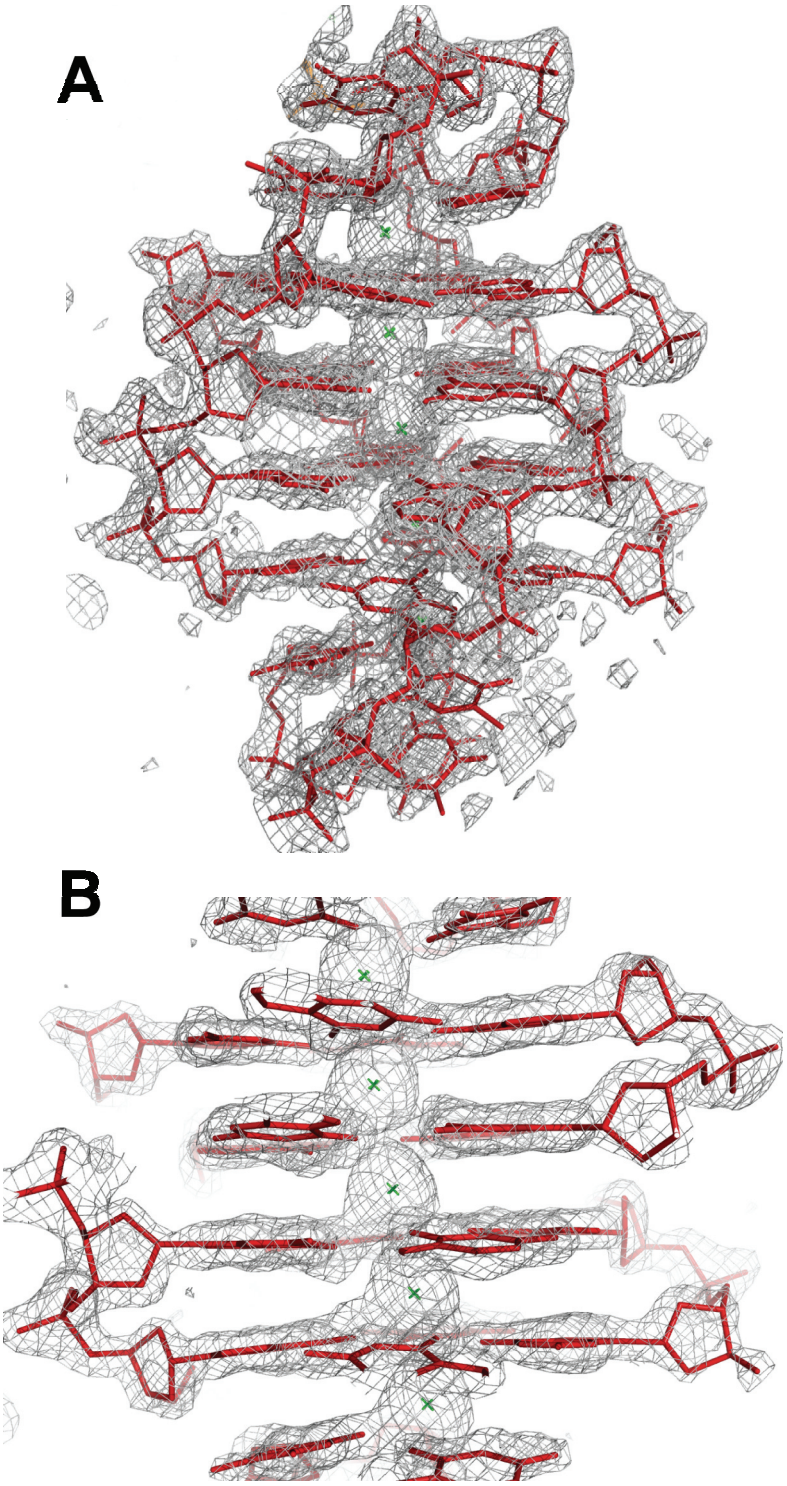


Figure 4-1. $2F_o - F_c$ map (1.0σ) of Tl^+-form of $d(G_4T_4G_4)_2$.
Nucleotide bases are shown in red. Tl^+ ions are denoted with green crosses.

A. Full view of a single G-quadruplex.

B. Expanded view of density surrounding five Tl^+ binding sites.

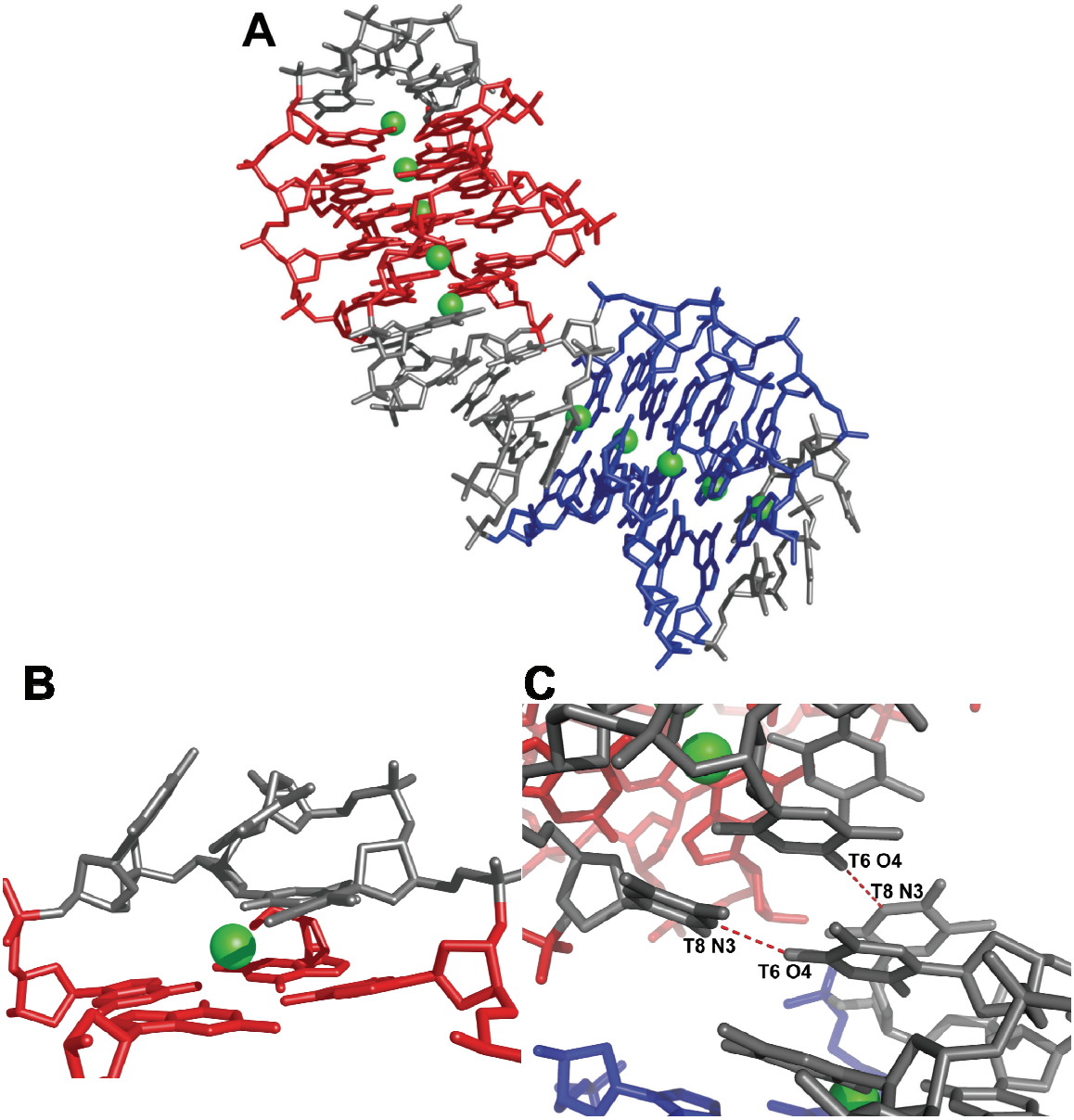


Figure 4-2. Characteristics of the crystal structure of the Tl^+-form of $d(G_4T_4G_4)_2$. Guanine bases are colored red or blue and thymines are colored gray. Tl^+ ions are colored green and are not drawn to scale.

A. The asymmetric unit contains two G-quadruplexes (red/gray and blue/gray), each with five bound Tl^+ ions (green).

B. The G-quadruplexes contain diagonally looped thymines.

C. Crystal packing is mediated by hydrogen bonding between two thymines (T6 and T8) on adjacent G-quadruplex loops.

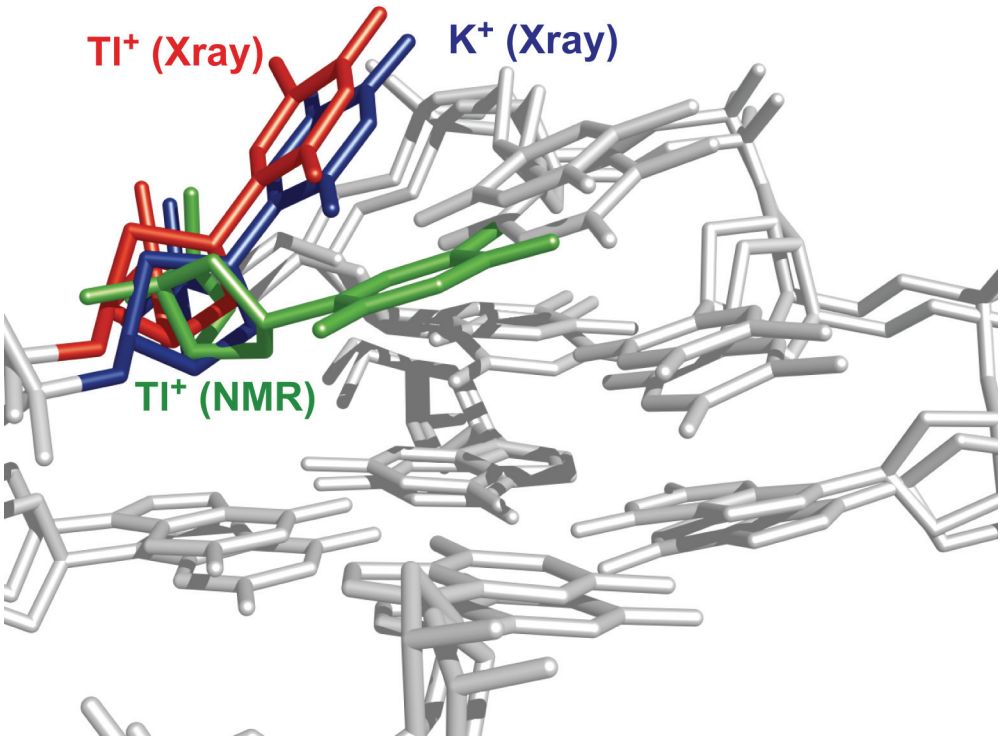


Figure 4-3. Structural differences associated with the thymine loops.
Overlay of the Tl^+ and K^+ crystal structures and Tl^+ solution structure of $d(G_4T_4G_4)_2$ showing the differences in T8 between the solution and crystal forms. T8 from the crystal structure of the Tl^+- and K^+-forms is colored red and blue, respectively. For simplicity, only T8 (green) is shown from the Tl^+ solution structure.

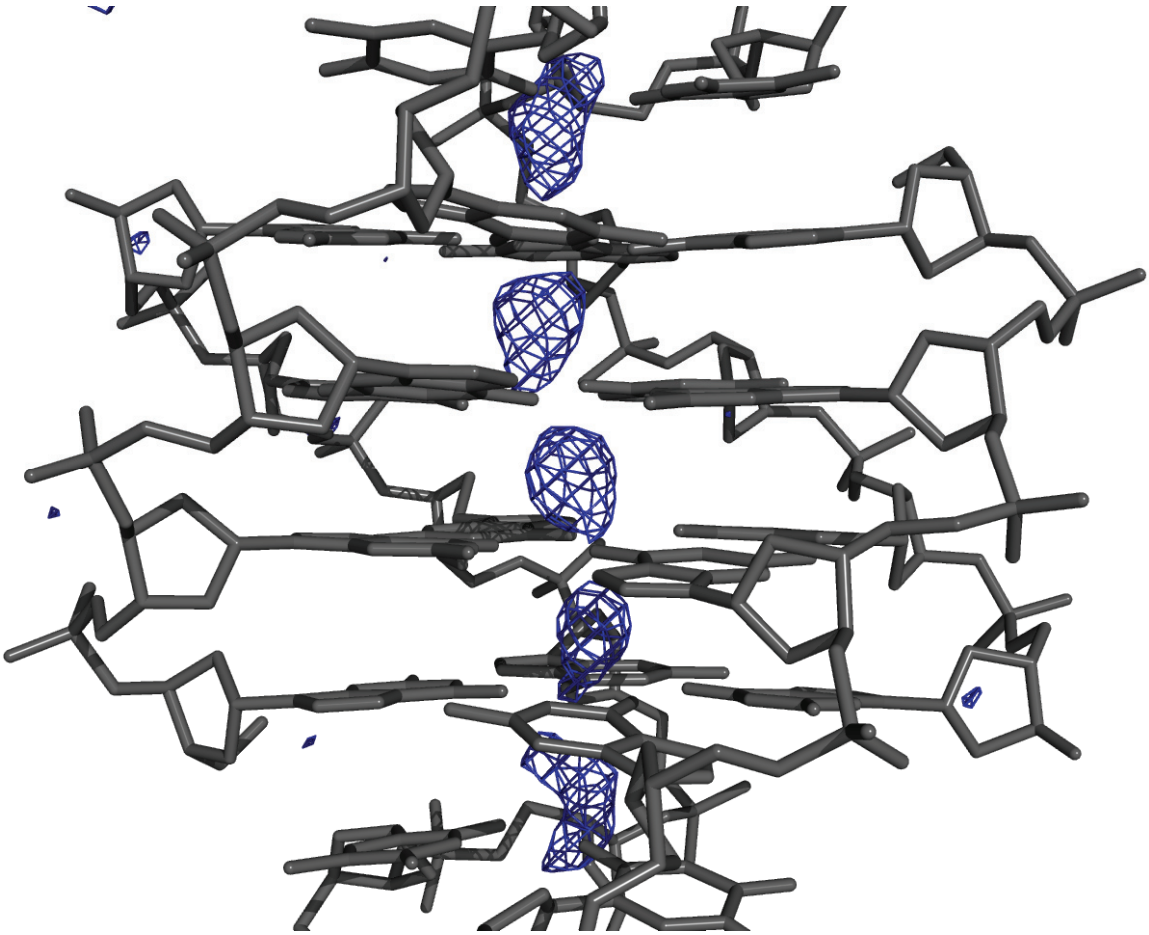


Figure 4-4. Anomalous density map (3.0σ) of the Tl^+ crystal structure of $d(G_4T_4G_4)_2$. The five strong peaks (blue) have been assigned to Tl^+ ions.

5 Assignment of ^{205}Tl Binding Sites Using ^{1}H–^{205}Tl Scalar Couplings

5.1 Introduction

Direct detection of ^{205}Tl has been used here and in other NMR studies to provide valuable information about monovalent ion stoichiometry, affinity, and exchange rate [67, 80-82]. However, direct detection techniques alone do not allow the site specific assignment of multiple resonances or the localization of newly discovered binding sites. Accomplishment of this important goal requires a heteronuclear NMR experiment, which will enable the correlation of individual ^{205}Tl resonances with nuclei located on the biomacromolecule of interest.

An obvious nucleus to utilize for detection of these ^{205}Tl correlations is ^{1}H because of its high gyromagnetic ratio (2.6752×10^{8} $T^{-1} \cdot s^{-1}$) and natural abundance (99.98%) [148]. Indeed, the existence of protons which are scalar coupled to the divalent metal surrogates ^{113}Cd and ^{199}Hg has been reported in several proteins [7-9, 12-17, 149]. Despite these examples of ^{1}H–M^{2+} scalar couplings, there have been no reports of ^{1}H–^{205}Tl *J*-correlations in biological systems. For this reason, we have developed a ^{1}H–^{205}Tl spin-echo difference experiment and used it to detect the presence of ^{1}H–^{205}Tl scalar couplings. By incorporating selective ^{205}Tl pulses, this experiment has been used to assign the $^{205}Tl^{+}$ cations coordinated within the G-quadruplex channel.

5.2 Materials and Methods

5.2.1 Materials and abbreviations.

Chemically synthesized DNA oligonucleotides, d(GGGGTTTTGGGG), were purchased from the W. M. Keck Facility, Yale University. Sep-Pak C_{18} desalting cartridges were from Waters, USA. Centricon 3 kDa NMWL centrifugal filter units were purchased from Millipore, USA. Acetonitrile, thallium nitrate ($TlNO_3$), potassium phosphate monobasic (KH_2PO_4), sodium phosphate monobasic (NaH_2PO_4), and phosphoric acid (H_3PO_4) were from Sigma-Aldrich, USA. Ethylenediaminetetraacetic acid (EDTA-d_{12}), deuterium oxide (D_2O), and acetic acid-d_4 (CD_3COOD) were purchased from Cambridge Isotope Laboratories, USA.

5.2.2 G-quadruplex formation.

DNA oligonucleotides d(GGGGTTTTGGGG) were desalted using Sep-Pak C_{18} cartridges and eluted with 40% (v/v) acetonitrile/water. After lyophilization, the DNA was dissolved to a final concentration of ~500 μM in 50 mM NaH_2PO_4. G-quadruplex formation was facilitated by heating the solution to 363 K for 15 minutes followed by slow cooling to 277 K. The solution was concentrated to about 2.5 mM G-quadruplex (~5 mM DNA strand) by centrifugation and the buffer exchanged to 50 mM $TlNO_3$, 100 μM EDTA-d_{12} and 10% D_2O. In some cases, G-quadruplex formation was also performed directly in the 50 mM $TlNO_3$ solution. The use of Na^+-containing solutions in the initial annealing step was done to reduce the amount of Tl^+ used. Similar ^{1}H and ^{205}Tl spectra were produced for G-quadruplexes formed in both manners. Two K^+ versions were also prepared—one with 50 mM KH_2PO_4 substituted for $TlNO_3$ and 10% D_2O and a second with 50 mM KH_2PO_4 and 100% D_2O. The sample pH was adjusted to 6.5 with H_2PO_4 or CD_3COOD.

5.2.3 NMR spectroscopy.

The ^{1}H–^{205}Tl spin-echo difference experiments were performed at 11.75 T (288 MHz ^{205}Tl) on a Varian Inova wide bore spectrometer and used a Nalorac indirect detection broadband probe with a homebuilt inductor stick for tuning the broadband channel to the ^{205}Tl frequency (288 MHz). A second high frequency amplifier was employed to achieve the ^{205}Tl frequency and allow simultaneous RF pulsing on ^{1}H and ^{205}Tl nuclei. Both ^{1}H and ^{205}Tl pulses were calibrated using standard direct detection methods. The delay between pulses in the 3–9–19 element was set to 200 μs to maximize excitation profile of the imino protons. Further details about experimental setup are provided in Appendix 2.

The spin-echo difference experiments were acquired in an interleaved manner using the phase cycle described in Figure 5-1. The application of the CYCLOPS phase cycling scheme [150] to the pulse sequence required the acquisition block to be set to 16 transients before switching to the alternate spectrum. Typically, 30,000–50,000 acquisitions were performed although signal can be observed above the background noise in as few as 10,000 acquisitions. The delay for *J*-coupling evolution was set to 15, 40, or 100 ms.

The magnetic susceptibility of $d(G_4T_4G_4)_2$ was measured using ^{1}H–^{13}C natural abundance HSQC experiments (adapted from [151]) performed at 14.1 T and 18.8 T on Varian Inova and Unity Plus spectrometers. The HSQC used the PEP technique [152] for sensitivity enhancement and States-TPPI [124, 125] for frequency discrimination in the indirect dimension. For each spectrum 2606 t_2 X 128 t_1 points were collected with a

sweepwidth of 8000 and 6000 Hz in the ^{1}H and ^{13}C dimensions. The ^{13}C carrier was placed at 130 ppm and 696 transients were acquired.

All NMR data were processed in NMRPipe [134]. NMR lineshapes and peak intensities were analyzed using NMRPipe and in-house written MATLAB code. HSQC data analysis was performed in Sparky [153].

Calculation of the G-quadruplex χ tensors was kindly performed by Christina Ragain and Dr. Jose Gascón in the laboratory of Professor Victor Batista at Yale University. First, hydrogen atoms were added to the PDB 1JRN prior to energy minimization in the presence of explicit solvent (water and K^{+} ions) using the AMBER99 force field in the program TINKER. QM/MM calculations were performed in *Gaussian03* [154] using DFT with B3LYP functional and 6-31g* basis sets. Calculations were performed on individual guanines, with each calculation yielding similar results. These results were then combined using the method of Bax and coworkers [155] to determine $\Delta\chi$ for the entire G-quadruplex. Example *Gaussian03* input scripts and the relevant results are provided in Appendix 3.

5.3 Results

To further investigate the nature of the downfield ^{205}Tl peaks, we implemented a heteronuclear ^{1}H–^{205}Tl spin-echo difference experiment (Figure 5-1) to look for potential H–Tl interactions [7, 8]. Because a number of the protons in close proximity to the $^{205}Tl^{+}$ ions exchange with solvent (Figure 5-2), a water-selective sinc pulse and gradient-tailored 3–9–19 pulse train [156] were incorporated into the pulse sequence. The experiment was performed in a two-part, interleaved fashion with either complete

recoupling of heteronuclear *J*-modulation when ϕ_4= –x, resulting in a ^{1}H signal intensity corresponding to

$$S_0 = e^{(-2\tau/T_2)} \tag{5-1}$$

or with the occurrence of heteronuclear *J*-modulation when ϕ_4= x, resulting in the ^{1}H signal intensity

$$S_1 = e^{(-2\tau/T_2)}\cos(2\pi J_{H-Tl}\tau) \tag{5-2}$$

where T_2 is the proton transverse relaxation time and τ is the scalar coupling refocusing delay. The scalar coupling constant, $J_{H–Tl}$, is then determined from

$$\frac{S_0\text{-}S_1}{S_0} = 1 - \cos(2\pi J_{\text{H-Tl}}\tau) = 2\sin^2(\pi J_{\text{H-Tl}}\tau) \tag{5-3}$$

To obtain the difference spectrum ($S_0 – S_1$), the two reference spectra (S_0 and S_1) (Figure 5-3) are subtracted from each other. The presence of any peaks in the difference spectrum (Figure 5-3) corresponds to ^{1}H resonances which are scalar coupled to the ^{205}Tl resonance selectively excited during experiment acquisition.

This spin-echo difference experiment was performed with the refocusing delay set to 40 ms and bandwidth selective square ^{205}Tl pulses (5.7 kHz, 3 kHz, 2.5 kHz, and 3 kHz) for each of the downfield ^{205}Tl peaks, 1–4 respectively (Figure 5-4A). Variations in the identity and magnitude of the scalar couplings are observed between $^{205}Tl^{+}$ peaks 2 and 3 and aromatic and imino protons of the G-quadruplex (Figure 5-4B–C). For peak 2, scalar couplings are observed to all G aromatic protons except G4 and G12 and to the imino protons for G1/G9 (assignment ambiguous), G2, and G4 (Figure 5-4B–C and Table 5-1). ^{1}H–^{205}Tl scalar couplings for peak 3 are observed to the G2, G3, G10, and G11 aromatic H8 protons and to the G2 and G10 imino protons (Figure 5-4B–C and Table

5-1). These couplings are small and range from 0.34–0.95 Hz (Table 5-1). No couplings between ^{1}H and ^{205}Tl were observed for ^{205}Tl peaks 1 and 4 (Figure 5-4B–C).

The spin-echo difference experiment was performed with ^{205}Tl selective pulses on peak 1 with two additional refocusing delays, 15 and 100 ms. The estimation of $J_{\text{H–Tl}}$ from these individual experiments is consistent with a cosine modulated coupling interaction (equation (1-3)). No signal was present in the $S_0 - S_1$ spectrum when K^+ is used in place of Tl^+, consistent with these couplings arising from a ^{1}H–^{205}Tl mediated effect (Figure 5-3). Further, no signal was observed when the large upfield ^{205}Tl peak is excited during experiment acquisition.

The potential for residual dipolar couplings (RDC) contributing to the observed *J*-coupling, due to the natural magnetic field alignment of the G-quadruplex, was addressed in two ways. First, we measured the $^{1}D_{\text{C–H}}$ value for the aromatic proton of guanine [157] (Figure 5-5) and estimated the $D_{\text{H–Tl}}$ value using the appropriate scaling factors. For the aromatic proton, H8, which is 7.7 Å from ^{205}Tl, both estimations indicate the contribution of the RDC to the *J*-coupling value ranges from negligible to ~1–2 % (Table 5-2). For the H1 imino proton (3.5 Å from ^{205}Tl), the RDC contribution could range from ~ 1% to ~18% (Table 5-2). For simplicity, the sign of $D_{\text{H–Tl}}$ was assumed to be positive in the above calculation; however the occurrence of negative $D_{\text{H–Tl}}$ values further supports the conclusion that its contribution to the observed scalar couplings is negligible because none of the peaks observed in the difference ($S_0 - S_1$) spectrum were negative. Second, the magnetic susceptibility, $\Delta\chi$, of the quadruplex was estimated using the method of Bothner-By [158] and that of Bax and coworkers [155] or calculated using quantum mechanical methods. Each method gave similar values for $\Delta\chi$. Therefore, for the

aromatic (H8) proton the *J*-values are essentially free from any residual dipolar contribution, whereas the RDC contribution to the observed imino (H1) *J*-coupling is small.

5.4 Discussion

The observation that small ^{1}H–^{205}Tl couplings exist presents possibilities to localize monovalent cation binding sites in macromolecules and to investigate how specific monovalent sites respond to solution perturbations, mutation, or ligand binding. In the experiments described here, ^{205}Tl selective pulses were incorporated into the ^{1}H–^{205}Tl spin-echo difference experiment so that the ^{205}Tl peaks could be assigned to monovalent binding sites within the G-quadruplex. By mapping the results of these experiments onto the structure of $d(G_4T_4G_4)_2$, the identities of two of the four ^{205}Tl peaks were determined. For peak 3, ^{1}H–^{205}Tl scalar couplings are only observed to the aromatic and imino protons of the inner two G-quartet planes (Figure 5-4B–C), so this peak was assigned to the innermost G-quadruplex binding site (Figure 5-6A). For the ^{205}Tl resonance corresponding to peak 2, couplings are observed to guanine aromatic and imino protons throughout all four G-quartet planes (Figure 5-4B–C). Assuming this peak represents a single ^{205}Tl binding site, the only assignment consistent with the observed couplings are the two symmetrically related outer binding sites (Figure 5-6B).

The assignment of peaks 2 and 3 to the respective outer and inner G-quadruplex cation sites is also consistent with the relative areas (2:1) of these peaks in the direct detection ^{205}Tl spectrum (Figure 3-1). Peaks 1 and 4 do not have any observed couplings to protons in this G-quadruplex and thus cannot be assigned in this fashion (Figure 5-4B–C). The couplings observed to ^{205}Tl peaks 2 and 3 are the first ^{1}H–^{205}Tl scalar couplings reported in a biological system.

The detection of small, unresolvable scalar couplings, such as those reported here for proton-metal interactions, requires the sensitivity of a difference experiment. The ^{1}H–^{205}Tl scalar couplings between guanine aromatic/imino protons and bound $^{205}Tl^{+}$ cations are all less than 1 Hz (Table 5-1), with the coupling to the G4 imino proton being the largest, 0.95 ± 0.06 Hz. The magnitude of these couplings is similar to those detected between either ^{113}Cd or ^{199}Hg and protons in rubredoxin [7].

Some of the expected couplings to imino protons from G3 and G11 and aromatic protons from G4 and G12 are absent. These 'missing' imino couplings to either ^{205}Tl peak (Figure 5-4C) could be caused by a combination of reduced signal-to-noise of these proton resonances and very small scalar coupling values. The absence of aromatic scalar couplings involving G4 and G12 to ^{205}Tl resonance 2 (Figure 5-4B) is likely due to increased conformational exchange at these sites (see Chapter 2 results and discussion).

These results indicate that $^{205}Tl^{+}$ is scalar coupled to both imino (H1) and aromatic (H8) protons, but they do not report on the pathway by which these two interactions occur. Besides covalent bonds, scalar couplings are known to follow hydrogen-bond, coordination, and though-space mediated pathways [7, 8]. A discussion of the possible pathways giving rise to these ^{1}H–^{205}Tl scalar couplings requires consideration of the availability of functional groups which could interact with Tl^{+}, the intricate hydrogen-bonding network of the G-quartets, and the most direct pathway(s) available for these interactions.

We have considered three explanations for the observed guanine imino (H1) couplings. The first involves a direct interaction between the coordinated $^{205}Tl^{+}$ cations and the imino protons (Figure 5-7A, blue). Based on the x-ray structure of the Tl^{+}-form

of d($G_4T_4G_4$)$_2$, the distance from the center of the metal cation to the H1 proton is about 3.5 Å. The couplings could also be mediated through the O6 carbonyl group, which coordinates the monovalent cation. This coordination would provide a mechanism for a 4-bond coupling (O6–C6, C6–N1, N1–H1) to the imino proton on the same guanine nucleotide (Figure 5-7A, red). An alternate pathway beginning with O6 and traversing the O6–H1 hydrogen bond would result in couplings to an imino proton on a nearby G nucleotide (Figure 5-7A, green). Of the last two possibilities, the former involves three covalent bonds; the later pathway is shorter, but includes a hydrogen bond. There are precedents for scalar couplings similar to both instances [159, 160].

The observation of ^{205}Tl scalar couplings to guanine H8 is unexpected given the distance and/or number of bonds that must be traversed. A direct interaction between ^{205}Tl and the H8 proton seems unlikely because the distance from the monovalent ion-binding site to this proton is 7.7 Å (Figure 5-7B, blue); the long distance over which couplings are observed also suggests these effects are due to electron orbital couplings and not a dipolar interaction. If the scalar coupling were mediated by the O6 carbonyl ligand, it is six covalent bonds (O6–C6, C6–C5, C5–N7, N7–C8, C8–H8) to the aromatic proton (Figure 5-7B, red). A total distance of six bonds (five covalent + one coordination) is also quite long, although scalar couplings have been observed through four [159, 161] and five bonds [7]. Another possibility considered here involves interaction of the $^{205}Tl^{+}$ with the N7 atom and continuation through two bonds (N7–C8, C8–H8) to the aromatic proton (Figure 5-7B, green). The distance from each of the three $^{205}Tl^{+}$ cations to the N7 atoms of scalar coupled G bases is about 5.2 Å. This pathway is plausible as the interaction of thallium with guanine N7 has been shown to be favorable [24, 162].

Finally, a combination of multiple pathways leading to an averaged observed coupling cannot be ruled out.

Two ^{205}Tl peaks (1 and 4) remain unassigned due to the absence of any observed ^{1}H–^{205}Tl scalar couplings. Given that these two peaks are sensitive to high concentrations of K^+ but not Cs^+ or TMA^+ (Figure 3-5 and Figure 3-4) and that anomalous density was observed within the loops and not along the G-quadruplex grooves (Figure 4-4), the most likely assignment for these peaks seems to be within the thymine loops. The presence of two ^{205}Tl peaks rather than one can likely be explained by the presence of conformational exchange (see Chapter 2 discussion) which is fast on the ^{1}H timescale and slow on the ^{205}Tl timescale. The slightly increased temperature sensitivity for these two peaks (Figure 3-2) seems consistent with their assignment to a region which lies somewhat peripheral to the G-quadruplex channel. The approximate areas of peaks 1 and 4 (relative to peaks 2 and 3) (Figure 3-1) indicate a high level of $^{205}Tl^+$ occupancy which is consistent with the anomalous density observed in the crystallographic map (Figure 4-4). The positioning of peaks 1 and 4 at opposite ends of the downfield region is somewhat curious given that they are both believed to result from loop binding. However, the large ^{205}Tl chemical shift range (~7000 ppm) [78] implies that this separation may not be so considerable after all.

The assignment of peaks 1 and 4 to the thymine loops may not seem consistent with the absence of scalar couplings to thymine protons, but this does not disprove loop binding. The elevated transverse relaxation rates associated with the thymine aromatic (H6) protons reduces the signal-to-noise, possibly masking any apparent ^{205}Tl–H6 proton

coupling. In addition, there may simply not be a significant amount of orbital interaction between ^{205}Tl and the thymine protons to observe these small couplings.

5.5 Conclusion

A ^{1}H–^{205}Tl spin-echo difference experiment has been developed and used to detect small ^{1}H–^{205}Tl scalar couplings. This NMR experiment is the first ^{205}Tl heteronuclear experiment reported. When the spin-echo difference experiment is performed using ^{205}Tl selective pulses, three Tl^{+} binding sites were determined by mapping the observed ^{1}H–^{205}Tl scalar couplings onto the G-quadruplex. The magnitude of the metal–proton scalar couplings is consistent with reports of ^{1}H–^{113}Cd couplings in divalent metal binding proteins. These results comprise the first report of ^{1}H–^{205}Tl scalar couplings in a biomacromolecule.

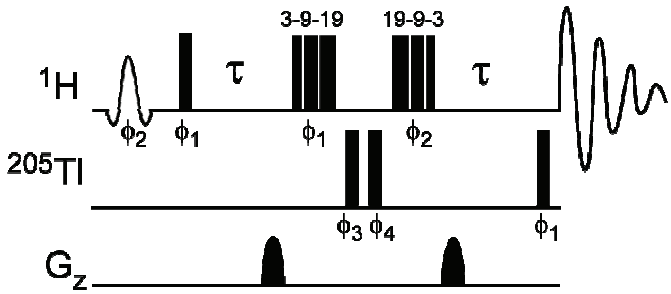


Figure 5-1. ^{1}H–^{205}Tl spin-echo difference pulse sequence.
The sequence is performed in two, interleaved parts. In the first half (S_0), φ_1 = {x,y,–x,–y}, φ_2 = {–x,–y,x,y}, φ_3 = {x}, φ_4 = {–x}, and φ_{rec} = {x,y,–x,–y}. For the S_1 experiment, φ_4= {x}. All pulses were additionally phase-cycled using the CYCLOPS scheme [150].

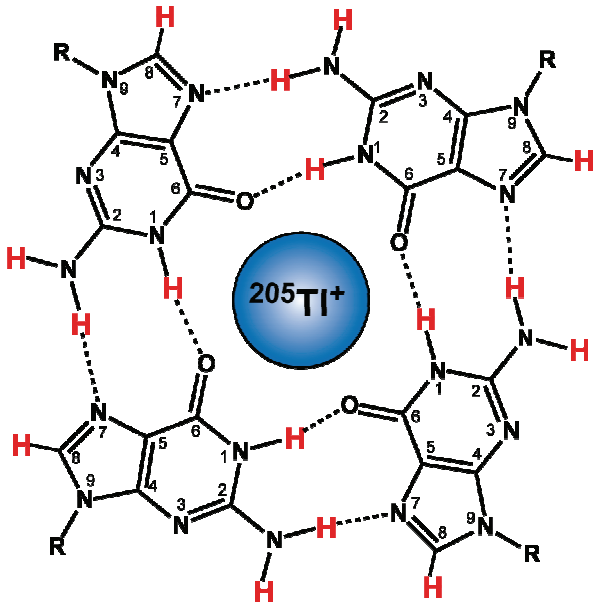


Figure 5-2. Potential protons available on G-quartets for detection of ^{1}H–^{205}Tl scalar couplings.

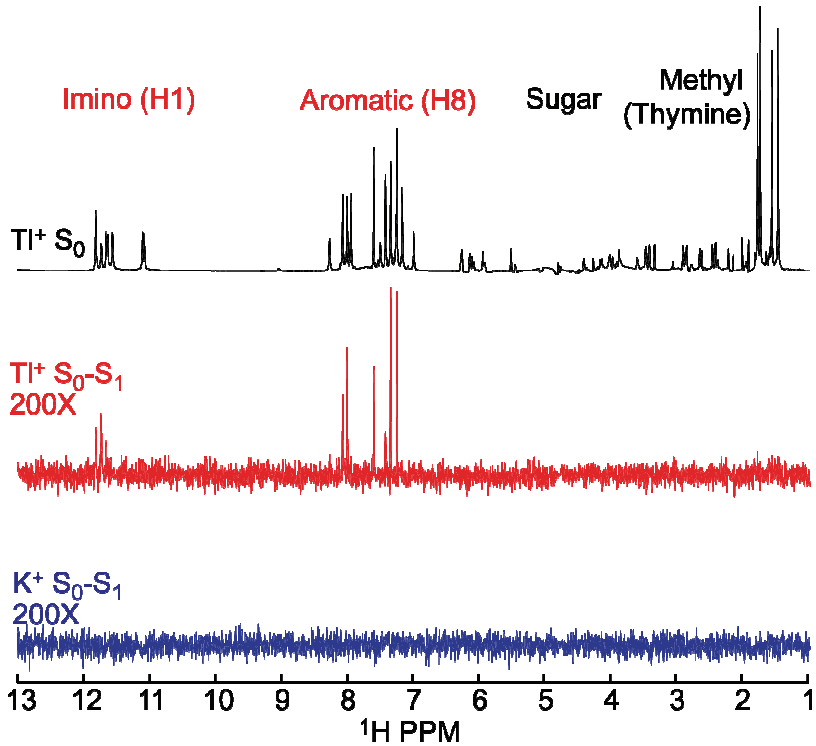


Figure 5-3. ^{1}H–^{205}Tl scalar couplings are observed to aromatic (H8) and imino (H1) protons.
A reference spectrum, S_0, for the Tl^+-form of $d(G_4T_4G_4)_2$ is shown in black. An example of a difference, $S_0 - S_1$, spectrum for the Tl^+-form of $d(G_4T_4G_4)_2$ is shown in red. A negative control, $S_0 - S_1$ spectrum of the K^+-form of $d(G_4T_4G_4)_2$, is shown in blue. All difference spectra, $S_0 - S_1$, are vertically expanded 200X.

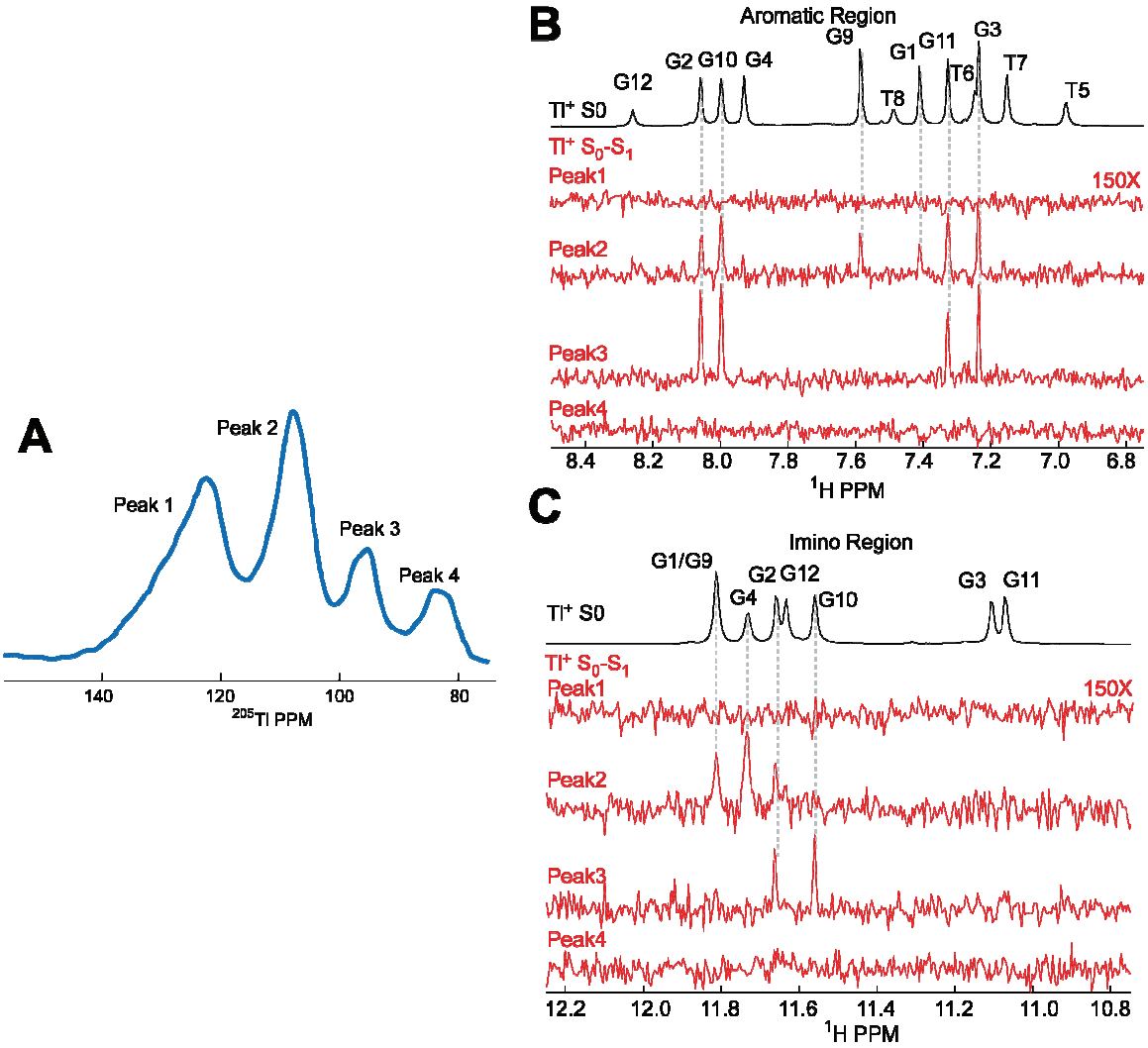


Figure 5-4. ^{1}H–^{205}Tl scalar couplings observed when selective ^{205}Tl pulses are used.

A. Selective ^{205}Tl pulses were used for each of the downfield ^{205}Tl peaks (1–4).

B. In red, an expanded region showing the ^{205}Tl scalar couplings to the aromatic protons for each of the downfield ^{205}Tl peaks. A reference spectrum, S_0, is shown in black.

C. In red, an expanded region showing the ^{205}Tl scalar couplings to the imino protons for each of the downfield ^{205}Tl peaks. A reference spectrum, S_0, is shown in black.

	Guanine Residue	Peak 2	Peak 3
Imino	G1/9	0.46 ± 0.04	–
	G2	0.54 ± 0.04	0.51 ± 0.06
	G4	0.95 ± 0.06	–
	G10	–	0.44 ± 0.03
Aromatic	G1	0.34 ± 0.06	–
	G2	0.44 ± 0.05	0.52 ± 0.03
	G3	0.49 ± 0.02	0.65 ± 0.01
	G9	0.34 ± 0.04	–
	G10	0.49 ± 0.04	0.56 ± 0.02
	G11	0.47 ± 0.03	0.40 ± 0.02

Table 5-1. Magnitude of ^{1}H–^{205}Tl scalar couplings (Hz) to individual ^{205}Tl peaks.

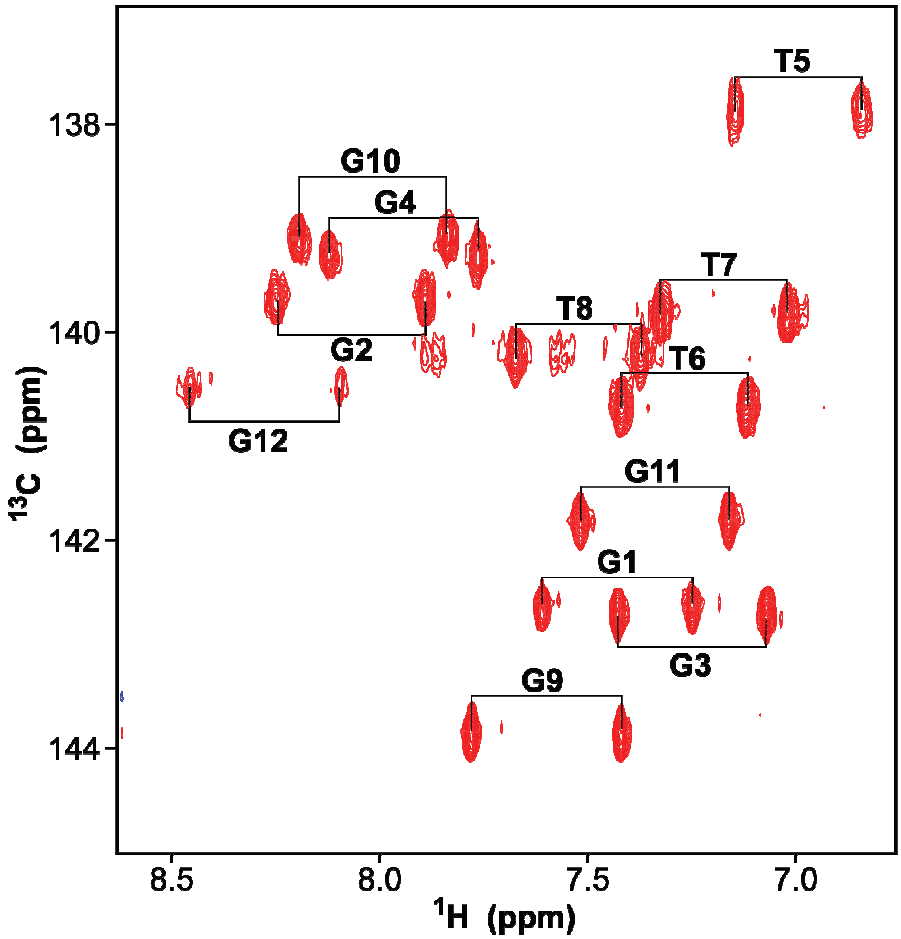


Figure 5-5. Measurement of $^{1}J_{H–C} + {}^{1}D_{H–C}$ using a natural abundance ^{13}C HSQC. The experiment was performed at 14.1 T (shown here) and 18.8 T (not shown).

	Guanine Residue	$D_{H–Tl}$ (11.75T)	Peak 2 (%)	Peak 3 (%)
Imino	G1/9	-0.103	22.4%	-
	G2	0.041	7.5%	7.9%
	G4	-0.021	2.2%	-
	G10	0.077	-	17.5%
Aromatic	G1	-0.003	0.8%	-
	G2	0.004	0.9%	0.7%
	G3	-0.007	1.5%	1.1%
	G9	-0.007	2.0%	-
	G10	0.007	1.5%	1.3%
	G11	-0.009	1.8%	2.1%

Table 5-2. Calculation of $D_{H–Tl}$ at 11.75 T.
The values were determined from experimental measurements of $^{1}D_{C–H}$ at 14.1 T and 18.8 T. The predicted contribution of dipolar couplings to $J_{H–Tl}$ measured for ^{205}Tl peaks 2 and 3 is shown in the right two columns.

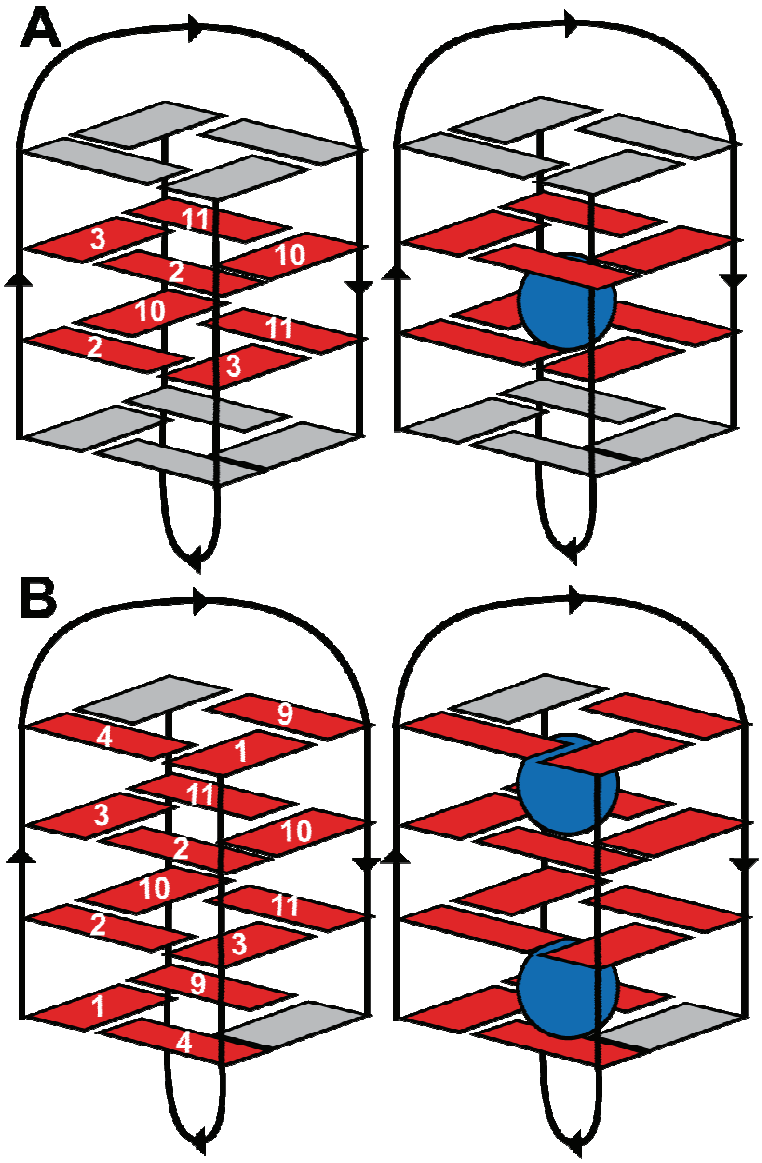


Figure 5-6. Location of ^{1}H–^{205}Tl scalar couplings for ^{205}Tl peaks 2 and 3.

A. On the left, the couplings from ^{205}Tl peak 3 are observed to imino and aromatic protons on bases (red) located within the middle two G-quartet planes. The assignment of peak 3 to a specific monovalent cation (blue) is shown on the right.

B. The *J*-couplings between ^{205}Tl peak 2 and imino or aromatic protons are located on bases found in all four G-quartet planes. The proposed assignment of peak 2 (blue spheres) is shown on the right.

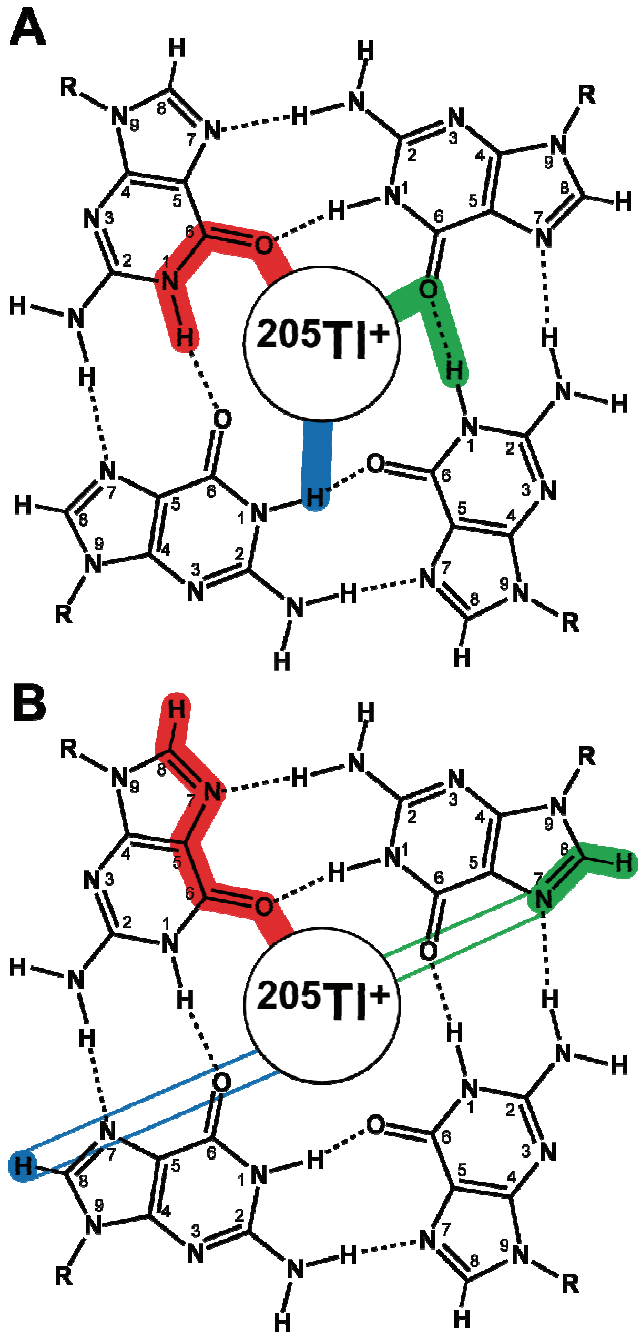


Figure 5-7. Possible ^{1}H–^{205}Tl scalar coupling mechanisms.

A. Possible mechanisms for imino protons involving direct (blue) and through bond (red and green) pathways.

B. Alternative mechanisms for scalar couplings between ^{205}Tl and aromatic protons are shown in blue, red, and green. Unfilled regions indicate atoms, which are not included in the pathway.

6 Implementation of ^{205}Tl NMR in RNA systems

6.1 Introduction

Having demonstrated that both ^{205}Tl direct detection and heteronuclear NMR methods can be used to study binding sites in a model system, $d(G_4T_4G_4)_2$, the next step in the development of ^{205}Tl NMR was to implement these methods in an RNA system of interest. Selection of such a system requires consideration of the limitations of both solution NMR in general and of those specific to ^{205}Tl NMR techniques. For instance, systems which are extremely stable and have shorter correlation times are generally more amenable to study by solution NMR. For the implementation of ^{205}Tl NMR techniques, the monovalent cation(s) of interest must also have a relatively slow exchange rate with the bulk solution.

We identified two RNA systems known to be well suited to solution NMR studies and that are either predicted or have been shown experimentally to bind monovalent cations. The first system is an engineered GAAA tetraloop-tetraloop receptor complex whose design and solution structure was recently reported by Butcher and coworkers [48]. The 30 kDa RNA complex is a homodimer of two helices each containing a GAAA tetraloop (nucleotides 20-23) and tetraloop receptor (nucleotides 5-9 and 34-39) separated by one helical turn (Figure 6-1A). Complex formation is divalent (Mg^{2+}) dependent [48] and is mediated by the docking of a tetraloop from one monomer into the tetraloop receptor of a second monomer (Figure 6-1A–B). A requirement for monovalent cations was not reported; however, based on its similarity to other RNA crystal structures

containing this tertiary motif [22, 23, 35], the binding of a monovalent cation in the tetraloop receptor is predicted.

The second RNA system is the 58 nucleotide fragment from *Escherichia coli* 23S ribosomal RNA which is bound by ribosomal protein L11 (L11 rRNA). This highly conserved region of ribosomal RNA is the site of translocation-associated GTP hydrolysis and the target of thiazole-containing antibiotics such as thiostrepton [163-169]. The L11 rRNA has also been shown to fold independently and bind monovalent cations [24, 41, 170, 171]. We have attempted the direct study of monovalent binding sites in both of these RNA systems using the previously described ^{205}Tl NMR methods.

6.2 Materials and Methods

6.2.1 Materials and abbreviations.

Nucleoside triphosphates (ATP, CTP, GTP, and UTP) were purchased as disodium salts from Sigma-Aldrich, Inc. Magnesium chloride ($MgCl_2$) and magnesium sulfate ($MgSO_4$) were also from Sigma-Aldrich, Inc. *EarI* restriction enzyme and inositol pyrophosphitase (IPPase) were from New England Biolabs. Dithiothreitol (DTT) and Triton X-100 were from American Bioanalytical Labs. T7 RNA polymerase was purified in-house [172]. The Amicon filtration unit was purchased from Millipore, Inc. DNA primers were chemically synthesized by the W. M. Keck Facility, Yale University.

6.2.2 Preparation of GAAA tetraloop-tetraloop receptor sample.

Studies of the GAAA tetraloop-tetraloop receptor complex were the result of a collaboration with the Butcher laboratory at the University of Wisconsin-Madison. The RNA sample was kindly prepared in this laboratory by Jared Davis in a manner similar to that used for structure determination [48]. The following protocol is summarized from this report [48].

RNA was prepared by *in vitro* transcription using chemically synthesized DNA template and T7 RNA polymerase. Transcription conditions included 40 mM Tris-HCl (pH 8.1), 1 mM spermidine, 10 mM dithiothreitol, 0.01% Triton X-100, 80 mg/ml of polyethylene glycol, 16 mM $MgCl_2$, 4 mM NTPs, 0.3 μM DNA, 1 U/ml IPPase, and 3000 U/ml of RNA polymerase. After incubation at 37°C for four hours, the reaction was quenched with 0.5 M EDTA and extracted with phenol/chloroform. The organic layer was further extracted with water. The resulting aqueous layers were then extracted with chloroform/isoamyl alcohol and ethanol precipitated. The precipitated RNA was purified by denaturing gel electrophoresis, eluted, and then purified over DEAE anion-exchange and G25 gel-filtration columns. The eluant was lyophilized and resuspended in water. The final NMR sample was 0.6 mM RNA (pH 6.8) with 12 mM $MgSO_4$, 50 mM $TlNO_3$, and 10% D_2O.

6.2.3 Preparation of L11 rRNA sample.

The 58 nucleotide L11 DNA sequence [24] was amplified by PCR from overlapping DNA primers and inserted immediately upstream of the anti-genomic form of the hepatitis delta (HDV) ribozyme. The L11-HDV sequence was then inserted into puc19 DNA plasmid. The purified plasmid was digested with *EarI* and used as the template for *in vitro* transcription. Transcription conditions included: 40 mM Tris-HCl (pH 7.5), 10 mM DTT, 4 mM spermidine, 20 mM $MgCl_2$, 0.05% (v/v) Triton X-100, 4 mM NTPs, 40 μg/ml DNA template, 1 U/ml IPPase, and T7 RNA polymerase and were incubated at 37°C for 4 hours. HDV ribozyme cleavage was facilitated by heating the samples to 75°C for 15 minutes before purification using denaturing polyacrylamide gel electrophoresis. The RNA was identified by UV shadowing, excised from the gel, and

eluted in 300 mM sodium acetate at 4°C overnight. The resulting solution was filtered and concentrated using an Amicon filtration unit with a 3 kDa NMWL membrane. The final RNA concentration in the sample was ~1.5 mM with 50 mM Na-cacodylate (pH 6.5), 50 mM $TlNO_3$, 10 mM $MgSO_4$, and 10% D_2O.

6.2.4 NMR spectroscopy.

^{205}Tl NMR experiments were performed at 11.75 T (288 MHz ^{205}Tl) using a Varian Inova wide bore spectrometer. Direct detection ^{205}Tl NMR experiments were performed with a Nalorac direct detection dual broadband probe using a pulse-acquire experiment with ~350,000 acquisitions, a 60 kHz spectral width, and a 400 ms recycle delay. ^{1}H NMR experiments were performed at 11.75 T (500 MHz ^{1}H) using either a Nalorac indirect detection broadband probe or a Varian triple resonance probe. The ^{1}H–^{1}H NOESY [118-120] (with mixing time, τ_m = 150 ms) experiment used WATERGATE [118] water suppression and States–TPPI [124, 125] frequency discrimination. The spectra were collected using 2438 t_2 x 360 t_1 points, and a spectral width of 10703 Hz in both dimensions. For each t_1 increment, 128 scans were acquired. The temperature for the NOESY was regulated at 283 K.

6.3 Results

The folding of the GAAA tetraloop-tetraloop receptor sample was verified by the presence of imino (H1) protons in a ^{1}H–^{1}H NOESY collected both before and after the addition of 50 mM $TlNO_3$ (data not shown and Figure 6-3A). The addition of Tl^+ had no significant effect on the chemical shifts of any well-resolved crosspeaks, indicating that no large scale conformational changes occur upon addition of Tl^+. This is not surprising given that the complex is able to form without the addition of any specific monovalent

cation [48]. Detection of minor chemical shift changes in other regions of the spectrum is likely precluded by spectral crowding.

The binding of Tl^+ to the tetraloop-tetraloop receptor was examined by direct detection ^{205}Tl NMR (Figure 6-3B). After 350,000 acquisitions, a single broad peak is observed at 26 ppm with a linewidth of ~1 kHz. Further, no resonances were observed when the experimental offset was placed ~90 ppm upfield of the broad peak. This indicates that any $^{205}Tl^+$ ions bound to the RNA are in fast exchange with the bulk solution. The exchange regime was not altered by lowering the acquisition temperature to 283 K. The relatively fast exchange rate (compared to $d(G_4T_4G_4)_2$) is most likely due to differing natures of the monovalent binding sites. In $d(G_4T_4G_4)_2$, the monovalent cations are trapped between successive G-quartet planes, whereas the predicted monovalent site in the tetraloop receptor is far more exposed. The aforementioned exchange rate likely precludes the detection of ^{1}H–^{205}Tl scalar couplings (data not shown).

The L11 rRNA contains a monovalent binding site that is less accessible to solution than the site in the tetraloop-tetraloop receptor. I was hopeful this would increase the bound lifetime of Tl^+, moving it into slow exchange on the ^{205}Tl chemical shift timescale. Accordingly, a sample was prepared following published protocols for L11 rRNA folding [24, 41]. However, preliminary ^{1}H NMR studies indicate that the RNA is not folded (Figure 6-4A). A ^{205}Tl NMR spectrum was also acquired (Figure 6-4B) and contains what appears to be two overlapped peaks at 35 ppm. The interpretation of this spectrum is not clear given that the RNA does not appear to be folded.

6.4 Discussion and Conclusions

We have attempted to study the monovalent binding site of the L11 binding portion of 23S ribosomal RNA and to detect a predicted monovalent binding site in the

GAAA tetraloop-tetraloop receptor complex by ^{205}Tl NMR methods. Bound ^{205}Tl resonances were not observed in the direct detection ^{205}Tl NMR spectra of the tetraloop-tetraloop receptor complex. This is most likely explained by the highly exposed location of the proposed monovalent binding site in the RNA complex. The possibility that Tl^+ does not bind to the tetraloop receptor cannot be specifically eliminated; however such an explanation would not be expected based on previous studies of this tertiary motif [22, 23, 114].

Our initial efforts to produce a homogenously folded L11 rRNA sample were not successful. Nevertheless, further exploration of this system as a candidate for ^{205}Tl NMR studies is warranted. This 58 nucleotide RNA fragment is known to fold in the presence of monovalent cations, including Tl^+, and does not require the L11 protein to adopt a stable conformation [24, 171]. These features, coupled with its relatively small size and buried monovalent binding site make it an ideal system for ^{205}Tl NMR studies.

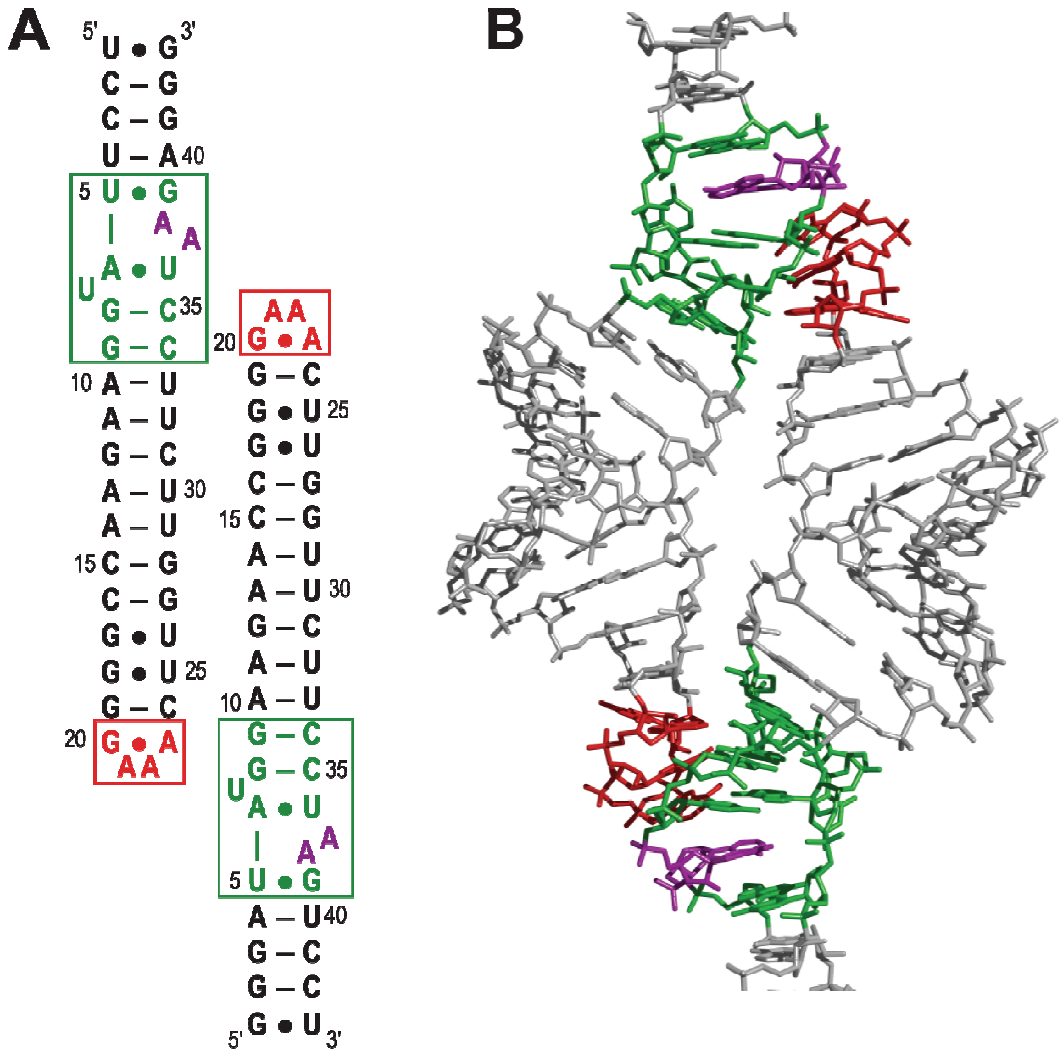


Figure 6-1. GAAA tetraloop-tetraloop receptor complex.

A. Schematic of the complex showing the GAAA tetraloop (red) and the tetraloop receptor (green and purple). The AA platform (purple) is expected to bind a monovalent cation. Figure adapted from [48].

B. A representative model from the solution structure of the GAAA tetraloop-tetraloop receptor complex reported by Butcher and coworkers (PDB 2ADT). The coloring is the same as (A).

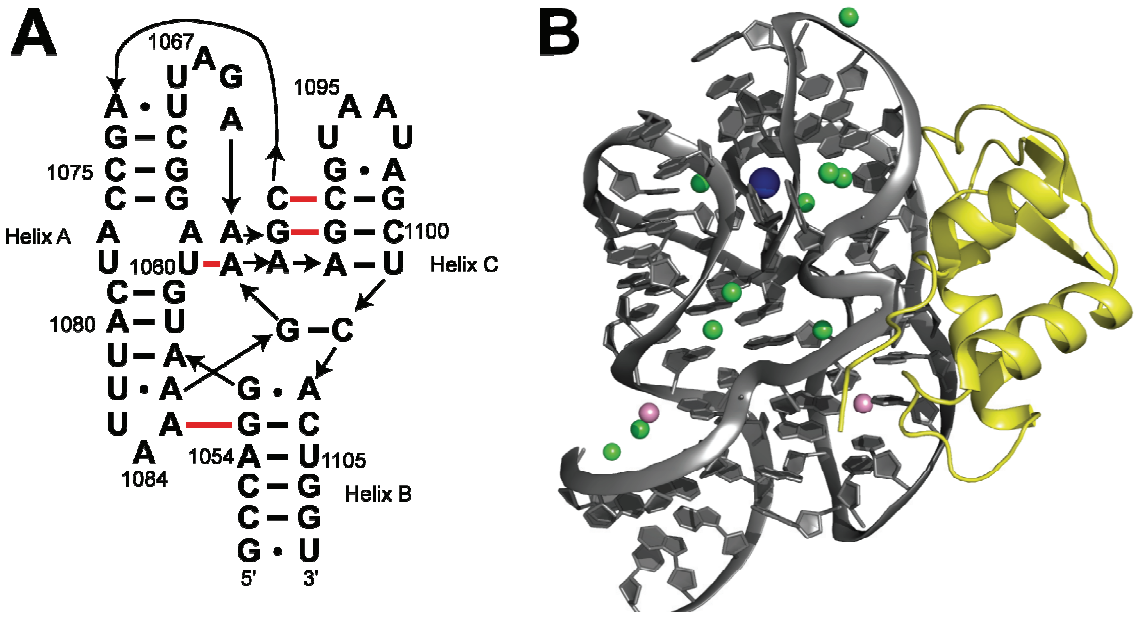


Figure 6-2. The L11 protein:rRNA complex.

A. The 58 nucleotide region of the 23S ribosomal RNA which binds the L11 protein. Tertiary contacts are denoted with red bars. Figure adapted from [24].

B. Crystal structure of the L11 protein:rRNA complex. The L11 protein and RNA are shown as yellow and gray ribbons, respectively. Mg^{2+} is colored green, K^{+} is blue, and $Os(NH_3)_6^{3+}$ is pink (PDB 1HC8).

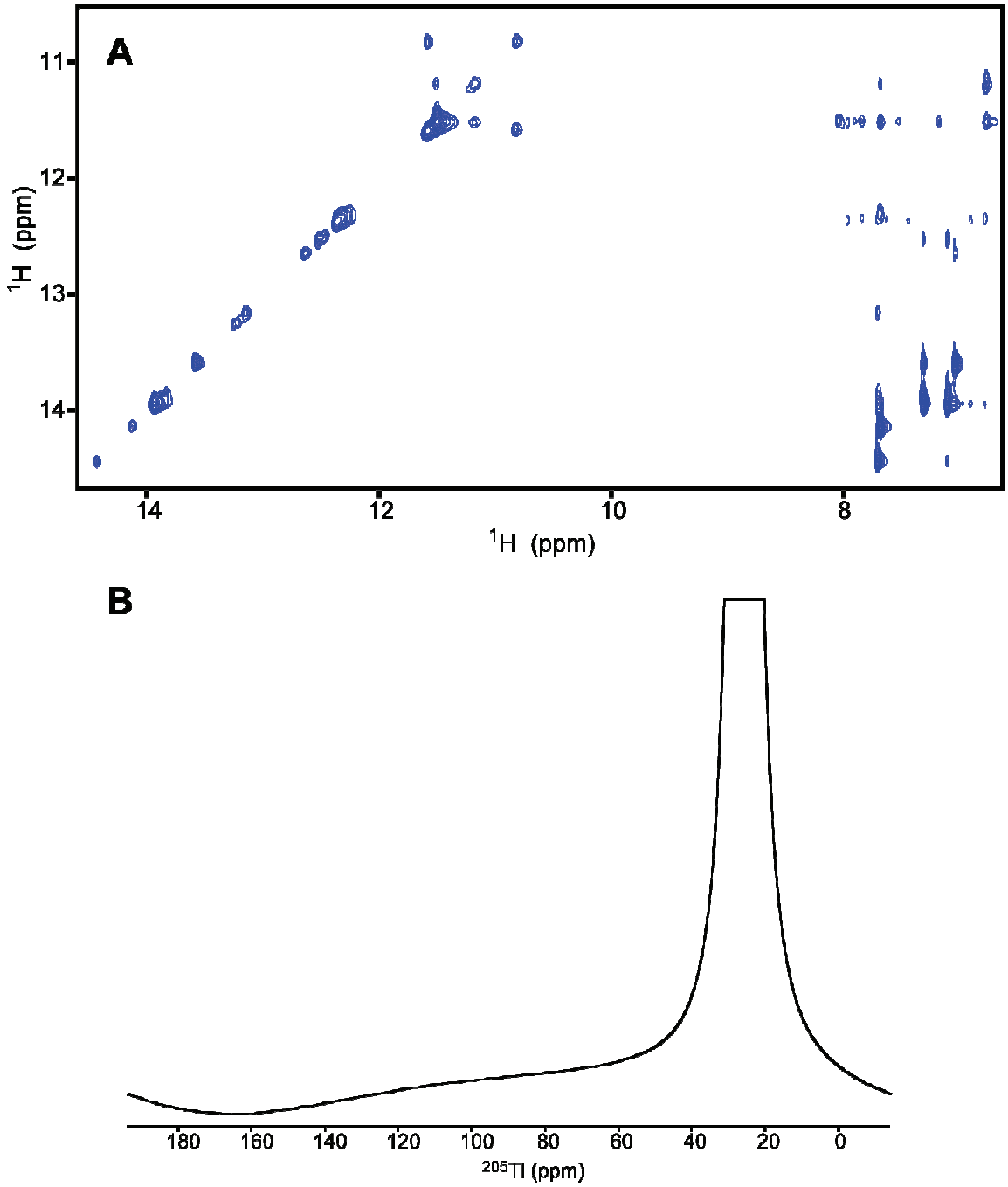


Figure 6-3. Effect of Tl^+ addition to the GAAA tetraloop-tetraloop receptor complex.

A. The presence of NOE crosspeaks involving imino protons indicates that the addition of Tl^+ does not disrupt complex formation.

B. A single ^{205}Tl resonance is observed at 25 ppm in a 1D ^{205}Tl NMR spectrum.

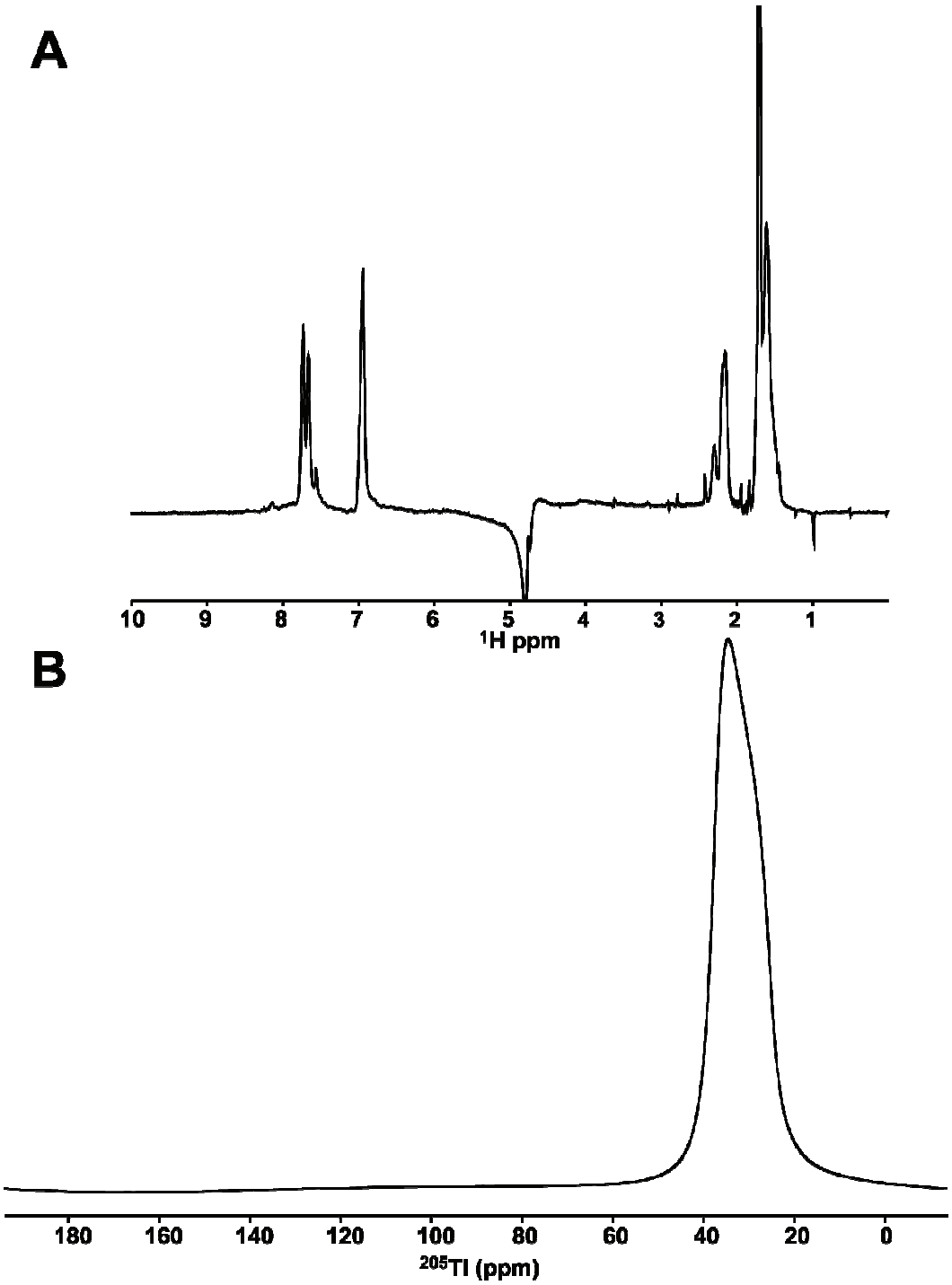


Figure 6-4. Preliminary studies of Tl^{+} binding to the L11 binding portion of the *Escherichia coli* 23S ribosomal RNA.

A. A 1D ^{1}H NMR spectrum of the RNA shows poor chemical shift dispersion, indicating that the RNA is likely unfolded.

B. The ^{205}Tl NMR spectrum contains what appear to be two overlapping peaks at 35 ppm.

7 Concluding Remarks

Monovalent cations play both structural and functional roles in many biological systems, including nucleic acids. The direct study of these cations has been limited largely to crystallography because the alkali metals have a nuclear spin (*I*) greater than ½, making their study by solution NMR difficult. One solution to this problem is the use of a spin-½ monovalent surrogate, such as $^{205}Tl^{+}$. Though there is a small precedent for the use of ^{205}Tl NMR to study K^{+} binding sites in proteins, few studies have been reported involving nucleic acids.

To this end, we have used ^{205}Tl NMR methods for the direct study of Tl^{+} binding to a model system, $d(G_4T_4G_4)_2$. The isomorphous nature of K^{+} and Tl^{+} was first verified by determining the solution and crystallographic structures of the Tl^{+}-form of $d(G_4T_4G_4)_2$. The similarity of each of these structures to their K^{+} counterparts includes even very dynamic regions of the G-quadruplex. The location of all Tl^{+} binding sites has been assigned within the crystallographic model and found to have a 1:1 correspondence to the location of K^{+} binding sites.

We have used direct detection ^{205}Tl NMR and heteronuclear ^{1}H–^{205}Tl NMR to characterize the binding of Tl^{+} to $d(G_4T_4G_4)_2$ in a site-specific manner. The power of direct detection ^{205}Tl NMR is demonstrated by the observation of a previously unobserved complexity in the association of monovalent cations with the G-quadruplex loops. To our knowledge, the results from the ^{1}H–^{205}Tl NMR experiment constitute the first heteronuclear ^{205}Tl NMR experiment reported and the first ^{1}H–^{205}Tl scalar coupling observed in a biological system.

The techniques presented herein can be readily extended to any system in which Tl^+ can be substituted for K^+, resulting in a coordinated cation in slow exchange on the ^{205}Tl chemical shift timescale. One advantage of ^{205}Tl NMR is that its large chemical shift range provides a generous timescale for the slow exchange regime. If ^1H chemical shift assignments are available and the Tl^+ ions have a bound lifetime which allows for experimentally detectable evolution of the ^1H–^{205}Tl scalar coupling ($J_{H–Tl}$), the ^1H–^{205}Tl spin-echo difference experiment can be used to identify residues residing near the monovalent binding site(s).

Information from this experiment could provide constraints for structure determination and allow monitoring of the response of monovalent cation sites to mutation or other perturbations. The relative simplicity of the difference spectrum means that this technique is feasible even when spectral overlap would normally be problematic in a one-dimensional ^1H spectrum. Further development of ^1H–^{205}Tl NMR spectroscopy could prove useful for the study of monovalent cations and their binding sites in a variety of biological and inorganic systems.

8 Footprinting of the *Azoarcus* Group I Intron

8.1 Introduction

The 3.1 Å resolution crystal structure of a self-splicing group I intron with both exons was reported by our laboratory in 2004 [23]. This intron occurs naturally in the pre-tRNA^{Ile} from the purple bacterium *Azoarcus sp. BH72* [173]. In the structure, both the 5′ and 3′ exons are base-paired to the internal guide sequence, forming the P1 and P10 helices, respectively. This conformation occurs on the intron splicing pathway immediately before the second splicing reaction.

After the crystallographic model was complete, it was compared to a phylogenetic model of a group I intron from the same organism [174]. One point of potentially significant divergence between the two structures was the angle between the P4-P6 and the P3-P9.0 domains, which is substantially more acute in the phylogenetic model (Figure 8-1A). The sharp angle was justified by observation of a solvent inaccessible Fe(II)-EDTA footprint in helix P6a (C100-C105) [174, 175]. In the pre-2S crystal structure, this region appears to be fully solvent exposed; however, both the P6a and P8a helices are making intermolecular crystal contacts which might alter the angle between the helices within the crystals (Figure 8-1B).

To explore the correlation between the crystal structure and the solution structure of the pre-2S construct, hydroxyl radical footprinting was performed. Hydroxyl radicals cleave oligonucleotides by extracting a proton from the ribose sugar, usually at the 5′ or 4′ position [176]. They are an effective probe of RNA and DNA solvent accessibility and are sequence independent [177-181]. Comparison of the hydroxyl radical protections to

solvent accessibility calculations was used to assess how closely the conformation observed in the crystal structure matches that which occurs in solution.

8.2 Materials and Methods

8.2.1 Materials and abbreviations.

Nucleoside triphosphates (ATP, CTP, GTP, and UTP) were purchased as disodium salts from Sigma-Aldrich, Inc. Guanosine 5′-monophosphate (GMP), cytidine 3′-monophosphate (Cp), iron(III) ethylenediaminetetraacetic acid (Fe(III)-EDTA), hydrogen peroxide (H_2O_2), thiourea, magnesium chloride ($MgCl_2$), and ethylenediaminetetraacetic acid (EDTA) were also from Sigma-Aldrich, Inc. *EarI* restriction enzyme, T7 polynucleotide kinase (PNK), and T7 RNA ligase were from New England Biolabs. Dithiothreitol (DTT) and Triton X-100 were from American Bioanalytical, Inc. γ–^{32}P ATP (5 mCi) was purchased from Perkin Elmer. The dCIRC (5′-AAGCCACACAAACCdAdGdACGGCC-3′) and CAT (5′-CAdT-3′) were from Dharmacon. T7 RNA polymerase and all S_p-nucleotide α-phosphorothioates (AαS, CαS, GαS, and UαS) were prepared in house [172, 182, 183].

8.2.2 RNA preparation.

The preparation of both the DNA plasmid (pucUP6+2) used for transcription and the U1A protein were performed by Anne Kosek and are described elsewhere [114, 184]. The pucUP6+2 plasmid was cut with *EarI* before in vitro transcription was performed using the following conditions: 40 mM Tris-HCl (pH 7.5), 10 mM DTT, 4 mM spermidine, 20 mM $MgCl_2$, 0.05% (v/v) Triton X-100, 40 μg/ml DNA plasmid, 1 mM NTPs, 8 mM GMP, 1 U/ml IPPase, and T7 RNA polymerase. Transcripts prepared for use as sequence markers also contained 0.5 mM of one of four phosphorothioate

analogues (AαS, CαS, GαS, and UαS). The transcripts were purified by electrophoresis and radioactively labeled using γ–^{32}P ATP and either T7 PNK (5′ end labeling) or Cp and RNA ligase (3′ end labeling). The dCIRC oligonucleotide was 5′ end labeled in a similar manner for footprinting. The labeled RNAs were gel purified and ethanol precipitated before use.

8.2.3 Hydroxyl radical footprinting.

The pre-2S crystallographic complex with U1A protein was formed at ~1 μM, with the labeled RNA slightly limiting. The complex was heated to 50°C and cooled slowly to room temperature. The folded and unfolded samples contained either 10 mM $MgCl_2$ or 10 mM EDTA. Uncleaved samples were also prepared in 10 mM $MgCl_2$ to control for degradation and any metal induced cleavage. Footprinting conditions were 5 mM DTT and 1 mM Fe(II)-EDTA for 80 min at 42°C. Reactions were quenched with 10 mM thiourea and ethanol precipitated. Samples contained between 2×10^5 and 4×10^5 c.p.m. and cleavage products were separated on sequencing gels. Transcripts containing one phosphorothioate analogue (AαS, CαS, GαS, or UαS) were cleaved in 10 mM I_2/ethanol and used as sequence markers. Gels were imaged using a Storm Phosphorimager (Molecular Dynamics).

8.2.4 Data analysis.

Protections were quantitated by normalizing each lane for loading and calculating a protection factor (Pf = unfolded intensity / folded intensity). A position was considered protected if Pf ≥ 1.5. Regions with significant non-Fe(II)-EDTA related degradation (C46, A97, and A109.06), GC compressions (C13-U16 and G75-G83), or at the very ends of the RNA could not be quantitated. Solvent accessibility calculations were

performed on the C4′ carbon of the crystal structure using the program ACCESS [185]. Solvent accessible areas < 1.4 $Å^2$ were considered protected.

8.3 Results

Within the pre-2S structure, there is excellent correlation between the degree of backbone protection and the predicted solvent accessibility (Figure 8-2). The protected C4′ protons are located almost exclusively along the intron core, signifying the dense packing of the ribose-phosphate backbone in these regions (Figure 8-3A–B and Table 8-1). The protections map precisely onto tertiary elements, including both of the TL/TLR contacts and the J8/7 region. Particularly noteworthy is J4/5 where the entire region showed strong protections. Protections were observed in the U1A binding loop, but only when protein was included in the splicing complex (data not shown). A total of 59 residues in the quantifiable regions were protected from reaction with hydroxyl radicals. Due to the heterogeneous ends produced from hydroxyl radical cleavage, these data are considered accurate to within one or two nucleotides. Based on the theoretical solvent accessibility of the C4′ proton calculated from the crystal structure, 60 residues were predicted to show protection (Figure 8-2).

Assuming a resolution of ±1 nucleotide, the only unpredicted protections (A129-G130 and C137) lie on the same side of the P7–P3 helical stack and are immediately adjacent to regions of predicted and observed protections (G122–A127 and G139–C140). Further, the calculated accessibility of the C4′ protons for A129 and G130 (1.4 and 1.7 $Å^2$) is only slightly larger than the limit for classifying a nucleotide as protected. Only six residues predicted to be solvent inaccessible were reactive (Figure 8-2), and each of these

is isolated and located within peripheral elements. Conformational dynamics in solution could readily account for these minor exceptions.

8.4 Discussion

Comparison of these data to that reported for an exon-less intron lacking an internal guide sequence reveals significant differences in only two regions, J4/5 and P6a [174]. These changes may reflect differences in the intron conformation in the presence or absence of its substrates. The J4/5 region was fully solvent accessible in the earlier work, while the entire region was protected in the pre-2S complex (Figure 8-2). Helices P1 and P10 are the principle tertiary contacts of J4/5, so it is reasonable that the absence of these helices would increase J4/5 solvent accessibility. Although protections were observed in J6/6a, none of the protections in P6a that were used to justify the acute angle between P6 and P8 were observed in the pre-2S complex (Figure 8-2) [175]. The solvent accessibility of P6a within the pre-2S complex is consistent with the orientation of P6 and P8 observed in the crystal structure, which argues that the P6-P8 angle is not a crystallization artifact.

8.5 Conclusion

Hydroxyl radical footprinting has been performed on the *Azoarcus* group I intron. The regions which are protected from radical cleavage are very consistent with calculations of solvent accessibility performed on the crystal structure. Taken together, these results indicate that the conformation observed in the crystal structure is very similar to that which exists in solution.

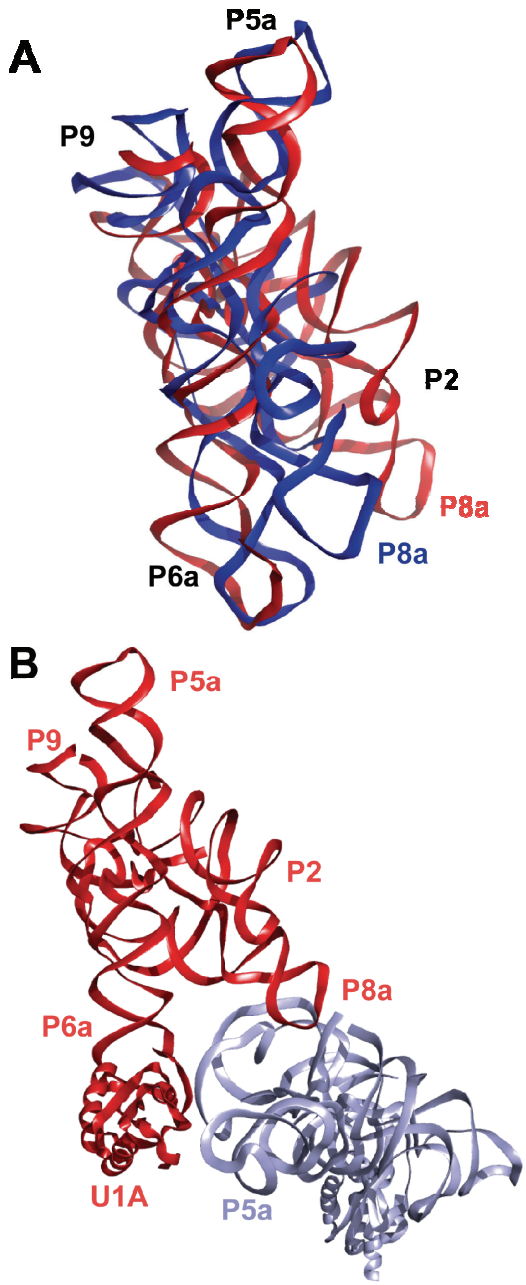


Figure 8-1. Comparison of the crystallographic structure and phylogenetic model of the *Azoarcus sp. BH72* group I intron.

A. The angle made by P6a and P8a helices in the phylogenetic model (blue) is significantly more acute than observed in the crystal structure (red) (PDB 1U6B).

B. In the crystal structure, both the P8a and P6a helices mediate intermolecular packing. The red and gray structures are related by the observed crystallographic symmetry [23].

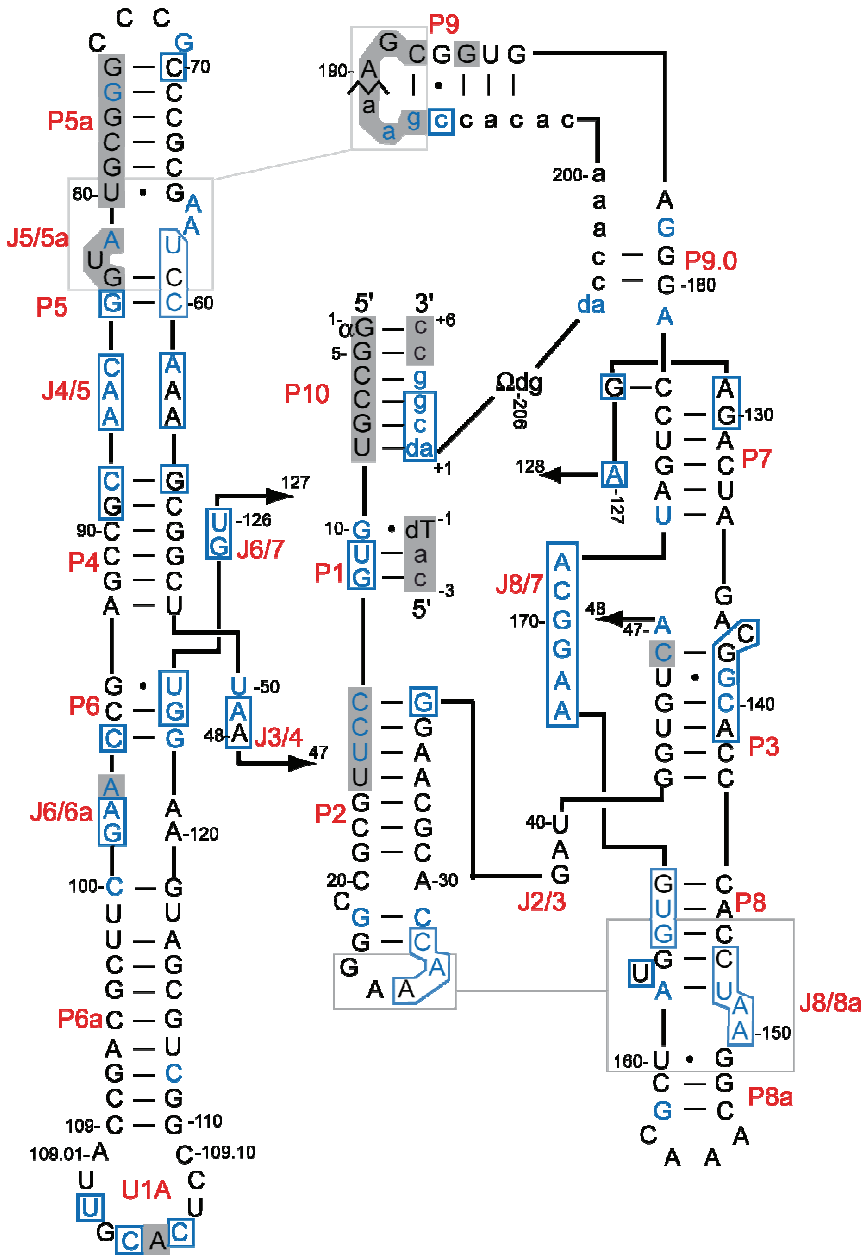


Figure 8-2. Comparison of hydroxyl radical footprinting of *Azoarcus sp. BH72* group I intron to calculated solvent accessibility.
The color of the letters indicates residues whose C4' proton is predicted to be protected (blue) or solvent accessible (black) within the crystal structure. Positions that were protected from hydroxyl radical reactivity in solution are enclosed in blue boxes. Gray squares indicate positions that could not be quantitated.

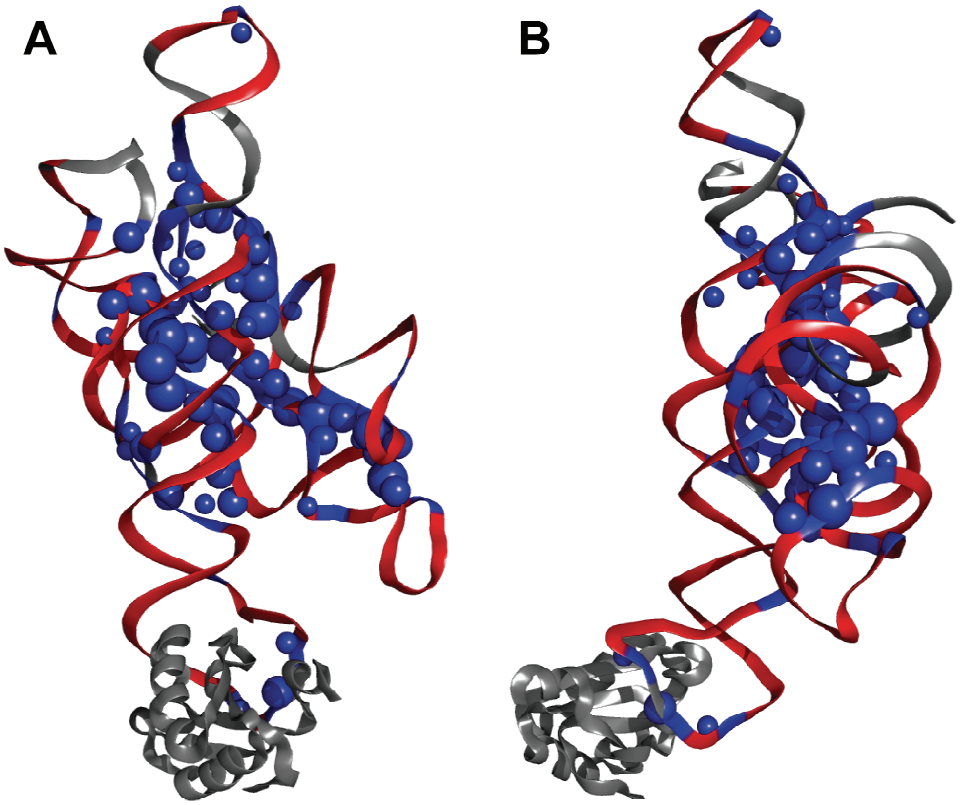


Figure 8-3. Three dimensional view of solvent protected regions on the *Azoarcus sp. BH72* group I intron.

A. Front view of hydroxyl radical protections mapped onto the pre-2S crystal structure. The backbone color indicates areas of predicted protection (blue) and predicted solvent accessibility (red). Areas that could not be quantitated are colored gray. Blue spheres indicate C4' atoms that were resistant to hydroxyl radicals in solution. The sphere size correlates with the degree of protection (small: 1.5–2.0, medium: 2.0–3.0, large: >3.0).

B. As in (B), but rotated 90° about the P4-P6 axis so the clustering of protected residues along the interior of the pre-2S structure can be visualized.

	Sequence	Pf		Sequence	Pf		Sequence	Pf
P10	1	–	P3	41	0.7 ± 0.1	P5a	78	–
	5	–		42	0.7 ± 0.1		79	–
	6	–		43	0.7 ± 0.1		80	–
	7	–		44	0.9 ± 0.0	J5/5a	81	–
	8	–		45	1.3 ± 0.3		82	–
	9	–		46	–	P5	83	–
P1	10	0.5 ± 0.0	J3/4	47	0.7 ± 0.5		84	1.5 ± 0.1
	11	3.1 ± 0.8		48	1.8 ± 0.2	J4/5	85	1.8 ± 0.6
	12	3.1 ± 0.3		49	1.7 ± 0.2		86	1.7 ± 0.7
P2	13	–		50	1.1 ± 0.0		87	2.0 ± 0.6
	14	–	P4	51	0.9 ± 0.1	P4	88	2.0 ± 0.7
	15	–		52	1.0 ± 0.1		89	1.5 ± 0.1
	16	–		53	1.2 ± 0.3		90	1.3 ± 0.2
	17	0.7 ± 0.2		54	1.4 ± 0.3		91	1.2 ± 0.3
	18	0.7 ± 0.1		55	1.4 ± 0.2		92	1.0 ± 0.1
	19	0.7 ± 0.1		56	1.7 ± 0.2		93	0.9 ± 0.1
	20	0.6 ± 0.1	J4/5	57	1.8 ± 0.3	P6	94	0.8 ± 0.1
	21	0.8 ± 0.1		58	2.2 ± 0.4		95	1.1 ± 0.1
	22	1.0 ± 0.1		59	2.4 ± 0.5		96	2.9 ± 0.3
	23	1.3 ± 0.1	P5	60	2.2 ± 0.0	J6/6a	97	–
	24	1.0 ± 0.0		61	2.2 ± 0.2		98	2.5 ± 0.9
	25	1.3 ± 0.0	J5/5a	62	1.6 ± 0.7		99	1.7 ± 0.4
	26	2.5 ± 0.0		63	1.4 ± 0.0	P6a	100	1.3 ± 0.0
	27	4.4 ± 0.2		64	1.5 ± 0.1		101	1.1 ± 0.1
	28	2.8 ± 0.5	P5a	65	1.5 ± 0.2		102	1.0 ± 0.0
	29	1.3 ± 0.2		66	1.5 ± 0.1		103	1.0 ± 0.0
	30	1.1 ± 0.0		67	1.4 ± 0.1		104	1.0 ± 0.0
	31	0.9 ± 0.0		68	1.4 ± 0.1		105	1.0 ± 0.1
	32	0.8 ± 0.0		69	1.4 ± 0.1		106	1.0 ± 0.2
	33	0.8 ± 0.1		70	1.5 ± 0.1		107	1.0 ± 0.3
	34	0.8 ± 0.1		71	1.0 ± 0.0	U1A	108	1.1 ± 0.5
	35	0.9 ± 0.0		72	0.9 ± 0.2		109	1.3 ± 0.2
	36	1.1 ± 0.1		73	0.9 ± 0.2		109.01	1.1 ± 0.4
	37	1.5 ± 0.1		74	0.9 ± 0.0		109.02	1.0 ± 0.1
J2/3	38	1.3 ± 0.1		75	–		109.03	1.6 ± 0.0
	39	1.2 ± 0.0		76	–		109.04	0.9 ± 0.2
	40	1.0 ± 0.1		77	–		109.05	2.0 ± 0.5

Table 8-1. Quantitation of *Azoarcus* group I intron footprinting.

	Sequence	Pf		Sequence	Pf		Sequence	Pf
U1A	109.06	–	P3	142	1.2 ± 0.0	P9.0	179	0.7 ± 0.0
	109.07	1.9 ± 0.5		143	1.0 ± 0.1		180	0.8 ± 0.1
	109.08	1.0 ± 0.4	P8	144	1.0 ± 0.1		181	0.8 ± 0.1
	109.09	1.0 ± 0.5		145	1.2 ± 0.1		182	0.3 ± 0.0
	109.10	0.9 ± 0.4		146	1.3 ± 0.2		183	0.3 ± 0.0
	110	0.9 ± 0.3		147	1.7 ± 0.1	P9	184	0.3 ± 0.0
	111	1.0 ± 0.1		148	2.6 ± 0.6		185	1.2 ± 0.1
P6a	112	0.9 ± 0.1	J8/8a	149	3.0 ± 0.2		186	–
	113	1.0 ± 0.2		150	2.0 ± 0.7		187	1.0 ± 0.2
	114	1.0 ± 0.2	P8a	151	1.5 ± 0.2		188	–
	115	1.0 ± 0.2		152	1.3 ± 0.1		189	–
	116	1.1 ± 0.4		153	1.3 ± 0.1		190	–
	117	1.1 ± 0.3		154	1.2 ± 0.1		191	–
	118	1.1 ± 0.3		155	0.8 ± 0.1		192	–
	119	1.0 ± 0.3		156	0.6 ± 0.0		193	–
J6/6a	120	1.0 ± 0.2		157	0.7 ± 0.1		194	2.9 ± 0.9
	121	1.1 ± 0.2		158	0.7 ± 0.1		195	1.1 ± 0.0
P6	122	1.4 ± 0.1		159	0.8 ± 0.1		196	1.0 ± 0.1
	123	2.0 ± 0.0		160	0.9 ± 0.1		197	0.3 ± 0.0
	124	3.3 ± 0.0	P8	161	1.2 ± 0.2		198	1.1 ± 0.1
J6/7	125	4.1 ± 0.0		162	1.5 ± 0.3		199	0.2 ± 0.0
	126	3.3 ± 0.2		163	1.2 ± 0.2	P9.0	200	1.1 ± 0.2
	127	6.6 ± 1.0		164	2.5 ± 1.0		201	1.3 ± 0.1
P7	128	4.2 ± 1.1		165	3.6 ± 0.5		202	0.9 ± 0.0
	129	2.2 ± 0.4		166	3.7 ± 1.3		203	1.0 ± 0.0
	130	1.6 ± 0.0	J8/7	167	2.4 ± 1.0		204	0.7 ± 0.1
	131	1.3 ± 0.3		168	2.0 ± 0.1		205	1.4 ± 0.0
	132	1.1 ± 0.3		169	2.2 ± 0.7		206	0.7 ± 0.0
	133	1.0 ± 0.1		170	2.6 ± 0.0	3'-exon	1	1.5 ± 0.1
	134	1.1 ± 0.2		171	3.1 ± 0.8		2	1.9 ± 0.0
	135	1.1 ± 0.3		172	2.0 ± 0.5		3	1.7 ± 0.0
P3	136	1.4 ± 0.7	P7	173	1.3 ± 0.9		4	0.3 ± 0.1
	137	2.3 ± 0.4		174	0.9 ± 0.0		5	–
	138	3.1 ± 1.4		175	0.6 ± 0.1		6	–
	139	2.8 ± 0.4		176	0.7 ± 0.1			
	140	2.6 ± 0.5		177	1.1 ± 0.1			
	141	2.1 ± 0.2		178	1.2 ± 0.1			

Table 8-1. Continued from previous page.

9 References

1. Wuthrich K., *NMR of Proteins and Nucleic Acids*. 1986, New York: John Wiley & Sons. 292.
2. Karplus M., *Contact electron-spin coupling of nuclear magnetic moments.* J. Chem. Phys., 1959. **30**: p. 11-18.
3. Prestegard J.H., al-Hashimi, H.M., and Tolman, J.R., *NMR structures of biomolecules using field oriented media and residual dipolar couplings.* Q. Rev. Biophys., 2000. **33**(4): p. 371-424.
4. Brunger A.T., Adams, P.D., Clore, G.M., DeLano, W.L., Gros, P., Grosse-Kunstleve, R.W., Jiang, J.S., Kuszewski, J., Nilges, M., Pannu, N.S., Read, R.J., Rice, L.M., Simonson, T., and Warren, G.L., *Crystallography & NMR System: A new software suite for macromolecular structure determination.* Acta Crystallogr D Biol Crystallogr, 1998. **54 (Pt 5)**: p. 905-21.
5. Vogel H.J., Drakenberg, T., Forsén, S., O'Neil, J.D.J., and Hofmann, T., *Structural differences in the two calcium binding sites of the porcine intestinal calcium binding protein: A multinuclear NMR study.* Biochemistry, 1985. **24**: p. 3870-3876.
6. Kellenbach E., Maler, B.A., Yamamoto, K.R., Boelens, R., and Kaptein, R., *Identification of the metal coordinating residues in the DNA binding domain of the glucocorticoid receptor by ^{113}Cd-^{1}H heteronuclear NMR spectroscopy.* FEBS, 1991. **291**: p. 367-370.
7. Blake P.R., Lee, B., Summers, M.F., Adams, M.W., Park, J.B., Zhou, Z.H., and Bax, A., *Quantitative measurement of small through-hydrogen-bond and 'through-space' ^{1}H-^{113}Cd and ^{1}H-^{199}Hg J couplings in metal-substituted rubredoxin from Pyrococcus furiosus.* J Biomol NMR, 1992. **2**(5): p. 527-33.
8. Blake P.R., Park, J.B., Adams, M.W.W., and Summers, M.F., *Novel Observation of Nh...S(Cys) Hydrogen-Bond-Mediated Scalar Coupling in Cd-113-Substituted Rubredoxin from Pyrococcus-Furiosus.* Journal of the American Chemical Society, 1992. **114**(12): p. 4931-4933.
9. Cass A.E.G., Galdes, A., Hill, H.A.O., McClelland, C.E., and Storm, C.B., *Heavy metal binding to biological molecules: Identification of ligands by observation of ^{199}Hg-^{1}H coupling.* FEBS Letters, 1978. **94**: p. 311-314.
10. Utschig L., Bryson, J., and O'Halloran, T., *Mercury-199 NMR of the metal receptor site in MerR and its protein-DNA complex.* Science, 1995. **268**: p. 380-385.
11. Coleman J.E., *Cadmium-113 nuclear magnetic resonance applied to metalloproteins.* Methods Enzymol, 1993. **227**: p. 16-43.
12. Neuhaus D., Wagner, G., Vasak, M., Kagi, J.H.R., and Wuthrich, K., *Cd-113-H-1 Spin-Spin Couplings in Homonuclear H-1 Correlated Spectroscopy of Metallothionein - Identification of the Cysteine H-1 Spin Systems.* Eur. J. Biochem., 1984. **143**(3): p. 659-667.
13. Frey M.H., Wagner, G., Vasak, M., Sorensen, O.W., Neuhaus, D., Worgotter, E., Kagi, J.H.R., Ernst, R.R., and Wuthrich, K., *Polypeptide Metal Cluster Connectivities in Metallothionein-2 by Novel H-1-Cd-113 Heteronuclear Two-*

Dimensional NMR Experiments. Journal of the American Chemical Society, 1985. **107**(24): p. 6847-6851.

14. Pan T. and Coleman, J., *GAL4 Transcription Factor is Not a "Zinc Finger" but Forms a $Zn(II)_2Cys_6$ Binuclear Cluster.* PNAS, 1990. **87**(6): p. 2077-2081.
15. Gardner K.H., Pan, T., Narula, S., Rivera, E., and Coleman, J.E., *Structure of the binuclear metal-binding site in the GAL4 transcription factor.* Biochemistry, 1991. **30**(47): p. 11292-302.
16. Gardner K.H. and Coleman, J.E., *^{113}Cd-^{1}H heteroTOCSY: a method for determining metal-protein connectivities.* J Biomol NMR, 1994. **4**(6): p. 761-74.
17. Live D., Armitage, I.M., Dalgarno, D.C., and Cowburn, D., *Two-Dimensional ^{1}H-^{113}Cd Chemical-Shift Correlation Maps by ^{1}H-Detected Multiple-Quantum NMR in Metal Complexes and Metalloproteins.* J Am Chem Soc, 1985. **107**: p. 1775-7.
18. Hess B. and Haeckel, R., *Interaction between potassium-, ammonium- and fructose-1,6-diphosphate activation of yeast pyruvate kinase.* Nature, 1967. **214**(90): p. 848-9.
19. Nowak T. and Mildvan, A.S., *Nuclear magnetic resonance studies of the function of potassium in the mechanism of pyruvate kinase.* Biochemistry, 1972. **11**(15): p. 2819-28.
20. Shiman R. and Draper, D.E., *Stabilization of RNA tertiary structure by monovalent cations.* J Mol Biol, 2000. **302**(1): p. 79-91.
21. Basu S. and Strobel, S.A., *Thiophilic metal ion rescue of phosphorothioate interference within the Tetrahymena ribozyme P4-P6 domain.* RNA, 1999. **5**(11): p. 1399-407.
22. Basu S., Rambo, R.P., Strauss-Soukup, J., Cate, J.H., Ferre-D'Amare, A.R., Strobel, S.A., and Doudna, J.A., *A specific monovalent metal ion integral to the AA platform of the RNA tetraloop receptor.* Nat Struct Biol, 1998. **5**(11): p. 986-92.
23. Adams P.L., Stahley, M.R., Kosek, A.B., Wang, J., and Strobel, S.A., *Crystal structure of a self-splicing group I intron with both exons.* Nature, 2004. **430**(6995): p. 45-50.
24. Conn G.L., Gittis, A.G., Lattman, E.E., Misra, V.K., and Draper, D.E., *A compact RNA tertiary structure contains a buried backbone-K^+ complex.* J Mol Biol, 2002. **318**(4): p. 963-73.
25. Mattai J., Hauser, H., Demel, R.A., and Shipley, G.G., *Interactions of metal ions with phosphatidylserine bilayer membranes: effect of hydrocarbon chain unsaturation.* Biochemistry, 1989. **28**(5): p. 2322-30.
26. Tessier C., Quinn, P., Koumanov, K., Trugnan, G., Rainteau, D., and Wolf, C., *Modulation of the phase heterogeneity of aminoglycerophospholipid mixtures by sphingomyelin and monovalent cations: maintenance of the lamellar arrangement in the biological membranes.* Eur Biophys J, 2004. **33**(6): p. 513-21.
27. Toner M., Vaio, G., McLaughlin, A., and McLaughlin, S., *Adsorption of cations to phosphatidylinositol 4,5-bisphosphate.* Biochemistry, 1988. **27**(19): p. 7435-43.
28. Coughlin R.T., Tonsager, S., and McGroarty, E.J., *Quantitation of metal cations bound to membranes and extracted lipopolysaccharide of Escherichia coli.* Biochemistry, 1983. **22**(8): p. 2002-7.

29. Roux M. and Bloom, M., *Ca^{2+}, Mg^{2+}, Li^{+}, Na^{+}, and K^{+} distributions in the headgroup region of binary membranes of phosphatidylcholine and phosphatidylserine as seen by deuterium NMR.* Biochemistry, 1990. **29**(30): p. 7077-89.
30. Moulik S.P. and Khan, D.P., *Complexation of reducing and nonreducing carbohydrates with hydroxides of some alkali and alkaline-earth metals.* Carbohydr Res, 1975. **41**: p. 93-104.
31. Kitamura N., Ikekita, M., Sato, T., Akimoto, Y., Hatanaka, Y., Kawakami, H., Inomata, M., and Furukawa, K., *Mouse Na^{+}/K^{+}-ATPase beta1-subunit has a K^{+}-dependent cell adhesion activity for beta-GlcNAc-terminating glycans.* Proc Natl Acad Sci U S A, 2005. **102**(8): p. 2796-801.
32. Kano K., Kitae, T., Shimofuri, Y., Tanaka, N., and Mineta, Y., *Complexation of polyvalent cyclodextrin ions with oppositely charged guests: entropically favorable complexation due to dehydration.* Chemistry, 2000. **6**(15): p. 2705-13.
33. Lerner L. and Torchia, D.A., *A multinuclear NMR study of the interactions of cations with proteoglycans, heparin, and Ficoll.* J Biol Chem, 1986. **261**(27): p. 12706-14.
34. Pyle A.M., *Catalytic Reaction Mechanisms and Structural Features of Group II Intron Ribozymes*, in *Nucleic Acids and Molecular Biology*, F. Eckstein and D.M. Lilley, Editors. p. 75-107.
35. Stahley M.R. and Strobel, S.A., *Structural evidence for a two-metal-ion mechanism of group I intron splicing.* Science, 2005. **309**(5740): p. 1587-90.
36. Nissen P., Hansen, J., Ban, N., Moore, P.B., and Steitz, T.A., *The structural basis of ribosome activity in peptide bond synthesis.* Science, 2000. **289**(5481): p. 920-30.
37. Pestka S., *Peptidyl-puromycin synthesis on polyribosomes from Escherichia coli.* Proc Natl Acad Sci U S A, 1972. **69**(3): p. 624-8.
38. Miskin R., Zamir, A., and Elson, D., *Inactivation and reactivation of ribosomal subunits: the peptidyl transferase activity of the 50 S subunit of Escherihia coli.* J Mol Biol, 1970. **54**(2): p. 355-78.
39. Hultin T. and Naslund, P.H., *Effects of thallium (I) on the structure and functions of mammalian ribosomes.* Chem Biol Interact, 1974. **8**(5): p. 315-28.
40. Hultin T. and Naslund, P.H., *Ion binding and ribosomal conformation and function. Experiments with the K^{+} analogue, Tl^{+}.* Acta Biol Med Ger, 1974. **33**(5-6): p. 753-60.
41. Conn G.L., Draper, D.E., Lattman, E.E., and Gittis, A.G., *Crystal structure of a conserved ribosomal protein-RNA complex.* Science, 1999. **284**(5417): p. 1171-4.
42. Basu S. and Strobel, S.A., *Biochemical detection of monovalent metal ion binding sites within RNA.* Methods, 2001. **23**(3): p. 264-75.
43. Abramovitz D.L. and Pyle, A.M., *Remarkable morphological variability of a common RNA folding motif: the GNRA tetraloop-receptor interaction.* J Mol Biol, 1997. **266**(3): p. 493-506.
44. Boudvillain M. and Pyle, A.M., *Defining functional groups, core structural features and inter-domain tertiary contacts essential for group II intron self-splicing: a NAIM analysis.* Embo J, 1998. **17**(23): p. 7091-104.

45. Konforti B.B., Liu, Q., and Pyle, A.M., *A map of the binding site for catalytic domain 5 in the core of a group II intron ribozyme.* Embo J, 1998. **17**(23): p. 7105-17.
46. Murray J.B., Seyhan, A.A., Walter, N.G., Burke, J.M., and Scott, W.G., *The hammerhead, hairpin and VS ribozymes are catalytically proficient in monovalent cations alone.* Chem Biol, 1998. **5**(10): p. 587-95.
47. Curtis E.A. and Bartel, D.P., *The hammerhead cleavage reaction in monovalent cations.* RNA, 2001. **7**(4): p. 546-52.
48. Davis J.H., Tonelli, M., Scott, L.G., Jaeger, L., Williamson, J.R., and Butcher, S.E., *RNA helical packing in solution: NMR structure of a 30 kDa GAAA tetraloop-receptor complex.* Journal of Molecular Biology, 2005. **351**(2): p. 371-382.
49. Qin P.Z., Butcher, S.E., Feigon, J., and Hubbell, W.L., *Quantitative Analysis of the Isolated GAAA Tetraloop/Receptor Interaction in Solution: A Site-Directed Spin Labeling Study.* Biochemistry, 2001. **40**(23): p. 6929-6936.
50. Jaeger L., Westhof, E., and Leontis, N.B., *TectoRNA: modular assembly units for the construction of RNA nano-objects.* Nucl. Acids Res., 2001. **29**(2): p. 455-463.
51. Downey C.D., Fiore, J.L., Stoddard, C.D., Hodak, J.H., Nesbitt, D.J., and Pardi, A., *Metal ion dependence, thermodynamics, and kinetics for intramolecular docking of a GAAA tetraloop and receptor connected by a flexible linker.* Biochemistry, 2006. **45**(11): p. 3664-73.
52. Hodak J.H., Downey, C.D., Fiore, J.L., Pardi, A., and Nesbitt, D.J., *Docking kinetics and equilibrium of a GAAA tetraloop-receptor motif probed by single-molecule FRET.* PNAS, 2005. **102**(30): p. 10505-10510.
53. Zhou Y. and MacKinnon, R., *Ion binding affinity in the cavity of the KcsA potassium channel.* Biochemistry, 2004. **43**(17): p. 4978-82.
54. Zhou Y. and MacKinnon, R., *The occupancy of ions in the K^+ selectivity filter: charge balance and coupling of ion binding to a protein conformational change underlie high conduction rates.* J Mol Biol, 2003. **333**(5): p. 965-75.
55. Nordenskiold L., Chang, D.K., Anderson, C.F., and Record, M.T., Jr., *^{23}Na NMR relaxation study of the effects of conformation and base composition on the interactions of counterions with double-helical DNA.* Biochemistry, 1984. **23**(19): p. 4309-17.
56. Cesare Marincola F., Denisov, V.P., and Halle, B., *Competitive Na(+) and Rb(+) binding in the minor groove of DNA.* J Am Chem Soc, 2004. **126**(21): p. 6739-50.
57. Halle B., Denisov, V.P., and Thomas L. James, V.D.a.U.S., *Magnetic relaxation dispersion studies of biomolecular solutions*, in *Methods in Enzymology*. 2002, Academic Press. p. 178-201.
58. Borzo M., Detellier, C., Laszlo, P., and Paris, A., *H-1, Na-23, and P-31 Nmr-Studies of the Self-Assembly of the 5'-Guanosine Monophosphate Dianion in Neutral Aqueous-Solution in the Presence of Sodium-Cations.* Journal of the American Chemical Society, 1980. **102**(3): p. 1124-1134.
59. Deng H. and Braunlin, W.H., *Kinetics of sodium ion binding to DNA quadruplexes.* J Mol Biol, 1996. **255**(3): p. 476-83.

60. Feigon J., Butcher, S.E., Finger, L.D., and Hud, N.V., *Solution nuclear magnetic resonance probing of cation binding sites on nucleic acids.* Methods Enzymol, 2001. **338**: p. 400-20.
61. Kieft J.S. and Tinoco, I., Jr., *Solution structure of a metal-binding site in the major groove of RNA complexed with cobalt (III) hexammine.* Structure, 1997. **5**(5): p. 713-21.
62. Hud N.V., Smith, F.W., Anet, F.A., and Feigon, J., *The selectivity for K^+ versus Na^+ in DNA quadruplexes is dominated by relative free energies of hydration: a thermodynamic analysis by ^1H NMR.* Biochemistry, 1996. **35**(48): p. 15383-90.
63. Gonzalez R.L., Jr. and Tinoco, I., Jr., *Identification and characterization of metal ion binding sites in RNA.* Methods Enzymol, 2001. **338**: p. 421-43.
64. Hud N.V., Schultze, P., and Feigon, J., *Ammonium ion as an NMR probe for monovalent cation coordination sites of DNA quadruplexes.* Journal of the American Chemical Society, 1998. **120**(25): p. 6403-6404.
65. Hud N.V., Schultze, P., Sklenar, V., and Feigon, J., *Binding sites and dynamics of ammonium ions in a telomere repeat DNA quadruplex.* J Mol Biol, 1999. **285**(1): p. 233-43.
66. Hud N.V., Sklenar, V., and Feigon, J., *Localization of ammonium ions in the minor groove of DNA duplexes in solution and the origin of DNA A-tract bending.* J Mol Biol, 1999. **286**(3): p. 651-60.
67. Loria J.P. and Nowak, T., *Conformational changes in yeast pyruvate kinase studied by $^{205}Tl^+$ NMR.* Biochemistry, 1998. **37**(19): p. 6967-74.
68. Pauling L., *The nature of the chemical bond and the structure of molecules and crystals; an introduction to modern structural chemistry*. 3d ed. The George Fisher Baker non-resident lectureship in chemistry at Cornell University ; [v. 18]. 1960, Ithaca, N.Y.,: Cornell University Press. 644 p.
69. Cox B.G. and Schneider, H., *Coordination and transport properties of macrocyclic compounds in solution*. Studies in physical and theoretical chemistry. 1992, Amsterdam ; New York: Elsevier. 420 p.
70. Draper D.E. and Misra, V.K., *RNA shows its metal.* Nat Struct Biol, 1998. **5**(11): p. 927-30.
71. Brown I.D., *What Factors Determine Cation Coordination Numbers.* Acta Crystallographica Section B-Structural Science, 1988. **44**: p. 545-553.
72. Douglas K.T., Bunni, M.A., and Baindur, S.R., *Thallium in biochemistry.* Int J Biochem, 1990. **22**(5): p. 429-38.
73. Basu S., Szewczak, A.A., Cocco, M., and Strobel, S.A., *Direct Detection of Monovalent Metal Ion Binding to a DNA G-quartet by ^{205}Tl NMR.* J Am Chem Soc, 2000. **122**(13): p. 3240-3241.
74. Bostian K., Betts, G.F., Man, W.K., and Hughes, M.N., *Thallium activation and inhibition of yeast aldehyde dehydrogenase.* FEBS Lett, 1975. **59**(1): p. 88-91.
75. Bostian K.A., Betts, G.F., Man, W.K., and Hughes, M.N., *Multiple binding of thallium and rubidium to potassium-activated yeast aldehyde dehydrogenase. Influences on tertiary structure, stability and catalytic activity.* Biochem J, 1982. **207**(1): p. 73-80.

76. Grisham C.M., Gupta, R.K., Barnett, R.E., and Mildvan, A.S., *Thallium-205 nuclear relaxation and kinetic studies of sodium and potassium ion-activated adenosine triphosphatase.* J Biol Chem, 1974. **249**(21): p. 6738-44.
77. Harris R.K., *Nuclear magnetic resonance spectroscopy : a physicochemical view.* 1989, Burnt Mill, Harlow, Essex, England, New York: Longman Scientific & Technical; J. Wiley. 260 p.
78. Hinton J.F., *Thallium-NMR Spectroscopy.* Magnetic Resonance in Chemistry, 1987. **25**(8): p. 659-669.
79. Hinton J.F., Metz, K.R., and Briggs, R.W., *Thallium NMR-Spectroscopy.* Progress in Nuclear Magnetic Resonance Spectroscopy, 1988. **20**: p. 423-513.
80. Reuben J. and Kayne, F.J., *Thallium-205 nuclear magnetic resonance study of pyruvate kinase and its substrates. Evidence for a substrate-induced conformational change.* J Biol Chem, 1971. **246**(20): p. 6227-34.
81. Hinton J.F., Whaley, W.L., Shungu, D., Koeppe, R.E., 2nd, and Millett, F.S., *Equilibrium binding constants for the group I metal cations with gramicidin-A determined by competition studies and Tl^+-205 nuclear magnetic resonance spectroscopy.* Biophys J, 1986. **50**(3): p. 539-44.
82. Hinton J.F., Young, G., and Millett, F.S., *Thallous ion interaction with gramicidin incorporated in micelles studied by thallium-205 nuclear magnetic resonance.* Biochemistry, 1982. **21**(4): p. 651-4.
83. Briggs R.W. and Hinton, J.F., *A thallium-205 nuclear magnetic resonance investigation of the thallium(I)-valinomycin complex.* Journal of Magnetic Resonance (1969), 1978. **32**(1): p. 155-160.
84. Williamson J.R., Raghuraman, M.K., and Cech, T.R., *Monovalent cation-induced structure of telomeric DNA: the G-quartet model.* Cell, 1989. **59**(5): p. 871-80.
85. Williamson J.R., *G-quartet structures in telomeric DNA.* Annu Rev Biophys Biomol Struct, 1994. **23**: p. 703-30.
86. Parkinson G.N., Lee, M.P., and Neidle, S., *Crystal structure of parallel quadruplexes from human telomeric DNA.* Nature, 2002. **417**(6891): p. 876-80.
87. Wang Y. and Patel, D.J., *Solution structure of the human telomeric repeat $d[AG_3(T_2AG_3)_3]$ G-tetraplex.* Structure, 1993. **1**(4): p. 263-82.
88. Sen D. and Gilbert, W., *Formation of Parallel 4-Stranded Complexes by Guanine-Rich Motifs in DNA and Its Implications for Meiosis.* Nature, 1988. **334**(6180): p. 364-366.
89. Evans T., Schon, E., Gora-Maslak, G., Patterson, J., and Efstratiadis, A., *S1-hypersensitive sites in eukaryotic promoter regions.* Nucleic Acids Res, 1984. **12**(21): p. 8043-58.
90. Kilpatrick M.W., Torri, A., Kang, D.S., Engler, J.A., and Wells, R.D., *Unusual DNA structures in the adenovirus genome.* J Biol Chem, 1986. **261**(24): p. 11350-4.
91. Han H. and Hurley, L.H., *G-quadruplex DNA: a potential target for anti-cancer drug design.* Trends Pharmacol Sci, 2000. **21**(4): p. 136-42.
92. Gowan S.M., Harrison, J.R., Patterson, L., Valenti, M., Read, M.A., Neidle, S., and Kelland, L.R., *A G-quadruplex-interactive potent small-molecule inhibitor of telomerase exhibiting in vitro and in vivo antitumor activity.* Mol Pharmacol, 2002. **61**(5): p. 1154-62.

93. Read M., Harrison, R.J., Romagnoli, B., Tanious, F.A., Gowan, S.H., Reszka, A.P., Wilson, W.D., Kelland, L.R., and Neidle, S., *Structure-based design of selective and potent G quadruplex-mediated telomerase inhibitors.* Proc Natl Acad Sci U S A, 2001. **98**(9): p. 4844-9.
94. Izbicka E., Wheelhouse, R.T., Raymond, E., Davidson, K.K., Lawrence, R.A., Sun, D., Windle, B.E., Hurley, L.H., and Von Hoff, D.D., *Effects of cationic porphyrins as G-quadruplex interactive agents in human tumor cells.* Cancer Res, 1999. **59**(3): p. 639-44.
95. Jing N., De Clercq, E., Rando, R.F., Pallansch, L., Lackman-Smith, C., Lee, S., and Hogan, M.E., *Stability-activity relationships of a family of G-tetrad forming oligonucleotides as potent HIV inhibitors. A basis for anti-HIV drug design.* J Biol Chem, 2000. **275**(5): p. 3421-30.
96. Jing N., Marchand, C., Liu, J., Mitra, R., Hogan, M.E., and Pommier, Y., *Mechanism of inhibition of HIV-1 integrase by G-tetrad-forming oligonucleotides in vitro.* J Biol Chem, 2000. **275**(28): p. 21460-7.
97. Cherepanov P., Este, J.A., Rando, R.F., Ojwang, J.O., Reekmans, G., Steinfeld, R., David, G., De Clercq, E., and Debyser, Z., *Mode of interaction of G-quartets with the integrase of human immunodeficiency virus type 1.* Mol Pharmacol, 1997. **52**(5): p. 771-80.
98. Rovnyak D., Baldus, M., Wu, G., Hud, N.V., Feigon, J., and Griffin, R.G., *Localization of Na-23(+) in a DNA quadruplex by high-field solid-state NMR.* Journal of the American Chemical Society, 2000. **122**(46): p. 11423-11429.
99. Wong A., Ida, R., and Wu, G., *Direct NMR detection of the "invisible" alkali metal cations tightly bound to G-quadruplex structures.* Biochemical and Biophysical Research Communications, 2005. **337**(1): p. 363-366.
100. Wong A. and Wu, G., *Selective binding of monovalent cations to the stacking G-quartet structure formed by guanosine 5'-monophosphate: a solid-state NMR study.* J Am Chem Soc, 2003. **125**(45): p. 13895-905.
101. Wu G., Wong, A., Gan, Z., and Davis, J.T., *Direct detection of potassium cations bound to G-quadruplex structures by solid-state ^{39}K NMR at 19.6 T.* J Am Chem Soc, 2003. **125**(24): p. 7182-3.
102. Eisenman G. and Horn, R., *Ionic selectivity revisited: The role of kinetic and equilibrium processes in ion permeation through channels.* Journal of Membrane Biology, 1983. **76**(3): p. 197-225.
103. Oka Y. and Thomas, C.A., Jr., *The cohering telomeres of Oxytricha.* Nucleic Acids Res, 1987. **15**(21): p. 8877-98.
104. Acevedo O.L., Dickinson, L.A., Macke, T.J., and Thomas, C.A., Jr., *The coherence of synthetic telomeres.* Nucleic Acids Res, 1991. **19**(12): p. 3409-19.
105. Hardin C.C., Henderson, E., Watson, T., and Prosser, J.K., *Monovalent cation induced structural transitions in telomeric DNAs: G-DNA folding intermediates.* Biochemistry, 1991. **30**(18): p. 4460-72.
106. Caceres C., Wright, G., Gouyette, C., Parkinson, G., and Subirana, J.A., *A thymine tetrad in d(TGGGGT) quadruplexes stabilized with Tl^+/Na^+ ions.* Nucleic Acids Res, 2004. **32**(3): p. 1097-102.
107. Schultze P., Hud, N.V., Smith, F.W., and Feigon, J., *The effect of sodium, potassium and ammonium ions on the conformation of the dimeric quadruplex*

formed by the Oxytricha nova telomere repeat oligonucleotide d(G(4)T(4)G(4)). Nucleic Acids Res, 1999. **27**(15): p. 3018-28.

108. Schultze P., Smith, F.W., and Feigon, J., *Refined solution structure of the dimeric quadruplex formed from the Oxytricha telomeric oligonucleotide d(GGGGTTTTGGGG).* Structure, 1994. **2**(3): p. 221-33.
109. Smith F.W. and Feigon, J., *Quadruplex structure of Oxytricha telomeric DNA oligonucleotides.* Nature, 1992. **356**(6365): p. 164-8.
110. Smith F.W. and Feigon, J., *Strand orientation in the DNA quadruplex formed from the Oxytricha telomere repeat oligonucleotide d($G_4T_4G_4$) in solution.* Biochemistry, 1993. **32**(33): p. 8682-92.
111. Haider S., Parkinson, G.N., and Neidle, S., *Crystal structure of the potassium form of an Oxytricha nova G-quadruplex.* J Mol Biol, 2002. **320**(2): p. 189-200.
112. Horvath M.P. and Schultz, S.C., *DNA G-quartets in a 1.86 A resolution structure of an Oxytricha nova telomeric protein-DNA complex.* J Mol Biol, 2001. **310**(2): p. 367-77.
113. Ban N., Nissen, P., Hansen, J., Moore, P.B., and Steitz, T.A., *The complete atomic structure of the large ribosomal subunit at 2.4 A resolution.* Science, 2000. **289**(5481): p. 905-20.
114. Adams P.L., Stahley, M.R., Gill, M.L., Kosek, A.B., Wang, J., and Strobel, S.A., *Crystal structure of a group I intron splicing intermediate.* Rna, 2004. **10**(12): p. 1867-87.
115. Li J., Correia, J.J., Wang, L., Trent, J.O., and Chaires, J.B., *Not so crystal clear: the structure of the human telomere G-quadruplex in solution differs from that present in a crystal.* Nucl. Acids Res., 2005. **33**(14): p. 4649-4659.
116. Kang C., Zhang, X., Ratliff, R., Moyzis, R., and Rich, A., *Crystal structure of four-stranded Oxytricha telomeric DNA.* Nature, 1992. **356**(6365): p. 126-31.
117. Gill M.L., Strobel, S.A., and Loria, J.P., *^{205}Tl NMR methods for the characterization of monovalent cation binding to nucleic acids.* J Am Chem Soc, 2005. **127**(47): p. 16723-32.
118. Piotto M., Saudek, V., and Sklenar, V., *Gradient-Tailored Excitation for Single-Quantum NMR-Spectroscopy of Aqueous-Solutions.* J. Biomol. NMR, 1992. **2**(6): p. 661-665.
119. Brown S.C., Weber, P.L., and Mueller, L., *Toward Complete H-1-NMR Spectra in Proteins.* J. Magn. Reson., 1988. **77**(1): p. 166-169.
120. Altieri A.S. and Byrd, R.A., *Randomization Approach to Water Suppression in Multidimensional NMR Using Pulsed-Field Gradients.* J. Magn. Reson. B, 1995. **107**(3): p. 260-266.
121. Cavanagh J. and Rance, M., *Suppression of Cross-Relaxation Effects in TOCSY Spectra via a Modified Dipsi-2 Mixing Sequence.* J. Magn. Reson., 1992. **96**(3): p. 670-678.
122. Rance M., *Improved Techniques for Homonuclear Rotating-Frame and Isotropic Mixing Experiments.* J. Magn. Reson., 1987. **74**(3): p. 557-564.
123. Shaka A.J., Lee, C.J., and Pines, A., *Iterative Schemes for Bilinear Operators - Application to Spin Decoupling.* Journal of Magnetic Resonance, 1988. **77**(2): p. 274-293.

124. States D.J., Haberkorn, R.A., and Ruben, D.J., *A Two-Dimensional Nuclear Overhauser Experiment with Pure Absorption Phase in 4 Quadrants.* J. Magn. Reson., 1982. **48**(2): p. 286-292.
125. Marion D. and Wuthrich, K., *Application of Phase Sensitive Two-Dimensional Correlated Spectroscopy (Cosy) for Measurements of H-1-H-1 Spin-Spin Coupling-Constants in Proteins.* Biochem. Biophys. Res. Comm., 1983. **113**(3): p. 967-974.
126. Sklenar V., Miyashiro, H., Zon, G., Miles, H.T., and Bax, A., *Assignment of the ^{31}P and ^{1}H resonances in oligonucleotides by two-dimensional NMR spectroscopy.* FEBS Lett, 1986. **208**(1): p. 94-8.
127. Celda B., Widmer, H., Leupin, W., Chazin, W.J., Denny, W.A., and Wuthrich, K., *Conformational studies of d-(AAAAATTTTT)$_2$ using constraints from nuclear overhauser effects and from quantitative analysis of the cross-peak fine structures in two-dimensional ^{1}H nuclear magnetic resonance spectra.* Biochemistry, 1989. **28**(4): p. 1462-71.
128. Varani G., Aboul-ela, F., and Allain, F.H.-T., *NMR investigation of RNA structure.* Progress in Nuclear Magnetic Resonance Spectroscopy, 1996. **29**: p. 51-127.
129. Bailey D.B., Ellis, P.D., and Fee, J.A., *Cadmium-113 nuclear magnetic resonance studies of cadmium-substituted derivatives of bovine superoxide dismutase.* Biochemistry, 1980. **19**(3): p. 591-6.
130. Briggs R.W. and Hinton, J.F., *Thallium-205 and proton nuclear magnetic resonance investigation of the complexation of thallium by the ionophores monensin and nigericin.* Biochemistry, 1978. **17**(25): p. 5576-82.
131. Forsen S. and Hoffman, R.A., *Exchange Rates by Nuclear Magnetic Multiple Resonance .3. Exchange Reactions in Systems with Several Nonequivalent Sites.* J. Chem. Phys., 1964. **40**(5): p. 1189-1196.
132. Forsen S. and Hoffman, R.A., *Study of Moderately Rapid Chemical Exchange Reactions by Means of Nuclear Magnetic Double Resonance.* J. Chem. Phys., 1963. **39**(11): p. 2892-2901.
133. Campbell I.D., Dobson, C.M., and Ratcliffe, R.G., *Fourier-Transform Proton NMR in H_2O - Method for Measuring Exchange and Relaxation Rates.* J. Magn. Reson., 1977. **27**(3): p. 455-463.
134. Delaglio F., Grzesiak, S., Vuister, G., Zhu, G., Pfeifer, J., and Bax, A., *NMRPipe: a multidimensional spectral processing system based on UNIX pipes.* J. Biomol. NMR, 1995. **6**: p. 277-293.
135. Levenberg K., *A Method for the Solution of Certain Problems in Least Squares.* Quart. Appl. Math., 1944. **2**: p. 164-168.
136. Marquardt D.W., *An Algorithm for Least-Squares Estimation of Nonlinear Parameters.* Journal of the Society for Industrial and Applied Mathematics, 1963. **11**(2): p. 431-441.
137. Gill P.E., Murray, W., and Wright, M.H., *Practical optimization*. 1981, London ; New York: Academic Press. xvi, 401 p.
138. Dingley A.J., Peterson, R.D., Grzesiek, S., and Feigon, J., *Characterization of the cation and temperature dependence of DNA quadruplex hydrogen bond*

properties using high-resolution NMR. Journal of the American Chemical Society, 2005. **127**(41): p. 14466-14472.

139. Howerton S.B., Sines, C.C., VanDerveer, D., and Williams, L.D., *Locating monovalent cations in the grooves of B-DNA.* Biochemistry, 2001. **40**(34): p. 10023-31.
140. Moulaei T., Maehigashi, T., Lountos, G.T., Komeda, S., Watkins, D., Stone, M.P., Marky, L.A., Li, J.S., Gold, B., and Williams, L.D., *Structure of B-DNA with cations tethered in the major groove.* Biochemistry, 2005. **44**(20): p. 7458-68.
141. Otwinowski Z., Minor, W., and Charles W. Carter, Jr., *[20] Processing of X-ray diffraction data collected in oscillation mode*, in *Methods in Enzymology*. 1997, Academic Press. p. 307-326.
142. Murshudov G.N., Vagin, A.A., and Dodson, E.J., *Refinement of macromolecular structures by the maximum-likelihood method.* Acta Crystallogr D Biol Crystallogr, 1997. **53**(Pt 3): p. 240-55.
143. Murshudov G.N., Vagin, A.A., Lebedev, A., Wilson, K.S., and Dodson, E.J., *Efficient anisotropic refinement of macromolecular structures using FFT.* Acta Crystallogr D Biol Crystallogr, 1999. **55**(Pt 1): p. 247-55.
144. Pannu N.S., Murshudov, G.N., Dodson, E.J., and Read, R.J., *Incorporation of prior phase information strengthens maximum-likelihood structure refinement.* Acta Crystallogr D Biol Crystallogr, 1998. **54**(Pt 6 Pt 2): p. 1285-94.
145. Winn M.D., Isupov, M.N., and Murshudov, G.N., *Use of TLS parameters to model anisotropic displacements in macromolecular refinement.* Acta Crystallogr D Biol Crystallogr, 2001. **57**(Pt 1): p. 122-33.
146. *The CCP4 suite: programs for protein crystallography.* Acta Crystallogr D Biol Crystallogr, 1994. **50**(Pt 5): p. 760-3.
147. Emsley P. and Cowtan, K., *Coot: model-building tools for molecular graphics.* Acta Crystallogr D Biol Crystallogr, 2004. **60**(Pt 12 Pt 1): p. 2126-32.
148. Cavanagh J., *Protein NMR spectroscopy : principles and practice*. 1996, San Diego: Academic Press. xxiii, 587 p.
149. Zerbe O., Pountney, D.L., Vonphilipsborn, W., and Vasak, M., *Vicinal Cd-113,(H-1)-H-Beta-Cysteine Coupling in Cd-Substituted Metalloproteins Follows a Karplus-Type Dependence.* Journal of the American Chemical Society, 1994. **116**(1): p. 377-378.
150. Hoult D.I. and Richards, R.E., *Critical Factors in Design of Sensitive High-Resolution Nuclear Magnetic-Resonance Spectrometers.* Proc. Royal Soc. of Lond. Ser. A, 1975. **344**(1638): p. 311-340.
151. Farrow N.A., Muhandiram, R., Singer, A.U., Pascal, S.M., Kay, C.M., Gish, G., Shoelson, S.E., Pawson, T., Forman-Kay, J.D., and Kay, L.E., *Backbone dynamics of a free and phosphopeptide-complexed Src homology 2 domain studied by ^{15}N NMR relaxation.* Biochemistry, 1994. **33**(19): p. 5984-6003.
152. Palmer A.G., 3rd, Fairbrother, W.J., Cavanagh, J., Wright, P.E., and Rance, M., *Improved resolution in three-dimensional constant-time triple resonance NMR spectroscopy of proteins.* J Biomol NMR, 1992. **2**(1): p. 103-8.
153. Goddard T. and Kneller, D.G., SPARKY 3, University of California, San Francisco.

154. Frisch M.J., Trucks, G.W., Schlegel, H.B., Scuseria, G.E., Robb, M.A., Cheeseman, J.R., Montgomery, J., J. A., Vreven, T., Kudin, K.N., Burant, J.C., Millam, J.M., Iyengar, S.S., Tomasi, J., Barone, V., Mennucci, B., Cossi, M., Scalmani, G., Rega, N., Petersson, G.A., Nakatsuji, H., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Klene, M., Li, X., Knox, J.E., Hratchian, H.P., Cross, J.B., Bakken, V., Adamo, C., Jaramillo, J., Gomperts, R., Stratmann, R.E., Yazyev, O., Austin, A.J., Cammi, R., Pomelli, C., Ochterski, J.W., Ayala, P.Y., Morokuma, K., Voth, G.A., Salvador, P., Dannenberg, J.J., Zakrzewski, V.G., Dapprich, S., Daniels, A.D., Strain, M.C., Farkas, O., Malick, D.K., Rabuck, A.D., Raghavachari, K., Foresman, J.B., Ortiz, J.V., Cui, Q., Baboul, A.G., Clifford, S., Cioslowski, J., Stefanov, B.B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Martin, R.L., Fox, D.J., Keith, T., Al-Laham, M.A., Peng, C.Y., Nanayakkara, A., Challacombe, M., Gill, P.M.W., Johnson, B., Chen, W., Wong, M.W., Gonzalez, C., and Pople, J.A., *Gaussian 03, Revision B.04*. 2003, Pittsburgh, PA: Gaussian, Inc.

155. Bryce D.L., Boisbouvier, J., and Bax, A., *Experimental and theoretical determination of nucleic acid magnetic susceptibility: Importance for the study of dynamics by field-induced residual dipolar couplings.* J. Am. Chem. Soc., 2004. **126**: p. 10820-10821.

156. Sklenar V., Piotto, M., Leppik, R., and Saudek, V., *Gradient-Tailored Water Suppression for H-1-N-15 HSQC Experiments Optimized to Retain Full Sensitivity.* Journal of Magnetic Resonance Series A, 1993. **102**(2): p. 241-245.

157. Kung H.C., Wang, K.Y., Goljer, I., and Bolton, P.H., *Magnetic alignment of duplex and quadruplex DNAs.* J. Magn. Reson. B, 1995. **109**: p. 323-325.

158. Lisicki M.A., Mishra, P.K., Bothner-By, A.A., and Lindsey, J.S., *Solution conformation of a porphyrin-quinone cage molecule determined by dipolar magnetic field effects in ultra-high-field NMR.* J. Phys. Chem., 1988. **92**: p. 3400-3403.

159. Grzesiek S., Cordier, F., and Dingley, A.J., *Scalar couplings across hydrogen bonds.* Methods Enzymol, 2001. **338**: p. 111-33.

160. Dingley A.J., Masse, J.E., Feigon, J., and Grzesiek, S., *Characterization of the hydrogen bond network in guanosine quartets by internucleotide 3hJ(NC)' and 2hJ(NN) scalar couplings.* J Biomol NMR, 2000. **16**(4): p. 279-89.

161. Liu A.Z., Majumdar, A., Hu, W.D., Kettani, A., Skripkin, E., and Patel, D.J., *NMR detection of N-H center dot center dot center dot O=C hydrogen bonds in C-13,N-15-labeled nucleic acids.* Journal of the American Chemical Society, 2000. **122**(13): p. 3206-3210.

162. Lee A.G., *The chemistry of thallium*. Topics in inorganic and general chemistry. 1971, Amsterdam, New York,: Elsevier Pub. Co. 336 p.

163. Agrawal R.K., Linde, J., Sengupta, J., Nierhaus, K.H., and Frank, J., *Localization of L11 protein on the ribosome and elucidation of its involvement in EF-G-dependent translocation.* J Mol Biol, 2001. **311**(4): p. 777-87.

164. Moazed D., Robertson, J.M., and Noller, H.F., *Interaction of elongation factors EF-G and EF-Tu with a conserved loop in 23S RNA.* Nature, 1988. **334**(6180): p. 362-4.

165. Ryan P.C., Lu, M., and Draper, D.E., *Recognition of the highly conserved GTPase center of 23 S ribosomal RNA by ribosomal protein L11 and the antibiotic thiostrepton.* J Mol Biol, 1991. **221**(4): p. 1257-68.
166. Thompson J., Musters, W., Cundliffe, E., and Dahlberg, A.E., *Replacement of the L11 binding region within E.coli 23 S ribosomal RNA with its homologue from yeast: in vivo and in vitro analysis of hybrid ribosomes altered in the GTPase centre.* Embo J, 1993. **12**(4): p. 1499-504.
167. Lu M. and Draper, D.E., *On the role of rRNA tertiary structure in recognition of ribosomal protein L11 and thiostrepton.* Nucleic Acids Res, 1995. **23**(17): p. 3426-33.
168. Thompson J., Cundliffe, E., and Stark, M., *Binding of thiostrepton to a complex of 23-S rRNA with ribosomal protein L11.* Eur J Biochem, 1979. **98**(1): p. 261-5.
169. Xing Y. and Draper, D.E., *Cooperative interactions of RNA and thiostrepton antibiotic with two domains of ribosomal protein L11.* Biochemistry, 1996. **35**(5): p. 1581-8.
170. Xing Y. and Draper, D.E., *Stabilization of a ribosomal RNA tertiary structure by ribosomal protein L11.* J Mol Biol, 1995. **249**(2): p. 319-31.
171. Wang Y.X., Lu, M., and Draper, D.E., *Specific ammonium ion requirement for functional ribosomal RNA tertiary structure.* Biochemistry, 1993. **32**(46): p. 12279-82.
172. Grodberg J. and Dunn, J.J., *ompT encodes the Escherichia coli outer membrane protease that cleaves T7 RNA polymerase during purification.* J Bacteriol, 1988. **170**(3): p. 1245-53.
173. Reinhold-Hurek B. and Shub, D.A., *Self-splicing introns in tRNA genes of widely divergent bacteria.* Nature, 1992. **357**(6374): p. 173-6.
174. Rangan P., Masquida, B., Westhof, E., and Woodson, S.A., *Assembly of core helices and rapid tertiary folding of a small bacterial group I ribozyme.* Proc Natl Acad Sci U S A, 2003. **100**(4): p. 1574-9.
175. Rangan P., Masquida, B., Westhof, E., and Woodson, S.A., *Architecture and folding mechanism of the Azoarcus Group I Pre-tRNA.* J Mol Biol, 2004. **339**(1): p. 41-51.
176. Balasubramanian B., Pogozelski, W.K., and Tullius, T.D., *DNA strand breaking by the hydroxyl radical is governed by the accessible surface areas of the hydrogen atoms of the DNA backbone.* Proc Natl Acad Sci U S A, 1998. **95**(17): p. 9738-43.
177. Celander D.W. and Cech, T.R., *Visualizing the higher order folding of a catalytic RNA molecule.* Science, 1991. **251**(4992): p. 401-7.
178. Latham J.A. and Cech, T.R., *Defining the inside and outside of a catalytic RNA molecule.* Science, 1989. **245**(4915): p. 276-82.
179. Cate J.H., Gooding, A.R., Podell, E., Zhou, K., Golden, B.L., Kundrot, C.E., Cech, T.R., and Doudna, J.A., *Crystal structure of a group I ribozyme domain: principles of RNA packing.* Science, 1996. **273**(5282): p. 1678-85.
180. Su L.J., Brenowitz, M., and Pyle, A.M., *An alternative route for the folding of large RNAs: apparent two-state folding by a group II intron ribozyme.* J Mol Biol, 2003. **334**(4): p. 639-52.

181. Swisher J., Duarte, C.M., Su, L.J., and Pyle, A.M., *Visualizing the solvent-inaccessible core of a group II intron ribozyme.* Embo J, 2001. **20**(8): p. 2051-61.
182. Arabshahi A. and Frey, P.A., *A Simplified Procedure for Synthesizing Nucleoside 1-Thiotriphosphates: dATPαS, dGTPαS, UTPαS, and dTTPαS.* Biochemical and Biophysical Research Communications, 1994. **204**(1): p. 150-155.
183. Ryder S.P. and Strobel, S.A., *Nucleotide analog interference mapping.* Methods, 1999. **18**(1): p. 38-50.
184. Ferre-D'Amare A.R. and Doudna, J.A., *Crystallization and structure determination of a hepatitis delta virus ribozyme: use of the RNA-binding protein U1A as a crystallization module.* J Mol Biol, 2000. **295**(3): p. 541-56.
185. Lee B. and Richards, F.M., *The interpretation of protein structures: estimation of static accessibility.* J Mol Biol, 1971. **55**(3): p. 379-400.

10 Appendix

10.1 Appendix 1 NMR Data

10.1.1 CNS annealing script.

```
{* Changes 2004-08-25 M. L. Gill *}
{* Increased "md.torsion.maxlength from 50 to 100 because the structure is a nucleic acid
*}
{* Changes 2004-08-26 M. L. Gill *}
{* Set complete cross validation variables "flg.cv.coup" and "flg.cv.cdih" to false
because *}
{* I haven't yet included this data.  Change to true when these data sets are added *}

{+ file: anneal_cv.inp +}
{+ directory: nmr_calc +}
{+ description: dynamical annealing with NOEs, coupling constants,
                chemical shift restraints starting from extended
                strands or pre-folded structures.  Includes
                complete cross-validation for NOEs,3-bond
                j-coupling, and dihedral angle restraints. +}
{+ authors: Gregory Warren, Michael Nilges, John Kuszewski,
           Marius Clore and Axel Brunger +}
{+ copyright: Yale University +}

{+ reference: Clore GM, Gronenborn AM, Tjandra N, Direct structure refinement
              against residual dipolar couplings in the presence of rhombicity
              of unknown magnitude., J. Magn. Reson., 131, In press, (1998) +}
{+ reference: Clore GM, Gronenborn AM, Bax A, A robust method for determining
              the magnitude of the fully asymmetric alignment tensor of
              oriented macromolecules in the absence of structural
              information., J. Magn. Reson., In press (1998) +}
{+ reference: Garrett DS, Kuszewski J, Hancock TJ, Lodi PJ, Vuister GW,
              Gronenborn AM, Clore GM, The impact of direct refinement against
              three-bond HN-C alpha H coupling constants on protein structure
              determination by NMR., J. Magn. Reson. Ser. B, 104(1),
              99-103, (1994) May +}
{+ reference: Kuszewski J, Qin J, Gronenborn AM, Clore GM, The impact of direct
              refinement against 13C alpha and 13C beta chemical shifts on
              protein structure determination by NMR., J. Magn. Reson. Ser. B,
              106(1), 92-6, (1995) Jan +}
{+ reference: Kuszewski J, Gronenborn AM, Clore GM, The impact of direct
              refinement against proton chemical shifts on protein structure
              determination by NMR., J. Magn. Reson. Ser. B, 107(3), 293-7,
              (1995) Jun +}
{+ reference: Kuszewski J, Gronenborn AM, Clore GM, A potential involving
              multiple proton chemical-shift restraints for
              nonstereospecifically assigned methyl and methylene protons.
              J. Magn. Reson. Ser. B, 112(1), 79-81, (1996) Jul. +}
{+ reference: Nilges M, Gronenborn AM, Brunger AT, Clore GM, Determination
              of three-dimensional structures of proteins by simulated
              annealing with interproton distance restraints: application
              to crambin, potato carboxypeptidase inhibitor and barley
              serine proteinase inhibitor 2. Protein Engineering 2,
              27-38, (1988) +}
{+ reference: Nilges M, Clore GM, Gronenborn AM,  Determination of
              three-dimensional structures of proteins from interproton
              distance data by dynamical simulated annealing from a random
              array of atoms. FEBS LEtt. 239, 129-136. (1988) +}
{+ reference: Rice LM, Brunger AT, Torsion Angle Dynamics: Reduced Variable
              Conformational Sampling Enhances Crystallographic Structure
              Refinement., Proteins, 19, 277-290 (1994) +}
{+ reference: Stein EG, Rice LM, Brunger AT, Torsion angle molecular
              dynamics: a new efficient tool for NMR structure calculation.,
              J. Mag. Res. Ser. B 124, 154-164 (1997) +}
```

```
{+ reference: Tjandra N, Garrett DS, Gronenborn AM, Bax A, Clore GM, Defining
              long range order in NMR structure determination from the
              dependence of heteronuclear relaxation times on rotational
              diffusion anisotropy. Nature Struct. Biol., 4(6), 443-9,
              (1997) June +}
{+ reference: Tjandra N, Omichinski JG, Gronenborn AM, Clore GM, Bax A, Use of
              dipolar 1H-15N and 1H-13C couplings in the structure
              determination of magnetically oriented macromolecules in
              solution. Nature Struct. Biol., 4(9), 732-8, (1997) Sept +}

! Data taken from: Qin J, Clore GM, Kennedy WP, Kuszewski J, Gronenborn AM,
!                  The solution structure of human thioredoxin complexed with
!                  its target from Ref-1 reveals peptide chain reversal.,
!                  Structure, 4(5), 613-620, 1996 May 15.

{- Guidelines for using this file:
   - all strings must be quoted by double-quotes
   - logical variables (true/false) are not quoted
   - do not remove any evaluate statements from the file -}

{- begin block parameter definition -} define(

{======================= molecular structure =========================}

{* parameter file(s) *}
{===>} par.1="./dna-rna-allatom.param";
{===>} par.2="";
{===>} par.3="";
{===>} par.4="";
{===>} par.5="";

{* structure file(s) *}
{===>} struct.1="1JRN2_cns.mtf";
{===>} struct.2="";
{===>} struct.3="";
{===>} struct.4="";
{===>} struct.5="";

{* input coordinate file(s) *}
{===>} pdb.in.file.1="G4T4G4_T10_mean.pdb";
{===>} pdb.in.file.2="";
{===>} pdb.in.file.3="";

{========================== atom selection ===========================}

{* input "backbone" selection criteria for average structure generation *}
{* for protein      (name n or name ca or name c)
   for nucleic acid (name O5' or name C5' or name C4' or name C3'
                     or name O3' or name P) *}
{===>} pdb.atom.select=(name O5' or name C5' or name C4' or name C3' or name O3' or name
P);

{====================== refinement parameters ========================}

{* if diffusion or susceptibility anistropy restraints are being used
   it is strongly recommended that a torsion, torsion, cartesian
   refinement scheme be used *}

{* type of molecular dynamics for hot phase *}
{+ choice: "torsion" "cartesian" +}
{===>} md.type.hot="torsion";

{* type of molecular dynamics for cool phase *}
{+ choice: "torsion" "cartesian" +}
{===>} md.type.cool="cartesian";

{* seed for random number generator *}
{* change to get different initial velocities *}
{===>} md.seed=82364;

{* select whether the number of structures will be either trial or
```

```
   accepted structures and whether to print only the trial, accepted,
   both sets of structures. *}
{+ list: The printing format is as follows:
         trial = pdb.out.name + _#.pdb , accepted = pdb.out.name + a_#.pdb +}

{* are the number of structures to be trials or accepted? *}
{+ choice: "trial" "accept" +}
{===>} flg.trial.struc="trial";
{* number of trial or accepted structures *}
{===>} pdb.end.count=100;

{* print accepted structures *}
{+ choice: true false +}
{===>} flg.print.accept=true;
{* print trial structures *}
{+ choice: true false +}
{===>} flg.print.trial=true;

{* calculate an average structure for either the trial or
   accepted structure.  If calculate accepted average is false then
   an average for the trial structures will be calculated. *}

{* calculate an average structure? *}
{+ choice: true false +}
{===>} flg.calc.ave.struct=true;
{* calculate an average structure for the accepted structures? *}
{+ choice: true false +}
{===>} flg.calc.ave.accpt=false;
{* minimize average coordinates? *}
{+ choice: true false +}
{===>} flg.min.ave.coor=false;

{=================== torsion dynamics parameters ====================}

{* maximum unbranched chain length *}
{* increase for long stretches of polyalanine or for nucleic acids *}
{===>} md.torsion.maxlength=100;

{* maximum number of distinct bodies *}
{===>} md.torsion.maxtree=4;

{* maximum number of bonds to an atom *}
{===>} md.torsion.maxbond=6;


{========== parameters for high temperature annealing stage ==========}

{* temperature (proteins: 50000, dna/rna: 20000) *}
{===>} md.hot.temp=20000;
{* number of steps (proteins: 1000, dna/rna: 4000) *}
{===>} md.hot.step=4000;
{* scale factor to reduce van der Waals (repel) energy term *}
{===>} md.hot.vdw=0.1;
{* scale factor for NOE energy term *}
{===>} md.hot.noe=150;
{* scale factor for dihedral angle energy term (proteins: 100, dna/rna: 5) *}
{===>} md.hot.cdih=5;
{* molecular dynamics timestep *}
{===>} md.hot.ss=0.015;

{======== parameters for the first slow-cool annealing stage =========}

{* temperature (cartesian: 1000, torsion: [proteins: 50000, dna/rna: 20000]) *}
{===>} md.cool.temp=1000;
{* number of steps *}
{===>} md.cool.step=1000;
{* scale factor for final van der Waals (repel) energy term
   (cartesian: 4.0, torsion: 1.0) *}
{===>} md.cool.vdw=4.0;
{* scale factor for NOE energy term *}
{===>} md.cool.noe=150;
```

```
{* scale factor for dihedral angle energy term *}
{===>} md.cool.cdih=200;
{* molecular dynamics timestep (cartesian: 0.005, torsion: 0.015) *}
{===>} md.cool.ss=0.005;
{* slow-cool annealing temperature step (cartesian: 25, torsion: 250) *}
{===>} md.cool.tmpstp=25;

{========= parameters for a second slow-cool annealing stage ==========}
{* cartesian slow-cooling annealing stage to be used only with torsion
   slow-cool annealing stage *}
{* this stage is only necessary when the macromolecule is a protein
   greater than 160 residues or in some cases for nucleic acids *}

{* use cartesian cooling stage? *}
{+ choice: true false +}
{===>} md.cart.flag=false;
{* temperature *}
{===>} md.cart.temp=3000;
{* number of steps *}
{===>} md.cart.step=3000;
{* scale factor for initial van der Waals (repel) energy term *}
{===>} md.cart.vdw.init=1.0;
{* scale factor for final van der Waals (repel) energy term *}
{===>} md.cart.vdw.finl=4.0;
{* scale factor for NOE energy term *}
{===>} md.cart.noe=150;
{* scale factor for dihedral angle energy term *}
{===>} md.cart.cdih=200;
{* molecular dynamics timestep *}
{===>} md.cart.ss=0.005;
{* slow-cool annealing temperature step *}
{===>} md.cart.tmpstp=25;

{=============== parameters for final minimization stage ==============}

{* scale factor for NOE energy term *}
{===>} md.pow.noe=75;
{* scale factor for dihedral angle energy term *}
{===>} md.pow.cdih=400;
{* number of minimization steps *}
{===>} md.pow.step=200;
{* number of cycles of minimization *}
{===>} md.pow.cycl=10;

{==================== complete cross validation =======================}

{* would you like to perform complete cross validation? *}
{+ choice: true false +}
{===>} flg.cv.flag=true;
{* the number of data partitions *}
{===>} nmr.cv.numpart=10;
{* for NOEs excluding h-bonds? *}
{+ choice: true false +}
{===>} flg.cv.noe=true;
{* for 3-bond J-coupling? *}
{+ choice: true false +}
{===>} flg.cv.coup=false;
{* for dihedral restraints? *}
{+ choice: true false +}
{===>} flg.cv.cdih=false;

{============================= noe data ===============================}

{- Important - if you do not have a particular data set then
   set the file name to null ("") -}

{* NOE distance restraints files. *}

{* restraint set 1 file *}
{===>} nmr.noe.file.1="std_fixT6_2.tbl";
{* restraint set 2 file *}
```

```
{===>} nmr.noe.file.2="cold_2.tbl";
{* restraint set 3 file *}
{===>} nmr.noe.file.3="methyl_fixT6_2.tbl";
{* restraint set 4 file *}
{===>} nmr.noe.file.4="exch_2.tbl";
{* restraint set 5 file *}
{===>} nmr.noe.file.5="";

{* NOE averaging modes *}

{* restraint set 1 *}
{+ choice: "sum" "cent" "R-6" "R-3" "symm" +}
{===>} nmr.noe.ave.mode.1="sum";
{* restraint set 2 *}
{+ choice: "sum" "cent" "R-6" "R-3" "symm" +}
{===>} nmr.noe.ave.mode.2="sum";
{* restraint set 3 *}
{+ choice: "sum" "cent" "R-6" "R-3" "symm" +}
{===>} nmr.noe.ave.mode.3="sum";
{* restraint set 4 *}
{+ choice: "sum" "cent" "R-6" "R-3" "symm" +}
{===>} nmr.noe.ave.mode.4="sum";
{* restraint set 5 *}
{+ choice: "sum" "cent" "R-6" "R-3" "symm" +}
{===>} nmr.noe.ave.mode.5="sum";

{======================== hydrogen bond data ==========================}

{* hydrogen-bond distance restraints file. *}
{===>} nmr.noe.hbnd.file="hbonds.tbl";

{* enter hydrogen-bond distance averaging mode *}
{+ choice: "sum" "cent" "R-6" "R-3" "symm" +}
{===>} nmr.noe.ave.mode.hbnd="sum";

{======================== 3-bond J-coupling data ========================}
{* the default setup is for the phi dihedral *}

{* Class 1 *}

{* 3-bond J-coupling non-glycine restraints file *}
{===>} nmr.jcoup.file.1="";
{* 3-bond J-coupling non-glycine potential *}
{+ choice: "harmonic" "square" "multiple" +}
{===>} nmr.jcoup.pot.1="harmonic";
{* 3-bond J-coupling non-glycine force value *}
{===>} nmr.jcoup.force.1.1=1;
{* 3-bond j-coupling multiple class force second value *}
{===>} nmr.jcoup.force.2.1=0;
{* 3-bond j-coupling Karplus coefficients *}
{* the default values are for phi *}
{* OLD values 6.98, -1.38, 1.72, -60.0 *}
{* MLG: changed to values for nucleic acids *}
{===>} nmr.jcoup.coef.1.1=5.00;
{===>} nmr.jcoup.coef.2.1=-0.54;
{===>} nmr.jcoup.coef.3.1=4.31;
{===>} nmr.jcoup.coef.4.1=0.0;

{* Class 2 *}

{* 3-bond j-coupling glycine restraints files *}
{* The potential for the glycine class must be multiple *}
{===>} nmr.jcoup.file.2="";
{* 3-bond J-coupling non-glycine potential *}
{+ choice: "harmonic" "square" "multiple" +}
{===>} nmr.jcoup.pot.2="multiple";
{* 3-bond J-coupling first force value *}
{===>} nmr.jcoup.force.1.2=1;
{* 3-bond j-coupling glycine or multiple force second value *}
{===>} nmr.jcoup.force.2.2=0;
{* 3-bond j-coupling Karplus coefficients *}
```

```
{* the default values are for glycine phi *}
{===>} nmr.jcoup.coef.1.2=6.98;
{===>} nmr.jcoup.coef.2.2=-1.38;
{===>} nmr.jcoup.coef.3.2=1.72;
{===>} nmr.jcoup.coef.4.2=0.0;

{================ 1-bond heteronuclear J-coupling data ================}

{* Class 1 *}

{* 1-bond heteronuclear j-coupling file *}
{===>} nmr.oneb.file.1="";
{* 1-bond heteronuclear j-coupling potential *}
{+ choice: "harmonic" "square" +}
{===>} nmr.oneb.pot.1="harmonic";
{* 1-bond heteronuclear j-coupling force value *}
{===>} nmr.oneb.force.1=1.0;

{=============== alpha/beta carbon chemical shift data ================}

{* Class 1 *}

{* carbon, alpha and beta, chemical shift restraints file *}
{===>} nmr.carb.file.1="";
{* carbon, alpha and beta, chemical shift restraint potential *}
{+ choice: "harmonic" "square" +}
{===>} nmr.carb.pot.1="harmonic";
{* carbon, alpha and beta, chemical shift restraint force value *}
{===>} nmr.carb.force.1=0.5;

{===================== proton chemical shift data =====================}

{* Class 1 *}

{* class 1 proton chemical shift restraints file *}
{===>} nmr.prot.file.1="";
{* class 1 proton chemical shift potential *}
{+ choice: "harmonic" "square" "multiple" +}
{===>} nmr.prot.pot.1="harmonic";
{* class 1 proton chemical shift force value *}
{===>} nmr.prot.force.1.1=7.5;
{* 2nd class 1 proton chemical shift force value for multi *}
{===>} nmr.prot.force.2.1=0;
{* class 1 proton chemical shift violation cutoff threshold *}
{===>} nmr.prot.thresh.1=0.3;

{* Class 2 *}

{* class 2 proton chemical shift restraints file *}
{===>} nmr.prot.file.2="";
{* class 2 proton chemical shift potential *}
{+ choice: "harmonic" "square" "multiple" +}
{===>} nmr.prot.pot.2="harmonic";
{* class 2 proton chemical shift force value *}
{===>} nmr.prot.force.1.2=7.5;
{* 2nd class 2 proton chemical shift force value for multi *}
{===>} nmr.prot.force.2.2=0;
{* class 2 proton chemical shift violation cutoff threshold *}
{===>} nmr.prot.thresh.2=0.3;

{* Class 3 *}

{* class 3 proton chemical shift restraints file *}
{===>} nmr.prot.file.3="";
{* class 3 proton chemical shift potential *}
{+ choice: "harmonic" "square" "multiple" +}
{===>} nmr.prot.pot.3="harmonic";
{* class 3 proton chemical shift force value *}
{===>} nmr.prot.force.1.3=7.5;
{* 2nd class 3 proton chemical shift force value for multi *}
{===>} nmr.prot.force.2.3=0;
```

```
{* class 3 proton chemical shift violation cutoff threshold *}
{===>} nmr.prot.thresh.3=0.3;

{* Class 4 *}

{* class 4 proton chemical shift restraints file *}
{===>} nmr.prot.file.4="";
{* class 4 proton chemical shift potential *}
{+ choice: "harmonic" "square" "multiple" +}
{===>} nmr.prot.pot.4="multiple";
{* class 4 proton chemical shift force value *}
{===>} nmr.prot.force.1.4=7.5;
{* 2nd class 4 proton chemical shift force value for multi *}
{===>} nmr.prot.force.2.4=0;
{* class 4 proton chemical shift violation cutoff threshold *}
{===>} nmr.prot.thresh.4=0.3;

{================ diffusion anisotropy restraint data =================}

{* fixed or harmonically restrained external axis *}
{+ choice: "fixed" "harm" +}
{===>} nmr.dani.axis="harm";

{* Class 1 *}

{* diffusion anisotropy restraints file *}
{===>} nmr.dani.file.1="";
{* diffusion anisotropy potential *}
{+ choice: "harmonic" "square" +}
{===>} nmr.dani.pot.1="harmonic";
{* diffusion anisotropy initial force value *}
{===>} nmr.dani.force.init.1=0.01;
{* diffusion anisotropy final force value *}
{===>} nmr.dani.force.finl.1=1.0;
{* diffusion anisotropy coefficients *}
{* coef: <Tc> <anis> <rhombicity> <wh> <wn> *}

{* Tc = 1/2(Dx+Dy+Dz) in <ns> *}
{===>} nmr.dani.coef.1.1=13.1;
{* anis = Dz/0.5*(Dx+Dy) *}
{===>} nmr.dani.coef.2.1=2.1;
{* rhombicity = 1.5*(Dy-Dx)/(Dz-0.5*(Dy+Dx)) *}
{===>} nmr.dani.coef.3.1=0.0;
{* wH in <MHz> *}
{===>} nmr.dani.coef.4.1=600.13;
{* wN in <MHz> *}
{===>} nmr.dani.coef.5.1=60.82;

{============= susceptability anisotropy restraint data ===============}

{* fixed or harmonically restrained external axis *}
{+ choice: "fixed" "harm" +}
{===>} nmr.sani.axis="harm";

{* Class 1 *}

{* susceptability anisotropy restraints file *}
{===>} nmr.sani.file.1="";
{* susceptability anisotropy potential *}
{+ choice: "harmonic" "square" +}
{===>} nmr.sani.pot.1="harmonic";
{* susceptability anisotropy initial force value *}
{===>} nmr.sani.force.init.1=0.01;
{* susceptability anisotropy final force value *}
{===>} nmr.sani.force.finl.1=50.0;
{* susceptability anisotropy coefficients *}
{* coef: <DFS> <axial > <rhombicity>;
   a0+a1*(3*cos(theta)^2-1)+a2*(3/2)*sin(theta)^2*cos(2*phi) *}

{* DFS = a0 *}
{===>} nmr.sani.coef.1.1=-0.0601;
```

```
{* axial = a0-a1-3/2*a2 *}
{===>} nmr.sani.coef.2.1=-8.02;
{* rhombicity = a2/a1 *}
{===>} nmr.sani.coef.3.1=0.4;

{======================== other restraint data ========================}

{* dihedral angle restraints file *}
{* Note: the restraint file MUST NOT contain restraints
         dihedral or end *}
{===>} nmr.cdih.file="dihedral.tbl";

{* DNA-RNA base planarity restraints file *}
{* Note: include weights as $pscale in the restraint file *}
{===>} nmr.plan.file="planarity3.tbl";
{* input planarity scale factor - this will be written into $pscale *}
{===>} nmr.plan.scale=5;

{* NCS-restraints file *}
{* example is in inputs/xtal_data/eg1_ncs_restrain.dat *}
{===>} nmr.ncs.file="ncs3.tbl";

{======================== input/output files ==========================}

{* base name for output coordinate files *}
{===>} pdb.out.name="G4T4G4";

{===============================================================================}
{         things below this line do not normally need to be changed         }
{         except for the torsion angle topology setup if you have           }
{         molecules other than protein or nucleic acid                      }
{===============================================================================}
flg.dgsa.flag=false;

 ) {- end block parameter definition -}

checkversion 1.1

evaluate ($log_level=quiet)

structure
   if  (&struct.1 # "") then
      @@&struct.1
   end if
   if  (&struct.2 # "") then
      @@&struct.2
   end if
   if  (&struct.3 # "") then
      @@&struct.3
   end if
   if  (&struct.4 # "") then
      @@&struct.4
   end if
   if  (&struct.5 # "") then
      @@&struct.5
   end if
end

if ( &BLANK%pdb.in.file.1 = false ) then
   coor @@&pdb.in.file.1
end if
if ( &BLANK%pdb.in.file.2 = false ) then
   coor @@&pdb.in.file.2
end if
if ( &BLANK%pdb.in.file.3 = false ) then
   coor @@&pdb.in.file.3
end if

parameter
   if (&par.1 # "") then
      @@&par.1
```

```
   end if
   if (&par.2 # "") then
      @@&par.2
   end if
   if (&par.3 # "") then
      @@&par.3
   end if
   if (&par.4 # "") then
      @@&par.4
   end if
   if (&par.5 # "") then
      @@&par.5
   end if
end

if ( $log_level = verbose ) then
  set message=normal echo=on end
else
  set message=off echo=off end
end if

parameter
   nbonds
      repel=0.80
      rexp=2 irexp=2 rcon=1.
      nbxmod=3
      wmin=0.01
      cutnb=6.0 ctonnb=2.99 ctofnb=3.
      tolerance=1.5
   end
end

{- Read experimental data -}

   @CNS_NMRMODULE:readdata ( nmr=&nmr;
                             flag=&flg;
                             output=$nmr; )

{- Read and store the number of NMR restraints -}

   @CNS_NMRMODULE:restraintnumber ( num=$num; )

{- Set mass values -}

do (fbeta=10) (all)
do (mass=100) (all)

evaluate ($nmr.trial.count = 0)    {- Initialize current structure number   -}
evaluate ($nmr.accept.count = 0)    {- Initialize number accepted            -}
evaluate ($nmr.counter        = 0)
evaluate ($nmr.prev.counter = -1)

@CNS_NMRMODULE:initave  ( ave=$ave;
                          ave2=$ave2;
                          cv=$cv;
                          ener1=$ener1;
                          ener2=$ener2;
                          flag=&flg;
                          nmr.prot=&nmr.prot; )

{- Zero the force constant of disulfide bonds. -}
parameter
   bonds ( name SG ) ( name SG ) 0. TOKEN
end

{- define a distance restraints for each disulfide bond, i.e.,
   treat it as if it were an NOE. -}
for $ss_rm_id_1 in id ( name SG ) loop STRM
  for $ss_rm_id_2 in id ( name SG and
                          bondedto ( id $ss_rm_id_1 )  ) loop STR2
    if ($ss_rm_id_1 > $ss_rm_id_2) then
```

```
      pick bond ( id $ss_rm_id_1 ) ( id $ss_rm_id_2 ) equil
      evaluate ($ss_bond=$result)
      noe
         assign ( id $ss_rm_id_1 ) ( id $ss_rm_id_2 ) $ss_bond 0.1 0.1
      end
    end if
  end loop STR2
end loop STRM

{- Count the number of residues and determine molecule type -}
identify (store9) (tag)
evaluate ($nmr.rsn.num = $SELECT)
identify (store9) ( tag and ( resn THY or resn CYT or resn GUA or
                              resn ADE or resn URI ))
evaluate ($nmr.nucl.num = $SELECT)

{- Improve geometry for torsion angle molecular dynamics -}
evaluate ($flag_tad=false)
if ( &md.type.hot = "torsion" ) then
   if ($nmr.nucl.num > 0) then
      flag exclude * include bond angl impr dihed vdw end
      minimize powell nstep=2000 drop=10.  nprint=100 end
   else
      flag exclude * include bond angl impr vdw end
      minimize powell nstep=2000 drop=10.  nprint=100 end
   end if
   evaluate ($flag_tad=true)
end if

if ( &md.type.cool="torsion") then
   evaluate ($flag_tad=true)
end if

if (&nmr.dani.axis = "harm") then
   do (harmonic=20.0) (resid 500 and name OO)
   do (harmonic=0.0) (resid 500 and name Z )
   do (harmonic=0.0) (resid 500 and name X )
   do (harmonic=0.0) (resid 500 and name Y )
   do (harmonic=0.0) (not (resid 500))
   restraints harmonic exponent=2 end
elseif (&nmr.sani.axis = "harm") then
   do (harmonic=20.0) (resid 500 and name OO)
   do (harmonic=0.0) (resid 500 and name Z )
   do (harmonic=0.0) (resid 500 and name X )
   do (harmonic=0.0) (resid 500 and name Y )
   do (harmonic=0.0) (not (resid 500))
   restraints harmonic exponent=2 end
end if

if (&flg.cv.flag=false) then
  if (&flg.cv.noe=true) then
    echo "Complete cross-validation for NOE, J-coupling and Dihedrals"
    echo "must be disabled if complete cross-validation is not used"
    abort
  elseif (&flg.cv.coup=true) then
    echo "Complete cross-validation for NOE, J-coupling and Dihedrals"
    echo "must be disabled if complete cross-validation is not used"
    abort
  elseif (&flg.cv.cdih=true) then
    echo "Complete cross-validation for NOE, J-coupling and Dihedrals"
    echo "must be disabled if complete cross-validation is not used"
    abort
  end if
end if

if (&flg.cv.flag=true) then
   evaluate ($cv.part.num=1)
   evaluate ($cvtemp = int(&pdb.end.count/&nmr.cv.numpart))
   if ($cvtemp < 1) then
      evaluate ($cvtemp = 1)
   end if
```

```
   evaluate ($pdb_end_count=&nmr.cv.numpart*$cvtemp)
else
   evaluate ($pdb_end_count=&pdb.end.count)
end if

do (refx=x) ( all )
do (refy=y) ( all )
do (refz=z) ( all )

set seed=&md.seed end

{- Begin protocol to generate structures -- loop until done -}
while ($pdb_end_count > $nmr.counter) loop main

   {- Set parameter values -}
   parameter
      nbonds
         repel=0.80
         rexp=2 irexp=2 rcon=1.
         nbxmod=3
         wmin=0.01
         cutnb=6.0 ctonnb=2.99 ctofnb=3.
         tolerance=1.5
      end
   end

   evaluate ($nmr.trial.count = $nmr.trial.count + 1)

   do (x=refx) ( all )
   do (y=refy) ( all )
   do (z=refz) ( all )

   if (&nmr.dani.axis = "fixed" ) then
      fix
         select=(resname ANI)
      end
   elseif (&nmr.sani.axis = "fixed" ) then
      fix
         select=(resname ANI)
      end
   end if

   do ( vx = maxwell(0.5) ) ( all )
   do ( vy = maxwell(0.5) ) ( all )
   do ( vz = maxwell(0.5) ) ( all )

   flags exclude *
         include bond angle dihe impr vdw
                 noe cdih coup oneb carb ncs dani
                 sani harm end

   {- repartition the data for multiple completely cross-validated
      refinements -}

   if ($nmr.prev.counter # $nmr.counter) then
     if (&flg.cv.flag=true) then
       if ($cv.part.num > &nmr.cv.numpart) then

         evaluate ($cv.part.num=1)
          @CNS_NMRMODULE:repartition ( cv=$cv;
                                       flag=&flg;
                                       nmr=&nmr; )

       else
         if (&flg.cv.noe=true) then
            noe cv = $cv.part.num end
         end if
         if (&flg.cv.coup=true) then
            coup cv = $cv.part.num end
         end if
         if (&flg.cv.cdih=true) then
```

```
            restraints dihed cv = $cv.part.num end
         end if
         evaluate ($cv.part.num=$cv.part.num+1)
       end if
     end if
   end if

   {- scaling of nmr restraint data during hot dynamics -}

   @CNS_NMRMODULE:scalehot ( md=&md;
                             nmr=&nmr;
                             input.noe.scale=&md.hot.noe;
                             input.cdih.scale=&md.hot.cdih; )

   {- Zero the force constant of disulfide bonds. -}
   parameter
      bonds ( name SG ) ( name SG ) 0. TOKEN
   end

   if ($flag_tad=true) then

      {- initialize torsion dynamics topology for this iteration -}

      dyna torsion
         topology
            maxlength=&md.torsion.maxlength
            maxtree=&md.torsion.maxtree
            maxbond=&md.torsion.maxbond
            {- All dihedrals w/ (force constant > 23) will be locked -}
            {- This keeps planar groups planar -}
            kdihmax = 23.
            @CNS_TOPPAR:torsionmdmods
         end
      end
   end if

{- High temperature dynamics -}

   if ( &md.type.hot = "torsion" ) then

      igroup
        interaction (chemical h* ) (all) weights * 1 vdw 0. elec 0. end
        interaction (not chemical h* ) (not chemical h*) weights * 1 vdw &md.hot.vdw
        end
      end

      dyna torsion
         cmperiodic=500
         vscaling = false
         tcoupling = true
         timestep = &md.hot.ss
         nstep = &md.hot.step
         nprint = 50
         temperature = &md.hot.temp
      end
   else
      evalutate ($md.hot.nstep1=int(&md.hot.step* 2. / 3. ))
      evalutate ($md.hot.nstep2=int(&md.hot.step* 1. / 3. ))
      noe asymptote * 0.1  end
      parameter  nbonds repel=1.   end end
      igroup
        interaction (chemical h* ) (all) weights * 1 vdw 0. elec 0. end
        interaction (not chemical h* ) (not chemical h*) weights * 1 angl 0.4 impr 0.1
                    vdw &md.hot.vdw end
      end

      dynamics cartesian
         cmperiodic=500
         vscaling = true
         tcoupling=false
         timestep=&md.hot.ss
```

```
         nstep=$md.hot.nstep1
         nprint=50
         temperature=&md.hot.temp
      end

      noe asymptote * 1.0  end
      igroup
        interaction (chemical h* ) (all) weights * 1 vdw 0. elec 0. end
        interaction (not chemical h* ) (not chemical h*) weights * 1 vdw &md.hot.vdw end
      end

      dynamics cartesian
         cmperiodic=500
         vscaling = true
         tcoupling=false
         timestep=&md.hot.ss
         nstep=$md.hot.nstep2
         nprint=50
         temperature=&md.hot.temp
      end

   end if

{- The first slow-cooling with torsion angle dynamics -}

   flags include plan end

   {- Increase the disulfide bond force constants to their full strength -}
   parameter
      bonds ( name SG ) ( name SG ) 1000. TOKEN
   end

   evaluate ($final_t = 0)

   evaluate ($ncycle = int((&md.cool.temp-$final_t)/&md.cool.tmpstp))
   evaluate ($nstep = int(&md.cool.step/$ncycle))

   evaluate ($ini_vdw =  &md.hot.vdw)
   evaluate ($fin_vdw =  &md.cool.vdw)
   evaluate ($vdw_step = ($fin_vdw-$ini_vdw)/$ncycle)

   if (&md.type.cool = "cartesian") then

      evaluate ($vdw_step = (&md.cool.vdw/&md.hot.vdw)^(1/$ncycle))
      evaluate ($ini_rad  = 0.9)
      evaluate ($fin_rad  = 0.8)
      evaluate ($rad_step = ($ini_rad-$fin_rad)/$ncycle)
      evaluate ($radius=    $ini_rad)

      do (vx=maxwell(&md.cool.temp)) ( all )
      do (vy=maxwell(&md.cool.temp)) ( all )
      do (vz=maxwell(&md.cool.temp)) ( all )

   end if

   {- set up nmr restraint scaling -}

   evaluate ($kdani.inter.flag=false)
   evaluate ($ksani.inter.flag=false)
   evaluate ($kdani.cart.flag=false)
   evaluate ($ksani.cart.flag=false)
   if (&md.cart.flag=true) then
      evaluate ($kdani.inter.flag=true)
      evaluate ($ksani.inter.flag=true)
      @CNS_NMRMODULE:scalecoolsetup ( kdani=$kdani;
                                      ksani=$ksani;
                                      nmr=&nmr;
                                      input.noe.scale=&md.cool.noe;
                                      input.cdih.scale=&md.cool.cdih;
                                      input.ncycle=$ncycle; )
      evaluate ($kdani.cart.flag=true)
```

```
      evaluate ($ksani.cart.flag=true)
   else
      @CNS_NMRMODULE:scalecoolsetup ( kdani=$kdani;
                                      ksani=$ksani;
                                      nmr=&nmr;
                                      input.noe.scale=&md.cool.noe;
                                      input.cdih.scale=&md.cool.cdih;
                                      input.ncycle=$ncycle; )
   end if

   evaluate ($bath  = &md.cool.temp)
   evaluate ($k_vdw = $ini_vdw)

   evaluate ($i_cool = 0)
   while ($i_cool <= $ncycle) loop cool
      evaluate ($i_cool = $i_cool + 1)

      igroup
         interaction (chemical h*) (all) weights * 1 vdw 0. elec 0. end
         interaction (not chemical h*) (not chemical h*) weights * 1 vdw $k_vdw end
      end

      if ( &md.type.cool = "torsion" ) then
        dynamics  torsion
           cmremove=true
           vscaling = true
           tcoup = false
           timestep = &md.cool.ss
           nstep = $nstep
           nprint = $nstep
           temperature = $bath
        end
      else
        dynamics  cartesian
           cmremove=true
           vscaling = true
           tcoup = false
           timestep = &md.cool.ss
           nstep = $nstep
           nprint = $nstep
           temperature = $bath
        end
      end if

      if (&md.type.cool = "cartesian") then
        evaluate ($radius=max($fin_rad,$radius-$rad_step))
        parameter  nbonds repel=$radius   end end
         evaluate ($k_vdw=min($fin_vdw,$k_vdw*$vdw_step))
      else
         evaluate ($k_vdw= $k_vdw + $vdw_step)
      end if
      evaluate ($bath  = $bath  - &md.cool.tmpstp)

      @CNS_NMRMODULE:scalecool ( kdani=$kdani;
                                 ksani=$ksani;
                                 nmr=&nmr; )

   end loop cool

{- A second slow-cooling with cartesian dyanmics -}

   evaluate ($flag_cart=false)
   if (&md.cart.flag=true) then
      if (&md.type.cool = "torsion") then

         evaluate ($flag_cart=true)

         dynamics torsion
            topology
               reset
            end
```

```
end

evaluate ($cart_nucl_flag=false)
if ($nmr.nucl.num > 0) then
   evaluate ($cart_nucl_flag=true)
   parameter
      nbonds
         repel=0
         nbxmod=5
         wmin=0.01
         tolerance=0.5
         cutnb=11.5 ctonnb=9.5 ctofnb=10.5
         rdie vswitch switch
      end
   end
   flags include elec end
end if

evaluate ($ncycle=int((&md.cart.temp-$final_t)/&md.cart.tmpstp))
evaluate ($nstep=int(&md.cart.step/$ncycle))

evaluate ($vdw_step=(&md.cart.vdw.finl/&md.cart.vdw.init)^(1/$ncycle))
evaluate ($ini_rad=0.9)
evaluate ($fin_rad=0.8)
evaluate ($rad_step=($ini_rad-$fin_rad)/$ncycle)
evaluate ($radius=$ini_rad)

{- set up nmr restraint scaling -}
@CNS_NMRMODULE:scalecoolsetup ( kdani=$kdani;
                                ksani=$ksani;
                                nmr=&nmr;
                                input.noe.scale=&md.cart.noe;
                                input.cdih.scale=&md.cart.cdih;
                                input.ncycle=$ncycle; )

do (vx=maxwell(&md.cart.temp)) ( all )
do (vy=maxwell(&md.cart.temp)) ( all )
do (vz=maxwell(&md.cart.temp)) ( all )

evaluate ($bath=&md.cart.temp)
evaluate ($k_vdw=&md.cart.vdw.init)

evaluate ($i_cool = 0)
while ($i_cool <= $ncycle) loop cart
   evaluate ($i_cool = $i_cool + 1)

   igroup
      interaction (chemical h*) (all) weights * 1 vdw 0. elec 0. end
      interaction (not chemical h*) (not chemical h*) weights * 1 vdw $k_vdw
      end
   end

   dynamics  cartesian
      vscaling = true
      tcoup = false
      timestep = &md.cart.ss
      nstep = $nstep
      nprint = $nstep
      temperature = $bath
   end

   if ($cart_nucl_flag=false) then
      evaluate ($radius=max($fin_rad,$radius-$rad_step))
      parameter  nbonds repel=$radius   end end
   end if
   evaluate ($k_vdw=min(&md.cart.vdw.finl,$k_vdw*$vdw_step))
   evaluate ($bath=$bath-&md.cart.tmpstp)

   @CNS_NMRMODULE:scalecool ( kdani=$kdani;
                              ksani=$ksani;
                              nmr=&nmr; )
```

```
         end loop cart

      end if
   end if


{- reset torsion angle topology -}
   if ( $flag_tad=true ) then
      if ($flag_cart=false) then
         dynamics torsion
            topology
               reset
            end
         end
      end if
   end if


{- Final minimization -}

{ turn on proton chemical shifts }

   flags include prot end

   noe
      scale * &md.pow.noe
   end

   restraints dihedral
      scale = &md.pow.cdih
   end


   igroup interaction ( all ) ( all ) weights * 1 end end

   evaluate ($count=0 )
   evaluate ($nmr.min.num=0.)
   while (&md.pow.cycl > $count) loop pmini

      evaluate ($count=$count + 1)
      minimize powell nstep=&md.pow.step drop=10.0 nprint=25 end
      evaluate ($nmr.min.num=$nmr.min.num + $mini_cycles)

   end loop pmini

   {- translate the geometric center of the structure to the origin -}
   if ($num.dani > 0. ) then
   elseif ($num.sani > 0. ) then
   else
      show ave ( x ) ( all )
      evaluate ($geom_x=-$result)
      show ave ( y ) ( all )
      evaluate ($geom_y=-$result)
      show ave ( z ) ( all )
      evaluate ($geom_z=-$result)
      coor translate vector=( $geom_x $geom_y $geom_z ) selection=( all ) end
   end if

   @CNS_NMRMODULE:printaccept ( ave=$ave;
                                ave2=$ave2;
                                cv=$cv;
                                ener1=$ener1;
                                ener2=$ener2;
                                flag=&flg;
                                md=&md;
                                nmr=&nmr;
                                num=$num;
                                output=$nmr;
                                pdb=&pdb;  )
```

```
end loop main

   @CNS_NMRMODULE:calcave ( ave=$ave;
                            ave2=$ave2;
                            cv=$cv;
                            ener1=$ener1;
                            ener2=$ener2;
                            flag=&flg;
                            md=&md;
                            nmr=&nmr;
                            num=$num;
                            output=$nmr;
                            pdb=&pdb;  )


stop
```

10.1.2 Input constraints

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                      Distance Constraints                        !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!! G1 !!!!!!!!!!
 assign (resid   1 and name H5' ) (resid   1 and name H2'')  3.8 0.9 2.0
 assign (resid   1 and name H3' ) (resid   1 and name H4' )  2.8 0.9 0.9
 assign (resid   1 and name H2' ) (resid   1 and name H2'')  1.9 0.9 0.9
 assign (resid   1 and name H3' ) (resid   1 and name H5'')  3.6 0.9 0.9
 assign (resid   1 and name H3' ) (resid   1 and name H2' )  2.1 0.9 0.9
 assign (resid   1 and name H4' ) (resid   1 and name H2' )  2.6 0.9 1.5
 assign (resid   1 and name H5' ) (resid   1 and name H4' )  1.8 0.9 2.0
 assign (resid   1 and name H2' ) (resid   1 and name H5' )  3.5 0.9 2.0
 assign (resid   1 and name H2'') (resid   1 and name H4' )  2.4 0.9 0.9
 assign (resid   1 and name H1' ) (resid   1 and name H2' )  2.2 0.9 0.9
 assign (resid   1 and name H5' ) (resid   1 and name H1' )  2.4 0.9 2.0
 assign (resid   1 and name H4' ) (resid   1 and name H8  )  3.8 0.9 0.9
 assign (resid   1 and name H4' ) (resid   1 and name H5'')  2.6 0.9 0.9
 assign (resid   1 and name H3' ) (resid   1 and name H5' )  2.5 0.9 0.9
 assign (resid   1 and name H5' ) (resid   1 and name H5'')  2.5 0.9 0.9
 assign (resid   1 and name H2'') (resid   1 and name H8  )  3.2 0.9 2.0
 assign (resid   1 and name H8  ) (resid   1 and name H2' )  3.5 0.9 1.5
 assign (resid   1 and name H3' ) (resid   1 and name H2'')  2.1 0.9 0.9
 assign (resid   1 and name H1' ) (resid   1 and name H8  )  1.5 0.9 1.5
 assign (resid   1 and name H2'') (resid   1 and name H1' )  1.9 0.9 0.9
 assign (resid   1 and name H1' ) (resid   2 and name H1' )  3.3 0.9 2.0
 assign (resid   1 and name H2'') (resid   2 and name H5'')  2.9 0.9 0.9
 assign (resid   1 and name H2'') (resid   2 and name H8  )  1.7 0.9 1.5
 assign (resid   1 and name H2' ) (resid   2 and name H5' )  3.0 0.9 1.5
 assign (resid   1 and name H1' ) (resid   2 and name H8  )  2.2 0.9 2.0
 assign (resid   1 and name H2' ) (resid   2 and name H8  )  1.8 0.9 0.9
 assign (resid   1 and name H1' ) (resid   2 and name H5'')  2.8 0.9 1.5
 assign (resid   1 and name H8  ) (resid   2 and name H8  )  3.7 0.9 2.0
 assign (resid   1 and name H5' ) (resid   7 and name H1' )  3.1 0.9 0.9
 assign (resid   1 and name H5' ) (resid   7 and name H5'')  2.7 0.9 1.5
 assign (resid   1 and name H5' ) (resid   7 and name H4' )  2.6 0.9 0.9
 assign (resid   1 and name H5' ) (resid   7 and name H6  )  1.9 0.9 2.0
 assign (resid   1 and name H4' ) (resid   7 and name H6  )  2.5 0.9 2.0
 assign (resid   1 and name H1' ) (resid   7 and name H6  )  3.7 0.9 2.0
 assign (resid   1 and name H2' ) (resid   7 and name H6  )  3.3 0.9 2.0
 assign (resid   1 and name H5' ) (resid   7 and name H5' )  3.7 0.9 0.9
 assign (resid   1 and name H3' ) (resid   2 and name H8  )  3.2 0.9 2.0
 assign (resid   1 and name H1' ) (resid   1 and name H8  )  2.5 0.9 0.9
 assign (resid   1 and name H21 ) (resid   1 and name H1  )  2.1 0.9 0.9
 assign (resid   1 and name H22 ) (resid   1 and name H1  )  3.4 0.9 0.9
 assign (resid   1 and name H21 ) (resid   1 and name H22 )  1.8 0.9 0.9
 assign (resid   1 and name H1' ) (resid   2 and name H8  )  3.8 0.9 0.9
 assign (resid   1 and name H22 ) (resid   9 and name H8  )  3.9 0.9 0.9
 assign (resid   1 and name H1  ) (resid  10 and name H1  )  2.2 0.9 0.9
 assign (resid   1 and name H1  ) (resid  10 and name H1  )  2.4 0.9 0.9
 assign (resid   1 and name H8  ) (resid   7 and name H7# )  3.7 0.9 0.9
```

```
 assign (resid   1 and name H5'') (resid   7 and name H7# )  3.6 0.9 1.1

!!!!!!!!!! G2 !!!!!!!!!!
 assign (resid   2 and name H3' ) (resid   1 and name H1' )  2.6 0.9 2.5
 assign (resid   2 and name H5' ) (resid   1 and name H2'')  2.4 0.9 1.5
 assign (resid   2 and name H1' ) (resid   1 and name H8  )  2.3 0.9 2.0
 assign (resid   2 and name H5'') (resid   1 and name H2' )  2.6 0.9 2.0
 assign (resid   2 and name H4' ) (resid   1 and name H8  )  2.8 0.9 2.0
 assign (resid   2 and name H4' ) (resid   2 and name H1' )  1.8 0.9 0.9
 assign (resid   2 and name H3' ) (resid   2 and name H2' )  2.0 0.9 0.9
 assign (resid   2 and name H4' ) (resid   2 and name H2'')  2.2 0.9 1.5
 assign (resid   2 and name H1' ) (resid   2 and name H5' )  2.8 0.9 2.0
 assign (resid   2 and name H2' ) (resid   2 and name H1' )  3.1 0.9 0.9
 assign (resid   2 and name H8  ) (resid   2 and name H1' )  2.6 0.9 2.0
 assign (resid   2 and name H4' ) (resid   2 and name H8  )  2.9 0.9 2.0
 assign (resid   2 and name H2' ) (resid   2 and name H8  )  3.7 0.9 0.9
 assign (resid   2 and name H2' ) (resid   2 and name H2'')  1.7 0.9 0.9
 assign (resid   2 and name H2'') (resid   2 and name H5' )  2.6 0.9 2.5
 assign (resid   2 and name H2' ) (resid   2 and name H4' )  2.4 0.9 1.5
 assign (resid   2 and name H3' ) (resid   2 and name H1' )  2.3 0.9 2.0
 assign (resid   2 and name H3' ) (resid   2 and name H2'')  1.9 0.9 0.9
 assign (resid   2 and name H5'') (resid   2 and name H8  )  3.6 0.9 2.0
 assign (resid   2 and name H8  ) (resid   2 and name H2'')  2.2 0.9 2.0
 assign (resid   2 and name H1' ) (resid   2 and name H2'')  2.1 0.9 0.9
 assign (resid   2 and name H5' ) (resid   2 and name H2' )  2.4 0.9 2.0
 assign (resid   2 and name H2'') (resid   2 and name H5'')  2.8 0.9 2.0
 assign (resid   2 and name H2'') (resid   3 and name H5'')  3.0 0.9 0.9
 assign (resid   2 and name H2' ) (resid  11 and name H8  )  2.7 0.9 0.9
 assign (resid   2 and name H8  ) (resid  11 and name H1' )  3.7 0.9 0.9
 assign (resid   2 and name H8  ) (resid  11 and name H8  )  2.4 0.9 0.9
 assign (resid   2 and name H5' ) (resid   2 and name H8  )  3.5 0.9 2.0
 assign (resid   2 and name H1' ) (resid   1 and name H8  )  4.0 0.9 0.9
 assign (resid   2 and name H1' ) (resid   2 and name H8  )  3.9 0.9 0.9
 assign (resid   2 and name H1  ) (resid   4 and name H1  )  2.9 0.9 0.9
 assign (resid   2 and name H1  ) (resid   3 and name H1  )  4.0 0.9 0.9

 !!!!!!!!!! G3 !!!!!!!!!!
 assign (resid   3 and name H5' ) (resid   2 and name H1' )  2.5 0.9 0.9
 assign (resid   3 and name H2' ) (resid   3 and name H1' )  2.0 0.9 0.9
 assign (resid   3 and name H2'') (resid   3 and name H8  )  2.1 0.9 2.0
 assign (resid   3 and name H3' ) (resid   3 and name H2'')  2.1 0.9 0.9
 assign (resid   3 and name H1' ) (resid   3 and name H2'')  2.2 0.9 0.9
 assign (resid   3 and name H8  ) (resid   3 and name H1' )  1.6 0.9 0.9
 assign (resid   3 and name H5' ) (resid   3 and name H4' )  2.4 0.9 0.9
 assign (resid   3 and name H8  ) (resid   3 and name H2' )  3.9 0.9 1.2
 assign (resid   3 and name H2' ) (resid   3 and name H2'')  2.0 0.9 0.9
 assign (resid   3 and name H3' ) (resid   3 and name H4' )  2.4 0.9 0.9
 assign (resid   3 and name H3' ) (resid   3 and name H2' )  2.0 0.9 0.9
 assign (resid   3 and name H4' ) (resid   3 and name H1' )  3.4 0.9 0.9
 assign (resid   3 and name H8  ) (resid   4 and name H8  )  3.0 0.9 2.0
 assign (resid   3 and name H3' ) (resid   4 and name H5'')  3.7 0.9 2.0
 assign (resid   3 and name H1' ) (resid   4 and name H4' )  2.6 0.9 2.0
 assign (resid   3 and name H2' ) (resid   4 and name H8  )  1.8 0.9 0.9
 assign (resid   3 and name H3' ) (resid   4 and name H8  )  2.2 0.9 2.0
 assign (resid   3 and name H8  ) (resid   4 and name H4' )  3.1 0.9 1.5
 assign (resid   3 and name H2' ) (resid   4 and name H5'')  3.6 0.9 2.0
 assign (resid   3 and name H1  ) (resid   3 and name H22 )  3.4 0.9 0.9
 assign (resid   3 and name H1' ) (resid   3 and name H8  )  2.5 0.9 0.9
 assign (resid   3 and name H1  ) (resid   3 and name H21 )  2.2 0.9 0.9
 assign (resid   3 and name H1  ) (resid  10 and name H1  )  3.6 0.9 0.9

!!!!!!!!!! G4 !!!!!!!!!!
 assign (resid   4 and name H5' ) (resid   3 and name H1' )  3.9 0.9 0.9
 assign (resid   4 and name H2' ) (resid   3 and name H8  )  3.8 0.9 2.0
 assign (resid   4 and name H8  ) (resid   3 and name H1' )  3.4 0.9 2.0
 assign (resid   4 and name H3' ) (resid   4 and name H2' )  2.0 0.9 0.9
 assign (resid   4 and name H8  ) (resid   4 and name H1' )  2.4 0.9 1.5
 assign (resid   4 and name H2'') (resid   4 and name H1' )  1.8 0.9 0.9
 assign (resid   4 and name H2' ) (resid   4 and name H1' )  2.0 0.9 0.9
 assign (resid   4 and name H4' ) (resid   4 and name H2' )  2.4 0.9 2.0
 assign (resid   4 and name H2'') (resid   4 and name H4' )  2.7 0.9 1.5
```

```
 assign (resid   4 and name H2' ) (resid   4 and name H2'')  1.6 0.9 0.9
 assign (resid   4 and name H5'') (resid   4 and name H8  )  3.7 0.9 1.5
 assign (resid   4 and name H4' ) (resid   4 and name H8  )  3.7 0.9 2.0
 assign (resid   4 and name H8  ) (resid   4 and name H2'')  1.8 0.9 2.0
 assign (resid   4 and name H3' ) (resid   4 and name H2'')  1.9 0.9 0.9
 assign (resid   4 and name H3' ) (resid   4 and name H1' )  2.2 0.9 1.6
 assign (resid   4 and name H3' ) (resid   4 and name H8  )  3.1 0.9 2.0
 assign (resid   4 and name H8  ) (resid   4 and name H2' )  2.9 0.9 0.9
 assign (resid   4 and name H4' ) (resid   4 and name H5' )  1.8 0.9 0.9
 assign (resid   4 and name H3' ) (resid   4 and name H4' )  2.5 0.9 0.9
 assign (resid   4 and name H1' ) (resid   5 and name H4' )  2.4 0.9 0.9
 assign (resid   4 and name H2' ) (resid   5 and name H6  )  2.1 0.9 0.9
 assign (resid   4 and name H1' ) (resid   5 and name H5'')  2.9 0.9 0.9
 assign (resid   4 and name H1' ) (resid   5 and name H6  )  2.3 0.9 2.0
 assign (resid   4 and name H1' ) (resid   5 and name H1' )  3.1 0.9 0.9
 assign (resid   4 and name H2'') (resid   5 and name H6  )  1.9 0.9 1.5
 assign (resid   4 and name H8  ) (resid   5 and name H6  )  2.4 0.9 2.0
 assign (resid   4 and name H1' ) (resid   4 and name H8  )  3.8 0.9 0.9
 assign (resid   4 and name H1' ) (resid   5 and name H6  )  3.9 0.9 0.9
 assign (resid   4 and name H1  ) (resid   5 and name H1' )  3.5 0.9 0.9
 assign (resid   4 and name H1  ) (resid   5 and name H1' )  3.5 0.9 0.9
 assign (resid   4 and name H1  ) (resid   2 and name H1  )  3.8 0.9 0.9
 assign (resid   4 and name H8  ) (resid   5 and name H7# )  3.5 0.9 0.9

!!!!!!!!!! T5 !!!!!!!!!!
 assign (resid   5 and name H5'') (resid   4 and name H2'')  2.5 0.9 1.5
 assign (resid   5 and name H1' ) (resid   4 and name H8  )  3.7 0.9 0.9
 assign (resid   5 and name H1' ) (resid   4 and name H2'')  3.3 0.9 1.5
 assign (resid   5 and name H4' ) (resid   4 and name H2' )  3.9 0.9 1.5
 assign (resid   5 and name H4' ) (resid   4 and name H2'')  2.8 0.9 2.0
 assign (resid   5 and name H4' ) (resid   5 and name H2' )  3.0 0.9 0.9
 assign (resid   5 and name H1' ) (resid   5 and name H2'')  2.1 0.9 0.9
 assign (resid   5 and name H4' ) (resid   5 and name H5' )  1.8 0.9 0.9
 assign (resid   5 and name H2' ) (resid   5 and name H2'')  1.9 0.9 0.9
 assign (resid   5 and name H6  ) (resid   5 and name H4' )  3.4 0.9 0.9
 assign (resid   5 and name H5'') (resid   5 and name H2'')  2.9 0.9 2.0
 assign (resid   5 and name H4' ) (resid   5 and name H1' )  2.1 0.9 0.9
 assign (resid   5 and name H1' ) (resid   5 and name H2' )  2.5 0.9 0.9
 assign (resid   5 and name H5'') (resid   5 and name H5' )  1.7 0.9 0.9
 assign (resid   5 and name H5'') (resid   5 and name H6  )  2.6 0.9 1.5
 assign (resid   5 and name H2' ) (resid   5 and name H5' )  3.8 0.9 1.2
 assign (resid   5 and name H2' ) (resid   5 and name H6  )  1.9 0.9 2.0
 assign (resid   5 and name H3' ) (resid   5 and name H5'')  2.4 0.9 0.9
 assign (resid   5 and name H2'') (resid   5 and name H4' )  2.3 0.9 0.9
 assign (resid   5 and name H4' ) (resid   5 and name H5'')  2.5 0.9 0.9
 assign (resid   5 and name H5'') (resid   5 and name H2' )  3.1 0.9 1.5
 assign (resid   5 and name H2'') (resid   5 and name H6  )  2.7 0.9 2.0
 assign (resid   5 and name H6  ) (resid   5 and name H1' )  2.7 0.9 0.9
 assign (resid   5 and name H3' ) (resid   5 and name H6  )  3.1 0.9 1.5
 assign (resid   5 and name H5'') (resid   5 and name H1' )  3.4 0.9 2.0
 assign (resid   5 and name H3' ) (resid   5 and name H4' )  2.3 0.9 0.9
 assign (resid   5 and name H3' ) (resid   5 and name H2' )  2.3 0.9 0.9
 assign (resid   5 and name H2'') (resid   5 and name H5' )  2.9 0.9 1.5
 assign (resid   5 and name H3' ) (resid   5 and name H1' )  3.7 0.9 1.3
 assign (resid   5 and name H3' ) (resid   5 and name H2'')  2.2 0.9 0.9
 assign (resid   5 and name H2' ) (resid   6 and name H6  )  2.2 0.9 0.9
 assign (resid   5 and name H2'') (resid   6 and name H5'')  2.7 0.9 0.9
 assign (resid   5 and name H4' ) (resid   6 and name H6  )  3.9 0.9 1.5
 assign (resid   5 and name H2' ) (resid   6 and name H4' )  3.0 0.9 0.9
 assign (resid   5 and name H3' ) (resid   6 and name H6  )  3.0 0.9 1.5
 assign (resid   5 and name H2' ) (resid   6 and name H5'')  3.3 0.9 0.9
 assign (resid   5 and name H2'') (resid   6 and name H6  )  2.1 0.9 1.5
 assign (resid   5 and name H2'') (resid   6 and name H4' )  3.1 0.9 1.5
 assign (resid   5 and name H3' ) (resid   4 and name H1' )  3.2 0.9 2.0
 assign (resid   5 and name H5' ) (resid   4 and name H2' )  3.4 0.9 2.0
 assign (resid   5 and name H5' ) (resid   4 and name H2'')  2.4 0.9 2.0
 assign (resid   5 and name H6  ) (resid   5 and name H5' )  3.6 0.9 0.9
 assign (resid   5 and name H5' ) (resid   5 and name H1' )  2.9 0.9 2.0
 assign (resid   5 and name H6  ) (resid   4 and name H8  )  3.7 0.9 0.9
 assign (resid   5 and name H1' ) (resid   5 and name H6  )  3.7 0.9 0.9
 assign (resid   5 and name H7# ) (resid   4 and name H2' )  3.8 0.9 1.2
```

```
 assign (resid   5 and name H7# ) (resid   5 and name H6  )  2.9 0.9 0.9
 assign (resid   5 and name H6  ) (resid   6 and name H7# )  3.8 0.9 0.9

!!!!!!!!!! T6 !!!!!!!!!!
 assign (resid   6 and name H6  ) (resid   5 and name H1' )  3.5 0.9 1.5
 assign (resid   6 and name H2'') (resid   6 and name H6  )  3.4 0.9 0.9
 assign (resid   6 and name H2'') (resid   6 and name H1' )  2.6 0.9 0.9
 assign (resid   6 and name H6  ) (resid   6 and name H2' )  2.8 0.9 1.5
 assign (resid   6 and name H2' ) (resid   6 and name H5'')  3.9 0.9 2.2
 assign (resid   6 and name H1' ) (resid   6 and name H6  )  2.2 0.9 1.5
 assign (resid   6 and name H5' ) (resid   6 and name H4' )  2.5 0.9 0.9
 assign (resid   6 and name H1' ) (resid   6 and name H2' )  2.4 0.9 0.9
 assign (resid   6 and name H3' ) (resid   6 and name H4' )  2.0 0.9 0.9
 assign (resid   6 and name H4' ) (resid   6 and name H2'')  2.4 0.9 2.0
 assign (resid   6 and name H3' ) (resid   6 and name H1' )  2.9 0.9 0.9
 assign (resid   6 and name H4' ) (resid   6 and name H2' )  2.7 0.9 0.9
 assign (resid   6 and name H6  ) (resid   6 and name H4' )  2.9 0.9 0.9
 assign (resid   6 and name H4' ) (resid   6 and name H5'')  1.9 0.9 0.9
 assign (resid   6 and name H6  ) (resid   6 and name H5'')  2.7 0.9 0.9
 assign (resid   6 and name H3' ) (resid   6 and name H2'')  3.2 0.9 0.9
 assign (resid   6 and name H3' ) (resid   6 and name H5'')  2.0 0.9 0.9
 assign (resid   6 and name H5' ) (resid   6 and name H2' )  2.8 0.9 1.5
 assign (resid   6 and name H1' ) (resid   6 and name H4' )  2.4 0.9 0.9
 assign (resid   6 and name H6  ) (resid   6 and name H5' )  2.8 0.9 0.9
 assign (resid   6 and name H5'') (resid   6 and name H5' )  2.5 0.9 0.9
 assign (resid   6 and name H5'') (resid   6 and name H2'')  2.8 0.9 1.0
 assign (resid   6 and name H1' ) (resid   6 and name H5' )  3.7 0.9 2.0
 assign (resid   6 and name H2'') (resid   6 and name H2' )  1.7 0.9 0.9
 assign (resid   6 and name H1' ) (resid   7 and name H6  )  3.6 0.9 0.9
 assign (resid   6 and name H1' ) (resid   6 and name H6  )  3.7 0.9 0.9
 assign (resid   6 and name H1' ) (resid   7 and name H6  )  3.5 0.9 0.9
 assign (resid   6 and name H6  ) (resid   6 and name H7# )  3.5 0.9 0.9
 assign (resid   6 and name H1' ) (resid   7 and name H7# )  2.2 0.9 1.5

!!!!!!!!!! T7 !!!!!!!!!!
 assign (resid   7 and name H3' ) (resid   7 and name H6  )  2.8 0.9 2.0
 assign (resid   7 and name H3' ) (resid   7 and name H2' )  2.1 0.9 0.9
 assign (resid   7 and name H5' ) (resid   7 and name H5'')  2.1 0.9 0.9
 assign (resid   7 and name H5'') (resid   7 and name H6  )  1.9 0.9 2.5
 assign (resid   7 and name H1' ) (resid   7 and name H2'')  2.1 0.9 0.9
 assign (resid   7 and name H3' ) (resid   7 and name H5'')  2.3 0.9 0.9
 assign (resid   7 and name H2' ) (resid   7 and name H1' )  2.5 0.9 0.9
 assign (resid   7 and name H2'') (resid   7 and name H6  )  2.4 0.9 2.0
 assign (resid   7 and name H3' ) (resid   7 and name H1' )  3.8 0.9 0.9
 assign (resid   7 and name H2' ) (resid   7 and name H4' )  2.3 0.9 2.0
 assign (resid   7 and name H2'') (resid   7 and name H4' )  3.1 0.9 0.9
 assign (resid   7 and name H2'') (resid   7 and name H5'')  2.7 0.9 1.5
 assign (resid   7 and name H1' ) (resid   7 and name H6  )  3.1 0.9 0.9
 assign (resid   7 and name H6  ) (resid   7 and name H4' )  2.4 0.9 2.0
 assign (resid   7 and name H3' ) (resid   7 and name H4' )  2.2 0.9 1.5
 assign (resid   7 and name H4' ) (resid   7 and name H1' )  3.8 0.9 0.9
 assign (resid   7 and name H5' ) (resid   7 and name H2' )  2.8 0.9 2.0
 assign (resid   7 and name H4' ) (resid   7 and name H5'')  2.1 0.9 0.9
 assign (resid   7 and name H5'') (resid   7 and name H1' )  3.5 0.9 1.5
 assign (resid   7 and name H5' ) (resid   7 and name H4' )  2.9 0.9 0.9
 assign (resid   7 and name H5' ) (resid   7 and name H2'')  2.9 0.9 2.0
 assign (resid   7 and name H3' ) (resid   7 and name H2'')  2.4 0.9 0.9
 assign (resid   7 and name H2' ) (resid   7 and name H2'')  1.8 0.9 0.9
 assign (resid   7 and name H2'') (resid   8 and name H5'')  2.4 0.9 1.5
 assign (resid   7 and name H2'') (resid   8 and name H6  )  2.2 0.9 1.5
 assign (resid   7 and name H1' ) (resid   8 and name H5'')  2.9 0.9 2.5
 assign (resid   7 and name H1' ) (resid   8 and name H6  )  3.9 0.9 2.0
 assign (resid   7 and name H3' ) (resid   1 and name H5' )  3.2 0.9 2.0
 assign (resid   7 and name H2' ) (resid   1 and name H5' )  3.4 0.9 2.0
 assign (resid   7 and name H2'') (resid   1 and name H5' )  4.0 0.9 2.0
 assign (resid   7 and name H2'') (resid   8 and name H5' )  2.2 0.9 1.5
 assign (resid   7 and name H1' ) (resid   7 and name H6  )  3.5 0.9 0.9
 assign (resid   7 and name H1' ) (resid   1 and name H22 )  3.2 0.9 0.9
 assign (resid   7 and name H1' ) (resid   1 and name H1  )  4.0 0.9 0.9
 assign (resid   7 and name H7# ) (resid   5 and name H4' )  3.9 0.9 0.9
 assign (resid   7 and name H7# ) (resid   6 and name H5' )  3.6 0.9 1.5
```

```
 assign (resid   7 and name H7# ) (resid   6 and name H5'')  3.5 1.5 0.9
 assign (resid   7 and name H6  ) (resid   7 and name H7# )  3.4 0.9 0.9
 assign (resid   7 and name H7# ) (resid   1 and name H1' )  3.8 0.9 1.5
 assign (resid   7 and name H6  ) (resid   8 and name H7# )  4.0 0.9 1.5
 assign (resid   7 and name H7# ) (resid   4 and name H1  )  3.9 0.9 0.9
 assign (resid   7 and name H7# ) (resid   1 and name H1  )  4.6 0.9 0.9
 assign (resid   7 and name H7# ) (resid  12 and name H1  )  5.7 0.9 0.9
 assign (resid   7 and name H1' ) (resid 101 and name H1  )  4.4 0.9 0.9

!!!!!!!!!! T8 !!!!!!!!!!
 assign (resid   8 and name H6  ) (resid   8 and name H2' )  4.0 0.9 0.9
 assign (resid   8 and name H2'') (resid   8 and name H1' )  2.0 0.9 0.9
 assign (resid   8 and name H2' ) (resid   8 and name H4' )  3.0 0.9 0.9
 assign (resid   8 and name H3' ) (resid   8 and name H2' )  2.1 0.9 0.9
 assign (resid   8 and name H1' ) (resid   8 and name H6  )  2.4 0.9 0.9
 assign (resid   8 and name H2' ) (resid   8 and name H2'')  1.8 0.9 0.9
 assign (resid   8 and name H5'') (resid   8 and name H6  )  2.7 0.9 2.0
 assign (resid   8 and name H5' ) (resid   8 and name H6  )  2.6 0.9 2.0
 assign (resid   8 and name H6  ) (resid   8 and name H2'')  2.2 0.9 2.0
 assign (resid   8 and name H3' ) (resid   8 and name H5'')  2.2 0.9 0.9
 assign (resid   8 and name H2' ) (resid   8 and name H1' )  2.2 0.9 0.9
 assign (resid   8 and name H3' ) (resid   8 and name H2'')  2.1 0.9 0.9
 assign (resid   8 and name H5' ) (resid   8 and name H2'')  3.4 0.9 1.5
 assign (resid   8 and name H5' ) (resid   8 and name H2' )  2.5 0.9 2.5
 assign (resid   8 and name H5' ) (resid   9 and name H8  )  3.4 0.9 0.9
 assign (resid   8 and name H3' ) (resid   8 and name H5' )  2.0 0.9 2.0
 assign (resid   8 and name H1' ) (resid   8 and name H6  )  3.7 0.9 0.9
 assign (resid   8 and name H6  ) (resid   8 and name H7# )  3.2 0.9 0.9
 assign (resid   8 and name H7# ) (resid   4 and name H1  )  4.8 0.9 0.9
 assign (resid   8 and name H7# ) (resid   1 and name H1  )  4.5 0.9 1.9
 assign (resid   8 and name H7# ) (resid  12 and name H1  )  4.8 0.9 0.9

!!!!!!!!!! G9 !!!!!!!!!!
 assign (resid   9 and name H3' ) (resid   9 and name H2'')  2.2 0.9 0.9
 assign (resid   9 and name H2'') (resid   9 and name H4' )  2.3 0.9 0.9
 assign (resid   9 and name H1' ) (resid   9 and name H2' )  2.2 0.9 0.9
 assign (resid   9 and name H8  ) (resid   9 and name H2' )  3.8 0.9 1.5
 assign (resid   9 and name H4' ) (resid   9 and name H2' )  2.7 0.9 0.9
 assign (resid   9 and name H3' ) (resid   9 and name H2' )  2.1 0.9 0.9
 assign (resid   9 and name H2'') (resid   9 and name H1' )  2.0 0.9 0.9
 assign (resid   9 and name H5'') (resid   9 and name H2'')  3.7 0.9 2.0
 assign (resid   9 and name H2'') (resid   9 and name H8  )  2.2 0.9 1.8
 assign (resid   9 and name H3' ) (resid   9 and name H1' )  2.5 0.9 2.0
 assign (resid   9 and name H2'') (resid   9 and name H2' )  1.7 0.9 0.9
 assign (resid   9 and name H5'') (resid   9 and name H4' )  1.8 0.9 0.9
 assign (resid   9 and name H5'') (resid   9 and name H1' )  3.0 0.9 1.5
 assign (resid   9 and name H1' ) (resid   9 and name H8  )  1.5 0.9 0.9
 assign (resid   9 and name H5' ) (resid   9 and name H1' )  2.3 0.9 2.5
 assign (resid   9 and name H3' ) (resid   9 and name H4' )  2.1 0.9 0.9
 assign (resid   9 and name H1' ) (resid  10 and name H8  )  3.2 0.9 1.5
 assign (resid   9 and name H8  ) (resid  10 and name H8  )  3.6 0.9 1.5
 assign (resid   9 and name H8  ) (resid  10 and name H1' )  2.6 0.9 0.9
 assign (resid   9 and name H2'') (resid  10 and name H8  )  2.7 0.9 1.5
 assign (resid   9 and name H2'') (resid  10 and name H4' )  3.6 0.9 1.5
 assign (resid   9 and name H2' ) (resid  10 and name H8  )  1.9 0.9 0.9
 assign (resid   9 and name H2' ) (resid  10 and name H2'#)  3.4 0.9 1.5
 assign (resid   9 and name H3' ) (resid  10 and name H2'#)  3.6 0.9 1.5
 assign (resid   9 and name H1' ) (resid   9 and name H8  )  2.4 0.9 0.9
 assign (resid   9 and name H1  ) (resid   9 and name H22 )  3.4 0.9 0.9
 assign (resid   9 and name H1' ) (resid  10 and name H8  )  3.8 0.9 0.9
 assign (resid   9 and name H1  ) (resid  11 and name H1  )  3.5 0.9 0.9
 assign (resid   9 and name H1  ) (resid  12 and name H8  )  3.9 0.9 0.9
 assign (resid   9 and name H1  ) (resid  12 and name H8  )  3.9 0.9 0.9

!!!!!!!!!! G10 !!!!!!!!!!
 assign (resid  10 and name H4' ) (resid   9 and name H8  )  3.6 0.9 1.5
 assign (resid  10 and name H5' ) (resid   9 and name H2' )  2.9 0.9 2.0
 assign (resid  10 and name H3' ) (resid   9 and name H8  )  3.7 0.9 2.0
 assign (resid  10 and name H2'') (resid   9 and name H8  )  3.2 0.9 2.0
 assign (resid  10 and name H3' ) (resid   9 and name H1' )  3.6 0.9 2.0
 assign (resid  10 and name H4' ) (resid  10 and name H1' )  1.8 0.9 0.9
```

```
 assign (resid  10 and name H3' ) (resid  10 and name H2'#)  2.8 0.9 0.9
 assign (resid  10 and name H8  ) (resid  10 and name H5' )  3.7 0.9 2.0
 assign (resid  10 and name H2'#) (resid  10 and name H5' )  3.6 0.9 2.0
 assign (resid  10 and name H1' ) (resid  10 and name H2'#)  1.9 0.9 0.9
 assign (resid  10 and name H1' ) (resid  10 and name H8  )  2.3 0.9 1.5
 assign (resid  10 and name H8  ) (resid  10 and name H2'#)  2.9 0.9 1.5
 assign (resid  10 and name H3' ) (resid  10 and name H1' )  2.2 0.9 2.0
 assign (resid  10 and name H3' ) (resid  10 and name H5' )  2.3 0.9 0.9
 assign (resid  10 and name H5'') (resid  10 and name H2'#)  3.6 0.9 2.0
 assign (resid  10 and name H3' ) (resid  10 and name H4' )  2.5 0.9 0.9
 assign (resid  10 and name H2'#) (resid   9 and name H2'')  3.9 0.9 0.9
 assign (resid  10 and name H5'') (resid  10 and name H8  )  3.6 0.9 1.5
 assign (resid  10 and name H3' ) (resid  10 and name H8  )  2.9 0.9 2.0
 assign (resid  10 and name H1' ) (resid   9 and name H8  )  2.9 0.9 0.9
 assign (resid  10 and name H22 ) (resid  10 and name H21 )  1.8 0.9 0.9
 assign (resid  10 and name H1  ) (resid  10 and name H21 )  2.2 0.9 0.9
 assign (resid  10 and name H1' ) (resid  10 and name H8  )  3.8 0.9 0.9
 assign (resid  10 and name H22 ) (resid  10 and name H1  )  3.5 0.9 0.9
 assign (resid  10 and name H1  ) (resid  11 and name H1  )  4.0 0.9 0.9
 assign (resid  10 and name H22 ) (resid   2 and name H8  )  3.6 0.9 0.9
 assign (resid  10 and name H1  ) (resid   3 and name H1  )  3.4 0.9 0.9

!!!!!!!!!! G11 !!!!!!!!!!
 assign (resid  11 and name H5' ) (resid  10 and name H1' )  2.0 0.9 0.9
 assign (resid  11 and name H5' ) (resid  10 and name H4' )  2.8 0.9 2.0
 assign (resid  11 and name H4' ) (resid  10 and name H1' )  3.5 0.9 2.0
 assign (resid  11 and name H8  ) (resid  11 and name H1' )  1.7 0.9 0.9
 assign (resid  11 and name H8  ) (resid  11 and name H4' )  3.3 0.9 2.0
 assign (resid  11 and name H8  ) (resid  11 and name H2' )  3.0 0.9 1.5
 assign (resid  11 and name H1' ) (resid  11 and name H2'')  2.1 0.9 0.9
 assign (resid  11 and name H3' ) (resid  11 and name H2'')  2.0 0.9 0.9
 assign (resid  11 and name H4' ) (resid  11 and name H2' )  2.7 0.9 1.5
 assign (resid  11 and name H5' ) (resid  11 and name H1' )  2.6 0.9 2.0
 assign (resid  11 and name H3' ) (resid  11 and name H4' )  2.4 0.9 0.9
 assign (resid  11 and name H4' ) (resid  11 and name H5'')  2.8 0.9 0.9
 assign (resid  11 and name H5' ) (resid  11 and name H2'')  2.6 0.9 2.0
 assign (resid  11 and name H3' ) (resid  11 and name H5'')  2.7 0.9 0.9
 assign (resid  11 and name H1' ) (resid  11 and name H2' )  2.3 0.9 0.9
 assign (resid  11 and name H3' ) (resid  11 and name H2' )  2.0 0.9 0.9
 assign (resid  11 and name H3' ) (resid  11 and name H1' )  2.7 0.9 1.5
 assign (resid  11 and name H1' ) (resid  11 and name H4' )  2.9 0.9 0.9
 assign (resid  11 and name H2'') (resid  11 and name H2' )  1.9 0.9 0.9
 assign (resid  11 and name H4' ) (resid  11 and name H2'')  2.6 0.9 0.9
 assign (resid  11 and name H8  ) (resid  11 and name H2'')  3.7 0.9 1.5
 assign (resid  11 and name H2'') (resid  12 and name H8  )  1.9 0.9 0.9
 assign (resid  11 and name H2'') (resid  12 and name H4' )  3.5 0.9 1.5
 assign (resid  11 and name H8  ) (resid  12 and name H8  )  3.5 0.9 0.9
 assign (resid  11 and name H1' ) (resid  12 and name H8  )  2.5 0.9 1.5
 assign (resid  11 and name H8  ) (resid  12 and name H1' )  2.9 0.9 1.5
 assign (resid  11 and name H2' ) (resid  12 and name H8  )  3.1 0.9 0.9
 assign (resid  11 and name H2' ) (resid  12 and name H1' )  3.8 0.9 2.0
 assign (resid  11 and name H2' ) (resid  12 and name H2' )  3.7 0.9 2.0
 assign (resid  11 and name H2' ) (resid  12 and name H2'')  3.4 0.9 2.0
 assign (resid  11 and name H8  ) (resid  12 and name H4' )  3.6 0.9 0.9
 assign (resid  11 and name H8  ) (resid   2 and name H2'')  3.7 0.9 0.9
 assign (resid  11 and name H2'') (resid  12 and name H1' )  3.3 0.9 2.0
 assign (resid  11 and name H1' ) (resid  11 and name H8  )  2.5 0.9 0.9
 assign (resid  11 and name H22 ) (resid  11 and name H1  )  3.4 0.9 0.9
 assign (resid  11 and name H21 ) (resid  11 and name H22 )  1.8 0.9 0.9
 assign (resid  11 and name H1  ) (resid  11 and name H21 )  2.2 0.9 0.9
 assign (resid  11 and name H1' ) (resid  12 and name H8  )  4.0 0.9 0.9
 assign (resid  11 and name H8  ) (resid   2 and name H8  )  3.5 0.9 0.9
 assign (resid  11 and name H21 ) (resid  10 and name H8  )  3.1 0.9 0.9

!!!!!!!!!! G12 !!!!!!!!!!
 assign (resid  12 and name H5'') (resid  11 and name H2'')  2.6 0.9 0.9
 assign (resid  12 and name H5' ) (resid  11 and name H2'')  3.1 0.9 0.9
 assign (resid  12 and name H5'') (resid  11 and name H1' )  3.9 0.9 0.9
 assign (resid  12 and name H5' ) (resid  11 and name H2' )  2.8 0.9 2.0
 assign (resid  12 and name H4' ) (resid  11 and name H1' )  2.5 0.9 2.0
 assign (resid  12 and name H5'') (resid  11 and name H8  )  3.2 0.9 2.0
```

```
 assign (resid  12 and name H2'') (resid  12 and name H2' )  1.6 0.9 0.9
 assign (resid  12 and name H1' ) (resid  12 and name H2'')  2.0 0.9 0.9
 assign (resid  12 and name H4' ) (resid  12 and name H8  )  3.8 0.9 0.9
 assign (resid  12 and name H3' ) (resid  12 and name H8  )  3.9 0.9 0.9
 assign (resid  12 and name H2'') (resid  12 and name H8  )  2.2 0.9 2.0
 assign (resid  12 and name H8  ) (resid  12 and name H1' )  2.7 0.9 0.9
 assign (resid  12 and name H1' ) (resid  12 and name H3' )  3.1 0.9 0.9
 assign (resid  12 and name H3' ) (resid  12 and name H2'')  2.9 0.9 0.9
 assign (resid  12 and name H5'') (resid  12 and name H8  )  3.3 0.9 2.0
 assign (resid  12 and name H2' ) (resid  12 and name H8  )  3.9 0.9 0.9
 assign (resid  12 and name H5' ) (resid  12 and name H1' )  2.6 0.9 2.0
 assign (resid  12 and name H1' ) (resid  12 and name H2' )  2.3 0.9 0.9
 assign (resid  12 and name H2' ) (resid  12 and name H3' )  2.7 0.9 0.9
 assign (resid  12 and name H5' ) (resid  11 and name H1' )  3.5 0.9 0.9
 assign (resid  12 and name H1' ) (resid  12 and name H8  )  3.8 0.9 0.9

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                    Hydrogen Bond Constraints                     !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

assign ( segid A and residue  1 and name N7 ) ( segid B and residue 104 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue  1 and name O6 ) ( segid B and residue 104 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue  1 and name H1 ) ( segid B and residue 109 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue  1 and name H2#) ( segid B and residue 109 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue 12 and name N7 ) ( segid B and residue 109 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue 12 and name O6 ) ( segid B and residue 109 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue 12 and name H1 ) ( segid B and residue 104 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue 12 and name H2#) ( segid B and residue 104 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue  2 and name N7 ) ( segid B and residue 110 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue  2 and name O6 ) ( segid B and residue 110 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue  2 and name H1 ) ( segid B and residue 103 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue  2 and name H2#) ( segid B and residue 103 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue 11 and name N7 ) ( segid B and residue 103 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue 11 and name O6 ) ( segid B and residue 103 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue 11 and name H1 ) ( segid B and residue 110 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue 11 and name H2#) ( segid B and residue 110 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue  3 and name N7 ) ( segid B and residue 102 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue  3 and name O6 ) ( segid B and residue 102 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue  3 and name H1 ) ( segid B and residue 111 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue  3 and name H2#) ( segid B and residue 111 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue 10 and name N7 ) ( segid B and residue 111 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue 10 and name O6 ) ( segid B and residue 111 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue 10 and name H1 ) ( segid B and residue 102 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue 10 and name H2#) ( segid B and residue 102 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue  4 and name N7 ) ( segid B and residue 112 and name H2#)
1.70 0.10 0.10
```

```
assign ( segid A and residue  4 and name O6 ) ( segid B and residue 112 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue  4 and name H1 ) ( segid B and residue 101 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue  4 and name H2#) ( segid B and residue 101 and name N7 )
1.70 0.10 0.10
assign ( segid A and residue  9 and name N7 ) ( segid B and residue 101 and name H2#)
1.70 0.10 0.10
assign ( segid A and residue  9 and name O6 ) ( segid B and residue 101 and name H1 )
1.70 0.10 0.10
assign ( segid A and residue  9 and name H1 ) ( segid B and residue 112 and name O6 )
1.70 0.10 0.10
assign ( segid A and residue  9 and name H2#) ( segid B and residue 112 and name N7 )
1.70 0.10 0.10

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                       Planarity Constraints                      !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

  group
   selection=(
        (residue 1 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 12 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 104 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 109 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5'')))
    weight=5.0
   end

  group
   selection=(
        (residue 2 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 11 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 103 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 110 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5'')))
    weight=5.0
   end

  group
   selection=(
        (residue 3 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 10 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 102 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 111 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5'')))
    weight=5.0
   end

  group
```

```
   selection=(
        (residue 4 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 9 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 101 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5''))
        or (residue 112 and (resname THY or resname CYT or resname GUA or resname ADE or
resname URI) and not (name C#' or name H#' or name O#P or name O#' or name P or name H#T
or name O#T or name H2'' or name H5'')))
    weight=5.0
   end
end
                              {* The planarity energy term needs to be turned on. *}
flags include plan end

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                        NCS Constraints                           !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

ncs restraint
   group
         equiv=(segid A)
         equiv=(segid B)
         weight=200
         sigb=1.0
   end
end

flags include ncs end

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                    Dihedral Constraints                          !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

assign (resid 1  and name O4') (resid  1 and name C1') (resid  1 and name N9) (resid  1
and name C4) 1   60 40 2 !{*syn*}
assign (resid 2  and name O4') (resid  2 and name C1') (resid  2 and name N9) (resid  2
and name C4) 1 -130 40 2 !{*anti*}
assign (resid 3  and name O4') (resid  3 and name C1') (resid  3 and name N9) (resid  3
and name C4) 1   60 40 2
assign (resid 4  and name O4') (resid  4 and name C1') (resid  4 and name N9) (resid  4
and name C4) 1 -100 40 2
assign (resid 5  and name O4') (resid  5 and name C1') (resid  5 and name N1) (resid  5
and name C2) 1 -130 40 2
assign (resid 6  and name O4') (resid  6 and name C1') (resid  6 and name N1) (resid  6
and name C2) 1 -150 40 2
assign (resid 7  and name O4') (resid  7 and name C1') (resid  7 and name N1) (resid  7
and name C2) 1 -170 40 2
assign (resid 8  and name O4') (resid  8 and name C1') (resid  8 and name N1) (resid  8
and name C2) 1 -150 40 2
assign (resid 9  and name O4') (resid  9 and name C1') (resid  9 and name N9) (resid  9
and name C4) 1   60 40 2
assign (resid 10 and name O4') (resid 10 and name C1') (resid 10 and name N9) (resid 10
and name C4) 1 -130 40 2
assign (resid 11 and name O4') (resid 11 and name C1') (resid 11 and name N9) (resid 11
and name C4) 1   60 40 2
assign (resid 12 and name O4') (resid 12 and name C1') (resid 12 and name N9) (resid 12
and name C4) 1 -130 40 2
assign (resid 4  and name O5') (resid  4 and name C5') (resid  4 and name C4') (resid  4
and name C3') 1   60 15 2
assign (resid 5  and name O5') (resid  5 and name C5') (resid  5 and name C4') (resid  5
and name C3') 1   80 15 2
assign (resid 6  and name O5') (resid  6 and name C5') (resid  6 and name C4') (resid  6
and name C3') 1   -180 15 2
assign (resid 8  and name O5') (resid  8 and name C5') (resid  8 and name C4') (resid  8
and name C3') 1   50 15 2
```

```
assign (resid 12 and name O5') (resid 12 and name C5') (resid 12 and name C4') (resid 12
and name C3') 1   60 25 2
assign (resid   4 and name O3') (resid   5 and name   P) (resid   5 and name O5') (resid
5 and name C5') 1  -85 40 2
assign (resid   5 and name O3') (resid   6 and name   P) (resid   6 and name O5') (resid
6 and name C5') 1  125 40 2
assign (resid   6 and name O3') (resid   7 and name   P) (resid   7 and name O5') (resid
7 and name C5') 1  -85 40 2
assign (resid   7 and name O3') (resid   8 and name   P) (resid   8 and name O5') (resid
8 and name C5') 1 -125 15 2
assign (resid   8 and name O3') (resid   9 and name   P) (resid   9 and name O5') (resid
9 and name C5') 1  180 40 2
assign (resid   9 and name O3') (resid  10 and name   P) (resid  10 and name O5') (resid
10 and name C5') 1 -100 40 2
assign (resid   5 and name   P) (resid   5 and name O5') (resid   5 and name C5') (resid
5 and name C4') 1  160 15 2
assign (resid   6 and name   P) (resid   6 and name O5') (resid   6 and name C5') (resid
6 and name C4') 1 -130 15 2
assign (resid   7 and name   P) (resid   7 and name O5') (resid   7 and name C5') (resid
7 and name C4') 1  180 40 2
assign (resid   8 and name   P) (resid   8 and name O5') (resid   8 and name C5') (resid
8 and name C4') 1 -130 15 2
assign (resid   9 and name   P) (resid   9 and name O5') (resid   9 and name C5') (resid
9 and name C4') 1  130 40 2
```

10.2 Appendix 2 Hardware Setup for ^{205}Tl NMR Experiments

10.2.1 ^{205}Tl direct detection experiments.

The Nalorac direct detection probe is used for ^{205}Tl direct detection experiments. To achieve the ^{205}Tl frequency, inductor I4 is used in CAP 1 (Figure 10-1). The ^{1}H channel is tuned as normal and the ^{205}Tl channel is tuned using an oscilloscope until the reflected power is less than 10% of the forward power. If the ^{1}H channel is not properly tuned, it will greatly increase the ^{205}Tl pulse length.

At 11.75 T, the ^{205}Tl frequency (288 MHz) is above the cut off for channel B, so the input and output must be routed through channel A (Figure 10-2). This means that the cables must be changed back to detect on ^{1}H, including to gradient shim. The spectrometer must be set to "Linear Fullband" for amplifier 2 in the VNMR configuration panel.

10.2.2 ^{1}H–^{205}Tl experiments.

The Nalorac indirect detection probe is used for ^{205}Tl direct detection experiments. To achieve the ^{205}Tl frequency, the homemade inductor is used in CAP 1 (Figure 10-3). The ^{1}H channel is tuned as normal and the ^{205}Tl channel (X1) is tuned using an oscilloscope until the reflected power is less than 10% of the forward power. The addition of the inductor will have a small, adverse effect on the ^{1}H tuning as the channels are slightly coupled to each other.

Performing simultaneous pulsing on ^{1}H and ^{205}Tl requires two high band amplifiers because of the previously mentioned frequency limitation on the lowband channel. The hookup for the first amplifier (^{1}H) is identical to the default configuration. The second amplifier (borrowed from Kurt Zilm) requires a very different setup. The

^{205}Tl signal must be driven by a separate interface cable, which requires that it enter the amplifier through channel 3. However, direct detection (required for pulse calibration) can only be performed using channels 1 and 2. For this reason, it is routed out of the amplifier as channel 2 (Figure 10-4). The spectrometer must be set to "Linear Fullband" for amplifiers 2 and 3 in the VNMR configuration panel. Direct detection is used to calibrate the ^{205}Tl pulse length.

Changes also have to be made to the inside of the second amplifier because the spectrometer will only route the signal for the third channel through the channel B input/outputs. Thus, channel B on the second amplifier has to be converted to channel A, requiring changes to be made inside the amplifier (Figure 10-5). The input and output for the longer amplifier region (channel B, right) are switched to the shorter amplifier region (channel A, middle). Lastly, two switches (#7) on the circuit boards (bottom left) have to be reversed.

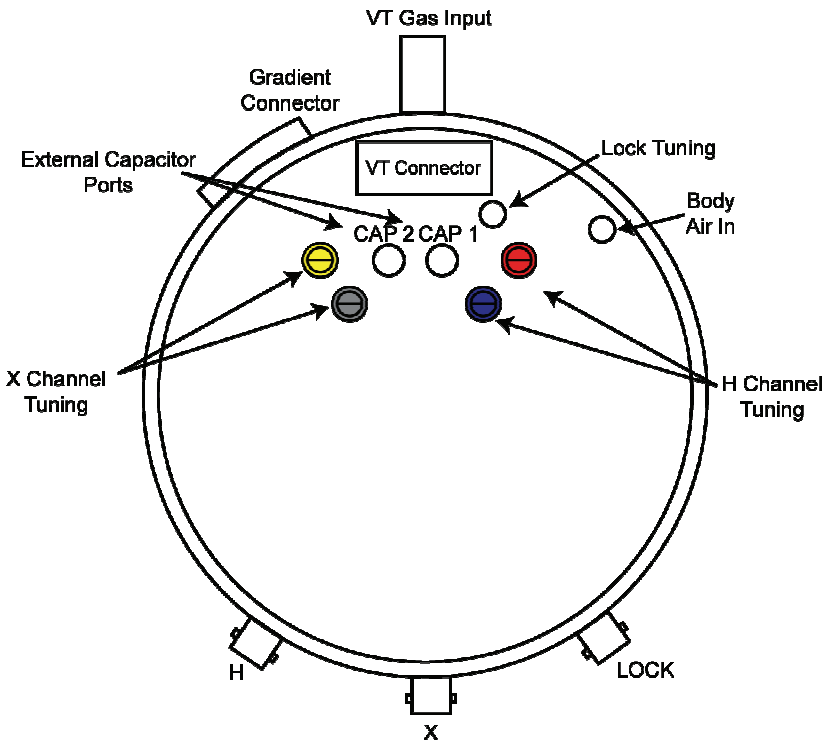


Figure 10-1. Bottom view of the Nalorac dual broad band probe used for ^{205}Tl direct detection experiments.

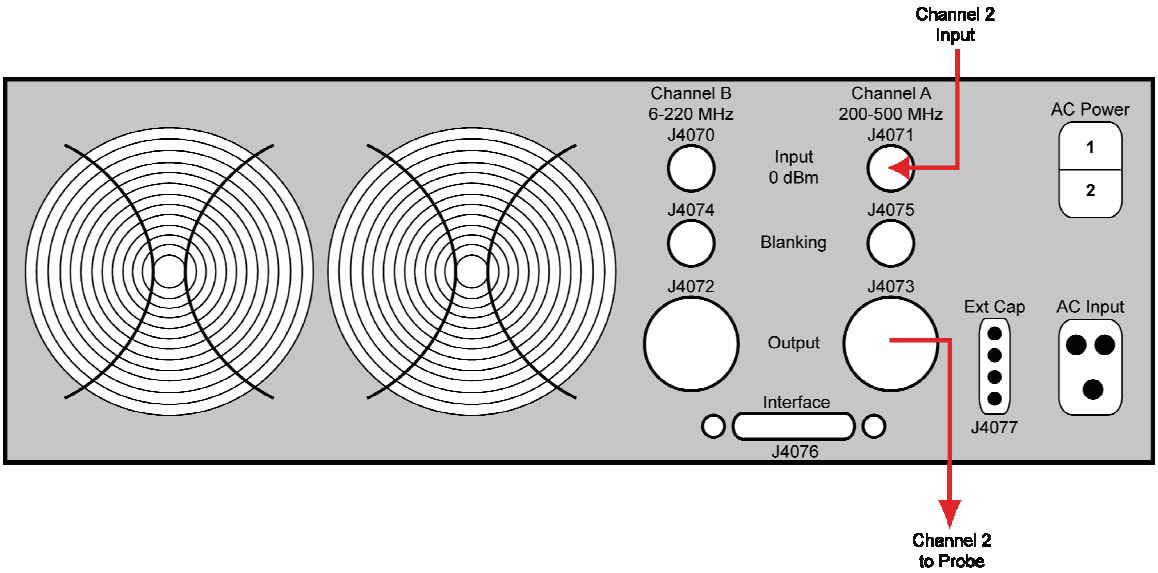


Figure 10-2. Back view of the amplifier setup for ^{205}Tl direct detection.

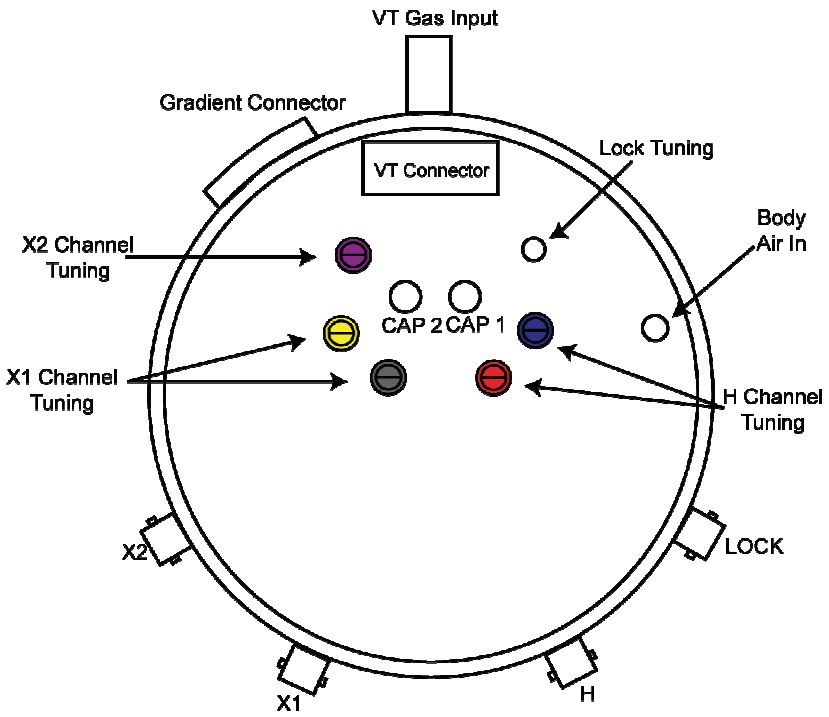


Figure 10-3. Bottom view of the Nalorac quad channel probe used for ^{1}H–^{205}Tl experiments.

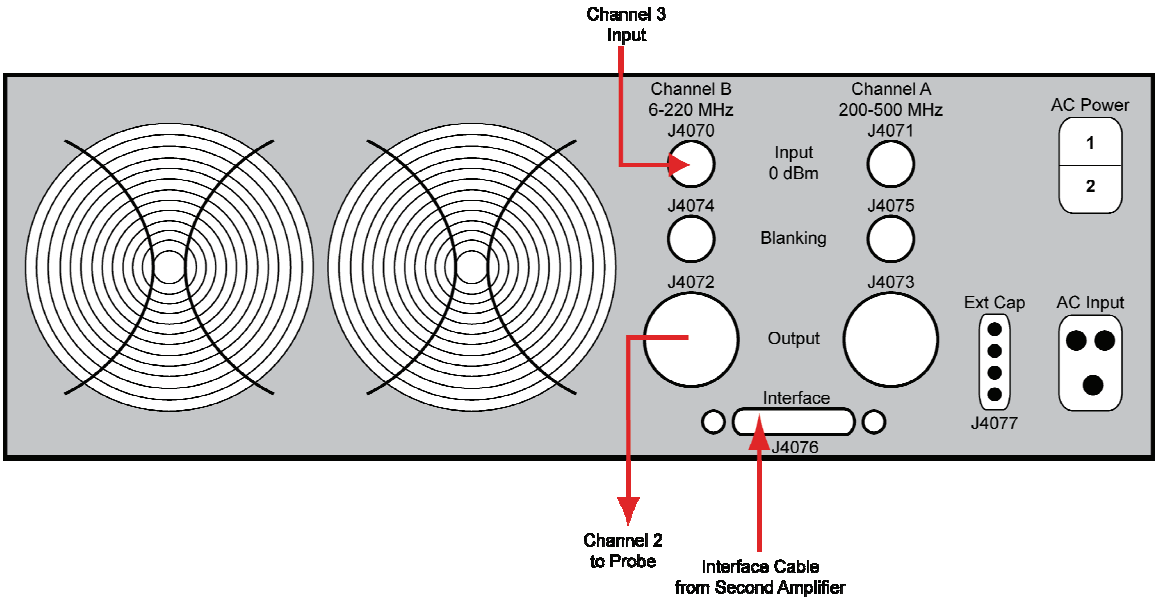


Figure 10-4. Back view of the setup for the second amplifier used for ^{1}H–^{205}Tl experiments.

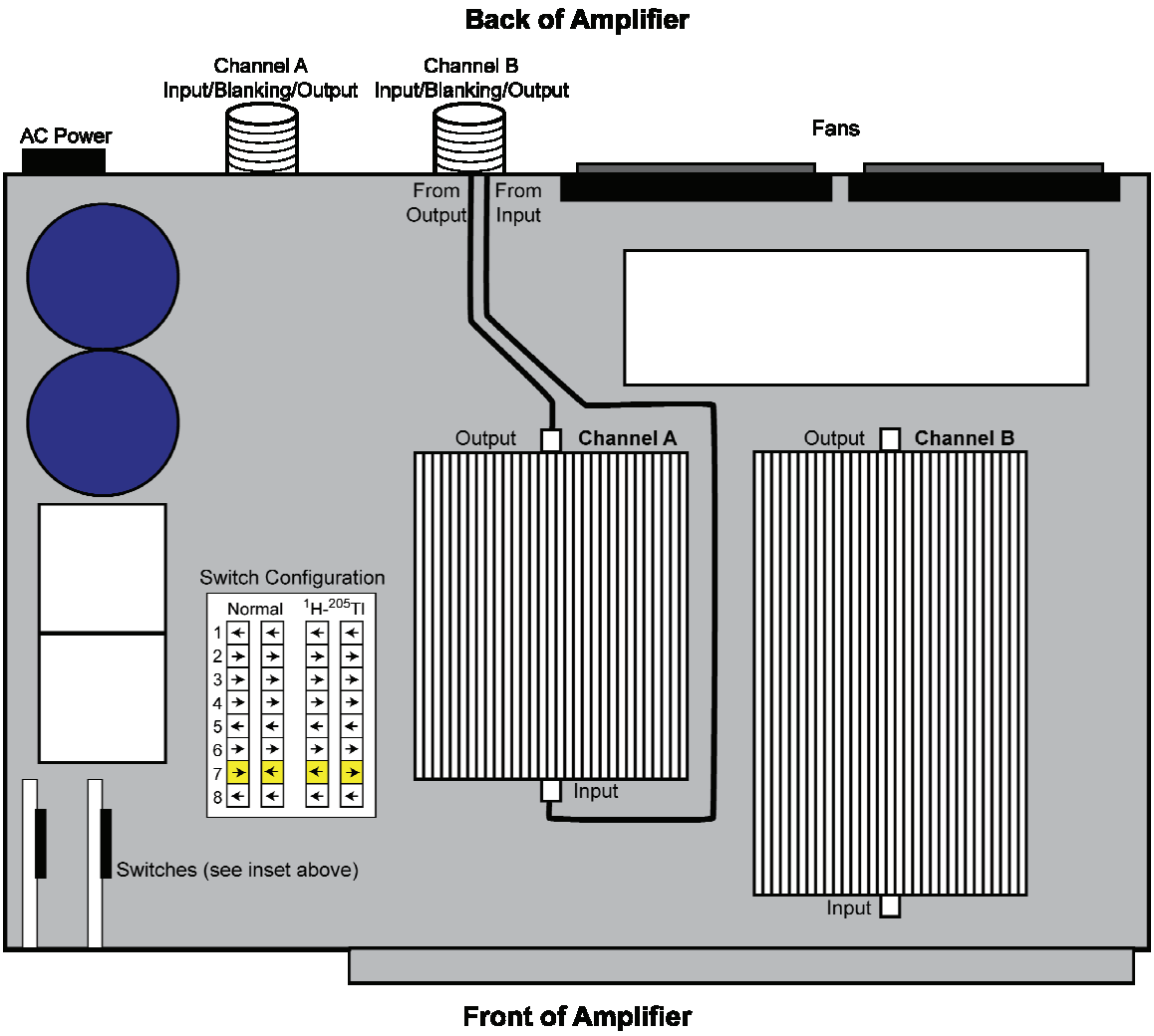


Figure 10-5. Inside of the second amplifier used for ^{1}H–^{205}Tl experiments.

10.3 Appendix 3 Calculation of Magnetic Susceptibility Tensor (Δχ)

10.3.1 Selected portions of the input file.

```
mem=10MW
%chk=xraymod
%Nproc=2
 # oniom=(b3lyp/6-31g*:amber)=embed geom=connectivity scf=direct
 nosymm NMR=PrintEigenvectors pop=chelpg maxdisk=10gb Iop(1/64=203) IOp(10/6=1)  test

 oniom from tinker file = xraymod.xyz

 5 1 0  1
 K-K+-1.00000      0    -0.457883   22.749133   53.513761 L
 K-K+-1.00000      0     2.669143   19.844931   59.213543 L
 K-K+-1.00000      0     4.006580   18.632929   62.062616 L
 K-K+-1.00000      0     1.379118   21.130066   56.297439 L
 K-K+-1.00000      0     5.457196   16.895779   64.671957 L

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
THE REMAINDER OF THE LOGFILE HAS BEEN REMOVED FOR BREVITY
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
```

10.3.2 Selected portions of the output file.

```
Entering Gaussian System, Link 0=g03
 Input=g4xray_c3magsep.com
 Output=g4xray_c3magsep.log
 Initial command:
 /usr/local/g03/l1.exe /home/christina/Gau-20661.inp -scrdir=/home/christina/
 Entering Link 1 = /usr/local/g03/l1.exe PID=     20662.

 Copyright (c) 1988,1990,1992,1993,1995,1998,2003, Gaussian, Inc.
                  All Rights Reserved.

 This is the Gaussian(R) 03 program.  It is based on the
 the Gaussian(R) 98 system (copyright 1998, Gaussian, Inc.),
 the Gaussian(R) 94 system (copyright 1995, Gaussian, Inc.),
 the Gaussian 92(TM) system (copyright 1992, Gaussian, Inc.),
 the Gaussian 90(TM) system (copyright 1990, Gaussian, Inc.),
 the Gaussian 88(TM) system (copyright 1988, Gaussian, Inc.),
 the Gaussian 86(TM) system (copyright 1986, Carnegie Mellon
 University), and the Gaussian 82(TM) system (copyright 1983,
 Carnegie Mellon University). Gaussian is a federally registered
 trademark of Gaussian, Inc.

 This software contains proprietary and confidential information,
 including trade secrets, belonging to Gaussian, Inc.

 This software is provided under written license and may be
 used, copied, transmitted, or stored only in accord with that
 written license.

 The following legend is applicable only to US Government
 contracts under DFARS:

                    RESTRICTED RIGHTS LEGEND

 Use, duplication or disclosure by the US Government is subject
 to restrictions as set forth in subparagraph (c)(1)(ii) of the
 Rights in Technical Data and Computer Software clause at DFARS
 252.227-7013.

 Gaussian, Inc.
 Carnegie Office Park, Building 6, Pittsburgh, PA 15106 USA

 The following legend is applicable only to US Government
```

```
contracts under FAR:

                    RESTRICTED RIGHTS LEGEND

Use, reproduction and disclosure by the US Government is subject
to restrictions as set forth in subparagraph (c) of the
Commercial Computer Software - Restricted Rights clause at FAR
52.227-19.

Gaussian, Inc.
Carnegie Office Park, Building 6, Pittsburgh, PA 15106 USA


---------------------------------------------------------------
Warning -- This program may not be used in any manner that
competes with the business of Gaussian, Inc. or will provide
assistance to any competitor of Gaussian, Inc.  The licensee
of this program is prohibited from giving any competitor of
Gaussian, Inc. access to this program.  By using this program,
the user acknowledges that Gaussian, Inc. is engaged in the
business of creating and licensing software in the field of
computational chemistry and represents and warrants to the
licensee that it is not a competitor of Gaussian, Inc. and that
it will not use this program in any manner prohibited above.
---------------------------------------------------------------


Cite this work as:
Gaussian 03, Revision B.04,
M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria,
M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven,
K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi,
V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega,
G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota,
R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao,
H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross,
C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev,
A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala,
K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg,
V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain,
O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari,
J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford,
J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz,
I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham,
C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill,
B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, and J. A. Pople,
Gaussian, Inc., Pittsburgh PA, 2003.

******************************************
Gaussian 03:  x86-Linux-G03RevB.04 2-Jun-2003
                 20-Sep-2005
******************************************
%mem=10MW
%chk=xraymod
%Nproc=2
Will use up to    2 processors via shared memory.
-----------------------------------------------------------------------
# oniom=(b3lyp/6-31g*:amber)=embed geom=connectivity scf=direct nosymm
 NMR=PrintEigenvectors pop=chelpg maxdisk=10gb Iop(1/64=203) IOp(10/6=
1) test
-----------------------------------------------------------------------
1/30=1,38=1,52=12,56=2,57=2,64=203,74=500/1;
2/15=1,17=6,18=5,40=1/2;
1/30=1,38=1,52=12,53=3172,64=203,74=500/20;
3/5=2,7=1,11=9,16=1,25=1,30=1/1;
4/20=11,24=3/2;
1/52=12,53=2032,64=203,74=500/20;
3/5=1,6=6,7=1,11=2,16=1,25=1,30=1,74=-5/1,2,8,3;
4//1;
5/5=2,38=5,94=2/2;
8/6=1,10=90,11=11,27=1342177280/1;
```

```
 10/6=1,13=100,31=1,45=16,75=2/2;
 6/7=2,8=2,9=2,10=2,15=8,20=3,28=1/1,2;
 1/30=1,52=12,53=1022,64=203,74=500/20;
 3/5=2,7=1,11=9,16=1,25=1,30=1/1;
 4/20=11,24=3/2;
 1/30=1,52=12,53=3014,64=203,74=500/20;
 99/9=1/99;
 ------------------------------------
 oniom from tinker file = xraymod.xyz
 ------------------------------------

&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
  PORTIONS OF THE LOGFILE HAVE BEEN REMOVED FOR BREVITY
&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&

Magnetic susceptibility tensor (au):
   XX=    -9.1756   YX=     1.7355   ZX=    -2.1460
   XY=     1.7355   YY=   -13.8616   ZY=     3.9016
   XZ=    -2.1460   YZ=     3.9016   ZZ=   -18.4263
   Isotropic susceptibility =   -13.8212
 Magnetic susceptibility tensor (10**(-30) J/T**2):
   XX=  -724.0499   YX=   136.9452   ZX=  -169.3433
   XY=   136.9452   YY= -1093.8267   ZY=   307.8752
   XZ=  -169.3433   YZ=   307.8752   ZZ= -1454.0256
   Isotropic susceptibility = -1090.6341
 Magnetic susceptibility tensor (cgs-ppm):
   XX=   -43.6033   YX=     8.2470   ZX=   -10.1981
   XY=     8.2470   YY=   -65.8718   ZY=    18.5407
   XZ=   -10.1981   YZ=    18.5407   ZZ=   -87.5635
   Isotropic susceptibility =   -65.6795
```

10.4 Appendix 4 Global Fast CPMG Perl Program

10.4.1 Sample input data from XMGR file.

```
# CurveFit 1.22
#
# title CPMG Dispersion Curve for S15
# function CPMG
# equation y=R2+Rex*(1-2*Tau*x*tanh(
# points       10
# X2   2.2971
# X2(red)      0.3282
#
# Parameter    Fitted_Value  Fitted_Error    Sim_value      Sim_error
# R2    15.5184 1.1331  15.4266 0.2121
# Rex   5.2497  0.9926  4.7454  0.2705
# kex   2.7806  0.1579  3.2180  0.1249
#
@ CLEAR STRING
@ TITLE "CPMG Dispersion Curve for S15"
@ SUBTITLE "y=R2+Rex*(1-2*Tau*x*tanh(1/(2*Tau*x)))"
@ VIEW XMIN 0.20
@ VIEW XMAX 0.90
@ VIEW YMIN 0.35
@ VIEW YMAX 0.85
@ XAXIS LABEL "1/tcp (1/ms)"
@ YAXIS LABEL "R2(tcp)  (1/s)"
@ XAXIS TICKLABEL FORMAT DECIMAL
@ YAXIS TICKLABEL FORMAT DECIMAL
@ XAXIS TICKLABEL CHAR SIZE 0.8
@ YAXIS TICKLABEL CHAR SIZE 0.8
@ WORLD XMIN   0.000E+00
@ WORLD XMAX   0.200E+01
@ XAXIS TICK MAJOR     0.500E+00
@ XAXIS TICK MINOR     0.250E+00
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.215
@ STRING DEF "R2       15.5184 +/-    0.2121"
@ STRING on
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.180
@ STRING DEF "Rex      5.2497 +/-     0.2705"
@ STRING on
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.145
@ STRING DEF "kex      2.7806 +/-     0.1249"
@ STRING on
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.110
@ STRING DEF "X2       2.2971"
@ STRING on
@ S0 TYPE xydy
@ S0  LINESTYLE 0
@ S0  SYMBOL 2
@ S0  SYMBOL SIZE 0.5
@ S0  SYMBOL FILL 1
@ S0  ERRORBAR LENGTH 0.5
```

```
@ S1 TYPE xy
@ S1  ERRORBAR LENGTH 0
@ S1  COLOR 1
        0.160E+01      0.164E+02      0.481E+00
        0.140E+01      0.166E+02      0.489E+00
        0.120E+01      0.174E+02      0.523E+00
        0.100E+01      0.175E+02      0.505E+00
        0.800E+00      0.179E+02      0.576E+00
        0.600E+00      0.191E+02      0.584E+00
        0.400E+00      0.191E+02      0.619E+00
        0.300E+00      0.198E+02      0.576E+00
        0.200E+00      0.197E+02      0.624E+00
        0.100E+00      0.205E+02      0.518E+00
&
        0.00000E+00    0.20768E+02    0.00000E+00
        0.20000E-01    0.20693E+02    0.00000E+00
        0.40000E-01    0.20617E+02    0.00000E+00
        0.60000E-01    0.20542E+02    0.00000E+00
        0.80000E-01    0.20466E+02    0.00000E+00
        0.10000E+00    0.20390E+02    0.00000E+00
        0.12000E+00    0.20315E+02    0.00000E+00
        0.14000E+00    0.20239E+02    0.00000E+00
        0.16000E+00    0.20164E+02    0.00000E+00
        0.18000E+00    0.20088E+02    0.00000E+00
        0.20000E+00    0.20013E+02    0.00000E+00
        0.22000E+00    0.19937E+02    0.00000E+00
        0.24000E+00    0.19862E+02    0.00000E+00
        0.26000E+00    0.19786E+02    0.00000E+00
        0.28000E+00    0.19711E+02    0.00000E+00
        0.30000E+00    0.19635E+02    0.00000E+00
        0.32000E+00    0.19560E+02    0.00000E+00
        0.34000E+00    0.19485E+02    0.00000E+00
        0.36000E+00    0.19410E+02    0.00000E+00
        0.38000E+00    0.19335E+02    0.00000E+00
        0.40000E+00    0.19261E+02    0.00000E+00
        0.42000E+00    0.19186E+02    0.00000E+00
        0.44000E+00    0.19113E+02    0.00000E+00
        0.46000E+00    0.19039E+02    0.00000E+00
        0.48000E+00    0.18967E+02    0.00000E+00
        0.50000E+00    0.18895E+02    0.00000E+00
        0.52000E+00    0.18823E+02    0.00000E+00
        0.54000E+00    0.18753E+02    0.00000E+00
        0.56000E+00    0.18683E+02    0.00000E+00
        0.58000E+00    0.18614E+02    0.00000E+00
        0.60000E+00    0.18546E+02    0.00000E+00
        0.62000E+00    0.18479E+02    0.00000E+00
        0.64000E+00    0.18413E+02    0.00000E+00
        0.66000E+00    0.18349E+02    0.00000E+00
        0.68000E+00    0.18285E+02    0.00000E+00
        0.70000E+00    0.18223E+02    0.00000E+00
        0.72000E+00    0.18161E+02    0.00000E+00
        0.74000E+00    0.18101E+02    0.00000E+00
        0.76000E+00    0.18042E+02    0.00000E+00
        0.78000E+00    0.17985E+02    0.00000E+00
        0.80000E+00    0.17929E+02    0.00000E+00
        0.82000E+00    0.17874E+02    0.00000E+00
        0.84000E+00    0.17820E+02    0.00000E+00
        0.86000E+00    0.17767E+02    0.00000E+00
        0.88000E+00    0.17716E+02    0.00000E+00
        0.90000E+00    0.17666E+02    0.00000E+00
        0.92000E+00    0.17617E+02    0.00000E+00
        0.94000E+00    0.17569E+02    0.00000E+00
        0.96000E+00    0.17523E+02    0.00000E+00
        0.98000E+00    0.17477E+02    0.00000E+00
        0.10000E+01    0.17433E+02    0.00000E+00
        0.10200E+01    0.17390E+02    0.00000E+00
        0.10400E+01    0.17348E+02    0.00000E+00
```

10.4.2 Perl program.

```
1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5
6  ##################################################################
7  ##################################################################
8  ##                Global CPMG Fit                                ##
9  ##   Simultaneously fits Xmgr residue files generated by         ##
10 ##   Curvefit to the fast exchange equation.                     ##
11 ##   This program uses the Curvefit results to get starting      ##
12 ##   input values for the Mathematica notebook.  The             ##
13 ##   notebook fits the data, generating a residuals file.        ##
14 ##   This residuals file can then be used to calculated X2       ##
15 ##                                                               ##
16 ##                Michelle L. Gill and J. Patrick Loria          ##
17 ##                Yale University, 12/09/2004                    ##
18 ##                Mathematica notebook was produced by           ##
19 ##                Michael Grey (Columbia University) and         ##
20 ##                subsequently modified by JPL                   ##
21 ##                                                               ##
22 ##  This software is provided "as is" without any express        ##
23 ##  warranty, etc.  If you make any significant improvements     ##
24 ##  to this program, please send the authors a copy with the     ##
25 ##  annotated improvements.                                      ##
26 ##################################################################
27 ##################################################################
28
29
30 ##################################################################
31 ##          Switches for running the program                     ##
32 ##################################################################
33
34 # Removed leading "-" and put switches in array
35 my @switches;
36
37 foreach my $a (@ARGV) {
38        chomp($a);
39        my @tmp1 = split /-/, $a;
40        my @tmp2 = split //, $tmp1[1];
41        foreach my $temp_switch (@tmp2) {
42              push @switches, $temp_switch;
43        }
44 }
45
46 my $force = 0;
47 my $calc_residuals = 0;
48 if (!@switches) {
49        print "\n\n\tDefault: running in safemode and generating Mathematica notebook.\n";
50        print "\tTry \"./global_cpMG_fast.pl -h\" for more information\n";
51        print "\t--including how to perform CHI^2 calculation.\n";
52 } else {
53        foreach my $switch (@switches) {
54              if ($switch =~ /h/ || $switch =~ /H/) {
55                    &prhelp();
56              } elsif ($switch =~ /f/ || $switch =~ /F/) {
57                    $force = 1;
58              } elsif ($switch =~ /r/ || $switch =~ /R/) {
59                    $calc_residuals = 1;
60              } else {
61                    print "\tThe switch ($switch) has no meaning.\n";
62                    print "\tTry \"./global_cpMG_fit.pl -h\" for more information.\n\n";
63                    exit;
64              }
65        }
66 }
67
68 sub prhelp() {
69    print "\n";
70    print "\tUsage:    ./global_cpMG_fit.pl -[Flags]\n";
71    print "\tDetails:  Generates a Mathematica notebook to globally\n";
72    print "\t          fit Curvefit-generated XMGR files to the fast\n";
73    print "\t          exchange equation.  A second run of the program\n";
```

```
74      print "\t        with the appropriate flag with calculate residuals.\n";
75      print "\tFlags:   -f  force mode, disables file checks\n";
76      print "\t            and doesn't check for generation of new\n";
77      print "\t            \"globalfast_results\" file.\n";
78      print "\t            (safemode is the default)\n";
79      print "\t        -r  calculate residuals from global fit\n";
80      print "\t            (can only be done after Mathematica notebook\n";
81      print "\t            has been evaluated and saved)\n";
82      print "\t        -h  displays this help information\n\n";
83      exit;
84  }
85
86  if ($calc_residuals == 0) {
87          print "\n";
88          print "\t*******************************************************\n";
89          print "\t*******************************************************\n";
90          print "\t**                Global CPMG Fit                **\n";
91          print "\t** Starting generation of 'cpMG_global_fast.nb'. **\n";
92          print "\t**                        by MGill and JPLoria   **\n";
93          print "\t*******************************************************\n";
94          print "\t*******************************************************\n";
95          print "\n";
96  } else {
97          print "\n";
98          print "\t*******************************************************\n";
99          print "\t*******************************************************\n";
100         print "\t**                Global CPMG Fit                **\n";
101         print "\t** Starting calculation of global residuals      **\n";
102         print "\t**                        by MGill and JPLoria   **\n";
103         print "\t*******************************************************\n";
104         print "\t*******************************************************\n";
105         print "\n";
106  }
107
108
109  ##################################################################
110  ##      Determine current directory and all *.xmgr files       ##
111  ##################################################################
112
113  use Cwd;
114  my $cur_dir = getcwd();
115  my @resi_files = <*.xmgr>;
116  @resi_files = glob("*.xmgr");
117
118  # Strip off the .xmgr suffix
119  foreach my $resi_num (@resi_files) {
120          my @xmgr_tmp = split /\./, $resi_num;
121          $resi_num = $xmgr_tmp[0];
122  }
123
124  # foreach $file (@xmgr_files) {
125  #       print "$file\n";
126  # }
127  # print "scalar $#xmgr_files\n";
128
129  if ($calc_residuals == 0) {
130          # Exit program if there are no *.xmgr files in this directory
131          if (!(@resi_files)) {
132                  print "\tCurrent working directory $cur_dir\n";
133                  print "\tcontains no *.xmgr files.\n";
134                  print "\tEither there are no xmgr files in this directory\n";
135                  print "\tor the files are misnamed.\n";
136                  print "\tThe program has exited.\n\n";
137                  exit;
138          }
139
140          # Can't do a global fit if only one residue file exists
141          if ($#resi_files == 0) {
142                  print "\tCurrent working directory $cur_dir\n";
143                  print "\tcontains only one *.xmgr file.\n";
144                  print "\tA global fit cannot be performed on only one file.\n";
145                  print "\tThe program has exited.\n\n";
146                  exit;
147          }
```

```
148
149         # Determine that we don't have any files from CPMGfit
150         my $file_type_error = 0;
151         my %file_errors;
152         foreach my $resi_num (@resi_files) {
153                 open (FILE_CHECK, "< $resi_num.xmgr");
154                 my $program_type = <FILE_CHECK>;
155
156                 if ($program_type =~ /CPMGfit/) {
157                         $file_type_error = 1;
158                         $file_errors{$resi_num} = 1;
159                 } else {
160                         $file_errors{$resi_num} = 0;
161                 }
162         }
163
164         # If there are CPMGfit files, print them out and exit the program
165         if ($file_type_error == 1) {
166                 print "\n\tIt looks like the following XMGR file(s) are from\n";
167                 print "\tCPMGfit: ";
168                 foreach my $resi_num (@resi_files) {
169                         if ($file_errors{$resi_num} == 1) {
170                                 print "$resi_num  ";
171                         }
172                 }
173                 print "\n";
174                 print "\tThese input files won't work with this program.\n";
175                 print "\tPlease reprocess your data using CurveFit.\n";
176                 print "\tIf you wish to ignore these files please rename\n";
177                 print "\tthem so they do not end in .xmgr.\n";
178                 print "\tProgram will now exit.\n\n";
179                 exit;
180         }
181 }
182
183 # Can't calculate residuals if the notebook hasn't been generated and evaluated
184 if (($calc_residuals == 1) && !(-e "globalfast_results")) {
185         print "\n\tYou cannot calculate residuals without first creating\n";
186         print "\tand evaluating the Mathematica notebook \"cpMG_global_fast.nb\".\n";
187         print "\tEvaluation of this notebook will produce the necessary file: \n";
188         print "\t\"globalfast_results\".\n";
189         print "\tThe program has exited.\n\n";
190         exit;
191 }
192
193
194 ##################################################################
195 ##  Safety checks to prevent ovewriting of *.in and *.nb files ##
196 ##################################################################
197
198 if ($force == 0) {
199         if ($calc_residuals == 0) {
200                 # If *.in files exist, ask if they can be overwritten
201                 my @in_files = <*.in>;
202                 @in_files = glob("*.in");
203                 if (@in_files) {
204                         print "\n\tThere are already *.in files in this directory.\n";
205                         print "\tIs it OK to overwrite these files? (y|n) ";
206                         my $overwrite = <STDIN>;
207                         if (!($overwrite =~ /Y/) && !($overwrite =~ /y/)) {
208                                 print "\n\tThe program exited without overwriting *.in files.\n\n";
209                                 exit;
210                         }
211                 }
212
213                 # Check for mathematica notebook
214                 if (-e "cpMG_global_fast.nb") {
215                         print "\n\tMathematica notebook cpMG_global_fast.nb already exists.\n";
216                         print "\tIs it OK to overwrite this file? (y|n) ";
217                         my $overwrite = <STDIN>;
218                         if (!($overwrite =~ /Y/) && !($overwrite =~ /y/)) {
219                                 print "\n\tThe program exited without overwriting notebook.\n\n";
220                                 exit;
221                         }
```

```
222             }
223         }
224
225         if ($calc_residuals == 1) {
226             # Check for residual calculation file
227             if (-e "globalfast_chisq") {
228                 print "\n\tResidual calculation file \"globalfast_chisq\" already exists.\n";
229                 print "\tIs it OK to overwrite this file? (y|n) ";
230                 my $overwrite = <STDIN>;
231                 if (!($overwrite =~ /Y/) && !($overwrite =~ /y/)) {
232                     print "\n\tThe program exited without overwriting residual file.\n\n";
233                     exit;
234                 }
235             }
236
237             # Check to see if Mathematica notebook has been generated but not evaluated
238             # Allows for about 1 min between evaluation(generating globalfast_results) and
239             # saving the notebook (hence the -M switch and the difference).  The value is
240             # in fractions of days
241
242             #if ((-M "cpMG_global_fast.nb") < (-M "globalfast_results")) {
243             if (((-M "globalfast_results")-(-M "cpMG_global_fast.nb")) > 0.0007) {
244                 print "\n\tThe notebook \"cpMG_global_fast.nb\" may be newer than the file\n";
245                 print "\t\"globalfast_results\".  This means you may have generated a new\n";
246                 print "\tMathematica notebook without evaluating and saving it.\n";
247                 print "\tDo you still want to force the residual calculation? (y|n) ";
248                 my $force_resid = <STDIN>;
249                 if (!($force_resid =~ /y/ || $force_resid =~ /Y/)) {
250                     print "\n\tThe program exited without calculating residuals.\n\n";
251                     exit;
252                 }
253             }
254         }
255 }
256
257
258 ##################################################################
259 ##     Check *.xmgr files.                                      ##
260 ##################################################################
261
262 # # Strip off the .xmgr suffix
263 # foreach $resi_num (@resi_files) {
264 #     @xmgr_tmp = split /\./, $resi_num;
265 #     $resi_num = $xmgr_tmp[0];
266 # }
267
268 # Sort by residue numbers
269 my $i = 0;
270 my %temp_resi;
271 foreach my $resi_num (sort { $a <=> $b } @resi_files) {
272     $temp_resi{$i} = $resi_num;
273     $i++;
274 }
275 for (my $j = 0; $j < $i; $j++) {
276     $resi_files[$j] = $temp_resi{$j};
277 }
278
279 #Print all residue file names
280 my $total_files = $#resi_files+1;
281 if ($calc_residuals == 0) {
282     print "\n\tXmgr files ($total_files) found:";
283     $i = 1;
284     foreach my $resi_num (@resi_files) {
285         printf " %3d", $resi_num;
286         if ($i%10 == 0) {
287             print "\n\t                    ";
288         }
289         $i++;
290     }
291     print "\n";
292
293     # Check to make sure files were read in correctly
294     print "\tIs this correct? (y|n) ";
295     my $correct = <STDIN>;
```

```
296
297         # Exit program if files are incorrect
298         if (!($correct =~ /Y/) && !($correct =~ /y/)) {
299             print "\n\tFiles have been determined to be incorrect.\n";
300             print "\tProgram will now exit.\n\n";
301             exit;
302         }
303     }
304
305     ##################################################################
306     ##    B0, Bref needed from user input only if generating      ##
307     ##    Mathematica notebook.                                   ##
308     ##################################################################
309     my $B0;
310     my $Bref;
311
312     if ($calc_residuals == 0) {
313         #TODO Add a check for field strength entry--must be real number
314         # Get B0 from user input, allow for input mistakes
315         my $correct_field = 0;
316         while ($correct_field == 0) {
317             print "\n\tEnter the magnetic field strength (B0, in Teslas): ";
318             $B0 = <STDIN>;
319             chomp($B0);
320             print "\tYou have entered $B0 Tesla(s).\n";
321             print "\tIs this correct? (y|n) ";
322             my $field_check = <STDIN>;
323             if ($field_check =~ /Y/) || ($field_check =~ /y/)) {
324                 $correct_field = 1;
325             }
326             if ($B0 <= 0) {
327                 print "\tB0 must be >= 0.  Please try again.\n";
328                 $correct_field = 0;
329             }
330
331         }
332
333         # Get reference field from user
334         $correct_field = 0;
335         while ($correct_field == 0) {
336             print "\n\tWould you like to use a reference field (Bref)? (y|n) ";
337             my $use_Bref = <STDIN>;
338             if ($use_Bref =~ /N/) || ($use_Bref =~ /n/)) {
339                 print "\tSetting Bref = B0.\n";
340                 $Bref = $B0;
341                 $correct_field = 1;
342             } elsif ($use_Bref =~ /Y/) || ($use_Bref =~ /y/)) {
343                 print "\tEnter the reference field strength (B0, in Teslas): ";
344                 $Bref = <STDIN>;
345                 chomp($Bref);
346                 print "\tYou have entered $Bref Tesla(s).\n";
347                 print "\tIs this correct? (y|n) ";
348                 my $field_check = <STDIN>;
349                 if ($field_check =~ /Y/) || ($field_check =~ /y/)) {
350                     $correct_field = 1;
351                 }
352                 if ($Bref <= 0) {
353                     print "\tBref must be >= 0.  Please try again.\n";
354                     $correct_field = 0;
355                 }
356             } else {
357                 print "\n\tYour answer doesn't make sense.  Please enter \"y\" or \"n\".";
358             }
359         }
360     }
361
362
363     ##################################################################
364     ##    Begin reading in information from all *.xmgr files      ##
365     ##################################################################
366     my %resi_ID;
367     my %points;
368     my %chi_sq_CF;
369     my %chi_sq_CF_red;
```

```
370 my %R20;
371 my %Rex0;
372 my %kex0;
373 my %tcp;
374 my %R2obs;
375 my %R2_chi;
376 my %R2err;
377 my %R2obsMAX;
378
379 foreach my $resi_num (@resi_files) {
380         open (XMGR, "< $resi_num.xmgr");
381         LINE: while (my $input = <XMGR>) {
382
383                 # Find the name and number of the residue
384                 if ($input =~ /# title    CPMG Dispersion Curve/) {
385                         chomp($input);
386                         my @split_line = split /\s+/, $input;
387                         my @abbrev_temp = split /\d/, $split_line[$#split_line];
388                         $resi_ID{$resi_num} = lcfirst($abbrev_temp[0]).$resi_num;
389                 }
390
391                 # Find the number of points to fit
392                 if ($input =~ /# points/) {
393                         chomp($input);
394                         my @split_line = split /\s+/, $input;
395                         $points{$resi_num} = $split_line[2];
396                 }
397
398                 # Find the X2 (CHI^2) value from Curvefit
399                 if (($input =~ /# X2/) && !($input =~ /red/)) {
400                         chomp($input);
401                         my @split_line = split /\s+/, $input;
402                         $chi_sq_CF{$resi_num} = $split_line[2];
403                 }
404
405                 # Find the X2 (CHI^2) (reduced) value from Curvefit
406                 if ($input =~ /# X2\(red\)/) {
407                         chomp($input);
408                         my @split_line = split /\s+/, $input;
409                         $chi_sq_CF_red{$resi_num} = $split_line[2];
410                 }
411
412                 # Find R20, Rex0, and kex0
413                 if ($input =~ /# R2/) {
414                         chomp($input);
415                         my @split_line = split /\s+/, $input;
416                         $R20{$resi_num} = $split_line[2];       #R20
417
418                         $input = <XMGR>;
419                         chomp($input);
420                         @split_line = split /\s+/, $input;
421                         $Rex0{$resi_num} = $split_line[2];      #Rex0
422
423                         $input = <XMGR>;
424                         chomp($input);
425                         @split_line = split /\s+/, $input;
426                         if ($split_line[1] =~ /Tau/) {
427                                 # Tau version of Curvefit
428                                 $kex0{$resi_num} = 1.0/$split_line[2];
429                         } else {
430                                 # kex0 version of Curvefit
431                                 $kex0{$resi_num} = $split_line[2];
432                         }
433                 }
434
435                 # Get values for tcp, R2obs, and R2err
436                 # NOTE tcp is originally in (ms)^(-1), must convert to (s)
437                 if ($input =~ /@ S1  COLOR/) {
438                         for (my $i = 0; $i < $points{$resi_num}; $i++) {
439                                 $input = <XMGR>;
440                                 chomp($input);
441                                 my @split_line = split /\s+/, $input;
442                                 $tcp{$resi_num}[$i] = 1/($split_line[1]*1000);          #tcp
443                                 $R2obs{$resi_num}[$i] = $split_line[2];         #R2obs
```

```
444                     if ($i == ($points{$resi_num}-1)) {
445                         $R2obsMAX{$resi_num}=$split_line[2];
446                     }
447                     $R2_chi{$resi_num}[$i] = $split_line[3];
448                     $R2err{$resi_num}[$i] = 1/$split_line[3];
449                 }
450             last LINE;      # Finished reading in data for this residue, go to the next one
451             }
452         }
453 }
454
455 ##################################################################
456 ##   Calculate average kex0, change amino acid abbreviations   ##
457 ##################################################################
458
459 #Calculate average kex0 for all residues
460 #kex is read in from files in (ms)^(-1), must convert to (s)
461 my $kex0_tot;
462 foreach my $resi_num (@resi_files) {
463         $kex0_tot += $kex0{$resi_num};
464 }
465 my $kex0_ave = ($kex0_tot * 1000) / ($#resi_files + 1);
466
467 #Check to see if number of points is equal for all residues
468 #for ($i = 0; $i < $#resi_files; $i++) {
469 #       $point_check = 0;
470 #       if ($points{$resi_files[$i]} == $points{$resi_files[$i+1]}) {
471 #               $point_check = 1;
472 #       }
473 #       if ($point_check == 0) {
474 #               print "\n\tResidues $resi_files[$i] and $resi_files[$i+1] do not contain the same number of points.\n";
475 #               print "\tA global fit cannot be performed.\n";
476 #               print "\tProgram will now exit.\n";
477 #               exit;
478 #       }
479 #}
480
481 # TODO What if the user doesn't use one letter abbreviations for amino acids?
482
483 # Change the amino acid abbreviation from upper case to lower case
484 # foreach my $resi_num (@resi_files) {
485 #       $resi_ID{$resi_num} = lcfirst $resi_ID{$resi_num};
486 # }
487
488
489 ##################################################################
490 ##   Generate *.in for single amino acids and globally, also   ##
491 ##   create *.errors file.  This is only done if the notebook  ##
492 ##   is being generated.                                       ##
493 ##################################################################
494
495 if ($calc_residuals == 0) {
496
497         open(ALL_IN, "> all_resi.in");              # Open all_resi.in for output
498         open(ALL_ERR, "> all_resi.error");          # Open all_resi.error for output
499
500         # Counter to determine position in residue list for generating
501         # Kronecker delta function
502         my $position = 0;
503
504         foreach my $resi_num (@resi_files) {
505
506                 # Print the *.in file for each residue
507                 open(RESI_IN, "> $resi_num.in");
508                 for (my $i = 0; $i < $points{$resi_num}; $i++) {
509                         printf RESI_IN "%11.9f     %4.2E\n", $tcp{$resi_num}[$i], $R2obs{$resi_num}[$i];
510                 }
511                 close(RESI_IN);
512
513                 # Now append information for each residue to the main *.in and *.error files
514                 for (my $i = 0; $i < $points{$resi_num}; $i++) {
515
516                         #Set up Kronecker delta function
517                         for (my $j = 0; $j < ($#resi_files+1); $j++) {
```

```
518                     if ($j == $position) { print ALL_IN "0  "; }
519                     else { print ALL_IN "1  "; }
520                 }
521
522                 # Print field strength, tcp, and R2obs
523                 printf ALL_IN "%4.1f  %11.9f  %4.2E\n", $B0, $tcp{$resi_num}[$i], $R2obs{$resi_num}[$i];
524
525                 # Print the error file
526                 printf ALL_ERR "%7.5f\n", $R2err{$resi_num}[$i];
527
528             }
529             $position++;
530         }
531
532         close(ALL_IN);
533         close(ALL_ERR);
534         print "\n\tResidue output files successfully written.\n";
535     }
536
537
538     ##################################################################
539     ##    The following text is used to perform the CHI^2         ##
540     ##    calculation.  This is only performed after the notebook ##
541     ##    is evaluated.                                           ##
542     ##################################################################
543
544     if ($calc_residuals == 1) {
545         # Chi-squared calculation
546         open(FITS, "< globalfast_results");
547         my %residuals;
548         my @all_residuals;
549
550         my $begin_table = 0;
551         while (my $input = <FITS>) {
552             # Find the line where the FitResiduals table begins
553             if ($input =~ /FitResiduals/) {
554                 $begin_table = 1;
555             }
556
557             # Once we reach the right line, start splitting the fields
558             if ($begin_table == 1) {
559                 chomp($input);
560                 #@one_line = split /[\{+|\}+|\,+|\s+]/, $input;
561                 my @one_line = split /\s+/, $input;
562
563                 # Removed commas, trailing and leading curly braces
564                 foreach my $field (@one_line) {
565                     if ($field =~ /\,/) {
566                         chop($field);
567                     }
568                     if ($field=~/\}\}/) {
569                         chop($field);
570                         chop($field);
571                     }
572                     if ($field=~/\{/) {
573                         $field = reverse($field);
574                         chop($field);
575                         $field = reverse($field);
576                     }
577                 }
578
579                 # Find the starting position for the first line
580                 # It probably isn't the very beginning of the line
581                 my $position;
582                 my $row = 0;
583                 for (my $i = 0; $i < scalar(@one_line); $i++) {
584                     if ($one_line[$i] =~ /->/) {
585                         $position = $i;
586                         $row = 1;
587                         last;
588                     }
589                 }
590
591                 # Push all numerical data into an array
```

```
592
593                 if ($row == 1) {
594                     # For the first line, put everything after the beginning
595                     # of the table into the array
596                     for (my $i = $position; $i < scalar(@one_line); $i++) {
597                         #if (!($one_line[$i] == "")) {
598                         if ($one_line[$i] =~ /\w/) {
599                             push @all_residuals, $one_line[$i];
600                         }
601                     }
602                     $row++;
603                 } else {
604                     # The entirety of every other line can be added
605                     for (my $i = 0; $i < scalar(@one_line); $i++) {
606                         #if (!($one_line[$i] == "")) {
607                         if ($one_line[$i] =~ /\w/) {
608                             push @all_residuals, $one_line[$i];
609                         }
610                     }
611                 }
612             }
613         }
614         close(FITS);
615
616         my $residual_position = 0;
617         foreach my $resi_num (@resi_files) {
618             for (my $i = 0; $i < $points{$resi_num}; $i++) {
619                 $residuals{$resi_num}[$i] = $all_residuals[$residual_position];
620                 $residual_position++;
621             }
622         }
623
624         # Now calculate the chi2 = (sum (residuals^2))/error^2
625         open(CHI2, "> globalfast_chisq");
626
627         my %BIG_resi_ID;
628         foreach my $resi_num (@resi_files) {
629             $BIG_resi_ID{$resi_num} = ucfirst $resi_ID{$resi_num};
630         }
631
632         foreach my $resi_num (@resi_files) {
633             print CHI2 "$BIG_resi_ID{$resi_num}\n";
634             print CHI2 "Taucp(s)  Residual    R2err\n";
635             my %chi_sq_MA;
636             for (my $i = 0; $i < $points{$resi_num}; $i++) {
637                 printf CHI2 " %7.5f   %7.4f  %7.4f\n", $tcp{$resi_num}[$i], $residuals{$resi_num}[$i], $R2_chi{$resi_num}[$i];
638                 $chi_sq_MA{$resi_num} += (($residuals{$resi_num}[$i])**2) / (($R2_chi{$resi_num}[$i])**2);
639             }
640             printf CHI2 "\nX^2(residue):      %8.4f\n", $chi_sq_CF{$resi_num};
641             printf CHI2 "X^2(residue_red):  %8.4f\n", $chi_sq_CF_red{$resi_num};
642             printf CHI2 "X^2(global):       %8.4f\n\n\n", $chi_sq_MA{$resi_num};
643         }
644         close(CHI2);
645         print "\n\tCHI^2 values calculated and printed to file \"globalfast_chisq\".\n\n";
646     }
647
648
649     ##################################################################
650     ##    The following text is used to generate the Mathematica  ##
651     ##    notebook from the input files.  This is only performed  ##
652     ##    when residuals are not calculated.                      ##
653     ##################################################################
654
655     if ($calc_residuals == 0) {
656         print "\tNow writing Mathematica notebook.\n";
657         # Now generate and load the Mathematica notebook
658         open(MATNB, "> cpMG_global_fast.nb");
659
660
661         # Print VERY LONG header text
662         {
663             print MATNB"(************** Content-type: application/mathematica **************
664                 CreatedBy='Mathematica 5.0'
665
```

```
666                 Mathematica-Compatible Notebook
667
668  This notebook can be used with any Mathematica-compatible
669  application, such as Mathematica, MathReader or Publicon. The data
670  for the notebook starts with the line containing stars above.
671
672  To get the notebook into a Mathematica-compatible application, do
673  one of the following:
674
675  * Save the data starting with the line of stars above into a file
676    with a name ending in .nb, then open the file inside the
677    application;
678
679  * Copy the data starting with the line of stars above to the
680    clipboard, then use the Paste menu command inside the application.
681
682  Data for notebooks contains only printable 7-bit ASCII and can be
683  sent directly in email or through ftp in text mode.  Newlines can be
684  CR, LF or CRLF (Unix, Macintosh or MS-DOS style).
685
686  NOTE: If you modify the data for this notebook not in a Mathematica-
687  compatible application, you must delete the line below containing
688  the word CacheID, otherwise Mathematica-compatible applications may
689  try to use invalid cache data.
690
691  For more information on notebooks and Mathematica-compatible
692  applications, contact Wolfram Research:
693    web: http://www.wolfram.com
694    email: info\@wolfram.com
695    phone: +1-217-398-0700 (U.S.)
696
697  Notebook reader applications are available free of charge from
698  Wolfram Research.
699  *******************************************************************)\n\n" ;
700        }
701
702        # Begin notebook format information
703        {
704              print MATNB "(*NotebookFileLineBreakTest
705  NotebookFileLineBreakTest*)
706  (*NotebookOptionsPosition[     55146,       1457]*)
707  (*NotebookOutlinePosition[     56125,       1488]*)
708  (*  CellTagsIndexPosition[     56081,       1484]*)
709  (*WindowFrame->Normal*)\n";
710        }
711
712        # Actual notebook text
713        {
714              print MATNB "\n\nNotebook[{
715
716  Cell[CellGroupData[{
717  Cell[\"Initialize\", \"Section\",
718    FontColor->RGBColor[0, 0, 1]],
719
720  Cell[BoxData[{
721      \\(<< Statistics`HypothesisTests`\\), \"\\[IndentingNewLine]\",
722      \\(<< Statistics`NonlinearFit`\\), \"\\[IndentingNewLine]\",
723      \\(<< Graphics`MultipleListPlot`\\), \"\\[IndentingNewLine]\",
724      \\(Off[General::spell1]\\), \"\\[IndentingNewLine]\",
725      \\(Off[General::spell]\\)}], \"Input\"]
726  }, Open  ]],\n\n";
727        }
728
729
730        # The equation for fitting can be carefully altered here if needed
731        # TODO Could enter two different equations and then have the user specify which one
732        {
733              print MATNB "Cell[CellGroupData[{
734
735  Cell[TextData[{
736    \"Define \",
737    Cell[BoxData[
738        \\(TraditionalForm\\`CPMG\\)]],
739    \" expressions\"
```

```
740  }], \"Section\",
741   Evaluatable->False,
742   FontColor->RGBColor[0, 0, 1]],
743
744  Cell[BoxData[
745     \\(\\(R2fast[R20_, Rex_, kex_, tcp_, B0_, Bref_] =
746        R20 + \\((Rex*\\((B0/
747                  Bref)\\)^2)\\)*\\((1 - \\((\\((2*Tanh[kex*tcp/2])\\)/\\((tcp*
748                   kex)\\))\\))\\);\\)\\)], \"Input\"]
749  }, Open  ]],\n\n";
750        }
751
752        # Generate Kronecker delta functions
753        {
754              print MATNB "Cell[CellGroupData[{
755
756  Cell[\"Functions for Nonlinear Fitting\", \"Section\",
757   FontColor->RGBColor[0, 0, 1]],
758
759  Cell[BoxData[
760     \\(\\(\\(\\[IndentingNewLine]\\)\\(globalfast[";
761
762              # Print AA_, AB_, AC_, etc...
763              # NOTE added second letter so more than 26 residues could be fit
764              my $big_alpha_1="A";
765              my $big_alpha_2="A";
766              my $limit;
767              for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
768                    if (((($#resi_files+1)-(26*($j-1))) < 26) {
769                          $limit = ($#resi_files+1)-26*($j-1);
770                    } else {
771                          $limit = 26;
772                    }
773
774                    for (my $i = 1; $i <= $limit; $i++) {
775                          print MATNB "$big_alpha_1$big_alpha_2\_, ";
776                          if ($big_alpha_2 =~ /Z/) {
777                                $big_alpha_2="A";
778                          } else {
779                                $big_alpha_2++;
780                          }
781                    }
782                    $big_alpha_1++;
783              }
784
785              print MATNB "\n\t\t";
786
787              # Print R20residue list...
788              foreach my $resi_num (@resi_files) {
789                    print MATNB "R20$resi_ID{$resi_num}\_, "
790              }
791              print MATNB "\n\t\t";
792
793              # Print Rexresidue list
794              foreach my $resi_num (@resi_files) {
795                    print MATNB "Rex$resi_ID{$resi_num}\_, "
796              }
797              print MATNB "\n\t\tkex_, tcp_, B0_, Bref_] =\n";
798
799              $big_alpha_1="A";
800              $big_alpha_2="A";
801              for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
802                    if (((($#resi_files+1)-(26*($j-1))) < 26) {
803                          $limit = ($#resi_files+1)-26*($j-1);
804                    } else {
805                          $limit = 26;
806                    }
807
808                    for (my $i = 1; $i <= $limit; $i++) {
809                          print MATNB "\t\tKroneckerDelta[$big_alpha_1$big_alpha_2]*R2fast[R20$resi_ID{$resi_files[($j-1)*26+$i-1]}\, R
$resi_files[($j-1)*26+$i-1]}\, kex, tcp, B0, Bref]";
810                    if (!((($j-1)*26+$i-1) == $#resi_files)) {
811                          print MATNB " + \n";
812                    } else {
```

```
813                     print MATNB ";\\)\\)\\)], \"Input\"]
814         }, Open  ]],\n\n"
815                 }
816                     if ($big_alpha_2 =~ /Z/) {
817                         $big_alpha_2="A";
818                     } else {
819                         $big_alpha_2++;
820                     }
821                 }
822                 $big_alpha_1++;
823             }
824         }
825
826         {
827             print MATNB "Cell[CellGroupData[{
828
829 Cell[TextData[{
830  \"Read in \",
831  Cell[BoxData[
832     \\(TraditionalForm\\`\\(\\(R2(1/\\[Tau]cp)\\)\\(\\ \\)\\)\\)]],
833  \"relaxation dispersion data\"
834 }], \"Section\",
835  FontColor->RGBColor[0, 0, 1]],
836
837 Cell[BoxData[{
838   \\(\\(datafast =
839     ReadList[\"\\<./all_resi.in\\>\", Number,
840       RecordLists \\[Rule] True];\\)\\), \"\\[IndentingNewLine]\",
841   \\(\\(\\(errorsfast =
842     ReadList[\"\\<./all_resi.error\\>\", Number];\\)\\(\\[IndentingNewLine]\\)
843   \\)\\), \"\\[IndentingNewLine]\",
844   \\(\\)}], \"Input\"]
845 }, Open  ]],\n\n";
846         }
847
848         # Set up the global fitting equation
849         {
850             print MATNB "Cell[CellGroupData[{
851
852 Cell[\"\\<\\
853 Perform Nonlinear Curvefitting and Plot the Data for multiple \\
854 residues\\
855 \\>\", \"Section\",
856  FontColor->RGBColor[0, 0, 1]],
857
858 Cell[BoxData[{
859   \\(fitglobalfast =
860     NonlinearRegress[datafast,
861       globalfast[";
862
863             # Print AA, AB, AC, etc...
864             # NOTE added second letter so more than 26 residues could be fit
865             my $big_alpha_1="A";
866             my $big_alpha_2="A";
867             my $limit;
868             for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
869                 if ((($#resi_files+1)-(26*($j-1))) < 26) {
870                     $limit = ($#resi_files+1)-26*($j-1);
871                 } else {
872                     $limit = 26;
873                 }
874
875                 for (my $i = 1; $i <= $limit; $i++) {
876                     print MATNB "$big_alpha_1$big_alpha_2\, ";
877                     if ($big_alpha_2 =~ /Z/) {
878                         $big_alpha_2="A";
879                     } else {
880                         $big_alpha_2++;
881                     }
882                 }
883                 $big_alpha_1++;
884             }
885
886             # Print R20residue list...
```

```
887             foreach my $resi_num (@resi_files) {
888                 print MATNB "R20$resi_ID{$resi_num}\, "
889             }
890             print MATNB "\n\t\t";
891
892             # Print Rexresidue list
893             foreach my $resi_num (@resi_files) {
894                 print MATNB "Rex$resi_ID{$resi_num}\, "
895             }
896             print MATNB "kex, tcp, B0, $Bref], \n\t\t{";
897
898             # Print AA_, AB_, AC_, etc...
899             # NOTE added second letter so more than 26 residues could be fit
900             $big_alpha_1="A";
901             $big_alpha_2="A";
902             #my $limit;
903             for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
904                 if (((($#resi_files+1)-(26*($j-1))) < 26) {
905                     $limit = ($#resi_files+1)-26*($j-1);
906                 } else {
907                     $limit = 26;
908                 }
909
910                 for (my $i = 1; $i <= $limit; $i++) {
911                     print MATNB "$big_alpha_1$big_alpha_2\, ";
912                     if ($big_alpha_2 =~ /Z/) {
913                         $big_alpha_2="A";
914                     } else {
915                         $big_alpha_2++;
916                     }
917                 }
918                 $big_alpha_1++;
919             }
920             print MATNB "B0, tcp},\n\t\t{";
921
922             # Print R20residue list...
923             foreach my $resi_num (@resi_files) {
924                 print MATNB "{R20$resi_ID{$resi_num}\, $R20{$resi_num}}, "
925             }
926             print MATNB "\n\t\t";
927
928             # Print Rexresidue list...
929             foreach my $resi_num (@resi_files) {
930                 print MATNB "{Rex$resi_ID{$resi_num}\, $Rex0{$resi_num}}, "
931             }
932
933             print MATNB "\n\t\t{kex, $kex0_ave}}, MaxIterations \\[Rule] 5000,
934         Weights \\[Rule] errorsfast,
935         RegressionReport \\[Rule] {BestFit, BestFitParameters,
936            ParameterCITable, EstimatedVariance, ANOVATable,
937            AsymptoticCorrelationMatrix, AsymptoticCovarianceMatrix,
938            FitResiduals}] >> globalfast_results\\), \"\\[IndentingNewLine]\",
939     \\(ReadList[\"\\<globalfast_results\\>\"]\\)}], \"Input\"]
940   }, Open  ]],\n\n";
941         }
942
943         # Extract values and fit to exchange curve
944         {
945             my $j = 1;
946             print MATNB "Cell[BoxData[{";
947
948             # Setup the R20 fits
949             foreach my $resi_num (@resi_files) {
950                 print MATNB "\n    \\(\\(fR20$resi_ID{$resi_num} =\n";
951                 print MATNB "        fitglobalfast\\[LeftDoubleBracket]3, 2, 1, ";
952                 print MATNB "$j, 1\\[RightDoubleBracket];\\)\\), \"\\[IndentingNewLine]\",";
953                 $j++;
954             }
955
956             # The Rex fits
957             foreach my $resi_num (@resi_files) {
958                 print MATNB "\n    \\(\\(fRex$resi_ID{$resi_num} =\n";
959                 print MATNB "        fitglobalfast\\[LeftDoubleBracket]3, 2, 1, ";
960                 print MATNB "$j, 1\\[RightDoubleBracket];\\)\\), \"\\[IndentingNewLine]\",";
```

```
961                 $j++;
962             }
963
964             print MATNB "\n    \\(\\(fkex = fitglobalfast\\[LeftDoubleBracket]3, 2, 1, ";
965             print MATNB "$j, 1\\[RightDoubleBracket];\\)\\[IndentingNewLine]\n\n";
966
967             print MATNB "\\[IndentingNewLine](*here, \\
968       the\\ xy\\ data\\ for\\ the\\ individual\\ residues\\ is\\ read\\ in\\ so\\ that\\ \\
969   it\\ can\\ be\\ plotted\\ *) \\[IndentingNewLine] (*This\\ means\\ that\\ in\\ \\
970   addition\\ to\\ the\\ global\\ data\\ file, \\
971       there\\ will\\ need\\ to\\ be\\ a\\ data*) \\), \"\\[IndentingNewLine]\",
972      \\( (*file\\ for\\ each\\ residue\\ *) \\)}], \"Input\"],\n\n";
973
974         }
975
976         # Set up the plots.  The plots are divided up so that only four residues are plotted
977         # per graph.
978         {
979             #Sort data by last R2obs point to make graph scaling pretty
980             sub by_value { $R2obsMAX{$b} <=> $R2obsMAX{$a} }
981             my $i = 0;
982             for my $R2MAX (sort by_value (keys(%R2obsMAX))) {
983                 $resi_files[$i] = $R2MAX;
984                 $i++;
985             }
986
987             print MATNB "Cell[BoxData[{\n";
988             foreach my $resi_num (@resi_files) {
989                 print MATNB "    \\(\\($resi_ID{$resi_num} = \n";
990                 print MATNB "        ReadList[\"\\<./$resi_num.in\\>\", Number, \n";
991                 print MATNB "          RecordLists \\[Rule] True];\\)\\), \"\\[IndentingNewLine]\", \n";
992             }
993             foreach my $resi_num (@resi_files) {
994                 print MATNB "    \\(\\(r2$resi_ID{$resi_num}\[tcp_] := \n";
995                 print MATNB "        fR20$resi_ID{$resi_num} + \n";
996                 print MATNB "          fRex$resi_ID{$resi_num} \\((1 - \n";
997                 print MATNB "              2*Tanh[fkex*tcp/2]/\\((fkex*tcp)\\))\\);\\)";
998                 if (!($resi_num == $resi_files[$#resi_files])) {
999                     print MATNB "\\), \"\\[IndentingNewLine]\",\n";
1000                } else {
1001                    print MATNB "        \\[IndentingNewLine]\\n ";
1002                }
1003            }
1004
1005            print MATNB "(*Here\\ the\\ actual\\ plotting\\ is\\ done . \\ I\\ think\\ 4\\ data\\ sets/
1006          graph\\ is\\ enough\\ so\\ if\\  > \\
1007       4\\ residues*) \n (*are\\ being\\ fit, \\
1008      the\\ graphing\\ should\\ be\\ divided\\ up\\ such\\ that\\ multiple\\ graphs\\ \\
1009  with\\ 4\\ res/
1010         graph*) \\[IndentingNewLine] (*are\\ plotted*) \\[IndentingNewLine]\\), \\
1011  \"\\[IndentingNewLine]\", \n";
1012
1013
1014
1015            for (my $graph_num = 1; $graph_num <= (($#resi_files-$#resi_files%4)/4 + 1); $graph_num++) {
1016                my $limit;
1017                if (($#resi_files+1)-(4*($graph_num-1)) < 4) {
1018                    $limit = ($#resi_files+1)-(4*($graph_num-1));
1019                } else {
1020                    $limit = 4
1021                }
1022                for (my $i = 1; $i <= $limit; $i++) {
1023                    print MATNB "     \\(\\(pFit$resi_ID{$resi_files[(($i-1)+(4*($graph_num-1)))]} =\n";
1024                    print MATNB "          Plot[r2$resi_ID{$resi_files[(($i-1)+(4*($graph_num-1)))]}\[tcp], {tcp, 0.0001, 0.011}, \n" ;
1025                    print MATNB "          PlotStyle \\[Rule] {RGBColor[";
1026                    if ($i == 1) {
1027                        print MATNB "1, 0, 0";
1028                    } elsif ($i == 2) {
1029                        print MATNB "0, 1, 0";
1030                    } elsif ($i == 3) {
1031                        print MATNB "0, 0, 1";
1032                    } else { print MATNB "1, 0, 1"; }
1033                    print MATNB "]}, \n";
1034                    print MATNB "         DisplayFunction \\[Rule] Identity];\\)\\), \"\\[IndentingNewLine]\",\n";
```

```
1035                    }
1036                    print "\n";
1037                    print MATNB "    \\(\\(pDatfast =
1038        MultipleListPlot[";
1039
1040
1041                    for (my $j = 1; $j <= $limit; $j++) {
1042                        print MATNB "$resi_ID{$resi_files[(($j-1)+(4*($graph_num-1)))]}\, ";
1043                    }
1044
1045                    print MATNB "\n         SymbolShape \\[Rule] {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4],
1046            PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
1047          DisplayFunction \\[Rule] Identity];\\)\\), \"\\[IndentingNewLine]\",
1048      \\(\\(Show[";
1049
1050
1051                    for (my $j = 1; $j <= $limit; $j++) {
1052                        print MATNB "pFit$resi_ID{$resi_files[(($j-1)+(4*($graph_num-1)))]}\, ";
1053                    }
1054
1055                    print MATNB "pDatfast,
1056        PlotLabel -> \"\\<Disperson Curves";
1057
1058                    # Make a hash with capital letter ID's for Plot titles
1059                    my %BIG_resi_ID;
1060                    foreach my $resi_num (@resi_files) {
1061                        $BIG_resi_ID{$resi_num} = ucfirst $resi_ID{$resi_num};
1062                    }
1063
1064                    for (my $j = 1; $j <= $limit; $j++) {
1065                        print MATNB " $BIG_resi_ID{$resi_files[(($j-1)+(4*($graph_num-1)))]}";
1066                    }
1067
1068                    #TODO Figure out how to put Y-axis lable sideways
1069                    #TODO Add command to export graphs
1070                    print MATNB "\\>\",
1071        AxesLabel \\[Rule] {\"\\<\\[Tau]aucp\\>\", \"\\<R2obs\\>\"},
1072        DisplayFunction \\[Rule] \$DisplayFunction];\\)\\";
1073
1074                    if ($graph_num == (($#resi_files-$#resi_files%4)/4 + 1)) {
1075                        print MATNB ")}], \"Input\"]\n";
1076                    } else {
1077                        print MATNB "[IndentingNewLine]\\), \"\\
1078  \\[IndentingNewLine]\",\n";
1079                    }
1080                }
1081        }
1082
1083        # Finish the notebook
1084        {
1085            print MATNB "},
1086  FrontEndVersion->\"5.0 for X\",
1087  ScreenRectangle->{{0, 1024}, {0, 768}},
1088  CellGrouping->Manual,
1089  WindowSize->{1006, 693},
1090  WindowMargins->{{Automatic, 1}, {Automatic, 0}},
1091  PrintingCopies->1,
1092  PrintingPageRange->{Automatic, Automatic},
1093  PrintingOptions->{\"PaperSize\"->{612, 792},
1094  \"PaperOrientation\"->\"Portrait\",
1095  \"PostScriptOutputFile\":>FrontEnd`FileName[{\$RootDirectory, \"home\", \"loria\", \\
1096  \"math\", \"global_cpmg\"}, \"cpmg.global.fast_loria.nb.ps\", CharacterEncoding -> \\
1097  \"iso8859-1\"],
1098  \"Magnification\"->1}
1099  ]
1100
1101
1102
1103
1104
1105  (*******************************************************************
1106  End of Mathematica Notebook file.
1107  *******************************************************************)" ;
1108        }
```

```
1109
1110        close(MATNB);
1111        # Run Mathematica and load notebook
1112        system("mathematica cpMG_global_fast.nb");
1113    }
```

10.4.3 Example Mathematica notebook.

Initialize

In[1]:=

```
<< Statistics`HypothesisTests`

<< Statistics`NonlinearFit`

<< Graphics`MultipleListPlot`

Off[General::spell1]

Off[General::spell]
```

Define CPMG expressions

In[6]:=

```
R2fast[R20_, Rex_, kex_, tcp_, B0_, Bref_] = R20 + (Rex * (B0 / Bref) ^2) * (1 - ((2 * Tanh[kex * tcp / 2]) / (tcp * kex)));
```

Functions for Nonlinear Fitting

In[7]:=

```
globalfast[AA_, AB_, AC_, AD_, AE_, AF_, AG_, AH_, AI_, AJ_, AK_, AL_, AM_, AN_, AO_, R20s15_, R20s16_, R20a19_, R20135_,
  R20q101_, R20s115_, R20s116_, R20a119_, R201135_, R20s215_, R20s216_, R20a219_, R201235_, R20s315_, R20q101600_,
  Rexs15_, Rexs16_, Rexa19_, Rexl35_, Rexq101_, Rexs115_, Rexs116_, Rexa119_, Rexl135_, Rexs215_, Rexs216_, Rexa219_,
  Rexl235_, Rexs315_, Rexq101600_, kex_, tcp_, B0_, Bref_] =
 KroneckerDelta[AA] * R2fast[R20s15, Rexs15, kex, tcp, B0, Bref] + KroneckerDelta[AB] * R2fast[R20s16, Rexs16, kex, tcp, B0, Bref] +
  KroneckerDelta[AC] * R2fast[R20a19, Rexa19, kex, tcp, B0, Bref] + KroneckerDelta[AD] * R2fast[R20135, Rexl35, kex, tcp, B0, Bref] +
  KroneckerDelta[AE] * R2fast[R20q101, Rexq101, kex, tcp, B0, Bref] + KroneckerDelta[AF] * R2fast[R20s115, Rexs115, kex, tcp, B0, Bref] +
  KroneckerDelta[AG] * R2fast[R20s116, Rexs116, kex, tcp, B0, Bref] + KroneckerDelta[AH] * R2fast[R20a119, Rexa119, kex, tcp, B0, Bref] +
  KroneckerDelta[AI] * R2fast[R201135, Rexl135, kex, tcp, B0, Bref] + KroneckerDelta[AJ] * R2fast[R20s215, Rexs215, kex, tcp, B0, Bref] +
  KroneckerDelta[AK] * R2fast[R20s216, Rexs216, kex, tcp, B0, Bref] + KroneckerDelta[AL] * R2fast[R20a219, Rexa219, kex, tcp, B0, Bref] +
  KroneckerDelta[AM] * R2fast[R201235, Rexl235, kex, tcp, B0, Bref] + KroneckerDelta[AN] * R2fast[R20s315, Rexs315, kex, tcp, B0, Bref] +
  KroneckerDelta[AO] * R2fast[R20q101600, Rexq101600, kex, tcp, B0, Bref];
```

Read in R2(1/τcp) relaxation dispersion data

In[8]:=

```
datafast = ReadList["./all_resi.in", Number, RecordLists → True];

errorsfast = ReadList["./all_resi.error", Number];
```

Perform Nonlinear Curvefitting and Plot the Data for multiple residues

In[10]:=

```
fitglobalfast = NonlinearRegress[datafast, globalfast[AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO,
    R20s15, R20s16, R20a19, R20135, R20q101, R20s115, R20s116, R20a119, R201135, R20s215, R20s216, R20a219, R201235,
    R20s315, R20q101600, Rexs15, Rexs16, Rexa19, Rexl35, Rexq101, Rexs115, Rexs116, Rexa119, Rexl135, Rexs215, Rexs216,
    Rexa219, Rexl235, Rexs315, Rexq101600, kex, tcp, B0, 14.1], {AA, AB, AC, AD, AE, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, B0, tcp},
   {{R20s15, 15.5184}, {R20s16, 12.1229}, {R20a19, 13.2859}, {R20135, 17.1943}, {R20q101, 22.0758}, {R20s115, 15.5184},
    {R20s116, 12.1229}, {R20a119, 13.2859}, {R201135, 17.1943}, {R20s215, 15.5184}, {R20s216, 12.1229}, {R20a219, 13.2859},
    {R201235, 17.1943}, {R20s315, 15.5184}, {R20q101600, 22.0758}, {Rexs15, 5.2497}, {Rexs16, 10.9557}, {Rexa19, 6.7544},
    {Rexl35, 9.0514}, {Rexq101, 47.7820}, {Rexs115, 5.2497}, {Rexs116, 10.9557}, {Rexa119, 6.7544}, {Rexl135, 9.0514},
    {Rexs215, 5.2497}, {Rexs216, 10.9557}, {Rexa219, 6.7544}, {Rexl235, 9.0514}, {Rexs315, 5.2497}, {Rexq101600, 47.7820},
    {kex, 2636.68348224513}}, MaxIterations → 5000, Weights → errorsfast,
   RegressionReport → {BestFit, BestFitParameters, ParameterCITable, EstimatedVariance, ANOVATable,
     AsymptoticCorrelationMatrix, AsymptoticCovarianceMatrix, FitResiduals}] >> globalfast_results

ReadList["globalfast_results"]
```

Out[11]=

$$\{\{\text{BestFit} \to \text{KroneckerDelta[AJ]}\left(15.4636 + 0.0267928\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AN]}\left(15.4636 + 0.0267928\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AA]}\left(15.4636 + 0.0267928\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AF]}\left(15.4636 + 0.0267928\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AC]}\left(12.9876 + 0.0350332\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AH]}\left(12.9876 + 0.0350332\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AL]}\left(12.9876 + 0.0350332\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AD]}\left(15.9799 + 0.0488133\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AI]}\left(15.9799 + 0.0488133\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AM]}\left(15.9799 + 0.0488133\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AB]}\left(11.4522 + 0.0569898\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AG]}\left(11.4522 + 0.0569898\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AK]}\left(11.4522 + 0.0569898\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AE]}\left(25.9229 + 0.229333\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right) +$$
$$\text{KroneckerDelta[AO]}\left(25.9229 + 0.229333\,B0^2\left(1 - \frac{0.000692612\,\text{Tanh}[1443.81\,tcp]}{tcp}\right)\right),$$

BestFitParameters → {R20s15 → 15.4636, R20s16 → 11.4522, R20a19 → 12.9876, R20l35 → 15.9799, R20q101 → 25.9229, R20s115 → 15.4636, R20s116 → 11.4522, R20a119 → 12.9876, R20l135 → 15.9799, R20s215 → 15.4636, R20s216 → 11.4522, R20a219 → 12.9876, R20l235 → 15.9799, R20s315 → 15.4636, R20q101600 → 25.9229, Rexs15 → 5.32667, Rexs16 → 11.3301, Rexa19 → 6.96495, Rexl35 → 9.70457, Rexq101 → 45.5936, Rexs115 → 5.32667, Rexs116 → 11.3301, Rexa119 → 6.96495, Rexl135 → 9.70457, Rexs215 → 5.32667, Rexs216 → 11.3301, Rexa219 → 6.96495, Rexl235 → 9.70457, Rexs315 → 5.32667, Rexq101600 → 45.5936, kex → 2887.62},

ParameterCITable →

	Estimate	Asymptotic SE	CI
R20s15	15.4636	0.398602	{14.6745, 16.2527}
R20s16	11.4522	0.416635	{10.6274, 12.2771}
R20a19	12.9876	0.268726	{12.4556, 13.5197}
R20l35	15.9799	0.412523	{15.1632, 16.7966}
R20q101	25.9229	1.13783	{23.6703, 28.1755}
R20s115	15.4636	0.398602	{14.6745, 16.2527}
R20s116	11.4522	0.416635	{10.6274, 12.2771}
R20a119	12.9876	0.268726	{12.4556, 13.5197}
R20l135	15.9799	0.412523	{15.1632, 16.7966}
R20s215	15.4636	0.398602	{14.6745, 16.2527}
R20s216	11.4522	0.416635	{10.6274, 12.2771}
R20a219	12.9876	0.268726	{12.4556, 13.5197}
R20l235	15.9799	0.412523	{15.1632, 16.7966}
R20s315	15.4636	0.398602	{14.6745, 16.2527}
R20q101600	25.9229	1.13783	{23.6703, 28.1755}
Rexs15	5.32667	0.64678	{4.0462, 6.60714}
Rexs16	11.3301	0.597414	{10.1474, 12.5129}
Rexa19	6.96495	0.387843	{6.19712, 7.73279}
Rexl35	9.70457	0.62101	{8.47512, 10.934}
Rexq101	45.5936	1.39423	{42.8334, 48.3539}
Rexs115	5.32667	0.64678	{4.0462, 6.60714}
Rexs116	11.3301	0.597414	{10.1474, 12.5129}
Rexa119	6.96495	0.387843	{6.19712, 7.73279}
Rexl135	9.70457	0.62101	{8.47512, 10.934}
Rexs215	5.32667	0.64678	{4.0462, 6.60714}
Rexs216	11.3301	0.597414	{10.1474, 12.5129}
Rexa219	6.96495	0.387843	{6.19712, 7.73279}
Rexl235	9.70457	0.62101	{8.47512, 10.934}
Rexs315	5.32667	0.64678	{4.0462, 6.60714}
Rexq101600	45.5936	1.39423	{42.8334, 48.3539}
kex	2887.62	131.378	{2627.52, 3147.72}

EstimatedVariance → 0.492143,

ANOVATable →

	DF	SumOfSq	MeanSq
Model	31	149100.	4809.68
Error	121	59.5493	0.492143,
Uncorrected Total	152	149160.	
Corrected Total	151	13242.	

In[12]:=

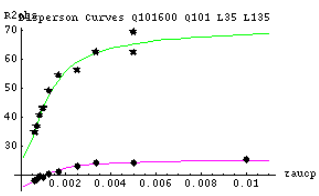
```
fR20s15 = fitglobalfast[[3, 2, 1, 1, 1]];

fR20s16 = fitglobalfast[[3, 2, 1, 2, 1]];

fR20a19 = fitglobalfast[[3, 2, 1, 3, 1]];

fR20135 = fitglobalfast[[3, 2, 1, 4, 1]];

fR20q101 = fitglobalfast[[3, 2, 1, 5, 1]];

fR20s115 = fitglobalfast[[3, 2, 1, 6, 1]];

fR20s116 = fitglobalfast[[3, 2, 1, 7, 1]];

fR20a119 = fitglobalfast[[3, 2, 1, 8, 1]];

fR201135 = fitglobalfast[[3, 2, 1, 9, 1]];

fR20s215 = fitglobalfast[[3, 2, 1, 10, 1]];

fR20s216 = fitglobalfast[[3, 2, 1, 11, 1]];

fR20a219 = fitglobalfast[[3, 2, 1, 12, 1]];

fR201235 = fitglobalfast[[3, 2, 1, 13, 1]];

fR20s315 = fitglobalfast[[3, 2, 1, 14, 1]];

fR20q101600 = fitglobalfast[[3, 2, 1, 15, 1]];

fRexs15 = fitglobalfast[[3, 2, 1, 16, 1]];

fRexs16 = fitglobalfast[[3, 2, 1, 17, 1]];

fRexa19 = fitglobalfast[[3, 2, 1, 18, 1]];

fRexl35 = fitglobalfast[[3, 2, 1, 19, 1]];

fRexq101 = fitglobalfast[[3, 2, 1, 20, 1]];

fRexs115 = fitglobalfast[[3, 2, 1, 21, 1]];

fRexs116 = fitglobalfast[[3, 2, 1, 22, 1]];

fRexa119 = fitglobalfast[[3, 2, 1, 23, 1]];

fRexl135 = fitglobalfast[[3, 2, 1, 24, 1]];

fRexs215 = fitglobalfast[[3, 2, 1, 25, 1]];

fRexs216 = fitglobalfast[[3, 2, 1, 26, 1]];

fRexa219 = fitglobalfast[[3, 2, 1, 27, 1]];

fRexl235 = fitglobalfast[[3, 2, 1, 28, 1]];

fRexs315 = fitglobalfast[[3, 2, 1, 29, 1]];

fRexq101600 = fitglobalfast[[3, 2, 1, 30, 1]];

fkex = fitglobalfast[[3, 2, 1, 31, 1]];

(*here, the xy data for the individual residues is read in so that it can be plotted *)
(*This means that in addition to the global data file, there will need to be a data*)

(*file for each residue *)
```

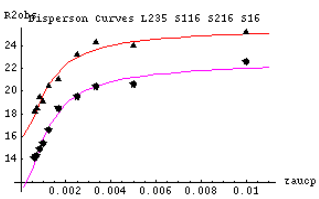
In[43]:=

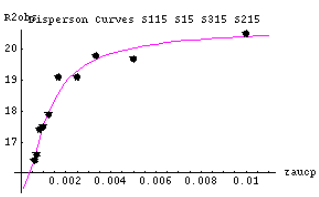
```
q101600 = ReadList["./101600.in", Number, RecordLists → True];

q101 = ReadList["./101.in", Number, RecordLists → True];

135 = ReadList["./35.in", Number, RecordLists → True];

1135 = ReadList["./135.in", Number, RecordLists → True];

1235 = ReadList["./235.in", Number, RecordLists → True];

s116 = ReadList["./116.in", Number, RecordLists → True];
```

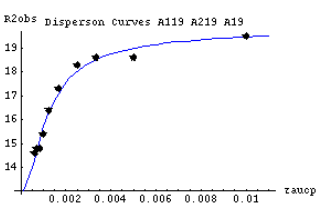
```
s216 = ReadList["./216.in", Number, RecordLists → True];

s16 = ReadList["./16.in", Number, RecordLists → True];

s115 = ReadList["./115.in", Number, RecordLists → True];

s15 = ReadList["./15.in", Number, RecordLists → True];

s315 = ReadList["./315.in", Number, RecordLists → True];

s215 = ReadList["./215.in", Number, RecordLists → True];

a119 = ReadList["./119.in", Number, RecordLists → True];

a219 = ReadList["./219.in", Number, RecordLists → True];

a19 = ReadList["./19.in", Number, RecordLists → True];

r2q101600[tcp_] := fR20q101600 + fRexq101600 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2q101[tcp_] := fR20q101 + fRexq101 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2135[tcp_] := fR20135 + fRexl35 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r21135[tcp_] := fR201135 + fRexl135 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r21235[tcp_] := fR201235 + fRexl235 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s116[tcp_] := fR20s116 + fRexs116 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s216[tcp_] := fR20s216 + fRexs216 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s16[tcp_] := fR20s16 + fRexs16 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s115[tcp_] := fR20s115 + fRexs115 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s15[tcp_] := fR20s15 + fRexs15 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s315[tcp_] := fR20s315 + fRexs315 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2s215[tcp_] := fR20s215 + fRexs215 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2a119[tcp_] := fR20a119 + fRexa119 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2a219[tcp_] := fR20a219 + fRexa219 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

r2a19[tcp_] := fR20a19 + fRexa19 (1 - 2 * Tanh[fkex * tcp / 2] / (fkex * tcp));

(*Here the actual plotting is done. I think 4 data sets/graph is enough so if > 4 residues*)(*are being fit,
 the graphing should be divided up such that multiple graphs with 4 res/graph*)
(*are plotted*)


pFitq101600 = Plot[r2q101600[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 0]}, DisplayFunction → Identity];

pFitq101 = Plot[r2q101[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 1, 0]}, DisplayFunction → Identity];

pFitl35 = Plot[r2135[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 0, 1]}, DisplayFunction → Identity];

pFitl135 = Plot[r21135[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 1]}, DisplayFunction → Identity];

pDatfast = MultipleListPlot[q101600, q101, l35, l135,
  SymbolShape → {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4], PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
  DisplayFunction → Identity];

Show[pFitq101600, pFitq101, pFitl35, pFitl135, pDatfast, PlotLabel -> "Disperson Curves Q101600 Q101 L35 L135",
 AxesLabel → {"τaucp", "R2obs"}, DisplayFunction → $DisplayFunction];


pFitl235 = Plot[r21235[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 0]}, DisplayFunction → Identity];

pFits116 = Plot[r2s116[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 1, 0]}, DisplayFunction → Identity];

pFits216 = Plot[r2s216[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 0, 1]}, DisplayFunction → Identity];

pFits16 = Plot[r2s16[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 1]}, DisplayFunction → Identity];

pDatfast = MultipleListPlot[l235, s116, s216, s16,
  SymbolShape → {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4], PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
  DisplayFunction → Identity];
```

```
Show[pFitl235, pFits116, pFits216, pFits16, pDatfast, PlotLabel -> "Disperson Curves L235 S116 S216 S16",
 AxesLabel → {"τaucp", "R2obs"}, DisplayFunction → $DisplayFunction];


pFits115 = Plot[r2s115[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 0]}, DisplayFunction → Identity];

pFits15 = Plot[r2s15[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 1, 0]}, DisplayFunction → Identity];

pFits315 = Plot[r2s315[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 0, 1]}, DisplayFunction → Identity];

pFits215 = Plot[r2s215[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 1]}, DisplayFunction → Identity];

pDatfast = MultipleListPlot[s115, s15, s315, s215,
  SymbolShape → {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4], PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
  DisplayFunction → Identity];

Show[pFits115, pFits15, pFits315, pFits215, pDatfast, PlotLabel -> "Disperson Curves S115 S15 S315 S215",
 AxesLabel → {"τaucp", "R2obs"}, DisplayFunction → $DisplayFunction];


pFital19 = Plot[r2al19[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 0]}, DisplayFunction → Identity];

pFita219 = Plot[r2a219[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 1, 0]}, DisplayFunction → Identity];

pFita19 = Plot[r2a19[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 0, 1]}, DisplayFunction → Identity];

pDatfast = MultipleListPlot[a119, a219, a19,
  SymbolShape → {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4], PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
  DisplayFunction → Identity];

Show[pFita119, pFita219, pFita19, pDatfast, PlotLabel -> "Disperson Curves A119 A219 A19", AxesLabel → {"τaucp", "R2obs"},
 DisplayFunction → $DisplayFunction];
```









Created by *Mathematica* (May 1, 2006)

10.5 Appendix 5 Global Full CPMG Perl Program

10.5.1 Sample input data from XMGR file.

```
# CPMGfit 1.0
#
# title Dispersion Plot for D83
# function Full_CPMG
# equation y=f(R20,papb,dw,kex)
# points       9
# X2    0.9933
# X2(red)      0.1987
#
# Parameter    Fitted_Value  Fitted_Error    Sim_value      Sim_error
# R20   12.0549 2.9231  11.4808 1.5623
# papb  0.9982E-01     0.5596E+00     0.4394E-01     0.1120E-01
# dw    0.8454  2.3993  1.0971  0.0317
# kex   2.2039  1.6019  2.1509  0.2499
#
#       Field   Rex     Rex_error      Half_point     Half_error
#
#       14.1000 30.0117 1.4036  0.6106  0.0498
#
@ CLEAR STRING
@ TITLE "Dispersion Plot for D83"
@ SUBTITLE "y=f(R20,papb,dw,kex)"
@ VIEW XMIN 0.20
@ VIEW XMAX 0.90
@ VIEW YMIN 0.35
@ VIEW YMAX 0.85
@ XAXIS LABEL "1/tcp (1/ms)"
@ YAXIS LABEL "R2(tcp)  (1/s)"
@ XAXIS TICKLABEL FORMAT DECIMAL
@ YAXIS TICKLABEL FORMAT DECIMAL
@ XAXIS TICKLABEL CHAR SIZE 0.8
@ YAXIS TICKLABEL CHAR SIZE 0.8
@ WORLD XMIN   0.000E+00
@ WORLD XMAX   0.200E+01
@ XAXIS TICK MAJOR    0.500E+00
@ XAXIS TICK MINOR    0.250E+00
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.215
@ STRING DEF "R20     12.0549 +/-    1.5623"
@ STRING on
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.180
@ STRING DEF "papb    0.9982E-01 +/- 0.1120E-01"
@ STRING on
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.145
@ STRING DEF "dw      0.8454 +/-     0.0317"
@ STRING on
@ WITH STRING
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.110
@ STRING DEF "kex     2.2039 +/-     0.2499"
@ STRING on
@ WITH STRING
```

```
@ STRING LOCTYPE VIEW
@ STRING CHAR SIZE 0.9
@ STRING 0.2, 0.075
@ STRING DEF "X2       0.9933"
@ STRING on
@ S0 TYPE xydy
@ S0  LINESTYLE 0
@ S0  COLOR 1
@ S0  SYMBOL 2
@ S0  SYMBOL SIZE 0.5
@ S0  SYMBOL FILL 1
@ S0  SYMBOL COLOR 1
@ S0  ERRORBAR LENGTH 0.5
@ S1 TYPE xy
@ S1  ERRORBAR LENGTH 0
@ S1  COLOR 1
        0.160E+01      0.160E+02      0.947E+00
        0.140E+01      0.179E+02      0.146E+01
        0.100E+01      0.211E+02      0.926E+00
        0.800E+00      0.234E+02      0.104E+01
        0.600E+00      0.270E+02      0.137E+01
        0.400E+00      0.320E+02      0.159E+01
        0.200E+00      0.359E+02      0.185E+01
        0.200E+00      0.376E+02      0.143E+01
        0.100E+00      0.401E+02      0.249E+01
&
        0.00000E+00    0.42067E+02    0.00000E+00
        0.40000E-02    0.41966E+02    0.00000E+00
        0.80000E-02    0.41865E+02    0.00000E+00
        0.12000E-01    0.41764E+02    0.00000E+00
        0.16000E-01    0.41663E+02    0.00000E+00
        0.20000E-01    0.41563E+02    0.00000E+00
        0.24000E-01    0.41462E+02    0.00000E+00
        0.28000E-01    0.41361E+02    0.00000E+00
        0.32000E-01    0.41260E+02    0.00000E+00
        0.36000E-01    0.41160E+02    0.00000E+00
        0.40000E-01    0.41059E+02    0.00000E+00
        0.44000E-01    0.40958E+02    0.00000E+00
        0.48000E-01    0.40857E+02    0.00000E+00
        0.52000E-01    0.40756E+02    0.00000E+00
        0.56000E-01    0.40656E+02    0.00000E+00
        0.60000E-01    0.40555E+02    0.00000E+00
        0.64000E-01    0.40454E+02    0.00000E+00
        0.68000E-01    0.40353E+02    0.00000E+00
        0.72000E-01    0.40253E+02    0.00000E+00
        0.76000E-01    0.40152E+02    0.00000E+00
        0.80000E-01    0.40051E+02    0.00000E+00
        0.84000E-01    0.39950E+02    0.00000E+00
        0.88000E-01    0.39850E+02    0.00000E+00
        0.92000E-01    0.39749E+02    0.00000E+00
        0.96000E-01    0.39648E+02    0.00000E+00
        0.10000E+00    0.39547E+02    0.00000E+00
        0.10400E+00    0.39446E+02    0.00000E+00
        0.10800E+00    0.39346E+02    0.00000E+00
        0.11200E+00    0.39245E+02    0.00000E+00
        0.11600E+00    0.39144E+02    0.00000E+00
        0.12000E+00    0.39043E+02    0.00000E+00
        0.12400E+00    0.38943E+02    0.00000E+00
        0.12800E+00    0.38842E+02    0.00000E+00
        0.13200E+00    0.38741E+02    0.00000E+00
        0.13600E+00    0.38640E+02    0.00000E+00
        0.14000E+00    0.38539E+02    0.00000E+00
        0.14400E+00    0.38439E+02    0.00000E+00
        0.14800E+00    0.38338E+02    0.00000E+00
        0.15200E+00    0.38237E+02    0.00000E+00
        0.15600E+00    0.38136E+02    0.00000E+00
```

10.5.2 Perl program.

```
#!/usr/bin/perl

use strict;
use warnings;

# CHANGE LOG
# 2006-02-12: Added ability to fit multiple field data for a
# single residue. Still need to correct generation of plots
# for different fields in mathematica notebook

#################################################################
#################################################################
##                Global CPMG Fit                ##
##   Simultaneously fits Xmgr residue files generated by     ##
##   Curvefit to the full exchange equation.                 ##
##   This program uses the CPMGfit results to get starting   ##
##   input values for the Mathematica notebook.  The         ##
##   notebook fits the data, generating a residuals file.    ##
##   This residuals file can then be used to calculated X2   ##
##                                                           ##
##                Michelle L. Gill and J. Patrick Loria  ##
##                Yale University, 12/09/2004            ##
##                Mathematica notebook was produced by   ##
##                Michael Grey (Columbia University) and ##
##                subsequently modified by JPL           ##
##                                                       ##
## This software is provided "as is" without any express     ##
## warranty, etc.  If you make any significant improvements  ##
## to this program, please send the authors a copy with the  ##
## annotated improvements.                                   ##
#################################################################
#################################################################


#################################################################
##            Switches for running the program              ##
#################################################################

# Removed leading "-" and put switches in array
my @switches;

foreach my $a (@ARGV) {
        chomp($a);
        my @tmp1 = split /-/, $a;
        my @tmp2 = split //, $tmp1[1];
        foreach my $temp_switch (@tmp2) {
                push @switches, $temp_switch;
        }
}

my $force = 0;
my $calc_residuals = 0;
if (!@switches) {
        print "\n\n\tDefault: running in safemode and generating\n";
        print "\tMathematica notebook.\n";
        print "\tTry \"./global_cpMG_full.pl -h\" for more information\n";
        print "\t--including how to perform CHI^2 calculation.\n";
} else {
        foreach my $switch (@switches) {
                if ($switch =~ /h/ || $switch =~ /H/) {
                        &prhelp();
                } elsif ($switch =~ /f/ || $switch =~ /F/) {
                        $force = 1;
                } elsif ($switch =~ /r/ || $switch =~ /R/) {
                        $calc_residuals = 1;
                } else {
                        print "\tThe switch ($switch) has no meaning.\n";
                        print "\tTry \"./global_cpMG_fit.pl -h\" for more information.\n\n";
                        exit;
                }
        }
}

```

```
74  sub prhelp() {
75     print "\n";
76     print "\tUsage:    ./global_cpMG_fit.pl -[Flags]\n";
77     print "\tDetails:  Generates a Mathematica notebook to globally\n";
78     print "\t         fit CPMGfit-generated XMGR files to the full\n";
79     print "\t         exchange equation.  A second run of the program\n";
80     print "\t         with the appropriate flag with calculate residuals.\n";
81     print "\tFlags:    -f  force mode, disables file checks\n";
82     print "\t             and doesn't check for generation of new\n";
83     print "\t             \"globalfull_results\" file.\n";
84     print "\t             (safemode is the default)\n";
85     print "\t         -r  calculate residuals from global fit\n";
86     print "\t             (can only be done after Mathematica notebook\n";
87     print "\t             has been evaluated and saved)\n";
88     print "\t         -h  displays this help information\n\n";
89     exit;
90  }
91
92  if ($calc_residuals == 0) {
93        print "\n";
94        print "\t*****************************************************\n";
95        print "\t*****************************************************\n";
96        print "\t**                Global CPMG Fit                **\n";
97        print "\t** Starting generation of 'cpMG_global_full.nb'. **\n";
98        print "\t**                      by MGill and JPLoria   **\n";
99        print "\t*****************************************************\n";
100        print "\t*****************************************************\n";
101        print "\n";
102  } else {
103        print "\n";
104        print "\t*****************************************************\n";
105        print "\t*****************************************************\n";
106        print "\t**                Global CPMG Fit                **\n";
107        print "\t** Starting calculation of global residuals      **\n";
108        print "\t**                      by MGill and JPLoria   **\n";
109        print "\t*****************************************************\n";
110        print "\t*****************************************************\n";
111        print "\n";
112  }
113
114
115  ##################################################################
116  ##      Determine current directory and all *.xmgr files       ##
117  ##################################################################
118
119  use Cwd;
120  my $cur_dir = getcwd();
121  my @resi_files = <*.xmgr>;
122  @resi_files = glob("*.xmgr");
123
124  # Strip off the .xmgr suffix
125  foreach my $resi_num (@resi_files) {
126        my @xmgr_tmp = split /\./, $resi_num;
127        $resi_num = $xmgr_tmp[0];
128  }
129
130  if ($calc_residuals == 0) {
131        # Exit program if there are no *.xmgr files in this directory
132        if (!(@resi_files)) {
133              print "\tCurrent working directory $cur_dir\n";
134              print "\tcontains no *.xmgr files.\n";
135              print "\tEither there are no xmgr files in this directory\n";
136              print "\tor the files are misnamed.\n";
137              print "\tThe program has exited.\n\n";
138              exit;
139        }
140
141        # Can't do a global fit if only one residue file exists
142        if ($#resi_files == 0) {
143              print "\tCurrent working directory $cur_dir\n";
144              print "\tcontains only one *.xmgr file.\n";
145              print "\tA global fit cannot be performed on only one file.\n";
146              print "\tThe program has exited.\n\n";
147              exit;
```

```
148         }
149
150         # Determine that we don't have any files from CPMGfit
151         my $file_type_error = 0;
152         my %file_errors;
153         foreach my $resi_num (@resi_files) {
154             open (FILE_CHECK, "< $resi_num.xmgr");
155             my $program_type = <FILE_CHECK>;
156
157             if ($program_type =~ /CurveFit/) {
158                 $file_type_error = 1;
159                 $file_errors{$resi_num} = 1;
160             } else {
161                 $file_errors{$resi_num} = 0;
162             }
163         }
164
165         # If there are CPMGfit files, print them out and exit the program
166         if ($file_type_error == 1) {
167             print "\n\tIt looks like the following XMGR file(s) are from\n";
168             print "\tCurveFit: ";
169             foreach my $resi_num (@resi_files) {
170                 if ($file_errors{$resi_num} == 1) {
171                     print "$resi_num  ";
172                 }
173             }
174             print "\n";
175             print "\tThese input files won't work with this program.\n";
176             print "\tPlease reprocess your data using CPMGfit.\n";
177             print "\tIf you wish to ignore these files please rename\n";
178             print "\tthem so they do not end in .xmgr.\n";
179             print "\tProgram will now exit.\n\n";
180             exit;
181         }
182     }
183
184     # Can't calculate residuals if the notebook hasn't been generated and evaluated
185     if (($calc_residuals == 1) && !(-e "globalfull_results")) {
186         print "\n\tYou cannot calculate residuals without first creating\n";
187         print "\tand evaluating the Mathematica notebook \"cpMG_global_full.nb\".\n";
188         print "\tEvaluation of this notebook will produce the necessary file: \n";
189         print "\t\"globalfull_results\".\n";
190         print "\tThe program has exited.\n\n";
191         exit;
192     }
193
194
195     ##################################################################
196     ##  Safety checks to prevent ovewriting of *.in and *.nb files ##
197     ##################################################################
198
199     if ($force == 0) {
200         if ($calc_residuals == 0) {
201             # If *.in files exist, ask if they can be overwritten
202             my @in_files = <*.in>;
203             @in_files = glob("*.in");
204             if (@in_files) {
205                 print "\n\tThere are already *.in files in this directory.\n";
206                 print "\tIs it OK to overwrite these files? (y|n) ";
207                 my $overwrite = <STDIN>;
208                 if (!($overwrite =~ /Y/) && !($overwrite =~ /y/)) {
209                     print "\n\tThe program exited without overwriting *.in files.\n\n";
210                     exit;
211                 }
212             }
213
214             # Check for mathematica notebook
215             if (-e "cpMG_global_full.nb") {
216                 print "\n\tMathematica notebook cpMG_global_full.nb already exists.\n";
217                 print "\tIs it OK to overwrite this file? (y|n) ";
218                 my $overwrite = <STDIN>;
219                 if (!($overwrite =~ /Y/) && !($overwrite =~ /y/)) {
220                     print "\n\tThe program exited without overwriting notebook.\n\n";
221                     exit;
```

```
222                 }
223             }
224         }
225
226         if ($calc_residuals == 1) {
227             # Check for residual calculation file
228             if (-e "globalfull_chisq") {
229                 print "\n\tResidual calculation file \"globalfull_chisq\" already exists.\n";
230                 print "\tIs it OK to overwrite this file? (y|n) ";
231                 my $overwrite = <STDIN>;
232                 if (!($overwrite =~ /Y/) && !($overwrite =~ /y/)) {
233                     print "\n\tThe program exited without overwriting residual file.\n\n";
234                     exit;
235                 }
236             }
237
238             # Check to see if Mathematica notebook has been generated but not evaluated
239             # Allows for about 1 min between evaluation(generating globalfull_results) and
240             # saving the notebook (hence the -M switch and the difference).  The value is
241             # in fractions of days
242
243             #if ((-M "cpMG_global_full.nb") < (-M "globalfull_results")) {
244             if (((-M "globalfull_results")-(-M "cpMG_global_full.nb")) > 0.0007) {
245                 print "\n\tThe notebook \"cpMG_global_full.nb\" may be newer than the fit file\n";
246                 print "\t\"globalfull_results\".  This means you may have generated a new\n";
247                 print "\tMathematica notebook without evaluating and saving it.\n";
248                 print "\tDo you still want to force the residual calculation? (y|n) ";
249                 my $force_resid = <STDIN>;
250                 if (!($force_resid =~ /y/ || $force_resid =~ /Y/)) {
251                     print "\n\tThe program exited without calculating residuals.\n\n";
252                     exit;
253                 }
254             }
255         }
256 }
257
258
259 ##################################################################
260 ##     Check *.xmgr files.                                      ##
261 ##################################################################
262
263 # Sort by residue numbers
264 my $i = 0;
265 my %temp_resi;
266 foreach my $resi_num (sort { $a <=> $b } @resi_files) {
267         $temp_resi{$i} = $resi_num;
268         $i++;
269 }
270 for (my $j = 0; $j < $i; $j++) {
271         $resi_files[$j] = $temp_resi{$j};
272 }
273
274 #Print all residue file names
275 my $total_files = $#resi_files+1;
276 if ($calc_residuals == 0) {
277         print "\n\tXmgr files ($total_files) found:";
278         $i = 1;
279         foreach my $resi_num (@resi_files) {
280             printf " %3d", $resi_num;
281             if ($i%10 == 0) {
282                 print "\n\t                    ";
283             }
284             $i++;
285         }
286         print "\n";
287
288         # Check to make sure files were read in correctly
289         print "\tIs this correct? (y|n) ";
290         my $correct = <STDIN>;
291
292         # Exit program if files are incorrect
293         if (!($correct =~ /Y/) && !($correct =~ /y/)) {
294             print "\n\tFiles have been determined to be incorrect.\n";
295             print "\tProgram will now exit.\n\n";
```

```
296             exit;
297         }
298 }
299
300 ##################################################################
301 ##    Begin reading in information from all *.xmgr files      ##
302 ##################################################################
303 my %resi_ID;
304 my %total_points;
305 my %points;
306 my %chi_sq_CF;
307 my %chi_sq_CF_red;
308 my %R20;
309 my %pa;
310 my %dw;
311 my %kex0;
312 my %total_fields;
313 my %B0_master;
314 my %B0;
315 my %tcp;
316 my %R2obs;
317 my %R2_chi;
318 my %R2err;
319 my %R2obsMAX;
320 foreach my $resi_num (@resi_files) {
321         open (XMGR, "< $resi_num.xmgr");
322         LINE: while (my $input = <XMGR>) {
323
324
325             # Find the name and number of the residue
326             if ($input =~ /^# title /) {
327                 chomp($input);
328                 my @split_line = split /\s+/, $input;
329                 #$resi_ID{$resi_num} = $split_line[$#split_line];
330                 my @abbrev_temp = split /\d/, $split_line[$#split_line];
331                 $resi_ID{$resi_num} = lcfirst($abbrev_temp[0]).$resi_num;
332             }
333
334             # Find the number of points to fit
335             if ($input =~ /^# points/) {
336                 chomp($input);
337                 my @split_line = split /\s+/, $input;
338                 $total_points{$resi_num} = $split_line[$#split_line];
339             }
340
341             # Find the X2 (CHI^2) value from Curvefit
342             if (($input =~ /^# X2/) && !($input =~ /red/)) {
343                 chomp($input);
344                 my @split_line = split /\s+/, $input;
345                 $chi_sq_CF{$resi_num} = $split_line[$#split_line];
346             }
347
348             # Find the X2 (CHI^2) (reduced) value from Curvefit
349             if ($input =~ /^# X2\(red\)/) {
350                 chomp($input);
351                 my @split_line = split /\s+/, $input;
352                 $chi_sq_CF_red{$resi_num} = $split_line[$#split_line];
353             }
354
355             # Find R20
356             if ($input =~ /^# R20/) {
357                 chomp($input);
358                 my @split_line = split /\s+/, $input;
359                 $R20{$resi_num} = $split_line[2];       #R20
360             }
361
362             # Get pa
363             if ($input =~ /^# papb/) {
364                 chomp($input);
365                 my %papb;
366                 my @split_line = split /\s+/, $input;
367                 $papb{$resi_num} = $split_line[2];     #papb
368                 $pa{$resi_num} = (1+sqrt(1-4*$papb{$resi_num}))/2; # papb = pa*(1-pa)
369             }
```

```
370
371             # dw
372             if ($input =~ /^# dw/) {
373                 chomp($input);
374                 my @split_line = split /\s+/, $input;
375                 $dw{$resi_num} = $split_line[2] * 1000; #delta Omega
376             }
377
378             # kex
379             if ($input =~ /^# kex/) {
380                 chomp($input);
381                 my @split_line = split /\s+/, $input;
382                 if ($split_line[1] =~ /Tau/) {
383                     # Tau version of Curvefit
384                     $kex0{$resi_num} = 1.0/($split_line[2] * 1000);
385                 } else {
386                     # kex0 version of Curvefit
387                     $kex0{$resi_num} = $split_line[2] * 1000;
388                 }
389             }
390
391             if ($input =~ /^#         Field/) {
392                 $input = <XMGR>;
393                 $total_fields{$resi_num}=0;
394                 until ($input =~ /^@/) {
395                     if (!($input =~ /^#\s*$/)) {
396                         chomp($input);
397                         my @split_line = split /\s+/, $input;
398                         $B0_master{$resi_num}[$total_fields{$resi_num}] = $split_line[1]; #B0 field
399                         $total_fields{$resi_num}++;
400                     }
401                     $input = <XMGR>;
402                 }
403             }
404
405             # Get values for tcp, R2obs, and R2err
406             # NOTE tcp is originally in (ms)^(-1), must convert to (s)
407             if ($input =~ /^@/) {
408                 while ($input =~ /^@/) {
409                     $input = <XMGR>;
410                 }
411                 my $B0_position = 0;
412                 for (my $k = 0; $k < $total_points{$resi_num}; $k++) {
413                     if ($input =~ /^&\s*$/) {
414                         $B0_position++;
415                         $input = <XMGR>;
416                     }
417                     chomp($input);
418                     my @split_line = split /\s+/, $input;
419                     $tcp{$resi_num}[$k] = 1/($split_line[1]*1000);            #tcp
420                     $R2obs{$resi_num}[$k] = $split_line[2];            #R2obs
421                     $R2_chi{$resi_num}[$k] = $split_line[3];
422                     $R2err{$resi_num}[$k] = 1/$split_line[3];
423                     $B0{$resi_num}[$k] = $B0_master{$resi_num}[$B0_position];
424                     $points{$resi_num}[$B0_position]++;
425                     if (exists $R2obsMAX{$resi_num}[$B0_position]) {
426                         if ($R2obs{$resi_num}[$k] > $R2obsMAX{$resi_num}[$B0_position]) {$R2obsMAX{$resi_num}[$B0_positi
k];}
427                     } else {$R2obsMAX{$resi_num}[$B0_position]=$R2obs{$resi_num}[$k];}
428                     $input = <XMGR>;
429                 }
430                 last LINE;        # Finished reading in data for this residue, go to the next one
431             }
432         }
433         close(XMGR);
434     }
435
436
437     ##################################################################
438     ##   Calculate average kex0, change amino acid abbreviations   ##
439     ##   Also check to see if a reference B0 field is required.    ##
440     ##################################################################
441
442
```

```
443  # TODO Check printing of graphs, R2obsMax may need to be redone and may need to separate points
444
445  my $kex0_ave;
446  my $pa_ave;
447  my $Bref;
448  if ($calc_residuals == 0) {
449      #Calculate average kex0 and pa for all residues
450      #kex is read in from files in (ms)^(-1), must convert to (s)
451      my $kex0_tot = 0;
452      my $pa_tot = 0;
453      foreach my $resi_num (@resi_files) {
454          $kex0_tot += $kex0{$resi_num};
455          $pa_tot += $pa{$resi_num};
456      }
457      $kex0_ave = $kex0_tot / ($#resi_files + 1);
458      $pa_ave = $pa_tot / ($#resi_files + 1);
459
460      # Check to see if B0 is equal for all residues
461      # If not, ask for a reference field
462      my $diff_field_check = 0;
463      foreach my $resi_num (@resi_files) {
464          if (exists $B0_master{$resi_num}[1]) {
465              $diff_field_check = 1;
466              last;
467          }
468      }
469      if ($diff_field_check == 0) {
470          for (my $i = 0; $i < $#resi_files; $i++) {
471              if (!($B0{$resi_files[$i]}[0]  == $B0{$resi_files[$i+1]}[0])) {
472                  $diff_field_check = 1;
473                  last;
474              }
475          }
476      }
477
478      if ($diff_field_check == 0) {
479          $Bref = $B0{$resi_files[0]}[0];
480          printf "\n\tAll fields (B0) are identical (%5.2f T).\n", $Bref;
481          print "\tNo reference field is needed\n";
482      } elsif ($diff_field_check == 1) {
483          my $correct_field = 0;
484          print "\n\tDifferent B0 fields found:\n";
485          print "\tResidue\t\tField(s)\n";
486
487          foreach my $resi_num (@resi_files) {
488              print "\t$resi_num\t\t";
489              my $i = 0;
490              while (exists $B0_master{$resi_num}[$i]) {
491                  printf "%5.2f", $B0_master{$resi_num}[$i];
492                  if (exists $B0_master{$resi_num}[$i+1]) {print ", ";}
493                  $i++;
494              }
495              print "\n";
496          }
497
498          while ($correct_field == 0) {
499              print "\n\tPlease enter a reference field (Bref, in Teslas): ";
500              $Bref = <STDIN>;
501              chomp($Bref);
502              printf "\tYou have entered %5.2f Tesla(s).\n", $Bref;
503              print "\tIs this correct? (y|n) ";
504              my $field_confirm = <STDIN>;
505
506              if (($field_confirm =~ /Y/) || ($field_confirm =~ /y/)) {
507                  $correct_field = 1;
508              }
509
510              if ($Bref <= 0) {
511                  print "\n\tBref must be >= 0.  Please try again.\n";
512                  $correct_field = 0;
513              }
514          }
515      }
516  }
```

```
517
518 # TODO What if the user doesn't use one letter abbreviations for amino acids?
519
520 ##################################################################
521 ##   Generate *.in for single amino acids and globally, also   ##
522 ##   create *.errors file.  This is only done if the notebook  ##
523 ##   is being generated.                                       ##
524 ##################################################################
525
526 if ($calc_residuals == 0) {
527
528         open(ALL_IN, "> all_resi.in");              # Open all_resi.in for output
529         open(ALL_ERR, "> all_resi.error");          # Open all_resi.error for output
530
531         # Counter to determine position in residue list for generating
532         # Kronecker delta function
533         my $position = 0;
534
535         foreach my $resi_num (@resi_files) {
536
537                 # Print the *.in file for each residue
538                 open(RESI_IN, "> $resi_num.in");
539                 for (my $i = 0; $i < $total_points{$resi_num}; $i++) {
540                         printf RESI_IN "%11.9f     %4.2E\n", $tcp{$resi_num}[$i], $R2obs{$resi_num}[$i];
541                 }
542                 close(RESI_IN);
543
544                 # Now append information for each residue to the main *.in and *.error files
545                 for (my $i = 0; $i < $total_points{$resi_num}; $i++) {
546
547                         #Set up Kronecker delta function
548                         for (my $j = 0; $j < ($#resi_files+1); $j++) {
549                                 if ($j == $position) { print ALL_IN "0  "; }
550                                 else { print ALL_IN "1  "; }
551                         }
552
553                         # Print field strength, tcp, and R2obs
554                         printf ALL_IN "%4.1f  %11.9f  %4.2E\n", $B0{$resi_num}[$i], $tcp{$resi_num}[$i], $R2obs{$resi_num}[$i];
555
556                         # Print the error file
557                         printf ALL_ERR "%7.5f\n", $R2err{$resi_num}[$i];
558
559                 }
560                 $position++;
561         }
562
563         close(ALL_IN);
564         close(ALL_ERR);
565         print "\n\tResidue output files successfully written.\n";
566 }
567
568 ##################################################################
569 ##     The following text is used to perform the CHI^2         ##
570 ##     calculation.  This is only performed after the notebook ##
571 ##     is evaluated.                                           ##
572 ##################################################################
573
574 if ($calc_residuals == 1) {
575         # Chi-squared calculation
576         open(FITS, "< globalfull_results");
577         my %residuals;
578         my @all_residuals;
579
580         my $begin_table = 0;
581         while (my $input = <FITS>) {
582                 # Find the line where the FitResiduals table begins
583                 if ($input =~ /FitResiduals/) {
584                         $begin_table = 1;
585                 }
586
587                 # Once we reach the right line, start splitting the fields
588                 if ($begin_table == 1) {
589                         chomp($input);
590                         my @one_line = split /\s+/, $input;
```

```
591
592                 # Removed commas, trailing and leading curly braces
593                 foreach my $field (@one_line) {
594                     if ($field =~ /\,/) {
595                         chop($field);
596                     }
597                     if ($field=~/\}\}/) {
598                         chop($field);
599                         chop($field);
600                     }
601                     if ($field=~/\{/) {
602                         $field = reverse($field);
603                         chop($field);
604                         $field = reverse($field);
605                     }
606                 }
607
608                 # Find the starting position for the first line
609                 # It probably isn't the very beginning of the line
610                 my $position;
611                 my $row = 0;
612                 for (my $i = 0; $i < scalar(@one_line); $i++) {
613                     if ($one_line[$i] =~ /->/) {
614                         $position = $i;
615                         $row = 1;
616                         last;
617                     }
618                 }
619
620                 # Push all numerical data into an array
621                 if ($row == 1) {
622                     # For the first line, put everything after the beginning
623                     # of the table into the array
624                     for (my $i = $position; $i < scalar(@one_line); $i++) {
625                         if ($one_line[$i] =~ /\w/) {
626                             push @all_residuals, $one_line[$i];
627                         }
628                     }
629                     $row++;
630                 } else {
631                     # The entirety of every other line can be added
632                     for (my $i = 0; $i < scalar(@one_line); $i++) {
633                         if ($one_line[$i] =~ /\w/) {
634                             push @all_residuals, $one_line[$i];
635                         }
636                     }
637                 }
638             }
639         }
640         close(FITS);
641
642         my $residual_position = 0;
643         foreach my $resi_num (@resi_files) {
644             for (my $i = 0; $i < $total_points{$resi_num}; $i++) {
645                 $residuals{$resi_num}[$i] = $all_residuals[$residual_position];
646                 $residual_position++;
647             }
648         }
649
650         # Now calculate the chi2 = (sum (residuals^2))/error^2
651         open(CHI2, "> globalfull_chisq");
652
653         my %BIG_resi_ID;
654         foreach my $resi_num (@resi_files) {
655             $BIG_resi_ID{$resi_num} = ucfirst $resi_ID{$resi_num};
656         }
657
658         foreach my $resi_num (@resi_files) {
659             my %chi_sq_MA;
660             print CHI2 "$BIG_resi_ID{$resi_num}\n";
661             print CHI2 "Taucp(s)  Residual    R2err\n";
662             for (my $i = 0; $i < $total_points{$resi_num}; $i++) {
663                 printf CHI2 " %7.5f   %7.4f  %7.4f\n", $tcp{$resi_num}[$i], $residuals{$resi_num}[$i], $R2_chi{$resi_num}[$i];
664                 $chi_sq_MA{$resi_num} += (($residuals{$resi_num}[$i])**2) / (($R2_chi{$resi_num}[$i])**2);
```

```
665             }
666             printf CHI2 "\nX^2(residue):      %8.4f\n", $chi_sq_CF{$resi_num};
667             printf CHI2 "X^2(residue_red):  %8.4f\n", $chi_sq_CF_red{$resi_num};
668             printf CHI2 "X^2(global):       %8.4f\n\n\n", $chi_sq_MA{$resi_num};
669         }
670         close(CHI2);
671         print "\n\tCHI^2 values calculated and printed to file \"globalfull_chisq\".\n\n";
672 }
673
674 ##################################################################
675 ##     The following text is used to generate the Mathematica  ##
676 ##     notebook from the input files.  This is only performed  ##
677 ##     when residuals are not calculated.                      ##
678 ##################################################################
679 print "\n";
680 if ($calc_residuals == 0) {
681         print "pause\n";
682         print "\tNow writing Mathematica notebook.\n";
683         # Now generate and load the Mathematica notebook
684         open(MATNB, "> cpMG_global_full.nb");
685
686
687         # Print VERY LONG header text
688         {
689             print MATNB"(************** Content-type: application/mathematica **************
690                 CreatedBy='Mathematica 4.2'
691
692                 Mathematica-Compatible Notebook
693
694 This notebook can be used with any Mathematica-compatible
695 application, such as Mathematica, MathReader or Publicon. The data
696 for the notebook starts with the line containing stars above.
697
698 To get the notebook into a Mathematica-compatible application, do
699 one of the following:
700
701 * Save the data starting with the line of stars above into a file
702   with a name ending in .nb, then open the file inside the
703   application;
704
705 * Copy the data starting with the line of stars above to the
706   clipboard, then use the Paste menu command inside the application.
707
708 Data for notebooks contains only printable 7-bit ASCII and can be
709 sent directly in email or through ftp in text mode.  Newlines can be
710 CR, LF or CRLF (Unix, Macintosh or MS-DOS style).
711
712 NOTE: If you modify the data for this notebook not in a Mathematica-
713 compatible application, you must delete the line below containing
714 the word CacheID, otherwise Mathematica-compatible applications may
715 try to use invalid cache data.
716
717 For more information on notebooks and Mathematica-compatible
718 applications, contact Wolfram Research:
719   web: http://www.wolfram.com
720   email: info\@wolfram.com
721   phone: +1-217-398-0700 (U.S.)
722
723 Notebook reader applications are available free of charge from
724 Wolfram Research.
725 *******************************************************************)\n\n\n\n\n" ;
726         }
727
728         # Begin notebook format information
729         {
730             print MATNB "(*NotebookFileLineBreakTest
731 NotebookFileLineBreakTest*)
732 (*NotebookOptionsPosition[     11159,       265]*)
733 (*NotebookOutlinePosition[     12138,       296]*)
734 (*  CellTagsIndexPosition[     12094,       292]*)
735 (*WindowFrame->Normal*)\n\n\n\n";
736         }
737
738         # Actual notebook text
```

```
739         {
740                 print MATNB "Notebook[{
741
742   Cell[CellGroupData[{
743   Cell[\"Initialize\", \"Section\",
744    FontColor->RGBColor[0, 0, 1]],
745
746   Cell[BoxData[{
747      \\(<< Statistics`HypothesisTests`\\), \"\\[IndentingNewLine]\",
748      \\(<< Statistics`NonlinearFit`\\), \"\\[IndentingNewLine]\",
749      \\(<< Graphics`MultipleListPlot`\\), \"\\[IndentingNewLine]\",
750      \\(Off[General::spell1]\\), \"\\[IndentingNewLine]\",
751      \\(Off[General::spell]\\)}], \"Input\"]
752   }, Open  ]],\n\n";
753         }
754
755
756         # The equation for fitting can be carefully altered here if needed
757         {
758                 print MATNB "Cell[CellGroupData[{
759
760   Cell[TextData[{
761    \"Define \",
762    Cell[BoxData[
763        \\(TraditionalForm\\`CPMG\\)]],
764    \" expressions\"
765   }], \"Section\",
766    Evaluatable->False,
767    FontColor->RGBColor[0, 0, 1]],
768
769   Cell[BoxData[{
770      \\(\\[Psi][\\[CapitalDelta]\\[Omega]_, kex_, B0_, Bref_] =
771         kex^2 - \\((\\[CapitalDelta]\\[Omega]*
772                 B0/Bref)\\)^2;\\)\\), \"\\[IndentingNewLine]\",
773      \\(\\[Zeta][\\[CapitalDelta]\\[Omega]_, kex_, B0_, Bref_,
774          pa_] = \\(-2\\)*\\((\\[CapitalDelta]\\[Omega]*B0/Bref)\\)*
775          kex*\\((pa - \\((1 - pa)\\))\\);\\)\\), \"\\[IndentingNewLine]\",
776      \\(\\(Dp[\\[CapitalDelta]\\[Omega]_, kex_, B0_, Bref_,
777          pa_] = \\((1/
778             2)\\)*\\((1 + \\((\\((\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0,
779                     Bref] + 2*\\((\\[CapitalDelta]\\[Omega]*B0/Bref)\\)^2)\\)/
780                 Sqrt[\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0,
781                       Bref]^2 + \\[Zeta][\\[CapitalDelta]\\[Omega], kex, B0,
782                       Bref, pa]^2])\\))\\);\\)\\), \"\\[IndentingNewLine]\",
783      \\(\\(Dm[\\[CapitalDelta]\\[Omega]_, kex_, B0_, Bref_,
784          pa_] = \\((1/
785             2)\\)*\\((\\(-1\\) + \\((\\((\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0,
786                     Bref] + 2*\\((\\[CapitalDelta]\\[Omega]*B0/Bref)\\)^2)\\)/
787                 Sqrt[\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0,
788                       Bref]^2 + \\[Zeta][\\[CapitalDelta]\\[Omega], kex, B0,
789                       Bref, pa]^2])\\))\\);\\)\\), \"\\[IndentingNewLine]\",
790      \\(\\(\\[Eta]p[\\[CapitalDelta]\\[Omega]_, kex_, B0_, Bref_, pa_,
791          tcp_] = \\((tcp/Sqrt[2])\\)*
792          Sqrt[\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0, Bref] +
793             Sqrt[\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0,
794                    Bref]^2 + \\[Zeta][\\[CapitalDelta]\\[Omega], kex, B0,
795                    Bref, pa]^2]];\\)\\), \"\\[IndentingNewLine]\",
796      \\(\\(\\[Eta]m[\\[CapitalDelta]\\[Omega]_, kex_, B0_, Bref_, pa_,
797          tcp_] = \\((tcp/Sqrt[2])\\)*
798          Sqrt[\\(-\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0, Bref]\\) +
799             Sqrt[\\[Psi][\\[CapitalDelta]\\[Omega], kex, B0,
800                    Bref]^2 + \\[Zeta][\\[CapitalDelta]\\[Omega], kex, B0,
801                    Bref, pa]^2]];\\)\\), \"\\[IndentingNewLine]\",
802      \\(\\(R2full[\\[CapitalDelta]\\[Omega]_, R20_, B0_, Bref_, kex_, pa_, tcp_] =
803         R20 + \\((1/
804                2)\\)*\\((kex - \\((\\((1/tcp)\\)*
805                    ArcCosh[\\((Dp[\\[CapitalDelta]\\[Omega], kex, B0, Bref, pa]*
806                          Cosh[\\[Eta]p[\\[CapitalDelta]\\[Omega], kex, B0,
807                             Bref, pa,
808                             tcp]])\\) - \\((Dm[\\[CapitalDelta]\\[Omega],
809                           kex, B0, Bref, pa]*
810                          Cos[\\[Eta]m[\\[CapitalDelta]\\[Omega], kex, B0,
811                             Bref, pa, tcp]])\\)])\\))\\);\\)\\)}], \"Input\"]
812   }, Open  ]],\n\n";
```

```
813         }
814
815         # Generate Kronecker delta functions
816         {
817                 print MATNB "Cell[CellGroupData[{
818
819 Cell[\"Functions for Nonlinear Fitting\", \"Section\",
820  FontColor->RGBColor[0, 0, 1]], \n\n";
821
822
823 print MATNB "Cell[BoxData[
824    \\(\\(globalfull[";
825
826                 # Print AA_, AB_, AC_, etc...
827                 # NOTE added second letter so more than 26 residues could be fit
828                 my $big_alpha_1="A";
829                 my $big_alpha_2="A";
830                 my $limit;
831                 for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
832                         if (((($#resi_files+1)-(26*($j-1))) < 26) {
833                                 $limit = ($#resi_files+1)-26*($j-1);
834                         } else {
835                                 $limit = 26;
836                         }
837
838                         for (my $i = 1; $i <= $limit; $i++) {
839                                 print MATNB "$big_alpha_1$big_alpha_2\_, ";
840                                 if ($big_alpha_2 =~ /Z/) {
841                                         $big_alpha_2="A";
842                                 } else {
843                                         $big_alpha_2++;
844                                 }
845                         }
846                         $big_alpha_1++;
847                 }
848
849                 print MATNB "\n\t\t";
850
851                 # Print chemical shift list
852                 foreach my $resi_num (@resi_files) {
853                         print MATNB "\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num}\_, "
854                 }
855
856                 # Print R20residue list...
857                 foreach my $resi_num (@resi_files) {
858                         print MATNB "R20$resi_ID{$resi_num}\_, "
859                 }
860
861                 print MATNB "kex_, pa_, tcp_, B0_, \\ Bref_] =\n";
862
863                 $big_alpha_1="A";
864                 $big_alpha_2="A";
865                 for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
866                         if (((($#resi_files+1)-(26*($j-1))) < 26) {
867                                 $limit = ($#resi_files+1)-26*($j-1);
868                         } else {
869                                 $limit = 26;
870                         }
871
872                         for (my $i = 1; $i <= $limit; $i++) {
873                                 print MATNB "\t\tKroneckerDelta[$big_alpha_1$big_alpha_2]*R2full[\\[CapitalDelta]\\[Omega]$resi_ID{$resi_files
    +$i-1]}\, R20$resi_ID{$resi_files[($j-1)*26+$i-1]}\, B0, Bref, kex, pa, tcp]";
874                         if (!((($j-1)*26+$i-1) == $#resi_files)) {
875                                 print MATNB " + \n";
876                         } else {
877                                 print MATNB "\\)\\(\\[IndentingNewLine]\\)
878    \\)\\)], \"Input\"]
879 }, Open  ]],\n\n";
880                         }
881                         if ($big_alpha_2 =~ /Z/) {
882                                 $big_alpha_2="A";
883                         } else {
884                                 $big_alpha_2++;
885                         }
```

```
886                     }
887                     $big_alpha_1++;
888                 }
889         }
890
891         {
892                 print MATNB "Cell[CellGroupData[{
893
894 Cell[TextData[{
895   \"Read in \",
896   Cell[BoxData[
897       \\(TraditionalForm\\`\\(\\(R2(1/\\[Tau]cp)\\)\\(\\ \\)\\)\\)]],
898   \"relaxation dispersion data\"
899 }], \"Section\",
900   FontColor->RGBColor[0, 0, 1]],
901
902 Cell[BoxData[
903     \\(\\(\\(\\[IndentingNewLine]\\)\\(\\(datafull =
904         ReadList[\"\\<$cur_dir/all_resi.in\\>\", Number,
905           RecordLists \\[Rule] True];\\)\\[IndentingNewLine]
906     \\(errorsfull =
907         ReadList[\"\\<$cur_dir/all_resi.error\\>\",
908           Number];\\)\\)\\)\\)], \"Input\"]
909 }, Open  ]],\n\n";
910         }
911
912         # Set up the global fitting equation
913         {
914
915                 print MATNB "Cell[BoxData[
916     \\(\\(\\(\\[IndentingNewLine]\\)\\(fitglobalfull =
917         NonlinearRegress[datafull,
918           globalfull[";
919
920                 # Print AA, AB, AC, etc...
921                 # NOTE added second letter so more than 26 residues could be fit
922                 my $big_alpha_1="A";
923                 my $big_alpha_2="A";
924                 my $limit;
925                 for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
926                         if ((($#resi_files+1)-(26*($j-1))) < 26) {
927                                 $limit = ($#resi_files+1)-26*($j-1);
928                         } else {
929                                 $limit = 26;
930                         }
931
932                         for (my $i = 1; $i <= $limit; $i++) {
933                                 print MATNB "$big_alpha_1$big_alpha_2\, ";
934                                 if ($big_alpha_2 =~ /Z/) {
935                                         $big_alpha_2="A";
936                                 } else {
937                                         $big_alpha_2++;
938                                 }
939                         }
940                         $big_alpha_1++;
941                 }
942
943                 # Print chemical shift list
944                 foreach my $resi_num (@resi_files) {
945                         print MATNB "\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num}\, "
946                 }
947
948                 # Print R20residue list...
949                 foreach my $resi_num (@resi_files) {
950                         print MATNB "R20$resi_ID{$resi_num}\, "
951                 }
952                 printf MATNB "\n\t\tkex, pa, tcp, B0, %5.2f], \n\t\t{", $Bref;
953
954                 # Print AA_, AB_, AC_, etc...
955                 # NOTE added second letter so more than 26 residues could be fit
956                 $big_alpha_1="A";
957                 $big_alpha_2="A";
958                 for (my $j=1; $j <= (($#resi_files - $#resi_files%26)/26+1); $j++) {
959                         if ((($#resi_files+1)-(26*($j-1))) < 26) {
```

```
960                     $limit = ($#resi_files+1)-26*($j-1);
961                 } else {
962                     $limit = 26;
963                 }
964
965                 for (my $i = 1; $i <= $limit; $i++) {
966                     print MATNB "$big_alpha_1$big_alpha_2\, ";
967                     if ($big_alpha_2 =~ /Z/) {
968                         $big_alpha_2="A";
969                     } else {
970                         $big_alpha_2++;
971                     }
972                 }
973                 $big_alpha_1++;
974             }
975             print MATNB "B0, tcp},\n\t\t{";
976
977             # Print chemical shift list...
978             foreach my $resi_num (@resi_files) {
979                 print MATNB "{\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num}\, $dw{$resi_num}}, "
980             }
981
982
983             # Print R20residue list...
984             foreach my $resi_num (@resi_files) {
985                 printf MATNB "{R20$resi_ID{$resi_num}\, %5.2f}, ", $R20{$resi_num};
986             }
987             #print MATNB "\n\t\t";
988
989             printf MATNB "\n\t\t{kex, %7.1f}, {pa, %4.2f}}, MaxIterations \\[Rule] 10000,
990         Weights \\[Rule] errorsfull,
991         RegressionReport \\[Rule] {BestFitParameters,
992           ParameterCITable, EstimatedVariance, ANOVATable,
993           AsymptoticCorrelationMatrix, FitResiduals}] >> globalfull_results\\n
994     ReadList[\"\\<globalfull_results\\>\"]\\)\\)\\)], \"Input\"],\n\n", $kex0_ave, $pa_ave;
995         }
996
997         # Extract values and fit to exchange curve
998         {
999             my $j = 1;
1000            print MATNB "Cell[BoxData[
1001    \\(\\(\\(\\[IndentingNewLine]\\)\\(";
1002
1003            # The Chemical shift fits
1004            foreach my $resi_num (@resi_files) {
1005                print MATNB "\\(f\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num} =\n";
1006                print MATNB "       fitglobalfull[\\([1, 2, ";
1007                print MATNB "$j, 2]\\)];\\)\\[IndentingNewLine]\n";
1008                $j++;
1009            }
1010
1011
1012            # Setup the R20 fits
1013            foreach my $resi_num (@resi_files) {
1014                print MATNB "    \\(fR20$resi_ID{$resi_num} =\n";
1015                print MATNB "       fitglobalfull[\\([1, 2, $j, 2]\\)];\\)\\[IndentingNewLine]\n";
1016                $j++;
1017            }
1018
1019            print MATNB "    \\(fkex = fitglobalfull[\\([1, 2, $j, 2]\\)];\\)\\[IndentingNewLine]\n";
1020            $j++;
1021            print MATNB "    \\(fpa = fitglobalfull[\\([1, 2, $j, 2]\\)];\\)\\)\\)\\)], \"Input\"],\n\n";
1022
1023    # TODO fix this so everything is separate for different fields
1024    # foreach my $resi_num (@resi_files) {
1025    #       my $i = 0;
1026    #       while (exists $B0_master{$resi_num}[$i]) {
1027    #           print "$resi_num\t$resi_ID{$resi_num}\t$B0_master{$resi_num}[$i]\t$points{$resi_num}[$i]\t$R2obsMAX{$resi_num}[$i]\n
1028    #           $i++;
1029    #       }
1030    # }
1031            print MATNB "Cell[BoxData[{\n";
1032            foreach my $resi_num (@resi_files) {
1033                print MATNB "    \\(\\(f\\[Psi]$resi_ID{$resi_num} :=
```

```
        fkex^2 - \\((f\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num})\\)^2;\\)\\), \"\\
\\[IndentingNewLine]\",
    \\(\\(f\\[Zeta]$resi_ID{$resi_num} := \\(-2\\)*f\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num}*
        fkex*\\((fpa - \\((1 - fpa)\\))\\);\\)\\), \"\\[IndentingNewLine]\",
    \\(\\(fDp$resi_ID{$resi_num} := \\((1/
            2)\\)*\\((1 + \\((\\((f\\[Psi]$resi_ID{$resi_num} +
                2*\\((f\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num})\\)^2)\\)/
              Sqrt[f\\[Psi]$resi_ID{$resi_num}^2 +
                f\\[Zeta]$resi_ID{$resi_num}^2])\\))\\);\\)\\), \"\\[IndentingNewLine]\", \n";
                if ($resi_num == $resi_files[$#resi_files]) {
                    print MATNB "    \\(\\(fDm$resi_ID{$resi_num} := \\((1/
            2)\\)*\\((\\(-1\\) + \\((\\((f\\[Psi]$resi_ID{$resi_num} +
                2*\\((f\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num})\\)^2)\\)/
              Sqrt[f\\[Psi]$resi_ID{$resi_num}^2 + f\\[Zeta]$resi_ID{$resi_num}^2])\\))\\);\\)\\)}], \"Input\"],\n\nCell
[BoxData[{\n";
                } else {
                    print MATNB "    \\(\\(fDm$resi_ID{$resi_num} := \\((1/
            2)\\)*\\((\\(-1\\) + \\((\\((f\\[Psi]$resi_ID{$resi_num} +
                2*\\((f\\[CapitalDelta]\\[Omega]$resi_ID{$resi_num})\\)^2)\\)/
              Sqrt[f\\[Psi]$resi_ID{$resi_num}^2 +
                f\\[Zeta]$resi_ID{$resi_num}^2])\\))\\);\\)\\(\\[IndentingNewLine]\\)
    \\)\\), \"\\[IndentingNewLine]\", \n";
                }
            }

            foreach my $resi_num (@resi_files) {
                my $tau_name = $resi_ID{$resi_num}."[tcp_]";
                print MATNB "    \\(\\(r2$tau_name :=
        fR20$resi_ID{$resi_num} + \\((1/
            2)\\)*\\((fkex - \\((\\((1/tcp)\\)*
              ArcCosh[\\((fDp$resi_ID{$resi_num}*
                  Cosh[\\((tcp/Sqrt[2])\\)*
                    Sqrt[f\\[Psi]$resi_ID{$resi_num} +
                      Sqrt[f\\[Psi]$resi_ID{$resi_num}^2 +
                        f\\[Zeta]$resi_ID{$resi_num}^2]]])\\) - \\((fDm$resi_ID{$resi_num}*
                  Cos[\\((tcp/Sqrt[2])\\)*
                    Sqrt[\\(-f\\[Psi]$resi_ID{$resi_num}\\) +
                      Sqrt[f\\[Psi]$resi_ID{$resi_num}^2 +
                        f\\[Zeta]$resi_ID{$resi_num}^2]]])\\)])\\))\\);\\)\\)";
                if ($resi_num == $resi_files[$#resi_files]) {
                    print MATNB "}], \\
\"Input\"],\n\nCell[BoxData[{\n";
                } else {
                    print MATNB ", \"\\
\\[IndentingNewLine]\", \n";
                }
            }
        }

        # Set up the plots.  The plots are divided up so that only four residues are plotted
        # per graph.
        {
            foreach my $resi_num (@resi_files) {
                print MATNB "    \\(\\($resi_ID{$resi_num} =
        ReadList[\"\\<$cur_dir/$resi_num.in\\>\", \\ Number, \\
         RecordLists \\[Rule] True];\\)\\)";
                if ($resi_num == $resi_files[$#resi_files]) {
                    print MATNB "}], \"Input\"],\n\nCell[BoxData[{\n";
                } else {
                    print MATNB ", \"\\[IndentingNewLine]\", \n";
                }
            }

            #Sort data by last R2obs point to make graph scaling pretty
            sub by_value { $R2obsMAX{$b} <=> $R2obsMAX{$a} }
            my $i = 0;
            for my $R2MAX (sort by_value (keys(%R2obsMAX))) {
                $resi_files[$i] = $R2MAX;
                $i++;
            }

            for (my $graph_num = 1; $graph_num <= (($#resi_files-$#resi_files%4)/4 + 1); $graph_num++) {
                my $limit;
                if (($#resi_files+1)-(4*($graph_num-1)) < 4) {
```

```
1107                    $limit = ($#resi_files+1)-(4*($graph_num-1));
1108                } else {
1109                    $limit = 4
1110                }
1111                for (my $i = 1; $i <= $limit; $i++) {
1112                    print MATNB "    \\(\\(pFit$resi_ID{$resi_files[(($i-1)+(4*($graph_num-1)))]} =\n";
1113                    print MATNB "        Plot[r2$resi_ID{$resi_files[(($i-1)+(4*($graph_num-1)))]}\[tcp], {tcp, 0.0001, 0.011}, \n";
1114                    print MATNB "        PlotStyle \\[Rule] {RGBColor[";
1115                    if ($i == 1) {
1116                        print MATNB "1, 0, 0";
1117                    } elsif ($i == 2) {
1118                        print MATNB "0, 1, 0";
1119                    } elsif ($i == 3) {
1120                        print MATNB "0, 0, 1";
1121                    } else { print MATNB "1, 0, 1"; }
1122                    print MATNB "]}, \n";
1123                    print MATNB "        DisplayFunction \\[Rule] Identity];\\)\\), \"\\[IndentingNewLine]\",\n";
1124                }
1125                print "\n";
1126                print MATNB "    \\(\\(pDatfull =
1127        MultipleListPlot[";
1128
1129
1130                for (my $j = 1; $j <= $limit; $j++) {
1131                    print MATNB "$resi_ID{$resi_files[(($j-1)+(4*($graph_num-1)))]}\, ";
1132                }
1133
1134                print MATNB "\n        SymbolShape \\[Rule] {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4],
1135            PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
1136            DisplayFunction \\[Rule] Identity];\\)\\), \"\\[IndentingNewLine]\",
1137        \\(\\(Show[";
1138
1139
1140                for (my $j = 1; $j <= $limit; $j++) {
1141                    print MATNB "pFit$resi_ID{$resi_files[(($j-1)+(4*($graph_num-1)))]}\, ";
1142                }
1143
1144                print MATNB "pDatfull,
1145        PlotLabel \\[Rule] \"\\<Disperson Curves";
1146
1147                # Make a hash with capital letter ID's for Plot titles
1148                my %BIG_resi_ID;
1149                foreach my $resi_num (@resi_files) {
1150                    $BIG_resi_ID{$resi_num} = ucfirst $resi_ID{$resi_num};
1151                }
1152
1153                for (my $j = 1; $j <= $limit; $j++) {
1154                    print MATNB " $BIG_resi_ID{$resi_files[(($j-1)+(4*($graph_num-1)))]}";
1155                }
1156
1157                # TODO Figure out how to put Y-axis lable sideways
1158                # TODO Add command to export graphs
1159                print MATNB "\\>\",
1160        AxesLabel \\[Rule] {\"\\<\\[Tau]aucp\\>\", \"\\<R2obs\\>\"},
1161        DisplayFunction \\[Rule] \$DisplayFunction];\\)\\";
1162
1163                if ($graph_num == (($#resi_files-$#resi_files%4)/4 + 1)) {
1164                    print MATNB ")}], \"Input\"]\n";
1165                } else {
1166                    print MATNB "[IndentingNewLine]\\), \"\\
1167    \\[IndentingNewLine]\",\n";
1168                }
1169            }
1170        }
1171
1172        # Finish the notebook
1173        {
1174            print MATNB "},
1175    FrontEndVersion->\"4.2 for X\",
1176    ScreenRectangle->{{0, 1024}, {0, 768}},
1177    CellGrouping->Manual,
1178    WindowSize->{1006, 693},
1179    WindowMargins->{{Automatic, 1}, {Automatic, 0}},
1180    PrintingCopies->1,
```

```
1181  PrintingPageRange->{Automatic, Automatic},
1182  PrintingOptions->{\"PaperSize\"->{612, 792},
1183  \"PaperOrientation\"->\"Portrait\",
1184  \"PostScriptOutputFile\":>FrontEnd`FileName[{\$RootDirectory, \"home\", \"loria\", \\
1185  \"math\", \"global_cpmg\"}, \"cpmg.global.full_loria.nb.ps\", CharacterEncoding -> \\
1186  \"iso8859-1\"],
1187  \"Magnification\"->1}
1188  ]
1189
1190
1191
1192
1193
1194  (*******************************************************************
1195  End of Mathematica Notebook file.
1196  *******************************************************************)" ;
1197        }
1198
1199        close(MATNB);
1200        # Run Mathematica and load notebook
1201        #system("mathematica cpMG_global_full.nb");
1202  }
```

10.5.3 Example Mathematica notebook.

```
<< Statistics`HypothesisTests`

<< Statistics`NonlinearFit`

<< Graphics`MultipleListPlot`

Off[General::spell1]

Off[General::spell]
```

```
ψ[Δω_, kex_, B0_, Bref_] = kex^2 - (Δω * B0 / Bref)^2;

ζ[Δω_, kex_, B0_, Bref_, pa_] = -2 * (Δω * B0 / Bref) * kex * (pa - (1 - pa));

Dp[Δω_, kex_, B0_, Bref_, pa_] =
 (1 / 2) * (1 + ((ψ[Δω, kex, B0, Bref] + 2 * (Δω * B0 / Bref)^2) / Sqrt[ψ[Δω, kex, B0, Bref]^2 + ζ[Δω, kex, B0, Bref, pa]^2]));

Dm[Δω_, kex_, B0_, Bref_, pa_] =
 (1 / 2) * (-1 + ((ψ[Δω, kex, B0, Bref] + 2 * (Δω * B0 / Bref)^2) / Sqrt[ψ[Δω, kex, B0, Bref]^2 + ζ[Δω, kex, B0, Bref, pa]^2]));

ηp[Δω_, kex_, B0_, Bref_, pa_, tcp_] =
 (tcp / Sqrt[2]) * Sqrt[ψ[Δω, kex, B0, Bref] + Sqrt[ψ[Δω, kex, B0, Bref]^2 + ζ[Δω, kex, B0, Bref, pa]^2]];

ηm[Δω_, kex_, B0_, Bref_, pa_, tcp_] =
 (tcp / Sqrt[2]) * Sqrt[-ψ[Δω, kex, B0, Bref] + Sqrt[ψ[Δω, kex, B0, Bref]^2 + ζ[Δω, kex, B0, Bref, pa]^2]];

R2full[Δω_, R20_, B0_, Bref_, kex_, pa_, tcp_] =
 R20 +
  (1 / 2) *
   (kex -
     ((1 / tcp) * ArcCosh[(Dp[Δω, kex, B0, Bref, pa] * Cosh[ηp[Δω, kex, B0, Bref, pa, tcp]]) -
         (Dm[Δω, kex, B0, Bref, pa] * Cos[ηm[Δω, kex, B0, Bref, pa, tcp]])])));
```

```
globalfull[AA_, AB_, Δωd83_, Δωq101_, R20d83_, R20q101_, kex_, pa_, tcp_, B0_, Bref_] =
 KroneckerDelta[AA] * R2full[Δωd83, R20d83, B0, Bref, kex, pa, tcp] +
  KroneckerDelta[AB] * R2full[Δωq101, R20q101, B0, Bref, kex, pa, tcp]
```

$$\left(R20d83+\frac{1}{2}\left(kex-\frac{1}{tcp}\,\mathrm{ArcCosh}\left[-\frac{1}{2}\left(-1+\frac{kex^2+\frac{B0^2\,\Delta\omega d83^2}{Bref^2}}{\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega d83^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega d83^2}{Bref^2}\right)^2}}\right)\right.\right.\right.$$

$$\mathrm{Cos}\left[\frac{tcp\sqrt{-kex^2+\frac{B0^2\,\Delta\omega d83^2}{Bref^2}+\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega d83^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega d83^2}{Bref^2}\right)^2}}}{\sqrt{2}}\right]+$$

$$\frac{1}{2}\left(1+\frac{kex^2+\frac{B0^2\,\Delta\omega d83^2}{Bref^2}}{\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega d83^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega d83^2}{Bref^2}\right)^2}}\right)$$

$$\left.\left.\left.\mathrm{Cosh}\left[\frac{tcp\sqrt{kex^2-\frac{B0^2\,\Delta\omega d83^2}{Bref^2}+\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega d83^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega d83^2}{Bref^2}\right)^2}}}{\sqrt{2}}\right]\right]\right)\right)\mathrm{KroneckerDelta}[AA]+$$

$$\left(R20q101+\frac{1}{2}\left(kex-\frac{1}{tcp}\,\mathrm{ArcCosh}\left[-\frac{1}{2}\left(-1+\frac{kex^2+\frac{B0^2\,\Delta\omega q101^2}{Bref^2}}{\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega q101^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega q101^2}{Bref^2}\right)^2}}\right)\right.\right.\right.$$

$$\mathrm{Cos}\left[\frac{tcp\sqrt{-kex^2+\frac{B0^2\,\Delta\omega q101^2}{Bref^2}+\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega q101^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega q101^2}{Bref^2}\right)^2}}}{\sqrt{2}}\right]+$$

$$\frac{1}{2}\left(1+\frac{kex^2+\frac{B0^2\,\Delta\omega q101^2}{Bref^2}}{\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega q101^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega q101^2}{Bref^2}\right)^2}}\right)$$

$$\left.\left.\left.\mathrm{Cosh}\left[\frac{tcp\sqrt{kex^2-\frac{B0^2\,\Delta\omega q101^2}{Bref^2}+\sqrt{\frac{4\,B0^2\,kex^2\,(-1+2\,pa)^2\,\Delta\omega q101^2}{Bref^2}+\left(kex^2-\frac{B0^2\,\Delta\omega q101^2}{Bref^2}\right)^2}}}{\sqrt{2}}\right]\right]\right)\right)\mathrm{KroneckerDelta}[AB]$$

Read in R2(1/τcp) relaxation dispersion data

In[14]:=

```
datafull = ReadList["/home/mlgill/programs/Perl/global_cpMG_full/all_resi.in", Number, RecordLists → True];
 errorsfull = ReadList["/home/mlgill/programs/Perl/global_cpMG_full/all_resi.error", Number];
```

In[16]:=

```
fitglobalfull = NonlinearRegress[datafull, globalfull[AA, AB, Δωd83, Δωq101, R20d83, R20q101, kex, pa, tcp, B0, 14.10],
    {AA, AB, B0, tcp}, {{Δωd83, 845.4}, {Δωq101, 1614.9}, {R20d83, 12.05}, {R20q101, 14.81}, {kex, 1742.1}, {pa, 0.91}},
    MaxIterations → 10000, Weights → errorsfull,
    RegressionReport → {BestFitParameters, ParameterCITable, EstimatedVariance, ANOVATable, AsymptoticCorrelationMatrix,
      FitResiduals}] >> globalfull_results
ReadList["globalfull_results"]
```

Out[17]=

```
{{BestFitParameters → {Δωd83 → 948.918, Δωq101 → 1491.52, R20d83 → 13.8214, R20q101 → 12.8458, kex → 1756.98, pa → 0.926557},

                                  Estimate      Asymptotic SE      CI
                        Δωd83     948.918       90.3973            {751.959, 1145.88}
                        Δωq101    1491.52       231.084            {988.027, 1995.}
  ParameterCITable →    R20d83    13.8214       1.31466            {10.957, 16.6858}   ,
                        R20q101   12.8458       1.58715            {9.38773, 16.3039}
                        kex       1756.98       288.369            {1128.68, 2385.29}
                        pa        0.926557      0.0180815          {0.887161, 0.965953}

                                                                   DF      SumOfSq      MeanSq
                                          Model                    6       11241.       1873.5
  EstimatedVariance → 0.947117, ANOVATable → Error                 12      11.3654      0.947117,
                                          Uncorrected Total        18      11252.3
                                          Corrected Total          17      1561.42

                                  1.          0.90487     0.184842    -0.150413   -0.525565   0.865956
                                  0.90487     1.          0.533086    -0.037719   -0.753276   0.975751
                                  0.184842    0.533086    1.          0.510484    -0.823888   0.629685
  AsymptoticCorrelationMatrix →   -0.150413   -0.037719   0.510484    1.          -0.530059   0.169678   ,
                                  -0.525565   -0.753276   -0.823888   -0.530059   1.          -0.851613
                                  0.865956    0.975751    0.629685    0.169678    -0.851613   1.

  FitResiduals → {-0.921416, 0.16802, 0.557523, 0.383272, 0.314438, 0.270511, -1.40091, 0.299092,
    0.0889945, 0.295741, 0.354457, -0.0489485, -1.47073, 0.0300807, 1.04826, 2.75744, 2.05744, -4.73591}}}
```

In[18]:=

```
fΔωd83 = fitglobalfull[[1, 2, 1, 2]];
 fΔωq101 = fitglobalfull[[1, 2, 2, 2]];
 fR20d83 = fitglobalfull[[1, 2, 3, 2]];
 fR20q101 = fitglobalfull[[1, 2, 4, 2]];
 fkex = fitglobalfull[[1, 2, 5, 2]];
 fpa = fitglobalfull[[1, 2, 6, 2]];
```

In[24]:=

```
fϕd83 := fkex^2 - (fΔωd83)^2;

fζd83 := -2 * fΔωd83 * fkex * (fpa - (1 - fpa));

fDpd83 := (1/2) * (1 + ((fϕd83 + 2 * (fΔωd83)^2) / Sqrt[fϕd83^2 + fζd83^2]));

fDmd83 := (1/2) * (-1 + ((fϕd83 + 2 * (fΔωd83)^2) / Sqrt[fϕd83^2 + fζd83^2]));


fϕq101 := fkex^2 - (fΔωq101)^2;

fζq101 := -2 * fΔωq101 * fkex * (fpa - (1 - fpa));

fDpq101 := (1/2) * (1 + ((fϕq101 + 2 * (fΔωq101)^2) / Sqrt[fϕq101^2 + fζq101^2]));

fDmq101 := (1/2) * (-1 + ((fϕq101 + 2 * (fΔωq101)^2) / Sqrt[fϕq101^2 + fζq101^2]));
```

In[32]:=

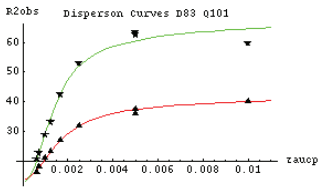
```
r2d83[tcp_] :=
 fR20d83 +
  (1/2) *
   (fkex -
     ((1/tcp) * ArcCosh[(fDpd83 * Cosh[(tcp/Sqrt[2]) * Sqrt[fϕd83 + Sqrt[fϕd83^2 + fζd83^2]]]) -
         (fDmd83 * Cos[(tcp/Sqrt[2]) * Sqrt[-fϕd83 + Sqrt[fϕd83^2 + fζd83^2]]])])));

r2q101[tcp_] :=
 fR20q101 +
  (1/2) *
   (fkex -
     ((1/tcp) * ArcCosh[(fDpq101 * Cosh[(tcp/Sqrt[2]) * Sqrt[fϕq101 + Sqrt[fϕq101^2 + fζq101^2]]]) -
         (fDmq101 * Cos[(tcp/Sqrt[2]) * Sqrt[-fϕq101 + Sqrt[fϕq101^2 + fζq101^2]]])]));
```

In[34]:=

```
d83 = ReadList["/home/mlgill/programs/Perl/global_cpMG_full/83.in", Number, RecordLists → True];

q101 = ReadList["/home/mlgill/programs/Perl/global_cpMG_full/101.in", Number, RecordLists → True];
```

In[36]:=

```
pFitd83 = Plot[r2d83[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[1, 0, 0]}, DisplayFunction → Identity];

pFitq101 = Plot[r2q101[tcp], {tcp, 0.0001, 0.011}, PlotStyle → {RGBColor[0, 1, 0]}, DisplayFunction → Identity];

pDatfull =
 MultipleListPlot[d83, q101,
  SymbolShape → {PlotSymbol[Triangle, 4], PlotSymbol[Star, 4], PlotSymbol[Diamond, 4], PlotSymbol[Box, 2]},
  DisplayFunction → Identity];

Show[pFitd83, pFitq101, pDatfull, PlotLabel → "Disperson Curves D83 Q101", AxesLabel → {"τaucp", "R2obs"},
 DisplayFunction → $DisplayFunction];
```



Created by *Mathematica* (May 1, 2006)