## Distribution Agreement

In presenting this dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I agree that the Library of the University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish, this dissertation may be granted by the professor under whose direction it was written, or in his absence, by the Dean of the Graduate School when such copying or publication is solely for scholarly purposes and does not involve financial gain. It is understood, that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

Signature

_____ _____

Li Sun Date

# Multidisciplinary Assessments of the Structure and Function of Co-enzyme $B_{12}$-Dependent Enzyme Ethanolamine Ammonia-lyase

By

Li Sun
Doctor of Philosophy
Physics

Advisor: Dr. Kurt Warncke

Approved for the Department by:

_____
Advisor

_____
Dr. Fereydoon Family
Committee Member

_____
Dr. Boi Hanh (Vincent) Huynh
Committee Member

_____
Dr. David Lynn
Committee Member

_____
Dr. James G. Nagy
Committee Member

Accepted:

_____
Lisa A. Tedesco, Ph.D. Dean of the Graduate School

_____
Date

# Multidisciplinary Assessments of the Structure and Function of Co-enzyme $B_{12}$-Dependent Enzyme Ethanolamine Ammonia-lyase

By

Li Sun
B.S., University of Science and Technology of China,
China 2003

Advisor
Kurt Warncke, Ph.D.

An Abstract of a Dissertation Submitted to the Faculty of the
James T. Laney Graduate School Studies of Emory University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
in
Physics
2009

# Abstract

## Multidisciplinary Assessments of the Structure and Function of Co-enzyme $B_{12}$-Dependent Enzyme Ethanolamine Ammonia-lyase

By Li Sun

The structure of the EutB protein from *Salmonella typhimurium*, which contains the active site of the coenzyme $B_{12}$ (adenosylcobalamin)-dependent enzyme, ethanolamine ammonia-lyase (EAL), has been predicted by using comparative modeling. Multiple three-dimensional template matching servers identified predominantly $\beta_8\alpha_8$, TIM-barrel proteins as templates. Model building was performed by using MODELLER. A conserved R160 in the active site was predicted to play a critical role in protein structure and catalysis. The protein chemical, kinetic, and electron paramagnetic resonance (EPR) and electron spin echo envelope modulation (ESEEM) spectroscopic properties of EAL with site-directed mutations in R160 have been characterized. R160I and R160E mutants fail to assemble into an EAL oligomer. The R160K and R160A mutants assemble, but R160A EAL is catalytically inactive. Simulations of the EPR spectra show that the $Co^{II}$-substrate radical pair separation distances are increased by 2.1 ±1.0 Å in R160K EAL relative to wild type, which corresponds to the predicted 1.6 Å change in arginine *versus* lysine side chain length. $^{14}N$ ESEEM from a hyperfine coupled protein nitrogen in wild type is absent in R160K EAL, which suggests that R160 interacts directly with the substrate radicals. ESEEM of the $^{2}H$-labeled substrate radical states in wild type and R160K EAL shows that the native separation distances among the substrate C1and C2, and coenzyme C5' reactant centers is conserved in the mutant protein. A ESEEM simulation toolbox, OPTESIM, was developed to facilitate the ESEEM analysis. OPTESIM allows automated numerical simulation of powder two- and three-pulse ESEEM for arbitrary number and type ($I$, $g_N$) of coupled nuclei. The EPR and ESEEM measurements evince a protein mediated force on the C5'-methyl center that is directed towards the reacting substrate species during the hydrogen atom transfer and radical rearrangement reactions. The results indicate that the positive charge at R160 side chain terminus is required for proper folding of EutB, assembly of a stable EAL oligomer, and that it is critical for catalysis. A consistent interpretation of the results is achieved by using the comparative model of EutB, which provides strong support for the model and the methodology of using this multidisciplinary approach in enzymological study.

# Multidisciplinary Assessments of the Structure and Function of Co-enzyme $B_{12}$-Dependent Enzyme Ethanolamine Ammonia-lyase

By

Li Sun
B.S., University of Science and Technology of China,
China 2003

Advisor
Kurt Warncke, Ph.D.

A Dissertation Submitted to the Faculty of the
James T. Laney Graduate School Studies of Emory University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy
in
Physics
2009

# Acknowledgement

I owe my deepest gratitude to my advisor, Prof. Kurt Warncke, for his generous encouragement, guidance and support from the ever beginning of my graduate study. These pages would be impossible without his kindly mentorship and friendship over my six years of long stay.

I am also heartily thankful to my committee members: Dr. Fereydoon Family, Dr. Boi Hanh (Vincent) Huynh, Dr. David Lynn, and Dr. James G. Nagy for their encouraging words, thoughtful criticism, time and attendance to my meetings.

I am grateful to my two previous coworkers, Olivia A. Groover and Dr. Jeffery M. Canfield. This project would be incomplete without their crucial contributions and thoughtful insights.

I would also like to thank my parents, my father Dexiu Sun and my mother Yunying Li for their constant support, encouragement and understanding.

I received numerous guidance and assistance from many people over the years of my study. It is impossible to list and thank all. The dissertation is dedicated to all of them.

# Table of Contents

# List of Figures

## Chapter 1 Introduction



## Chapter 2 Bioinformatics study of EAL by comparative modeling of EutB structure

## Chapter 3 Comprehensive spectroscopic, and Biochemical studies on the EAL mutants



## Chapter 4 Numerical simulation and nuclear parameter optimization of ESEEM with OPTESIM toolbox

# List of Tables

## Chapter 1 Introduction



## Chapter 2 Bioinformatics study of EAL by comparative modeling of EutB structure



## Chapter 3 Comprehensive spectroscopic, and Biochemical studies on the EAL mutants



## Chapter 4 Numerical simulation and nuclear parameter optimization of ESEEM with OPTESIM toolbox

# Chapter 1

# Introduction

## §1.1 Coenzyme $B_{12}$ dependent enzyme, ethanolamine ammonia-lyase

The bacterial enzyme, ethanolamine ammonia-lyase (EAL, EC 4.3.1.7; also known as ethanolamine deaminase) [1, 2] is a coenzyme $B_{12}$ (adenosylcobalamin)-dependent enzyme that catalyzes the conversion of aminoethanol to acetaldehyde and ammonia [3] by using highly reactive radical intermediates. EAL from *Salmonella typhimurium* is composed of a 453-residue, 49.4 kDa EutB protein subunit and a 286-residue, 32.1 kDa EutC protein subunit, that are coded by the *eutb* and *eutc* genes, respectively [4]. EutB contains the active site [5]. The enzyme molecule has a molecular mass of approximately 500 kDa, which indicates that the two subunits are present in a $EutB_6EutC_6$ stoichiometry [6].

EAL belongs to a larger enzyme family, known as adenosylcobalamin (AdoCbl, coenzyme $B_{12}$)-dependent enzyme superfamily. This adenosylcobalamin-dependent enzyme superfamily shares the feature of utilizing adenosylcobalamin for the generation of electron-deficient radicals to catalyze atom migration or elimination reactions [7-9]. This enzyme superfamily is categorized into three classes. Class I [7] coenzyme $B_{12}$-dependent enzymes catalyze the carbon skeleton rearrangement reaction, and include glutamate mutase [10], isobutyryl-CoA mutase [11], 2-methylene-glutarate mutase [12] and methyl-malonyl-CoA mutase [13, 14]. Class II [15] coenzyme $B_{12}$-dependent enzymes catalyze hydroxyl or amino group migration, followed by elimination, and include EAL, glycerol dehydratase [16, 17], propane-1,2-diol dehydratase [18, 19], and ribonucleotide triphosphate reductase [20]. Class III [21] coenzyme $B_{12}$-dependent enzymes, catalyze the amino migration reaction, and include lysine 2,3-aminomutase [22] and ornithine 4,5-aminomutase [23].

This adenosylcobalamin (AdoCbl, coenzyme $B_{12}$)-dependent enzyme superfamily shares the feature of utilizing adenosylcobalamin for the generation of electron-deficient radicals to catalyze atom migration or elimination reactions [7-9]. Adenosylcobalamin is a water-soluble vitamin, and essential for all cells [8, 24]. Adenosylcobalamin, as shown in Figure 1.1, is a substituted corrin-Co(III) complex in which the cobalt atom is bound to the four nitrogen atoms of the corrin ring, an axial group 5'-deoxyadenosyl, and 5,6-dimethylbenzimidazole (DMBI). The metal-alkyl bond between cobalt ($Co^{III}$) and C5' of the 5'-deoxyadenosyl axial ligand is the first metal-carbon bond discovered [24]; its homolytic cleavage generates electron-deficient radicals for the catalytic functions.

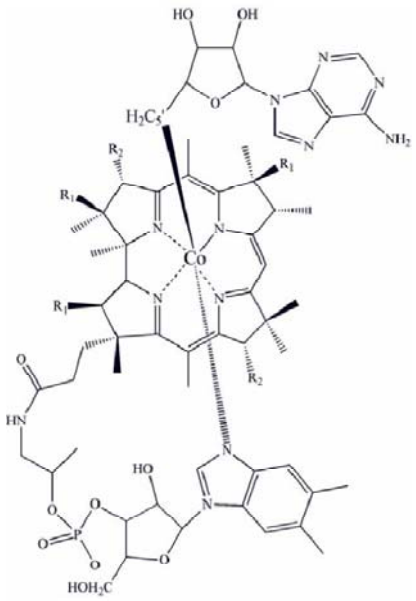


**Figure 1.1 Structure of adenosylcobalamin (AdoCbl, coenzyme $B_{12}$) [24].**

Figure 1.2 shows the minimal catalytic mechanism proposed for EAL [1, 2]. Following the homolytic cleavage of the metal-alkyl bond between cobalt ($Co^{III}$) and C5' of the 5'-deoxyadenosyl axial ligand, the C5' radical center migrates to the substrate binding site and abstracts a hydrogen atom from the carbinol carbon (C1) of the substrate,

forming a substrate radical with unpaired spin density localized on C1 [25]. The subsequent radical rearrangement reaction produces the product radical, which is predicted to be in the C1-carbinolamine form [26, 27] , with the unpaired electron localized on C2. After the rearrangement, a hydrogen atom transfer from the the 5'-methyl group of 5'-deoxyadenosine to C2 of the product radical forms a diamagnetic product species and regenerates the 5'-deoxyadenosyl radical, which recombines with $Co^{II}$ to complete the catalytic cycle [28].

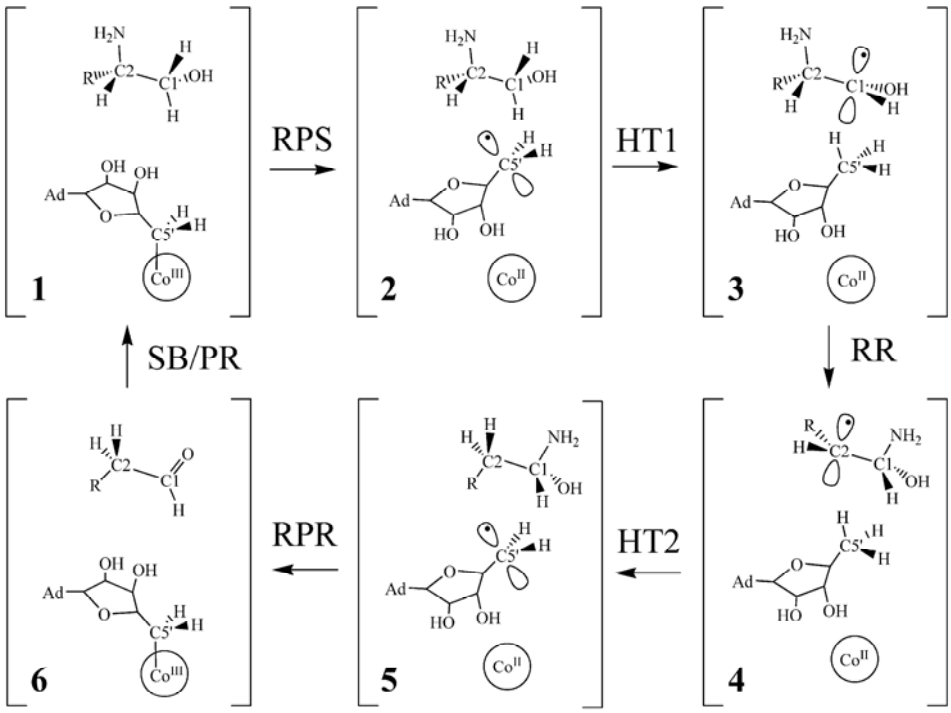


**Figure 1.2. Minimal mechanism of catalysis for coenzyme $B_{12}$-dependent EAL.** The forward direction of reaction is indicated by the arrows. Steps of the cycle are denoted RPS (radical pair separation), HT1 (first hydrogen atom transfer), RR (Radical rearrangement), HT2 (second hydrogen atom transfer), RPR (radical pair recombination), and SB/PR (substrate binding/product release). The brackets represent the active site region in the protein interior. The adenine group of the 5'-deoxyadenosyl β-axial ligand is denoted as Ad. The cobalt ion and its formal oxidation states are depicted, but the corrin ring and α-axial ligand of the coenzyme are not shown. The substrates are aminoethanol (R=H) or (*S*)-2-amino-propanol (R=$CH_3$). The forms of the substrate are as follows: States 1 and 2, bound substrate; State 3, substrate radical; State 4, product radical; State 5, bound carbinolamine intermediate; State 6, bound aldehdyde and ammonia products.

## §1.2 Previous spectroscopic study on EAL

In EAL, the $Co^{II}$-substrate radical pair intermediate accumulates during steady-state turnover on the substrates, (*S*)-2-aminopropanol [25], and aminoethanol. These radical pair states are stably cryotrapped [25, 29] for investigation by different techniques of EPR spectroscopy. In this section, previous spectroscopic studies on the EAL radical pair states are reviewed and structural conclusions are presented.

### §1.2.1 Continuous-wave EPR studies on EAL

Continuous-wave electron paramagnetic resonance (CW-EPR, other abbreviations include ESR and EMR), is a spectroscopic technique which measures the absorption of electromagnetic radiation by molecules, ions, or atoms possessing electrons with unpaired spins (S>0). Discovered by Yevgeny Zavoisky and Brebis Bleaney independently during the 1940s, CW-EPR has been an indispensable tool for studying free radicals and paramagnetic centers, and it has been applied to many fields spanning over chemistry, physics, and biology. The basis of CW-EPR is the electron Zeeman effect. In the presence of an external magnetic field of strength $B_0$, the electron magnetic moment aligns either parallel or antiparallel to the field, corresponding to $m_s$= -½, or $m_s$= +½ states, respectively. In the simplest single electron S=1/2 system, the energy of the electron is given by

$$E = g\beta_e B_0 m_s \qquad \text{[Eq. 1.1]}$$

in which $g$ is the electron $g$-factor, and $\beta_e$ is the Bohr magneton. The energy difference of the two $m_s$ states is $\Delta E = g\beta_e B_0$. The electron is induced to transit between the two energy levels, by absorbing or emitting electromagnetic radiation that satisfies the resonance condition:

$$h\nu = \Delta E = g\beta_e B_0 \qquad \text{[Eq. 1.2]}$$

where $h$ is the Plank constant, and $\nu$ is the electromagnetic radiation frequency. In thermal equilibrium, the population ratio of the two $m_s$ states is given by the Boltzmann distribution:

$$\frac{n_{+1/2}}{n_{-1/2}} = \exp\left(-\frac{\Delta E}{kT}\right) = \exp\left(-\frac{h\nu}{kT}\right) \qquad \text{[Eq. 1.3]}$$

With this ratio being less than 1 at thermal equilibrium, and the existence of electron spin-lattice relaxation, which causes the spin to return from $m_s = +½$, to $m_s = –½$ state, an overall absorption of the electromagnetic radiation is observed at the resonance condition.

In a more complex system with multiple unpaired electrons and coupled nuclei, the multiple energy differences between the system's energy states, or, the resonance conditions similar to that in Eq. 1.2 are usually derived from the system's Hamiltonian, which has the form:

$$\mathbf{H = EZ + NZ + HF + EX + EDIP} \qquad \text{[Eq. 1.4]}$$

where **EZ** is the electron Zeeman term, **NZ** is the nuclear Zeeman term, **HF** is the electron-nuclear hyperfine coupling term, **EX** is the electron spin-spin isotropic exchange term, and **EDIP** is the electron spin-spin dipolar interaction term.

Simulation of the CW-EPR spectra yields electron spin-spin isotropic exchange constant (*J*), and distance between to bi-radical with Ångström accuracy. Previous CW-EPR studies on $Co^{II}$-substrate radical pair intermediate state in EAL show a $Co^{II}$-C1 separation of 11 ±1 Å in the (*S*)-2-aminopropanol-generated $Co^{II}$-substrate radical pair [30, 31] , and 9.3 ±1 Å in the aminoethanol-generated $Co^{II}$-substrate radical pair [30, 32], with the assumption of the separation being along the cobalt $g_{||}$ axis. The approximately 2 Å shift of the C1 position in the (*S*)-2-aminopropanol-generated radical relative to the aminoethanol-generated radical is consistent with the methyl substitution at C2 in (*S*)-2-aminopropanol, because the extra methyl group in aminopropanol causes this substrate to position itself approximately 2 Å away from the cofactor.

### §1.2.2 Electron spin echo envelope modulation (ESEEM) studies on EAL

Electron spin echo (ESE) method is a pulsed technique of EPR spectroscopy [33]. Linearly polarized microwave pulses with the oscillating magnetic field ($B_1$) perpendicular to the external magnetic field ($B_0$) are used to manipulate the electron magnetization vector. Figure 1.4 shows the geometry of $B_0$ and $B_1$. In the absence of the electromagnetic wave pulses, electron magnetization that is not aligned with $B_0$ will perform Larmor precession along $B_0$ axis with a frequency of

$$\omega_L = \frac{2\mu_e}{\hbar} B_0 \qquad \text{[Eq. 1.5]}$$

where $\mu_e$ is the electron magnetic moment. Since the linear polarized electromagnetic wave can be decomposed into two counter-rotating circular polarized electromagnetic waves, if this electromagnetic wave also has a frequency of $\omega_L$ (the resonance condition), the electron magnetization with Larmor precession and the oscillating magnetic field $B_1$ in the lab frame are static in the frame attaching to one of the circular polarized electromagnetic wave, which is often referred to as the rotating frame. Therefore, in the rotating frame, the electromagnetic wave pulses with specific widths rotate the electron magnetization vector by specific angles (typically, $\pi/2$ or $\pi$) about the $B_1$ axis.

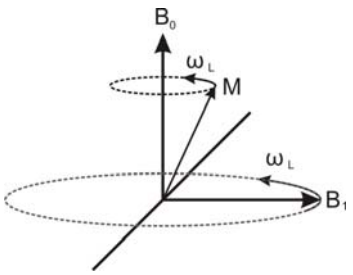


**Figure 1.3. $B_0$ and $B_1$ fields in lab frame under resonance condition. M** represents electron magnetization vector.

Figure 1.4A shows the schematic pulse timing diagram for typical one-dimensional two-pulse ESEEM experiments. Figure 1.5 shows the magnetization vector manipulation by the two pulses. Due the inhomogeneous broadening of EPR lineshapes, not all electron magnetization vectors will precess with the Larmor frequency, after their rotation to the x-y plane by the first ($\pi/2$) pulse. Therefore, the off-resonance electron species will cause the electron magnetization vectors to "fan" out for the duration of $\tau$ between the two pulses. After the reversal of the magnetization vectors on the x-y plane by the second ($\pi$) pulse, those vectors "refocus" at time $2\tau$, to form the electron spin echo. ESEEM is the waveform of the echo amplitudes versus $\tau$, which records the distribution of the off-resonance electron species. Since hyperfine (hf) and superhyperfine (shf) couplings of nearby nuclei cause the effective local field of the electron to change, and hence the inhomogeneous broadening, ESEEM is a sensitive method for detecting the local environment of an unpaired electron with hf or shf coupled nuclei.

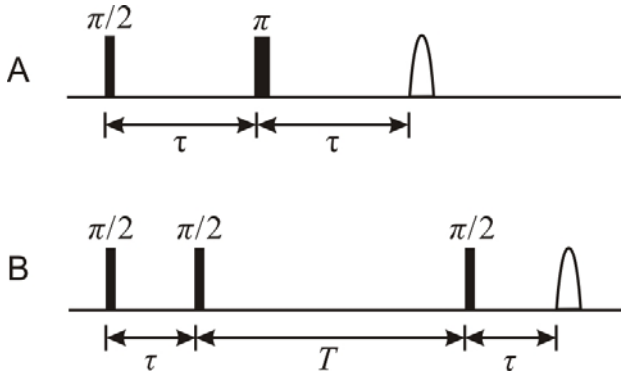


**Figure 1.4. Pulse timing diagrams for the two- and three-pulse ESEEM pulse sequences.** A. Two-pulse sequence: $\pi/2 - \tau - \pi - \tau -$ *detection*. B. Three-pulse sequence: $\pi/2 - \tau - \pi/2 - T - \pi/2 - \tau -$ *detection*.

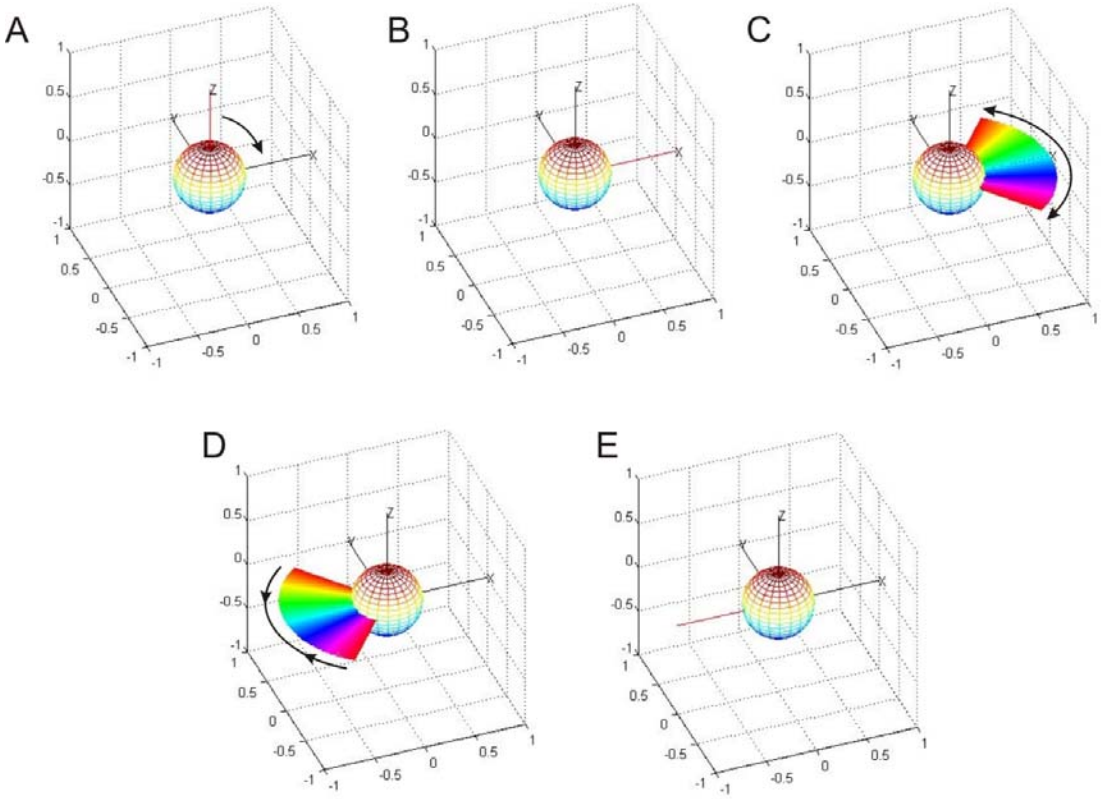


**Figure 1.5. Magnetization vector manipulation by two pulse ESEEM sequence.** $B_0$ is along z axis, $B_1$ is along y axis. A. before the first (π/2) pulse; B. right after the first (π/2) pulse; C. between the first (π/2) pulse and the second (π) pulse; D. after the second (π) pulse; E. echo formation at t = 2τ.

In additional to the two-pulse sequence, the three-pulse sequence is widely used in ESEEM experiments. Figure 1.4B shows the schematic pulse timing diagram for one-dimensional three-pulse ESEEM experiments. Three π/2 pulses are applied at $t = 0$, $\tau$ and $\tau + T$, and the echo of interest appears at $t = 2\tau+T$. The magnetization vector diagram is shown in Figure 1.6. The echo amplitude is recorded with varying $T$, while τ is fixed. Since the on resonance magnetization vectors are aligned to the y axis after the second

pulse, which are subject to longitudinal relaxation time ($T_1$), compared to the much shorter phase memory decay time ($T_M$) of the two-pulse sequence echo, three-pulse ESEEM offers supreme resolution for hf or shf coupled nuclei.

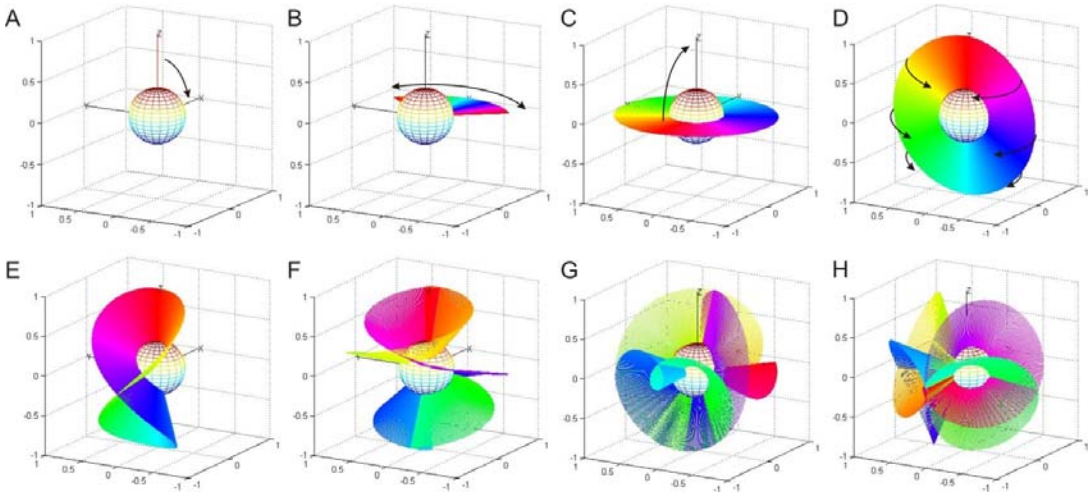


**Figure 1.6. Magnetization vector manipulation by three pulse ESEEM sequence.** $B_0$ is along z axis, $B_1$ is along y axis. A. before the first pulse; B. between the first and the second pulses; C. before the second pulse; D. right after the second pulse; E. between the second and the third pulses ($t = 2\tau$); F. between the second and the third pulses ($t>2\tau$); G. right after the third pulse; F. echo formation at $t = 2\tau+T$; net magnetization is along –x direction.

ESEEM spectroscopy has been applied to (*S*)-2-aminopropanol-generated and aminoethanol-generated radical intermediate states in EAL in previous studies [30, 34-38]. The magnetic coupling of the (*S*)-2-aminopropanol-generated radicals with a protein nitrogen nucleus has been characterized by using $^{14}N$ ESEEM [34, 35]. A three-dimensional model with sub-Ångström (Figure 1.7) of the C1, C2, and C5'-methyl group reactant center geometry in each of the $Co^{II}$-radical pair states are developed by multi-

frequency powder [36, 37] and orientation-selection [30, 38] ESEEM studies of the hyperfine interaction between the radicals and $^{2}$H-labeled hydrogen sites on the C5'-methyl group of 5'-deoxyadenosine, combining the $Co^{II}$-C1 distance information obtained by continuous-wave EPR studies. Figure 1.8 shows a diagram of the geometry of the aminoethanol-generated $Co^{II}$-substrate radical pair, with a full cofactor depicted.

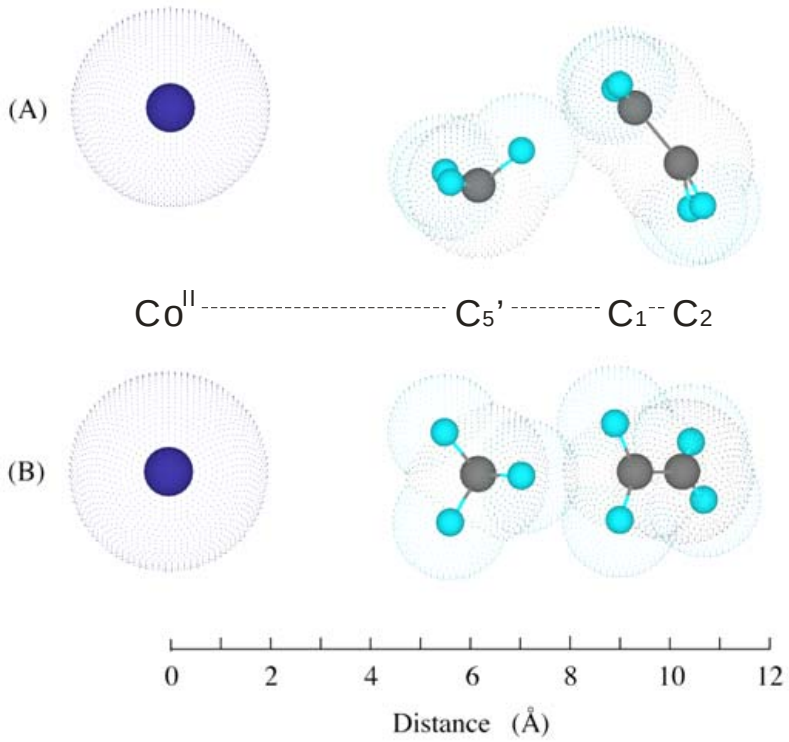


**Figure 1.7. Model for the structure of the reactant centers in the active site of the $Co^{II}$–substate radical pair state in EAL [39].** (A) View along the line perpendicular to the $Co^{II}$–C1–C2 plane. (B) View at 90° rotation about the $Co^{II}$–C1 axis, relative to view in (A).

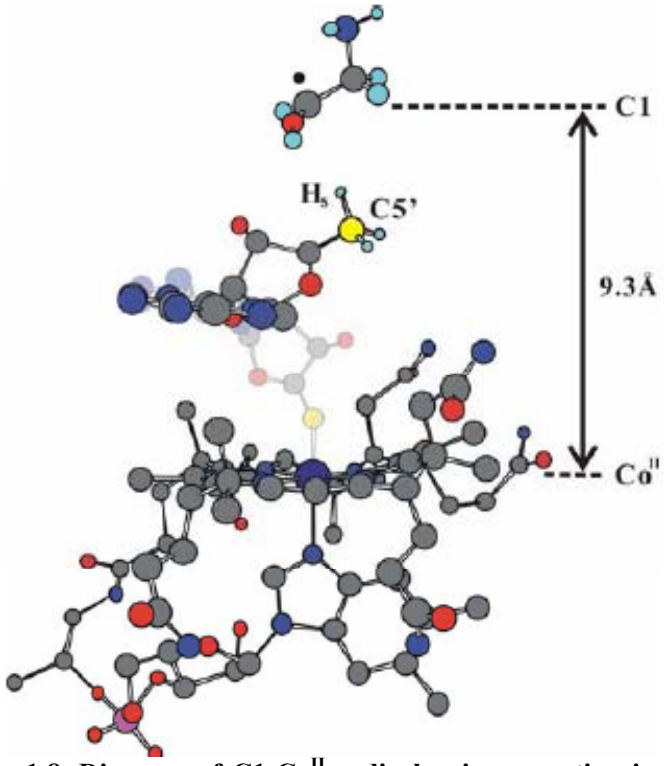


**Figure 1.8. Diagram of C1-$Co^{II}$ radical pair separation in $Co^{II}$-substrate radical pair intermediate state of EAL.**

### §1.2.3 Electron-Nuclear Double Resonance (ENDOR) studies on EAL

ENDOR (Electron-Nuclear Double Resonance) is an EPR technique that allows NMR transitions being detected in paramagnetic systems, for which those transitions are not straightforwardly detected by conventional NMR spectroscopy, due to the wide line broadending in the presence of unpaired electrons. The principle of ENDOR is to use a fixed microwave radiation to saturate electron Zeeman transitions, while applying another radio frequency radiation for the detection of nuclei near the unpaired electron. Since the development by Feher [40] in 1956, ENDOR spectroscopy has become a powerful technique for the added resolution to conventional EPR spectroscopy.

For the simplest case of one electron (S = ½) and one nucleus (I = ½) with an isotropic hyperfine constant $a$ in a strong external magnetic field $B_0$ along z axis, the spin Hamiltonian is given as:

$$\mathbf{H} = g\beta_e B_0 m_S - g_n \beta_N B_0 m_I + h a m_S m_I \qquad \text{[Eq. 1.6]}$$

where $g$ and $g_n$ are the electron and nuclear g factor; $\beta_e$ and $\beta_N$ are the Bohr and nuclear magneton , respectively; $m_S$ and $m_I$ are the respective magnetic spin quantum numbers, taking values of –½ and +½. The energy level diagram is shown in Figure 1.9. The electron spin transition selection rules ( $\Delta m_S = \pm 1$ , $\Delta m_I = 0$ ) indicate the allowed EPR frequencies are $\nu_{ERP1} = g\beta_e B_0 / h + a/2$ , and $\nu_{ERP2} = g\beta_e B_0 / h - a/2$ , each connecting state 1 and 3, state 2 and 4, respectively. Suppose $\nu_{ERP1}$ microwave radiation

is applied to the system, an increase of the $\nu_{ERP1}$ microwave radiation powers results in an increase of the EPR absorption signal, until the spin relaxation between state 1 and 3 does not compete with the pumping of $\nu_{ERP1}$ microwave radiation. At this point, the populations in state 1 and 3 are equal, and the system is saturated by the microwave radiation. More microwave radiation will not increase EPR absorption. Importantly, the actual spin relaxation between state 1 and 3 may be taking four routes, which include:

*a.* Direct electron spin relaxation (1 → 3);
*b.* Nuclear spin relaxation (3 → 4), followed by cross relaxation (4 → 1);
*c.* Nuclear spin relaxation (3 → 4), followed by electron relaxation (4 → 2), and nuclear spin relaxation (2 → 1).
*d.* Cross relaxation (3 → 2), followed by nuclear spin relaxation (2 → 1).

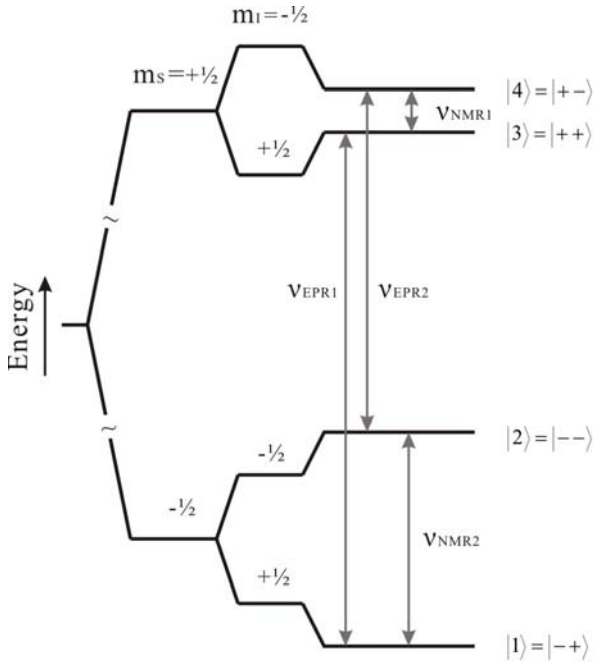


**Figure 1.9. Energy level diagram of one unpaired electron with a nucleus of I=1/2, with electron Zeeman, nuclear Zeeman and hyperfine coupling splittings.** The four energy states are denoted by number and signs of respective $m_S$ and $m_I$. Allowed EPR and NMR transitions are indicated.

In the presence of a radio frequency radiation corresponding to allowed NMR transition, for example, with a frequency of $\nu_{NMR1} = g_N \beta_N B_0 / h - a/2$, the transition rate between the state 3 and 4 is enhanced. As a result, the effective relaxation rate between state 3 and 1 is increased with relaxation routes *b* and *c* being enhanced. Thus, in the presence of radio frequency radiations corresponding to NMR transition frequencies, the saturated EPR absorption signal will exhibit a change in signal amplitude.

In the previous study on EAL (L. Sun, J. M. Canfield, K. Warncke, unpublished results), solvent exchangeable $^1$H in (S)-2-aminopropanol-derived substrate radical were analyzed by powder Q-band ENDOR technique. **Figure 1.10** shows the derivative ENDOR spectrum of the (S)-2-aminopropanol-derived radical around free $^1$H frequency at 1.206 T. Top spectrum is obtained with sample in $^1H_2O$, and the middle spectrum is obtained with sample equilibrated with $^2H_2O$. Spectroscopic features of solvent exchangeable $^1$H that are coupled to the substrate radical will be absent in the $^2H_2O$ spectrum. The difference of $^1H_2O$ and $^2H_2O$ spectra is also presented in the bottom with simulation overlaid. Two different solvent exchangeable $A_\perp$ features were assigned to two different dipolar coupled $^1$H nuclei. Parameters for the simulation are listed in Table 1.1. The effective electron-$^1$H distances of $r$=1.98 Å and $r$=2.06 Å are calculated by using point-dipole approximation. The distances are comparable to the distance between the unpaired spin density at C1 and the hydroxyl hydrogen calculated from molecular mechanics (1.9 Å). We have assigned the $r$=1.98 Å coupling to the hydroxyl hydrogen, and the $r$=2.06 Å coupling to either a proton that is hydrogen-bonded to the hydroxyl oxygen or a rotamer configuration of the hydroxyl hydrogen. Additionally, three more weakly-coupled, solvent exchangeable $^1$H features are revealed under higher resolution

conditions, shown in **Figure 1.10**. Parameters of their simulation are presented in Table 1.1 as well. Effective electron-$^{1}$H distances are calculated as 3.37 Å, 3.95 Å, and 4.66 Å by using point-dipole approximation. The 3.37 Å and 3.95 Å distances correspond to amine hydrogen and amine hydrogen-bonded proton. The origin of the 4.66 Å feature was yet unknown, although the weak coupled parameter indicated a remote solvent exchangeable proton from the protein. Figure 1.11 shows a diagram of the assignment of the feature in an (S)-2-aminopropanol-derived substrate radical model.

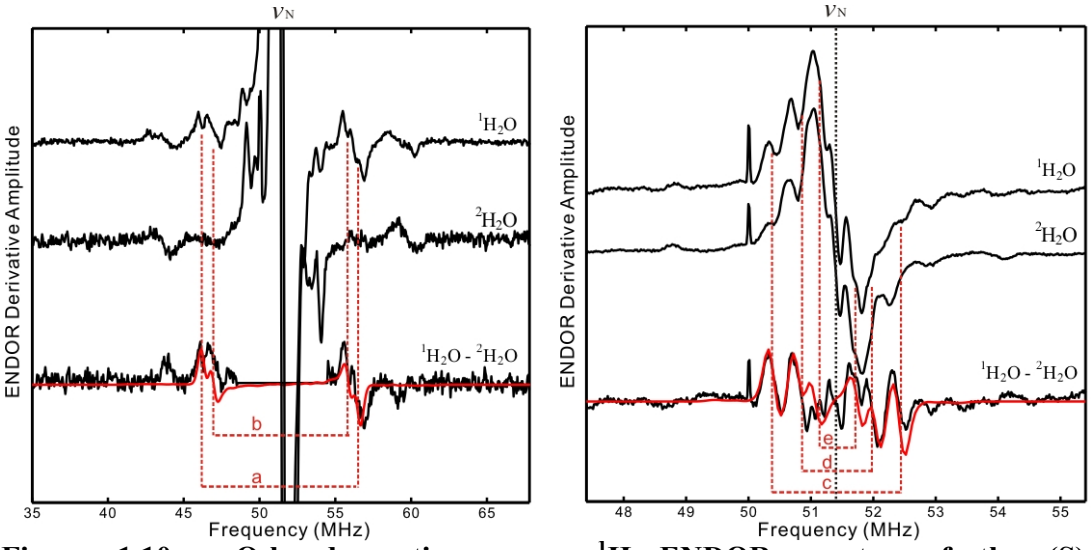


**Figure 1.10. Q-band continues-wave $^{1}$H ENDOR spectra of the (S)-2-aminopropanol-derived substrate radical in EAL.** $^{1}H_2O$ and $^{2}H_2O$ refer to samples prepared in natural abundance aqueous buffer and in buffer extensively exchanged with $^{2}H_2O$. Simulations are overlaid with difference spectra, with features in table indicate. Conditions: $B_0$, 1.206 T; $\nu_{MW}$, 34.14 GHz; T, 10 K; FM depth, 200 kHz (Left), 50 kHz (Right).

**Table 1.1. ENDOR simulation parameters.**

| Feature | $A_{\perp}$ (MHz) | Corresponding $r$ | Linewidth (MHz) |
|---|---|---|---|
| a | -10.23 | 1.98 Å | 0.28 |
| b | -9.06 | 2.06 Å | 0.28 |
| c | -2.06 | 3.37 Å | 0.14 |
| d | -1.28 | 3.95 Å | 0.14 |
| e | -0.78 | 4.66 Å | 0.14 |

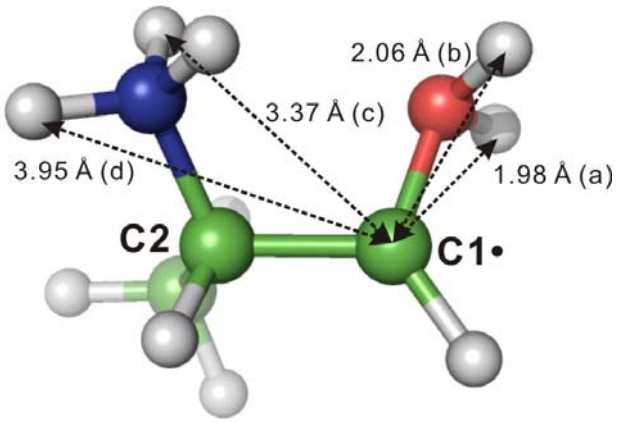


**Figure 1.11. Diagram of solvent exchangeable $^{1}$H features in (S)-2-aminopropanol-derived substrate radical.**

# Chapter 2

# Bioinformatics study of EAL by comparative modeling of EutB structure

## §2.1 Background and introduction

### §2.1.1 Co-enzyme $B_{12}$-dependent enzyme family protein structures

High-resolution structures of the Class I enzymes, methylmalonyl-CoA mutase, glutamate mutase and methylene glutarate mutase, and of the Class II enzymes, diol dehydratase, glycerol dehydratase and ribonucleotide triphosphate reductase, have been determined [16-19, 41-46]. The active site-containing large subunits were found to have the $\beta_8\alpha_8$, TIM-barrel fold, with the exception of the ribonucleotide triphosphate reductase, which has a $(\alpha/\beta)_{10}$ structure and the PFL-like glycyl radical enzyme fold characteristic of other classes of ribonucleotide reductases [46]. The $\beta_8\alpha_8$, TIM-barrel fold is present in approximately 10% of structurally-characterized proteins [47]. A recent bioinformatics study [48] has shown that the members of the S-adenosylmethionine (AdoMet, or SAM)-dependent radical enzyme superfamily [49-51], which use an iron-sulfur center-mediated reduction of AdoMet to generate the 5'-deoxyadenosyl initiator radical, are also α/β-barrel proteins. The AdoMet radical enzymes include over 600 unique sequences, and have a common, $\alpha_6\beta_6$, or three-quarters-barrel structure, with the full, $\beta_8\alpha_8$-barrel found in fewer members [48]. A molecular level description of the radical migration, hydrogen atom (or proton/electron) transfer, and radical rearrangement reactions in these systems is critical for defining biological functions and developing biomedical and technology opportunities.

At the time that the studies that form the basis of this dissertation chapter were conducted, the X-ray crystallographic structure of EAL had not been reported. We therefore used structural proteomics techniques of comparative modeling [52] to obtain a model of the active site-containing EutB protein of EAL from *S. typhimurium* [53, 54],

with several remarkable structural feature predictions. Although the X-ray crystallographic structure of EutB protein was reported recently [55], the comparative model of EutB remains significant in providing imperative research information, which is absent from the X-ray crystallographic structure. First, for the first time, the clear structural similarity between coenzyme $B_{12}$ dependent EAL and the other two Class II coenzyme $B_{12}$ dependent enzymes, specifically, diol dehydratase and glycerol dehydratase were shown by the comparative model. This structural analogy provides a unique perspective of comparative understanding of the mechanism for both coenzyme $B_{12}$ dependent classes. Second, the X-ray crystallographic structure of EutB protein lacks the cofactor $B_{12}$ and substrate. The comparative model provides exact active-site locality and positioning of the cofactor $B_{12}$ in the binding pocket. Another significance of this study is that, upon availability of the X-ray crystallographic structure, which verifies this comparative model of EutB, the methodology of using comparative modeling is proven viable in studying protein with no known structure.

### §2.1.2 Comparative modeling

Comparative modeling is the method of constructing an atomic model of a "target" protein amino acid sequence by using an experimentally determined three-dimensional structure of a related protein as a structural reference ("template"). Comparative modeling is based on the principle that protein tertiary structure is more conserved than protein primary structure and secondary structure [56]. The identification of templates is therefore based on either high level of target-template sequence similarity, or a similar fold shared by the target and template proteins. Constructing a model based on amino

acid sequence similarity is often referred as homology modeling. Applicability of homology modeling is directly related to the sequence identity. With more than 70% sequence identity, a homology model has about 2 Å root mean square deviation (RMSD) between the matched $C_\alpha$ atoms. However, with less than 30% sequence identity, the RMSD between the matched $C_\alpha$ atoms is typically more than 4 Å and increases exponentially with decreasing sequence identity. Therefore, homology modeling is not applicable to target sequence with less than 20% sequence identity to any protein sequence with experimentally determined structure. In this case, fold recognition, or protein threading methods can be used. The fold recognition method relates the target sequence to a specific fold using statistical knowledge, including prediction of target sequence secondary structures and search against database of proteins with solved structure. Figure 1 shows a flowchart of comparative modeling, with the general steps described as follows.

Step A1: Target protein sequence is obtained from credible publications or online databases.

Step A2: Target protein sequence is input into search engines to perform sequence similarity search. The process is referred as "FASTA" or "BLAST" search, which is offered by most of the online sequence databases [57]. Structure prediction programs, for example MODELLER [58], have the function of performing sequence similarity search, though the databases are often dated or only containing a subset. The sequence similarity search is based on the alignments of the target sequence with all the existing sequences inside the databases. A similarity score is assessed for each of the existing sequences. This score is based on the similarity of each of the amino acid pairs in the alignment. An

identical or chemically similar amino acids pair gives a positive score based on observed substitution frequencies in homologous sequences, while alignment gaps or a pair of amino acids with dissimilar chemical properties give negative scores.

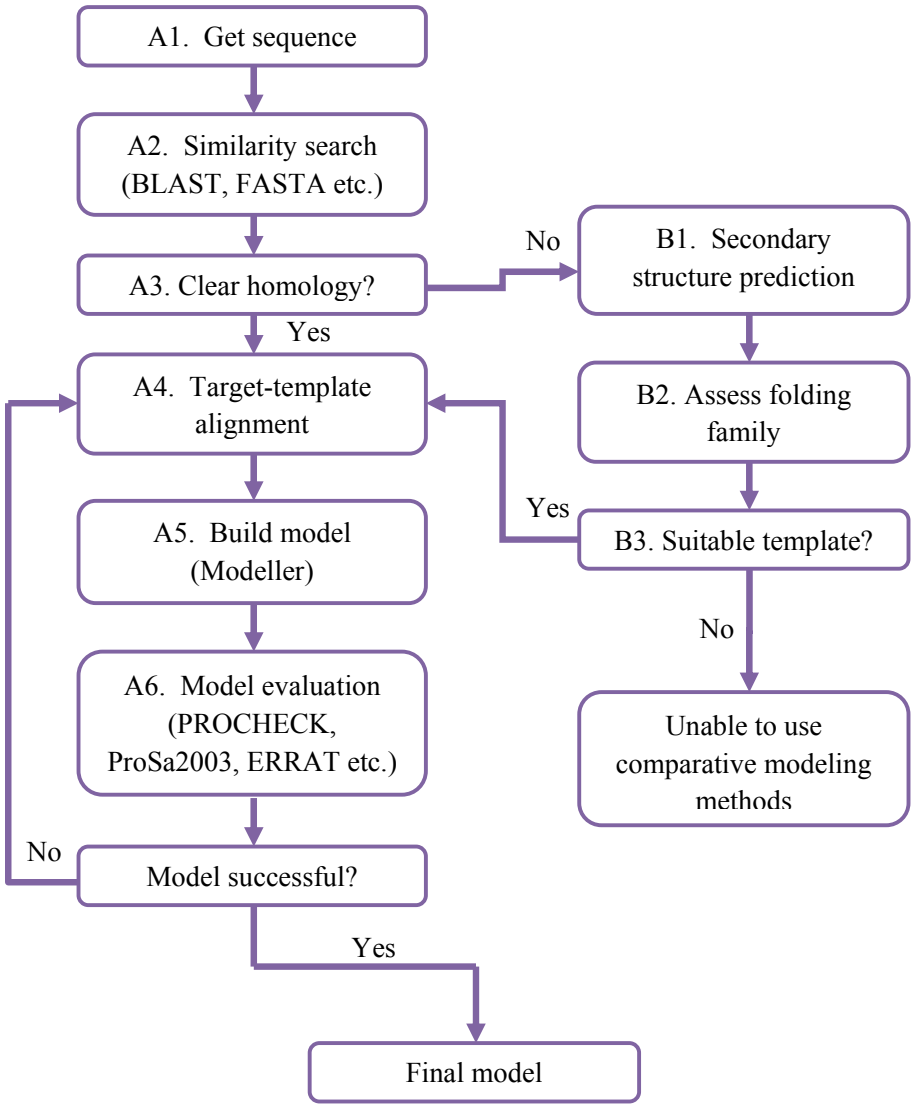


**Figure 2.1. Flowchart of comparative modeling.**

Step A3 and A4. If templates of sequence identity higher than 40% with the target sequence are found in step A2, the template-target alignment is performed based on sequence information. Though homology modeling is applicable to sequence identity lower than 40%, the constructed model is not reliable and may contain serious errors. Secondary structure prediction and fold recognition are used in this case. For high levels of the sequence similarity, different amino acid substitution matrices are used to refine the alignment, which are manually adjusted by eliminating gaps in secondary structure region and matching conserved residues both in template and target sequences.

Step A5. Using the alignment and template PDB files as inputs, models are built automatically by Modeller, which involves four phases: (1) framework construction, (2) side-chain building, (3) loop simulation, and (4) energy minimization. First, the main-chains of the templates are superimposed and assigned as the main-chain guideline of the target protein. The backbone of the target protein is constructed by minimization of the violation of stereochemical restrains. In the second phase, the side-chains are built by using this backbone as a framework. The loop regions are further simulated by *ab initio* methods in the third phase. In the last phase, using molecular mechanics, the target models are refined by minimizing the energy.

Step A6. The models built in step A5 are evaluated on the basis of the stereochemistry, the energy profile (the energy of each residue in the sequence) and the residue environment. PROCHECK [59] is used to evaluate the model stereochemistry. Bond lengths, bond angles, dihedral angles, packing and solvent accessibility are evaluated against the standards. Problematic residues can be identified by PROCHECK. Prosa2003 [60] and VERIFY3D [61] are used to evaluate the model energy profile by the

stereochemical profile and the mean force potential energy. Abnormal energy profile regions suggest incorrect alignment. Errat 2.0 [62] is used to evaluate the residue environment. The alignment is improved based on the evaluation results. Step A5 is repeated and models are rebuilt on the basis of the new alignment if necessary.

Step B1 and B2. If no known protein is identified to have a sequence similarity with the target sequence of higher than 40% in step A2, novel techniques are used to determine the distantly related protein templates and align the templates with the target. Secondary structures of the target protein are predicted first by knowledge-based methods. The possible topologies of the target protein are assessed based on the secondary structure prediction. This process of secondary structure prediction and fold topology assessment is often known as "threading", which has been implemented by a number of web servers. Multiple servers are often used to achieve a consensual prediction and to increase the chance of finding potential templates. Each of the servers will give possible templates and the template-target alignments. Templates are selected, and their alignments are improved manually according to the consensual prediction.

Step B3. Templates are selected based on both their server scores and their functional relations with the target protein. The improved server alignments are used to build target models. Models are evaluated and their alignments are improved, as described above.

## §2.2 Comparative modeling of EutB

### §2.2.1 Structural relationship of EutB protein to diol dehydratase and glycerol dehydratase

We have used structural proteomics techniques of comparative modeling [52] to obtain a model of the active site-containing EutB protein of EAL from *S. typhimurium* [53, 54]. Comparative modelling of EutB presents a challenge, because of the relatively large, 453 residue size, and because a sequence similarity search with PSI-BLAST [63] found that EutB displays no significant sequence identity to proteins of known structure in the Protein Data Bank (PDB). Thus, modeling by using "sequence homology" methods was not viable [56], and secondary structure prediction and three-dimensional conformation matching, or threading, were required to identify templates for the EutB target [56]. Remarkably, multiple fold recognition servers identified the active site-containing large subunits of diol dehydratase [16, 17] and glycerol dehydratase [18, 19], two other Class II coenzyme $B_{12}$-dependent enzymes, as templates. The program, MODELLER [58], was used to build the model of EutB, based on the large subunit templates from diol dehydratase (from *Klebsiella oxytoca*; PDB: 1eex:A) and glycerol dehydratase (from *Klebsiella pneumoniae*; PDB: 1mmf:A).

***Protein Sequences***

Protein sequences were obtained from the enzyme sequence databases, BRENDA [64] and SWISSPROT [65]. The databases were accessed in July, 2005. The EMBL (http://www.embl.org) accession numbers for EutB sequences from different bacterial

species are as follows: Q9KWC6 (*Agrobacterium rhizogenes*), Q59781(*Rhodococcus erythropolis*), Q8XUQ9 (*Ralstonia solanacearum*), Q889M2 (*Pseudomonas syringae pv. tomato*), Q8PK12 (*Xanthomonas axonopodis pv. citri*), Q89QX8 (*Bradyrhizobium japonicum*), Q97FL8 (*Clostridium acetobutylicum*), Q8Y7U5 (*Listeria monocytogenes*), Q8XLZ2 (*Clostridium perfringens*), Q892C9 (*Clostridium tetani*), Q8RH36 (*Fusobacterium nucleatum subsp. nucleatum*), P19264 (*Salmonella typhimurium*). P19264 from *S. typhimurium* (NCBI gene identification number: gi|6686281) displayed the strongest sequence homology to the EutB sequence reported by Faust and Babior [54], as later modified [1], which corresponds to the enzyme that we have used to perform our experimental studies [66-69]. P19264 was selected for use in the multiple sequence alignment and modelling.

***Sequence Similarity Search***

The P19264 EutB sequence from *S. typhimurium* was used as the target in a PSI-BLAST [63] search for sequence similarity in the NCBI non-redundant database (http://www.ncbi.nih.gov). The NCBI non-redundant database contains all non-redundant GenBank CDS translations, PDB, SwissProt, PIR, and PRF entries, excluding environmental samples (update: June 15, 2005; number of letters: 854,568,496; number of sequences: 2,521,256). All PSI-BLAST parameters were default.

The PSI-BLAST search for homology with the target EutB sequence P19264 returned 49 sequences in the first iteration, with scores ranging from 926 to 34.3 and *e*-values ranging from 0.0 to 8.7. The 46 sequences with *e*-value <0.1 were all for EutB

from the EAL family. The other 3 sequences, with *e*-values of 0.92, 1.9 and 8.7, have sequence identities with the target sequence for 35, 18, and 18 identical residues out of 453 residues, respectively (8% identity). The 46 sequences with *e*-value <0.1 were selected for the second iteration of PSI-BLAST search. The second iteration returned 17 new sequences with *e*-values ranging from 1.5 to 9.6. However, only short sequence identities were displayed among the set members. The PSI-BLAST results indicate that no similar sequences are available for sequence-based homology modelling of EutB.

***Secondary Structure Prediction and Fold Recognition***

The absence of significant sequence identity between EutB P19264 and all proteins of known structure led us to seek structural similarity by using the fold recognition approach, achieved with the GeneSilico Metaserver [70]. Results from the following multiple fold recognition servers were used: 3D-PSSM [71], FFAS03 [72], FUGUE 2.0 [73], MGenTHREADER [74], Pcons2 [75], Pcons5 [75], and Sam-T02 [76, 77]. The top hits of the FFAS03, Sam-T02, 3D-PSSM, FUGUE 2.0, Pcons2, and Pcons5 servers, and 10 of the 11 total hits in the range of "unsure" to "reliable", had the $\beta_8\alpha_8$, TIM-barrel fold. The complete server results are presented in Table 2.1. Remarkably, a clear relation emerged between EutB and the large subunits of the Class II "eliminase" coenzyme $B_{12}$-dependent enzymes, diol dehydratase and glycerol dehydratase, which also have the $\beta_8\alpha_8$, TIM-barrel fold. The Sam-T02 server found 1eex:A (large subunit, diol dehydratase) as the top hit, with a high confidence score of $4.4 \times 10^{-6}$ (for the Sam-T02 server, the score criteria are as follows: reliable, <0.0048; unsure, <1.183; unreliable, >1.183). The Pcons5 server found 1dio:A (large subunit, diol dehydratase)

and 1eex:A as its first and second hits, with scores of 1.5052 and 1.1273, respectively (Pcons5 criteria: reliable, >2.17; unsure, >1.03; unreliable, <1.03). The FUGUE 2.0 server found 1dio:A,

| Server | PDB Code | Score | Score Criteria | Protein | SCOP 1.69 Class | Fold |
|---|---|---|---|---|---|---|
| **FFAS03** | 1mmf:A | -9.03 | reliable <-38.2<br>unsure < -9.2 | Diol Dehydratase *(Klebsiella pneumoniae)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1dio:A | -8.84 | unreliable > -9.2 | Diol Dehydratase *(Klebsiella oxytoca)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| **MGenThreader** | 1efu:B | 0.577 | reliable> 0.65<br>unsure > 0.546<br>unreliable < 0.546 | Elongation Factor Complex Ef-Tu/Ef-Ts *(Escherichia coli)* | Alpha and beta proteins (a+b) & All alpha proteins (multiple domains) | Elongation factor Ts (EF-Ts), dimerisation domain & RuvA C-terminal domain-like & Elongation factor Ts (EF-Ts), dimerisation domain |
| | 1qax:A | 0.525 | | HMG-COA Reductase *(Pseudomonas mevalonii)* | Alpha and beta proteins (a+b) & All alpha proteins (multiple domains) | Ferredoxin-like & Substrate-binding domain of HMG-CoA reductase |
| **Sam-T02** | 1eex:A | 4.40E-06 | reliable< 0.0048<br>unsure < 1.183<br>unreliable > 1.183 | Diol Dehydratase *(Klebsiella oxytoca)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1leh:A | 1.3 | | Leucine Dehydrogenase *(Bacillus sphaericus)* | Alpha and beta proteins (a/b) (multiple domains) | NAD(P)-binding Rossmann-fold domains & Aminoacid dehydrogenase-like, N-terminal domain |
| **3D-PSSM** | 1f6y:A | 0.6 | reliable< 0.0065<br>unsure < 0.255 | Corrinoid/Iron-Sulfur Protein Methyltransferase *(Moorella thermoacetica)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1aj2 | 1.4 | unreliable > 0.255 | Dihydropteroate Synthase *(Escherichia coli)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| **FUGUE 2.0** | 1dio:A | 3.96 | reliable> 21.03<br>unsure > 3.69 | Diol Dehydratase *(Klebsiella oxytoca)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1ka2:A | 3.44 | unreliable < 3.69 | Carboxypeptidase Apo-Mg *(Pyrococcus furiosus)* | Alpha and beta proteins (a+b) | Zincin-like |
| **Pcons2** | 1aj2 | 1.49 | | Dihydropteroate Synthase *(Escherichia coli)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1ad1 | 1.404 | | Dihydropteroate Synthetase (Staphylococcus aureus) | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1f6y | 1.248 | | Corrinoid/Iron-Sulfur Protein Methyltransferase *(Moorella thermoacetica)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1dio:A | 0.897 | | Diol Dehydratase *(Klebsiella oxytoca)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1r30:A | 0.793 | reliable> 2.17<br>unsure > 1.03<br>unreliable < 1.03 | Biotin Synthase *(Escherichia coli)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| **Pcons5** | 1dio:A | 1.5052 | | Diol Dehydratase *(Klebsiella oxytoca)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1eex:A | 1.1273 | | Diol Dehydratase *(Klebsiella pneumoniae)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1iwp:A | 1.0572 | | Glycerol Dehydratase *(Klebsiella pneumoniae)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1f6y:A | 0.9377 | | Corrinoid/Iron-Sulfur Protein Methyltransferase *(Moorella thermoacetica)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1aj2 | 0.8763 | | Dihydropteroate Synthase *(Escherichia coli)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |
| | 1r30:A | 0.8543 | | Biotin Synthase *(Escherichia coli)* | Alpha and beta proteins (a/b) | TIM beta/alpha-barrel |

**Table 2.1. Individual server results from fold recognition servers under GeneSilico Metaserver.**

with a score 3.96 (FUGUE 2.0 criteria: reliable >21.03; unsure, >3.69; unreliable, <3.69), and the FFAS03 server found 1mmf:A (glycerol dehydratase) and 1dio:A as its first two hits, with scores of -9.03 and -8.84 (FFAS03 criteria: reliable, <-38.2; unsure, <-9.2; unreliable, >-9.2).

The fold recognition server results indicate that the EAL large subunit has the $\beta_8\alpha_8$, TIM-barrel fold, and that a close three-dimensional structural similarity exists among the large subunits of EAL, diol dehydratase and glycerol dehydratase. This structural similarity is consistent with the common catalytic mechanism among the EAL, diol dehydratase and glycerol dehydratase enzyme families,[15] which belong to the cobalamin (vitamin $B_{12}$)-dependent enzyme superfamily.

The Pcons2 and Pcons5 servers also identified a weak structural relationship between EutB and biotin synthase (BioB; PDB: 1r30). Biotin synthase is an AdoMet-dependent (radical SAM) enzyme with the full barrel, $(\alpha\beta)_8$ structure [78]. The server results are consistent with the prevalence of the α/β barrel structure in enzymes that use the 5'-deoxyadenosyl radical to initiate radical catalysis [48, 78, 79].

***Multiple Sequence Alignment***

Multiple sequence alignments were generated by using the ClustalW program [80] for eleven sequences of the large, active site-containing subunit from the following three enzyme families: (a) EAL (Pfam database [81] accession number, PF06751; EMBL accession numbers: Q9KWC6, Q59781, Q8XUQ9, Q889M2, Q8PK12, Q89QX8, Q97FL8, Q8Y7U5, 18XLZ2, Q892C9, Q8RH36, and P19264), (b) diol dehydratase

(EMBL accession numbers: Q9ZFF1 and Q59470), and (c) glycerol dehydratase (EMBL accession numbers: P37450, P45514, Q8Z5M2 and Q59476). The alignments of EutB with the large subunits from diol dehydratase and glycerol dehydratase were merged manually, according to the optimal multiple alignment indicated by threading for EutB (P19264) with the large subunits of diol dehydratase (Q59470) and glycerol dehydratase (Q59476).

Figure 2.2 shows the multiple sequence alignment for EutB and the large subunits of diol dehydratase and glycerol dehydratase from different species, obtained by using the ClustalW [82] program. The multiple sequence alignment is based on the secondary structure prediction and threading results for EutB. The multiple sequence alignment includes 354 residues (residues 79-432) of the 453 residue EutB sequence. This is because the threading results indicate the absence of reliable alignments for EutB residues 1-78 and 433-453. We attribute the deviation in structural relationships outside of the 354 residue EutB sequence, at least in part, to the different number of subunits and subunit-subunit interactions among the enzymes [15]. EAL has two subunits, α (EutB) and β (EutC) [54] whereas diol dehydratase and glycerol dehydratase each have three subunits, denoted α, β and γ [15]. In addition, both diol dehydratase and glycerol dehydratase crystallize as a dimer [16, 18, 19], which is suggestive of a $(\alpha\beta\gamma)_2$ functional oligomer. As described above, the functional oligomer for EAL in solution appears to be a hexamer of αβ pairs [83, 84]. Deviations in the structure outside of the core $\beta_6\alpha_6$- or $\beta_8\alpha_8$-fold have also been noted across families that belong to the radical-SAM enzyme superfamily [48].

**Figure 2.2. Multiple sequence alignment of EAL EutB (Pfam database accession number: PF06751) and diol dehydratase and glycerol dehydratase**. Each sequence is identified by the EMBL accession number (left column). Identical residues in the multiple sequence alignment are marked in yellow and conserved residues are marked in grey. The consensus secondary structure prediction (E: β-strand; H: α-helix) for *S. typhimurium* EutB P19264 is shown below its sequence, along with the prediction confidence score (7: highest; 0: lowest) for each residue. The ruler that is presented on top of the multiple sequence alignment indicates the amino acid numbering for *S. typhimurium* EutB P19264. The regions of the β-strands corresponding to the $\beta_8\alpha_8$-barrel are indicated in the Notes row. Secondary structures of diol dehydratase (PDB: 1eex:A) and glycerol dehydratase (PDB: 1mmf:A) are also shown.

The vertical yellow bands in Figure 2.2 highlight the residues that are identical among the different Class II coenzyme $B_{12}$-dependent enzymes. The sequence identity between the structure-based alignment for EutB and the large subunits of diol dehydratase and glycerol dehydratase is 7%. The low value indicates strong evolutionary divergence among the three enzyme families, if they arose from a common ancestral gene [47]. Blocks of sequence-identical residues occur predominantly within the eight β-barrel strands and in the loop regions that follow the C-termini of the β-strands in the barrel. These are the positions that surround and form the active site and the binding region of the coenzyme $B_{12}$ cofactor. There are six conserved glycine residues, which are recognized to facilitate the turns that link the β- and α- secondary structure elements in $\beta_6\alpha_6$, TIM-barrrel structures [85, 86].

***Template X-ray Crystallographic Structures***

The EMBL accession numbers for the diol dehydratase large subunit are Q9ZFF1 (*K. pneumoniae*) and Q59470 (*K. oxytoca*), and for the glycerol dehydratase large subunit are P37450 (*S. typhimurium*), P45514 (*Citrobacter freundii*), Q8Z5M2 (*S. typhi*) and Q59476 (*K. pneumoniae*).

The Protein Data Bank contains the following seven structures for diol dehydratase from *K. oxytoca*, which differ in the type of bound cofactor and the absence or presence of substrates: PDB accession codes, 1dio, 1eex, 1egm, 1egv, 1iwb, 1uc4 and 1uc5. The Protein Data Bank contains the following two structures for glycerol dehydratase, both from *K. pneumoniae*: PDB accession codes, 1iwp and 1mmf.

***Model Generation***

The multiple sequence alignment was used as input to MODELLER [58] 7v7 and 8v1 for generation of the EutB structure model. MODELLER 8v1 was used to obtain the final EutB model. MODELLER builds three-dimensional protein models by using the method of satisfaction of spatial restraints, and is based on the optimization of an effective energy function over empirical pairwise $C_\alpha$ and $C_\beta$ probability distribution functions and backbone $\psi$, $\varphi$ angle, bond length and bond angular preferences [58, 87, 88], MODELLER was used to generate a structural model for EutB, which was based on the threading results obtained with the templates diol dehydratase 1eex:A (as indicated by the Sam-T02 server) and glycerol dehydratase 1mmf:A (as indicated by the FFAS03 server). The best models (criteria described below) were selected for further modeling of loop regions. Loop regions were optimized by MODELLER, which uses conjugate gradients and molecular dynamics with simulated annealing with a pseudo-energy function, including the CHARMM-22 molecular mechanics force field [89].

The models were evaluated by using the programs PROCHECK [59], VERIFY3D [60, 90], PROSA2003 [61], and ERRAT 2.0 [62], and by visual inspection, by using the computer graphics programs, PyMOL (Delano Scientific LLC, San Francisco, CA) and SYBYL (Tripose Inc., St. Louis, MO). The EutB modelling proceeded in three stages, as follows: (1) In the initial stage of modelling, models with alternative alignments were made, based on the multiple sequence alignment, and re-evaluated in an effort to flatten regions of very high energy and to decrease the overall effective energy. Iteration of this procedure led to a terminal stage 1 model for EutB that represented the best model prior

to computer graphics-guided manual intervention. (2) In the intermediate stage of model refinement, visual inspection of computer graphics representations of the templates and target identified gaps in the sequences in the β-strands and α-helices of the $\beta_8\alpha_8$ structure of EutB. The gaps were moved to the loop regions that linked the β-strands and α-helices. The multiple sequence alignment was modified according to these results. Iteration of this procedure led to a terminal stage 2 model for EutB that represented the best match to the templates 1eex:A and 1mmf:A in the absence of the inclusion of substrate or cofactor. (3) The final stage of model refinement was achieved by inclusion of the propanediol substrate and adeninylpentyl-cobalamin cofactor that are present in the diol dehydratase 1eex:A structure [17]. The propanediol and cofactor atoms were modelled as hard spheres and the molecular frameworks were held rigid. The substrate and cofactor therefore supplied only steric exclusion constraints. No polar or hydrogen bond interactions between substrate and cofactor and the protein were incorporated. We consider the terminal stage 3 model as the best representation of the structure of EutB.

Figure 2.3A displays the overall match of the α-helix and β-strand secondary structure elements of the EutB model with 1eex:A and 1mmf:A in the core $\beta_8\alpha_8$ structure. In general, the loop regions are also well matched, although MODELLER is typically less successful in building extended loop regions [58]. An example is the loop region, E333 to R343, which is visible on the upper right of Figure 2.3A. The conformation of this loop suggests that the abutting α-helices actually extend two to three residues into the loop. Figure 2.3B shows the individual β-strands and α-helices that compose the $\beta_6\alpha_6$, TIM-barrel fold of EutB.

The EutB model was evaluated by using different programs. Model stereochemistry was evaluated by using PROCHECK [59]. The model incorporating substrate and adeninylpentyl-cobalamin cofactor showed 98.4% of residues in allowed regions of the $\psi$, $\varphi$ or Ramachandran, plot, of which 85.4% were in favored regions. The five residues that are in disallowed regions (V195, T230, V326, T385, and Q405) are present in loop segments outside of the β- and α- secondary structure, and therefore do not have a significant impact on the structure of the model. We consider the PROCHECK results to be very good at the resolution of the current EutB model. The PROCHECK output for the EutB model is presented in Figure 2.4.

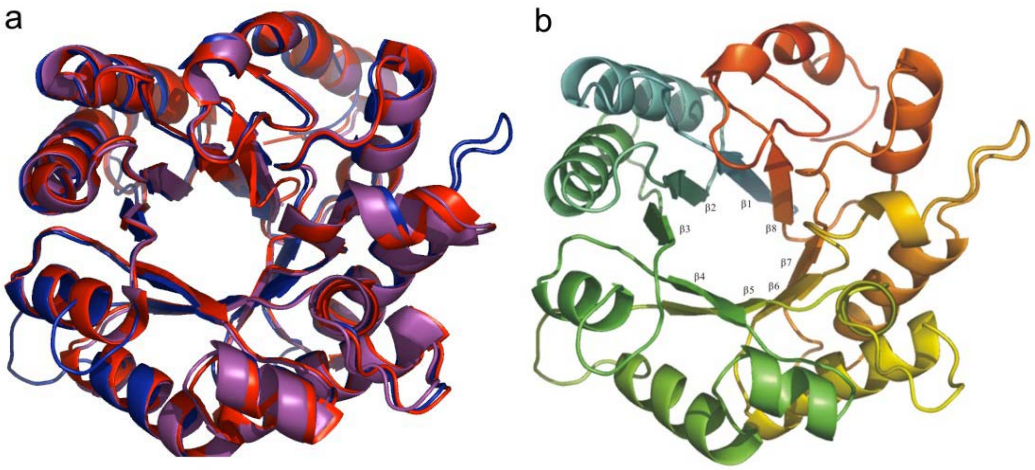


**Figure 2.3 Model for the structure of EutB from *S. typhimurium* EAL and overlap with large subunits from diol dehydratase (1eex:A) and glycerol dehydratase (1mmf:A).** The view is down the $\beta_8\alpha_8$-barrel from the C-terminal end. (a) Overlay of the secondary structure representations of the $\beta_8\alpha_8$ region of EutB (blue), 1eex:A (diol dehydratase; red) and 1mmf:A (glycerol dehydratase; magenta). The sequences are truncated to clearly present the $\beta_8\alpha_8$ region. The displayed residues are as follows: EutB, T156-S423; 1eex:A, S139-G399; 1mmf:A, P139-G339. (b) Secondary structure representation of EutB model, T156-S423, showing β-strands 1-8.

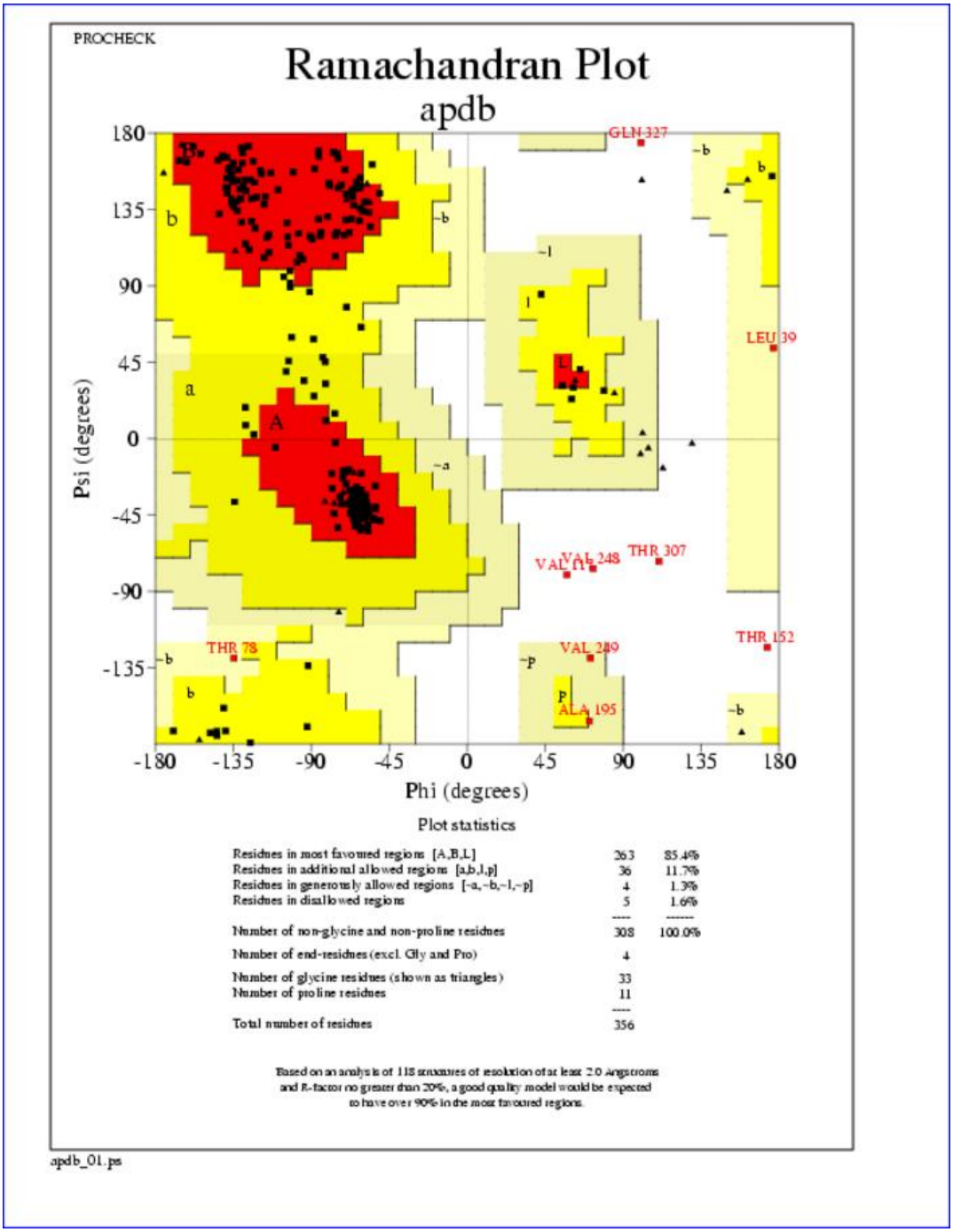


**Figure 2.4. Results from PROCHECK analysis of the EutB model.**

VERIFY3D is a widely used program for the assessment of the reliability of X-ray crystallographic, nuclear magnetic resonance (NMR) and calculated protein structures [60, 90]. VERIFY3D compares the environment of each residue in the model with the environment expected on the basis of high-resolution X-ray crystallographic structures, and a calculated potential of mean force is used to quantitate the results [60, 90]. The EutB model had an average VERIFY3D score of 0.27 and a total score of 95.17 for the 354 residues included in the multiple sequence alignment (Figure 2.2). The average and total scores increase to 0.29 and 103.0 with the omission of the last ten residues, which score poorly. The average and total VERIFY3D scores indicate that the EutB model is acceptable and of reasonable quality [60, 90].

### §2.2.2 Comparison with X-ray structure and model improvement

The X-ray crystallographic structure of EutB from *Listeria monocytogenes* was deposited in the Protein Data Bank [55] two years after our original comparative model was published. Figure 2.5 shows a comparison of the original comparative model with the X-ray crystallographic structure. The original comparative model shows considerable overlap with the X-ray crystallographic structure, and is correct about the structural predictions, which are presented in §2.3.

Although the X-ray crystallographic structure provides atomic resolution (2.15 Å), the coenzyme and substrate are absent in this structure, which still limits the understanding of enzyme mechanism of EAL. By utilizing the structural relation between EutB protein and diol dehydratase α subunit from the original study [91], an

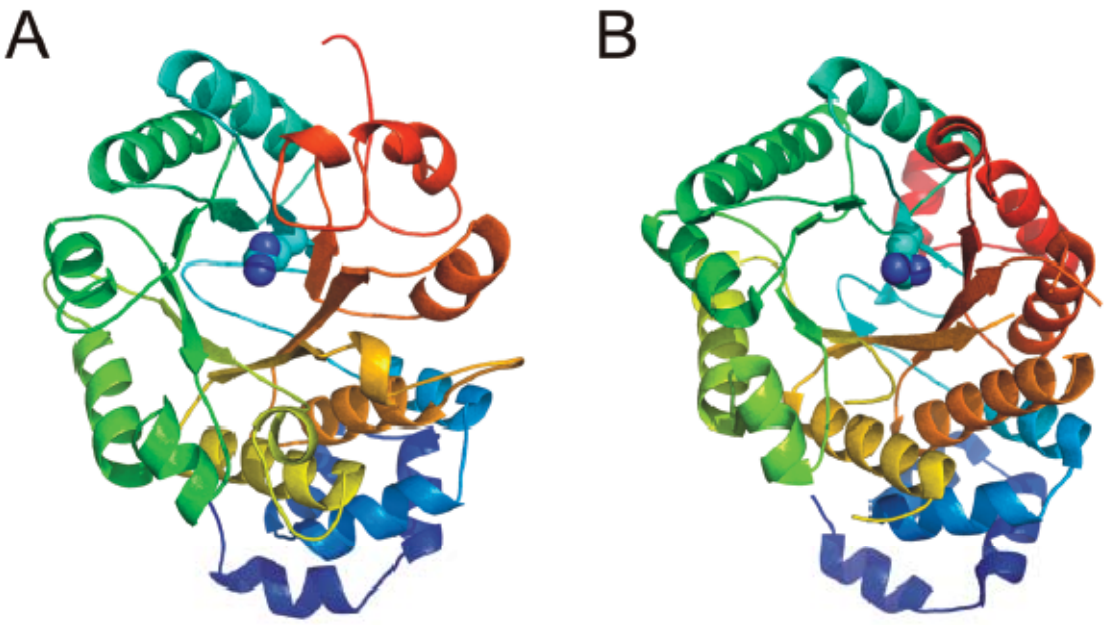


**Figure 2.5 Comparison of the original comparative model and X-ray crystallographic structures of the EutB protein of EAL.** (A) Original comparative model [91] of the EutB protein from *S. typhimurium* showing the β8α8 region (residues 79-432). The R160 side chain is depicted in van der Waals surface mode. (B) X-ray crystallographic structure of the EutB protein (chain A) from *L. monocytogenes*, PDB accession code 2qez, showing the β8α8 region (residues 79-432).

improved EutB comparative model with atomic accuracy and inclusion of cofactor and substrate is built, with the following procedure : (1) diol dehydratase α subunit structure containing cofactor and substrate (1eex:A) is aligned with EutB crystallographic structure (2qez:A) by pairwise 3D DALI algorithm [92]; (2) EutB sequence from *S. typhimurium* (P19264) is aligned with EutB crystallographic structure sequence from *Listeria monocytogenes* (Q8Y7U5) by pairwise sequence alignment algorithm; (3) the two alignments are combined and used as the multiple sequence alignment (Figure 2.6) for constructing the improved EutB model of sequence from *S. typhimurium*. Comparison of

the originally published model with the improved EutB model shows the remarkable predictive power of comparative modeling. The original comparative model was correct about the overall structure of the protein and assignment and positioning of secondary structure elements. Table 2.2 shows a detailed comparison of each secondary structure element in the originally published model and the improved EutB model. The only major defects in the original model are that the $\beta_3$ and $\beta_7$ strands are shifted by 2 and 4 residues respectively, though the original recognition of active-site residues remains largely correct (details in §2.3.2). This research confirms comparative modeling as a viable and important tool for studying proteins with unknown structures.

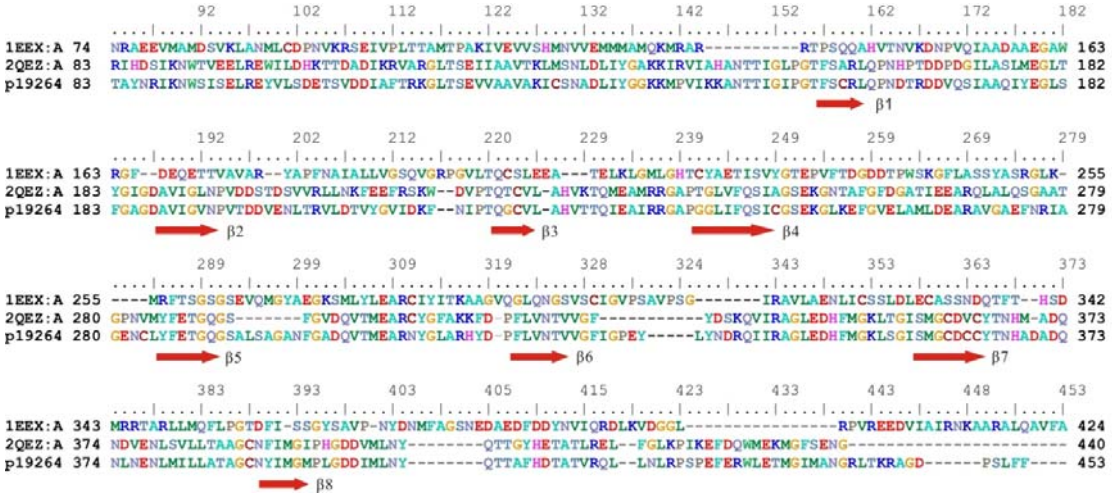


**Figure 2.6. Improved multiple sequence alignment for EAL of *S. typhimurium* EutB (P19264) with *L. monocytogenes* EutB(Q8Y7U5) and diol dehydratase (1eex:A)sequence.** The ruler that is presented on top of the multiple sequence alignment indicates the amino acid numbering for *S. typhimurium* EutB P19264. The regions of the β-strands corresponding to the $\beta_8\alpha_8$-barrel are indicated.

**Table 2.2 Comparison of secondary structure elements in the originally published model and the improved EutB model.**

| Structure Elements | Original EutB model | Improved model | Remarks |
|---|---|---|---|
| M1-D78 | Not available in original model | | |
| $\alpha_{N6}$ | E86-K89 | E86-K89 | Match |
| $\alpha_{N7}$ | I93-L101 | I93-L101 | Match |
| $\alpha_{N8}$ | L119-L127 | L119-I128 | Match |
| $\alpha_{N9}$ | N131-K140 | N131-K140 | Match |
| Loop "Cap" | M141-T156 | M141-G155 | Match |
| $\beta_1$ | F157-Q162 | F157-Q162 | Match |
| $\alpha_1$ | V170-F183 | V170-F183 | Match |
| $\beta_2$ | D187-N193 | D187-N193 | Match |
| $\alpha_2$ | V199-G211 | V199-F216 | Position matches, V212-F216 incorrectly modeled as loop |
| $\beta_3$ | T220-V224 | T220-V224 | Shifted by 2 residues |
| $\alpha_3$ | V228-G238 | V228-G238 | Match |
| $\beta_4$ | G241-C249 | G241-C249 | Match |
| $\alpha_4$ | E252-E274 | E252-E274 | Match |
| $\beta_5$ | Y285-G291 | Y285-G291 | Match |
| $\alpha_5$ | Q304-Y318 | Q304-Y318 | Match |
| $\beta_6$ | F321-V326 | F321-V326 | Match |
| $\alpha_6$ | R343-S355 | D338-S355 | Position matches, D338-I342 incorrectly modeled as loop |
| $\beta_7$ | C362-H368 | S358-C364 | Shifted by 4 residues |
| $\alpha_7$ | Q373-L383 | Q373-A386 | Position matches, A384-A386 incorrectly modeled as loop |
| $\beta_8$ | N389-M394 | N389-M394 | Match |
| $\alpha_8$ | D399-L419 | D399-L419 | Match |
| $\alpha_{C1}$ | P425-G435 as loop | P425-G435 | Incorrectly modeled as loop |

## §2.3 Predicted EutB structural features

The comparative model has shown that EutB contains the $\beta_8\alpha_8$, TIM-barrel fold, in common with other active site-containing large subunits of coenzyme $B_{12}$-dependent enzymes of known structure. The EutB model, and a multiple sequence alignment among the Class II enzymes, EAL, diol dehydratase and glycerol dehydratase from different species, reveals the following structural features: (a) a "cap" loop segment that closes the N-terminal region of the β-barrel, (b) a set of amino acid side chains in the active site region that are proposed to hydrogen bond with the substrate-derived radical species, (c) a β-barrel-internal guanidinium group from R160 that overlaps the position of the active site potassium ion found in the diol and glycerol dehydratases, and (d) a shared cobalamin cofactor binding site topography at the C-terminal region of the β-barrel.

### §2.3.1 N-Terminal Region of the β-Barrel: Cap Structure

The multiple sequence alignment in Figure 2.2 shows 12 residues prior to the β1 strand, IKKANTTIGIPG, in the EutB sequence (I144 to G155), which are not aligned with residues in diol dehydratase 1eex:A or in glycerol dehydratase 1mmf:A. As shown in Figure 2.7a, the orininal published EutB model [91] predited the 12 residue sequence to form a loop, or "cap", that covers the N-terminal regions of the β-strands that form the barrel. A similar cap-like structure is observed in 1eex:A of diol dehydratase (Figure 2.7b) and in 1mmf:A of glycerol dehydratase (not shown). However, the sequence order of the loop is significantly different in diol dehydratase, arising from residues beyond the β8 strand (residues I505 to L517 in 1eex:A Q9ZFF). The 1eex:A sequence folds back to the N-terminal regions of the β-barrel to form the cap. The common loop cap structure

appears to contribute to protection of the radical reactants in the interior of the β-barrel from redox-active species in the exterior medium. The prediction about the presence, the sequence and position of this cap structure is verified to be startlingly correct as the 2QEZ X-ray crystallographic structure Figure 2.7c becomes available.

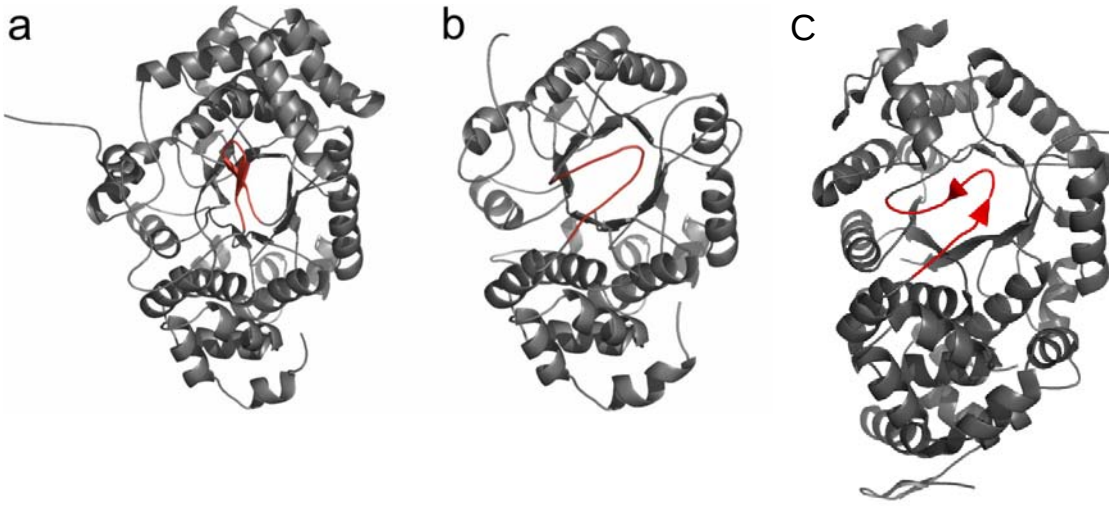


**Figure 2.7. Secondary structure representation of the loop "cap" structures that cover the N-terminal end of the $\beta_8\alpha_8$-barrel 1eex:A from diol dehydratase, in EutB model, and in 2QEZ EAL X-ray structure from *L. monocytogenes*.** The view is down the $\beta_8\alpha_8$-barrel from the N-terminal end. The loop cap regions are highlighted in red. (a) 1eex:A, residues I505-L517. (b) predicted EutB model [91], residues I144-G155. (c) 2QEZ:A, residues V143-G155.

## §2.3.2 Features of the Active Site

Figure 2.8 displays the arrangement of amino acid side chains that compose the substrate binding site in the improved EutB model. The view also includes the propanediol substrate and the potassium ion ($K^+$) from the 1eex:A structure in the positions specified by the template-target match. In 1eex:A, the potassium ion is coordinated by Q141, E170, E221, S361 and the two oxygen atoms of the propanediol substrate [16, 17]. As shown in Figure 2.8, the side chains of R160, Q162, F245, Y285, E287, N324, D362, and M394 fill the space in the EutB model that is occupied by the

potassium ion and its ligation field in 1eex:A. There is no evidence for an arrangement of potassium ion binding residues in the EutB model. This is consistent with the absence of a monovalent cation requirement for *S. typhimurium* EAL activity, and with the absence of hyperfine coupling detectable with ESEEM spectroscopy between the substrate radical and alkali cations, addressed by using buffers containing $Li^+$, $Na^+$, $K^+$, or $Rb^+$ (J. M. Canfield, K. Warncke, unpublished results). Noteworthy, to active site side chain arrangement in Figure 2.8 is modeled with the 2QEZ X-ray crystallographic structure; therefore the positioning of the side chain groups is accurate to an atomic scale. Nevertheless, the predicted active site in the original model (Figure 6 in [91]) shows great accuracy in side chain positioning, especially for the residues of R160, F245, Q162, and E287; although H368 and M392 were placed incorrectly close to the substrate bind region.

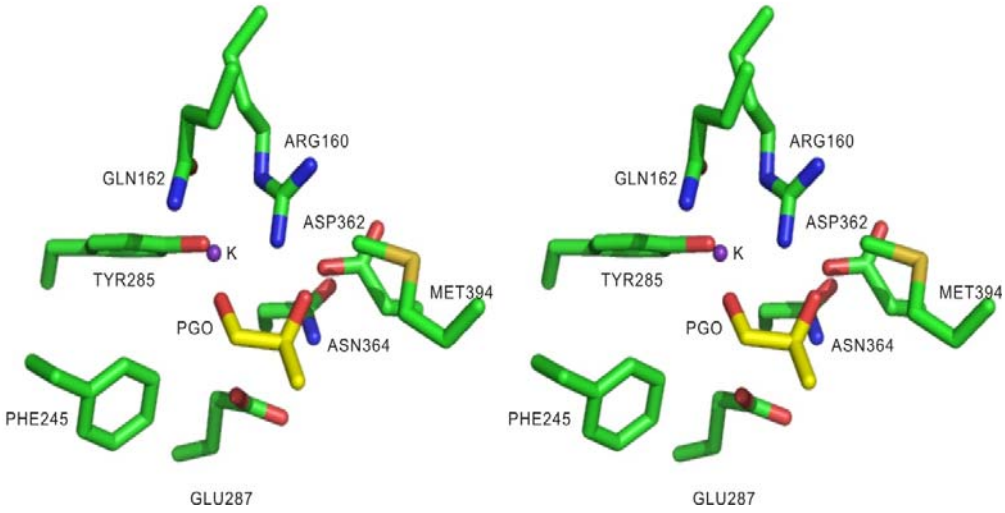


**Figure 2.8. Stick representation of side chains in the substrate binding pocket in the active site in the EAL EutB model in stereo view.** The substrate propanediol and potassium ion (K, violet sphere) are positioned in accord with the template-target match of the EutB model to the diol dehydratase 1eex:A structure.

Figure 2.8 shows that the E287 carboxylate oxygens are 4.0 and 5.9 Å from O1 of the substrate. In the active site of diol dehydratase, the sidechain of E170 is hydrogen bonded to the O2 and O1 hydroxyl groups of the substrate [16, 17]. The $k_{cat}$ value of the E170A mutant is $<2.7 \times 10^{-5}$ that of the native enzyme ($k_{cat}/K_M$ not reported for E170A) [15]. A computational study, which treated the substrate and hydrogen bonding side chains by using density functional theory, and portions of the surrounding 1eex:A structure by using molecular mechanics, suggested that the interaction of O1-H with the deprotonated carboxyl group of E170 provides a crucial hydrogen bond that lowers the barrier for the rearrangement reaction [93]. Computational studies of Radom and coworkers had predicted a requirement for this interaction, which delocalizes charge in the transition state [94]. In addition, a requirement for hydrogen bonding to, and partial protonation of, the migrating hydroxyl group (O2-H) to provide additional, synergistic charge delocalization during migration, was predicted [95]. Computational studies performed on models for the EAL rearrangement reaction [96-99], including one with imidazolyl and carboxyl moieties to mimic possible in situ interactions [98], indicated that this interaction is necessary to compute a rearrangement energy barrier consistent with experimental constraints. Therefore, we propose that, in EutB of EAL, the E287 carboxyl group hydrogen bonds to the O1 hydroxyl group.

Figure 2.8 shows additional groups in the active site region. The amide nitrogen of Q162 is 3.9 Å from O1 and 5.1 Å from O2 of the substrate, which suggests possible hydrogen bond interactions. The side chain amide heteroatoms of N403 and N324 are >5.7 Å from O1 and O2; it may also be considered as candidates for hydrogen bond interactions. In diol dehydratase, the 1eex:A side chains of Q296 and D335 form

hydrogen bonds with the substrate hydroxyl groups [16, 17]. The aromatic ring of F245 in the EutB model appears to play a significant steric role, forming close contact with the adenine group from the cofactor. As considered above, the dual role of F245 in cofactor site and substrate site formation suggests involvement in the substrate binding "trigger" of cobalt-carbon bond cleavage.

The striking juxtaposition of the van der Waals spheres of the EutB R160 sidechain guanidinium moiety and the potassium ion in 1eex:A suggests a functional analogy between these two positively charged centers. Toraya and coworkers have proposed that the potassium ion plays a primarily structural role in the rearrangement reaction in diol dehydratase [15], essentially acting as a pivot for small displacements of the carbon backbone of the diol substrate during the catalytic cycle. In the Class I coenzyme $B_{12}$-dependent enzymes, methylmalonyl-CoA mutase and glutamate mutase, arginine side chains are important for correctly positioning the substrates for catalysis [43, 44, 100]. We had proposed that the guanidinium moiety of R160 in EutB hydrogen bonds to both the 2-amino group and the 1-hydroxyl group of the substrate, in a scaffolding role. In addition, hydrogen bonding of the positively charged guanidinium group to the migrating amino group could provide "partial protonation" and enhanced electrostatic contributions to catalysis of the rearrangement reaction in EAL. In Chapter 3, site-directed mutation experiments were used to test these proposals.

### §2.3.3 C-Terminal Region of the β-Barrel: Cobalamin Binding Site

The X-ray crystallographic structures of diol dehydratase [16] and glycerol dehydratase [18, 19] show that the cobalamin binds to the C-terminal regions of the β-strands that form barrel, with the corrin macrocycle plane approximately perpendicular to the long axis of the barrel. In the diol dehydratase structure, 1dio:A, determined with cyano-cobalamin as cofactor [16], 12 residues were reported to be involved in hydrogen bonding interactions with the acetamide and propionamide substituents on the corrin. The diol dehydratase structure incorporating the adeninylpentyl-cobalamin, 1eex:A, allowed 6 additional interactions with the adenine ring to be specified [17]. Table 2.3 lists these residues, and the sequence-related residues from EutB from the multiple sequence alignment in Figure 2.6 for the improved EutB model incorporating 2QEZ X-ray crystallographic structure information. Table 2.3 also includes 6 residues that contribute to the cofactor binding site in the EutB model, and the sequence related residues from 1eex:A. As shown in Table 1, of the 17 hydrogen bond interactions specified for the large subunit-cofactor interaction in diol dehydratase, 11 are from the backbone, and 4 of the 6 side chain interactions are indirect (mediated by water molecules) [16, 17]. The large number of backbone interactions is consistent with the low sequence identity (22%) among 1eex:A and EutB for the 23 total residues in Table 2.3. The contour of the polypeptide backbone therefore appears to play a dominant role in determining specific interactions between protein and cofactor. Worth mentioning, 22 of the 23 residues (except for T392) are correctly predicted in the original EutB model ([91], Table 1), prior to the knowledge of X-ray crystallographic structure.

**Table 2.3. Residues involved in the formation of the cobalamin cofactor binding site in the EALEutB model (P19264) and in diol dehydratase large subunit (1eex:A).** Residue identity (yellow highlight) and conservation (grey) are indicated in Columns 1 and 2.

| Residue 1eex:A | Residue EutB | EutB Region[a] | Hydrogen Bond[b] | HydrogenBond Interaction[c] | Cofactor Contact[d] | Assignment Reference |
|---|---|---|---|---|---|---|
| T172 | N193 | β2 | sc | direct | ring, N29 | 16, this work |
| V173 | P194 | β2 | - | - | - | this work |
| A176 | D197 | β2 | bb, NH | $H_2O$ | ring, O34 | 16, this work |
| L203 | V224 | β3 | - | - | - | this work |
| E204 | L225 | β3 | bb, NH | $H_2O$ (2) | ring, N33 | 16, this work |
| E205 | A226 | β3 | sc | direct | ring, N40 | 16, this work |
| T222 | F245 | β4 | bb, =O | direct | ring, N40 | 16 |
| S224 | S247 | β4 | sc | $H_2O$/dir | ring,O39;Ad, N3A | 16, 17, this work |
| V225 | I248 | β4 | bb, =O | $H_2O$ | Ad, N1A | 17, this work |
| Y226 | C249 | β4 | sc | $H_2O$ | ring, O39 | 16, this work |
| D234 | E257 | loop 4-5 | bb, =O | direct | ring, N45 | 16 |
| G235 | F258 | loop 4-5 | bb, =O | $H_2O$ | ring, O39 | 16 |
| D237 | V260 | loop 4-5 | sc | $H_2O$ (2) | ring, O39 | 16 |
| T259 | E287 | β5 | - | - | - | this work |
| S260 | T288 | β5 | - | - | - | this work |
| G261 | G289 | β5 | bb, =O | direct | Ad, N6A | 17, this work |
| S262 | Q290 | β5 | - | - | - | this work |
| G263 | G291 | β5 | - | - | - | this work |
| S264 | S292 | β5 | sc | $H_2O$ | Ad, N1A | 17, this work |
| Q267 | S295 | loop 4-5/ 5 | bb, =O | $H_2O$ | ring, O51 | 16 |
| S299 | V326 | β6 | bb, =O | direct | Ad, N6A | 17 |
| S301 | G328 | β6 | bb, NH | direct | Ad, N7A | 17 |
| M373 | T392 | β8 | bb, =O | direct | ring, N62 | 16, this work |

[a]Secondary structure region: β-strand, α-helix or loop number in $\beta_8\alpha_8$ region. [b]Protein hydrogen bond donor: bb, backbone amide (NH) or carbonyl (=O); sc, side chain. [c]Direct: direct hydrogen bond between protein and cofactor; $H_2O$: indirect hydrogen bond through one or two (2) water molecules. [d]Hydrogen bond contact group on cofactor, as designated in 1dio:A or 1eex:A; ring, amide substituent on corrin macrocycle; Ad, heteroatom in adenine.

A notable difference between 1eex:A and the EutB model arises from the sidechain of EutB F245 (multiple sequence alignment residue T222 in 1eex:A). The side chain forms a sterically-compatible, well-defined hydrophobic contact surface with the adenine plane of the cofactor. The F245 side chain also forms part of the substrate binding site, as described below. This suggests the intriguing possibility that F245 may be involved in the "substrate trigger", which transduces substrate binding into labialization and cleavage of the cobalt-carbon bond by an as yet unknown mechanism [15]. In the Class I coenzyme $B_{12}$-dependent enzyme, methylmalonyl-CoA mutase, substrate binding has been proposed to dislodge and force the phenol side chain of Y89 of subunit A into unfavorable steric contact with the 5'-deoxyadenosyl moiety, with a consequent proposed weakening of the cobalt-carbon bond [42].

Figure 2.9 shows views of the active site region above the cobalamin ring in the diol dehydratase 1eex:A template structure and in the EutB model. Figure 2.9A shows the relative orientation of the adeninylpentyl-cobalamin cofactor and protein surface at the C-terminal region of the barrel in 1eex:A [17]. Figure 2.9B shows the same view of 1eex:A, but with the cobalamin cofactor removed. The arrow in Figure 2.9B indicates the adenine binding niche. The diol dehydratase substrate, propanediol (propylene glycol), is visible through a narrow aperture in the dome-like protein surface above the center of the cobalamin ring. The aperture leads into a small pocket, in which the substrate resides. Our experimental studies for EALshow that, following cobalt-carbon bond cleavage, the C5' radical center of the 5'-deoxyadenosyl moiety migrates 6 ±1 Å from cobalt ($Co^{II}$) to this opening, and subsequently abstracts the hydrogen atom from the substrate [66, 67]. Figure 2.9C shows the view from the same perspective in the EutB

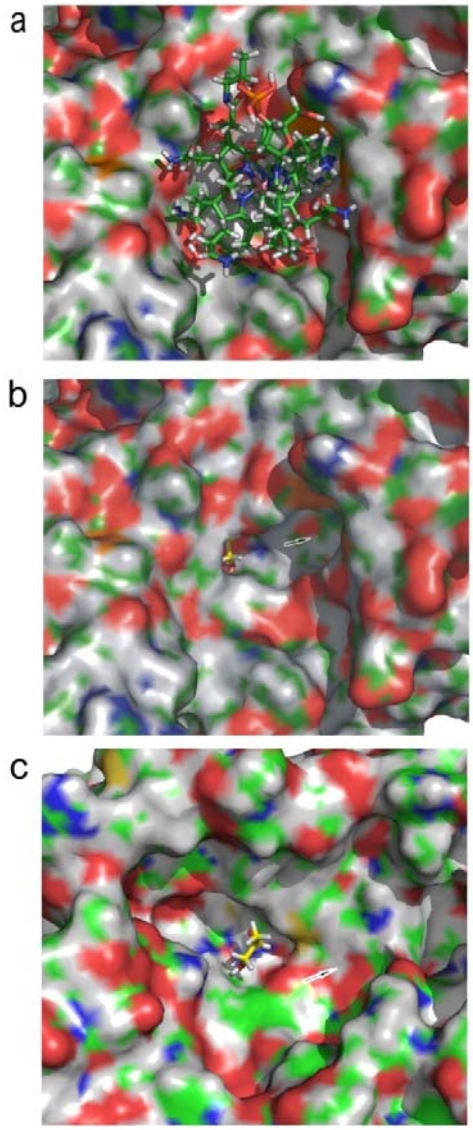


**Figure 2.9. Surface representation of the residues at the C-terminal end of the $\beta_8\alpha_8$-barrel that form the active site region in the EALEutB model and in 1eex:A from diol dehydratase.** The protein atoms are colored as follows: carbon (green), hydrogen (white), oxygen (red), nitrogen (blue), and sulfur (yellow). (**a**) 1eex:A. The adeninylpentyl-cobalamin cofactor is shown in stick representation to provide context for the view. (**b**) 1eex:A. The cofactor has been removed to show the adenine binding niche (arrow) and the aperture that leads into the substrate binding pocket. The substrate binding pocket is occupied by propanediol, shown in stick representation (carbon atoms are yellow). (**c**) EutB model. The adenine binding niche (arrow), aperture, and the substrate binding pocket are present.

model. The EutB model displays an adenine binding niche (arrow), and the aperture that leads into the substrate binding pocket. In Figure 2.9C, the propanediol substrate from the diol dehydratase structure has been added in the position corresponding to the template-target match of the EutB model and 1eex:A structure (Figure 2.3). The 2-aminopropanol substrate of EAL[101] and propanediol are isosteric. In the EutB model, the substrate is also centered in a cavity region (the substrate binding pocket). The distances from cobalt to the propanediol C1 and C2 atoms in 1eex:A are 8.6 Å and 9.2 Å, respectively [16, 17]. Our experimental distance values for the cobalt-C1 and cobalt-C2 distances in EALare 10.9-11.5 Å and 8.5-10.5 Å, respectively [67, 68]. The relative location and extent of the substrate binding pocket in the EutB model is therefore compatible with the experimentally determined distances. As shown in Figure 2.9C, the extent of the substrate binding pocket in EutB is slightly larger than in 1eex:A, which would allow positioning of the substrate at the approximately 1 Å farther distance from cobalt that is observed in ethanolamine ammonia-lyase. Comparison of Figure 2.9B and Figure 2.9C shows that the key elements of the active site region that are observed in the template 1eex:A structure are also present in the EutB model.

# Chapter 3

# Comprehensive spectroscopic, and Biochemical studies on the EAL mutants

## §3.1 Background and introduction

The major conclusion made from the comparative model of EutB protein in the Chapter 2 is the hypothesis that the R160 residue organizes the protein around the substrate binding site, and that it has the same "scaffolding" function for the substrate during the rearrangement reaction that was proposed [2, 16, 17] for $K^+$ in diol dehydratase. . The substrate and potassium ion ($K^+$) are from the diol dehydratase 1eex:A structure [17], which was used as the template for the EutB model. The substrate and $K^+$ are positioned according to the template (1eex:A)-target (EutB) match, and the substrate position is consistent with the dimensions of the substrate binding cavity in the EutB model [5]. Additionally, there is no evidence for protein ligands that form a $K^+$ binding site in the EutB model. Instead, the guanidinium group of R160 overlaps the van der Waals sphere of the $K^+$. Therefore, protonated R160 from EutB may assume the functionality of the $K^+$ in diol dehydratase. Arginine side chains have been proposed to position substrates for catalysis in the coenzyme $B_{12}$-dependent mutase enzymes [102-104]. In addition, the guanidinium group of R160 is positioned near the migrating group, as shown in Figure 2.8. *Ab initio* calculations on simple model systems show that the barrier to amine migration is lowered if the migrating group is "partially protonated" during rearrangement [96, 97]. Therefore, R160 in EutB was also proposed to accelerate the rearrangement reaction by both general acid and electrostatic catalysis [5].

In this chapter, we test the predicted roles for R160, and thus the comparative model of EutB, by using techniques of site directed mutagenesis, enzyme kinetics, and a comprehensive EPR spectroscopic protocol for assessing the impact of the mutations on reactant and protein structure. Specifically, the protein chemical, enzyme kinetic and EPR

spectroscopic properties of EAL with mutations at position 160 of EutB (R160K, R160A, R160E and R160I) are characterized and compared to the properties of wild type (WT) EAL. The R160I and R160E mutants failed to assemble into an EAL oligomer that could be isolated by the standard enzyme purification procedure. The R160K and R160A mutants assemble, but R160A is catalytically inactive and reacts with substrates to form uncoupled $Co^{II}$ and unidentified protein-based radical species. R160K displays catalytic turnover of aminoethanol, and the formation of $Co^{II}$-radical pair intermediate states. EPR and ESEEM spectroscopies and simulations reveal that the native $Co^{II}$-radical pair separation distances are increased by 2.1 ±1.0 Å in R160K, but that the geometry of the C1, C2, and C5'-methyl group reactant centers is conserved. The consistent interpretation of the results that is achieved by using the EutB model provides strong support for the model in the region of the active site.

## §3.2 Construction of a site-directed mutagenesis system and biochemical assessments of the EAL mutants

***Site-directed mutagenesis.***

The 8.5 kb plasmid containing the *S. typhimurium* EALcoding sequence was extracted from the *Eschericia coli* overexpression strain [4] with Qiagen QIAprep© Spin Miniprep kit (Qiagen Co., Valencia CA). Site-directed mutagenesis was performed on the arginine 160 position within the *eutb* sequence by using the GeneTailor™ Site-Directed Mutagenesis Kit (Invitrogen Co., Carlsbad, CA). Five plasmids, including four mutations (R160A, R160K, R160E, and R160I) and a WT control, were constructed. The original plasmid was first methylated with DNA methylase, and amplified individually with five pairs of overlapping primers, which either contained the WT Arg160 or the four desired mutations. The list of primers used is presented in Table 3.1. Platinum® Taq polymerase (Invitrogen Co., Carlsbad, CA) was used for the PCR reactions, and followed by transformation into the competent *E.coli* DH5α™ $T1^R$ strain that circularizes the linear mutated DNA and digests the methylated templates. Positive transformations were selected by ampicillin resistance on LB plates. The mutations on the new plasmids were confirmed by nucleotide sequencing (performed by Cogenics Co., Morrisville, NC).

**Table 3.1. Sequences of primers used for site-directed mutagenesis.** The forward primers contain the mutations at position 21-23. The same backward primer is used in all mutations and in the wild type control.

| Mutation Primers | Nucleotide sequence (5' to 3') |
|---|---|
| R160A Forward | ttc cgg gca cct tta gct gcg ctt tgc agc cga a |
| R160K Forward | ttc cgg gca cct tta gct gca aat tgc agc cga a |
| R160I Forward | ttc cgg gca cct tta gct gca ttt tgc agc cga a |
| R160E Forward | ttc cgg gca cct tta gct gcg aat tgc agc cga a |
| WT Control Forward | ttc cgg gca cct tta gct gcc gtt tgc agc cga a |
| Backward Primer | gca gct aaa ggt gcc cgg aat acc gat ggt |

***Enzyme preparation.***

DH5α™ T1$^{R}$ *E. coli* srtains containing expression plasmids for WT and the four mutated *eutb* were grown in a 10 L volume in a BioFlo® 110 fermentor (New Brunswick Scientific Co., Inc., Edison, NJ) and harvested following standard protocols [6]. Enzymes were purified as previously described [6], except that the enzymes were dialyzed against a final buffer containing 100 mM HEPES (pH 7.50), 10 mM potassium chloride, 5 mM dithiothreitol, and 10% glycerol [105]. Polyacrylamide gel electrophoresis (PAGE) was performed under both non-denaturing conditions [106] and denaturing conditions [107], as previously described. Protein was stained by using Coomassie blue dye.

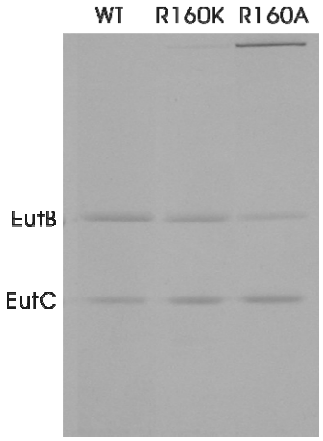


**Figure 3.1. Sodium dodecylsulfate polyacrylamide gel electrophoresis gel of purified wild type, R160K and R160A ethanolamine ammonia-lyase.** The 10% polyacrylamide gels were stained with Coomassie blue. The bands for EutB and EutC are indicated.

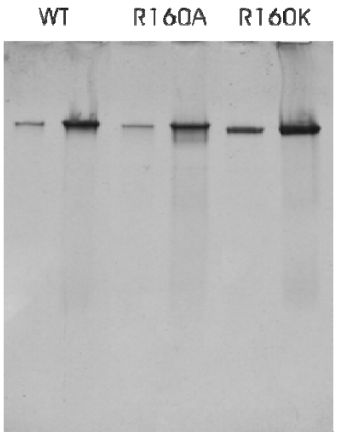


**Figure 3.2. Non-denaturing polyacrylamide gel electrophoresis gel of purified wild type, R160A, and R160K ethanolamine ammonia-lyase.** The 8% polyacrylamide gels were stained with Coomassie blue. Lanes for the wild type (WT), R160A and R160K EAL are indicated. The left and right lanes for each protein molecule correspond to 10 μg and 20 μg protein, respectively.

The impact of the mutations on EutB folding and EAL oligomer assembly can be qualitatively rationalized, as follows. The EutB model predicts that R160 is buried in the protein at the active site. The polar microenvironment around R160 indicates that the side chain will be in the guanidinium form [5]. This is supported by a study of amino acid side chain ionization in proteins, which predicts that arginine in the buried and deeply buried sites in proteins are ionized with 94% and 86% probabilities, respectively [108]. Replacing the arginine side chain with the isobutyl side chain of isoleucine in R160I EutB will therefore result in a significant Coulombic destabilization of the folded EutB state. The unionized side chain of glutamic acid in R160E EAL may produce a similar effect. It is unlikely that the negatively charged carboxylate moiety would be stable in folded EutB, and it may stabilize the unfolded state relative to the folded state. Replacing R160 with alanine, which has the small methyl sidechain, may allow water molecules to partially compensate for the loss of the arginine side chain. For a given conformation of their *n*-alkyl moieties, the *n*-butylamine side chain of lysine is shorter than the *n*-propylguanidine side chain of arginine by 1.6 Å. Therefore, the R160K mutation is reasonably conservative with respect to volume and local hydrophobicity. In addition, *in situ* protonation of the lysine amino group to create the ammonium charge center would contribute charge and polar interactions characteristic of the guanidinium group. The R160K mutation therefore maintains the structural integrity of EutB, leading to correct assembly of a stable EAL oligomer.

***Enzyme activity assay.***

EAL enzyme activity was measured by using the coupled assay with alcohol dehydrogenase and NADH [109], in which oxidation of NADH, which is proportional to the production of acetaldehyde by EAL, was monitored at $\lambda$=340 nm. For the WT EAL, a stock of 1.25 mg/ml EAL and 30.4 μM adenosylcobalamin was first incubated at 25° C for 15 minutes to assure completion of holoenzyme formation [110]. Alternatively, for mutant R160A and R160K EAL, the stock concentration was typically 7 mg/ml mutant EAL and 160 M adenosylcobalamin. A 20 μL aliquot of the incubated stock was then diluted to a total volume of 500 μL with 50 mM potassium phosphate (pH 7.5) and added to a 500 μL solution containing 50 mM potassium phosphate (pH 7.5), 4.3 U/ml alcohol dehydrogenase, and series of concentrations of aminoethanol, corresponding to approximately 0.2 to 10 times the value of the Michaelis constant, $K_M$. The reaction velocity ($v$) was then measured immediately by using the $A_{340}$ value, with a conversion factor of 6.2 $A_{340}$/μmol product [109]. The maximum reaction velocity ($V_{max}$) and $K_M$ were calculated from fits of Hanes-Woolf plots [111].

Enzyme activity assays were carried out on WT, R160A and R160K EAL with aminoethanol as substrate. Table 3.2 summarizes the results. R160K EAL shows a 30–fold increase in $K_M$ for aminoethanol relative to WT enzyme. This suggests that the affinity of substrate for EAL is weakened in R160K relative to WT.

**Table 3.2. Steady-state enzyme kinetic parameters for wild type and mutant EALfor ethanolamine as substrate.**

| **Enzyme** | $K_m$ (μM) | $k_{cat}$ ($s^{-1}$) | $k_{cat}/K_m$ ($M^{-1} \cdot s^{-1}$) |
|---|---|---|---|
| WT | 13.6±2.9 | 52.2±3.7 | $(3.8\pm1.1)\times10^{6}$ |
| R160K | 410±50 | 8.5±0.6 | $(2.1\pm0.2)\times10^{4}$ |
| R160A | - | $<10^{-5}$ | - |

Table 3.2 also shows that $k_{cat}$ for $^{1}$H-aminoethanol is decreased by 6.1-fold for R160K relative to WT. R160A displays no detectable catalytic turnover with aminoethanol ($k_{cat} < 10^{-5}$ $s^{-1}$). The value of the specificity constant [112], $k_{cat}/K_M$, is therefore decreased by 180-fold in R160K EAL relative to WT. The results show that replacement of the native positively-charged guanidinium side chain with the 1.6 Å-shorter, positively-charged ammonium side chain of R160K leads to a modest decrease in catalytic performance. However, removal of the molecular volume and positive charge of the arginine sidechain eliminates catalysis in EAL.

***R160A enzyme activity assay in the presence of guanidinium-HCl.***

The specific activities of R160A and WT for the substrate aminoethanol were determined in the presence of 0.1, 1.0, 10.0 and 100.0 mM guanidinium-hydrochloride. The enzyme activity assays were performed following the same procedure described in the enzyme activity assay section, except that the substrate concentration was 10 mM and the final concentrations of R160A and WT protein in the assay were 0.143 mg/ml and 0.025 mg/ml, respectively. A guanidinium-HCl solution was adjusted to pH 7.5, and added to the EAL and the adenosylcobalamin mixture for a 10 minute pre-incubation, prior to addition of alcohol dehydrogenase and the substrate, aminoethanol.

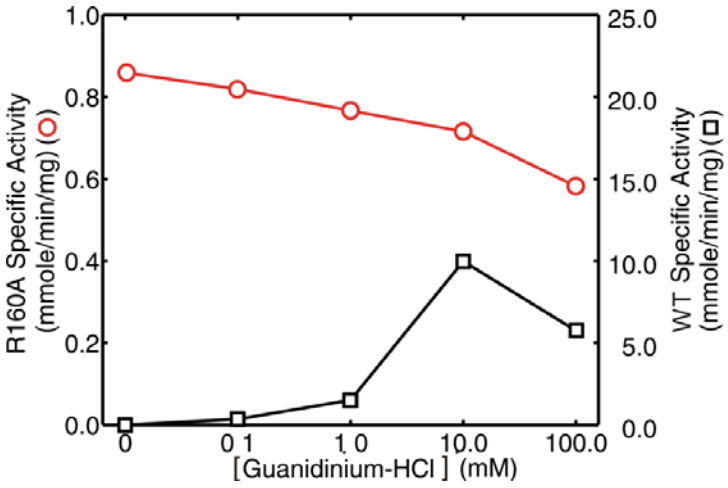


**Figure 3.3. Specific activity of wild type and R160A EAL as a function of guanidinium-hydrochloride concentration.** The specific activities of wild type and R160A EAL are plotted on different scales, as indicated on the left and right ordinate axes, respectively.

Figure 3.3 shows that externally added guanidinium–hydrochloride (HCl) leads to significant R160A EAL enzyme activity under $V_{max}$ conditions. As shown in Figure 3.3, the activity of R160A EAL increases with guanidinium-HCl concentration and reaches a maximum at 10 mM, which represents 2.3% of the activity of the WT EAL control. The structure of the guanidinium ion resembles the guanidinium group at the terminus of the arginine side chain. The results in Figure 3.3 indicate that the externally added guanidinium binds in the vacant region of the arginine side chain terminus that is created by the alanine mutation in the active site of EutB. The resurrection of enzyme activity in R160A EAL by externally added guanidinium-HCl provides strong support that a positively charged arginine side chain in the active site is catalytically essential.

## §3.3 Spectroscopic assessments by EPR and ESEEM techniques

### §3.3.1 Theory and protocols

Previous EPR studies have shown that the $Co^{II}$-C1 separation is 11 ±1 Å in the (*S*)-2-aminopropanol-generated $Co^{II}$-substrate radical pair [30, 31] and 9.3 ±1 Å in the aminoethanol-generated $Co^{II}$-substrate radical pair [30, 32]. These distances span the active site region, from the cobalamin to the substrate binding site. As shown by the approximately 2 Å shift of the C1 position in the 2-aminopropan-1-ol-1-yl radical relative to the aminoethan-1-ol-1-yl radical, which is caused by the methyl substitution at C2, the $Co^{II}$-C1 separation is a sensitive indicator of structural perturbations. Sub-Ångström resolution of the distances between the C1, C2 and C5'-methyl centers in the local region of the substrate binding site has been achieved by electron spin echo envelope modulation (ESEEM) and electron-nuclear double resonance (ENDOR) spectroscopies. Multi-frequency powder [36, 37] and orientation-selection [30, 38] ESEEM studies of the hyperfine interaction between the radicals and $^{2}H$-labeled hydrogen sites on the C5'-methyl group of 5'-deoxyadenosine have led to a three-dimensional model of the C1, C2, and C5'-methyl group reactant center geometry in each of the $Co^{II}$-radical pair states. $^{13}C$ ENDOR results are consistent with this model [113]. The magnetic coupling of the substrate radicals with a protein nitrogen nucleus has also been characterized by using $^{14}N$ ESEEM [34, 35]. The EPR and ESEEM measurements are sensitive to structural perturbations in the active site, on length scales from 0.1-10 Å. Therefore, the spectroscopies are useful as probes of the effects of the site-directed mutations on the reactant center geometry and reactant-protein interactions.

***Paramagnetic sample preparation.***

Aminoethanol (natural abundance, $^{1}H_4$-aminoethanol) and (*S*)-2-aminopropanol (Aldrich Chemical Co., St. Louis, MO), 1,1,2,2-$^{2}H_4$-aminoethanol ($^{2}H_4$-aminorthanol; Cambridge Isotope Laboratories, Inc.), 1,2-$^{13}C_2$-aminoethanol, 1-$^{13}$C-aminoethanol (Cambridge Isotope Laboratories, Inc., Andover, MA), and adenosylcobalamin (Sigma Chemical Co., St. Louis, MO) were purchased from commercial sources. Reactions were performed in air-saturated 10 mM potassium phosphate (pH = 7.5) buffer. All manipulations were performed under dim red safe-lighting on ice (273 K; aminoethanol reactions), or at room temperature [295 K; (*S*)-2-aminopropanol reactions]. The concentration of wild-type and mutant EALranged from 10 to 30 mg/mL, which is equivalent to 20 – 60 μM for a holoenzyme molecular mass of 500,000 g/mol [6]. Adenosylcobalamin (180 – 540 μM) was added to each sample to satisfy a 1.5:1 cobalamin/active site ratio. The active site:holoenzyme stoichiometry is 6 [114, 115]. After the addition of adenosylcobalamin to the mixture of enzyme and substrate, the solution was injected into a 4 mm o.d. EPR tube, which was immersed immediately into liquid nitrogen-chilled isopentane (≈140 K) and transferred to liquid nitrogen (77 K). The total time from mixing to plunging into isopentane was less than 15 s. Excess substrate was added to ensure steady-state turnover at the time of cryotrapping, as previously described [37, 116].

***Continuous-Wave EPR Spectroscopy.***

X-band continuous-wave EPR spectra were collected by using a commercial Bruker E500 EPR spectrometer equipped with Bruker ER 049X X-band microwave bridge and Oxford ESR 900 liquid helium cryostat, and Oxford ITC 503S temperature controller. All continuous-wave EPR spectra were obtained at 6 K.

Simulations of the $Co^{II}$-substrate radical pair EPR spectra were carried out by using the program MENO [117], as implemented in MATLAB with the addition of a fitting optimization algorithm [30]. The Hamiltonian can be formulated as follows [118]:

$$H = (\vec{\mu}_1 + \vec{\mu}_2) \bullet \vec{B}_0 - hJ\vec{S}_1 \bullet \vec{S}_2 + \frac{\mu_0}{4\pi}\left[\frac{\vec{\mu}_1 \bullet \vec{\mu}_2}{R^3} - \frac{3(\vec{\mu}_1 \bullet \vec{R})(\vec{\mu}_2 \bullet \vec{R})}{R^5}\right] + h\left[A_\perp (S_{1x}I_{1x} + S_{1y}I_{1y}) + A_\parallel S_{1z}I_{1z}\right] \quad \text{[Eq. 3.1]}$$

where $h$ is the Plank's constant, $\mu_0$ is the permittivity of free space, the subscripts *1* and *2* denote $Co^{II}$ and the organic radical, respectively, $S_1=S_2=½$, $I_1=7/2$, $R$ is the electron magnetic dipole-dipole separation distance, $J$ is the isotropic exchange coupling constant, and the $A$ are principal components of the axial cobalt hyperfine tensor. The electron spin magnetic moments for $Co^{II}$ and the organic radical are given by the following expression:

$$\vec{\mu}_j = \beta_e \left(g_{jx}S_{jx}\hat{x} + g_{jy}S_{jy}\hat{y} + g_{jz}S_{jz}\hat{z}\right) \quad \text{[Eq. 3.2]}$$

The vector $\vec{R}$ from the radical towards $Co^{II}$ is aligned with the $g_z$ axis of $Co^{II}$, and the cobalt hyperfine and *g* principal axis systems are coincident. The following parameters were fixed in the simulations: $g_{1x}=g_{1y}=g_{\perp}=2.26$, $g_{\|}=2.0$, and $g_{2xyz}=2.0$, and $A_{\perp}=30$ MHz and $A_{\|}=309$ MHz [30, 119]. The adjustable parameters in the simulation were *R*, *J* and the anisotropic line widths for $Co^{II}$ and radical. A spherical average of simulated spectra at all possible magnetic field orientations, $\vec{B}_0$, gives the powder spectrum.

Simulations of EPR spectra were fitted to the experimental data by using the Nelder–Mead simplex direct search method [120, 121], as implemented in MATLAB's "fminsearch" function. Optimal parameters are obtained by minimizing the mean squared deviation between simulations and the experimental data. For continuous-wave EPR simulations, the mean squared deviations between simulations and the experimental data of both the first and second derivative spectra were calculated. The product of these two mean squared deviations was set as the objective function for optimization.

***ESEEM Spectroscopy.***

Pulsed-EPR experiments were performed by using a laboratory-constructed wideband pulsed-EPR spectrometer that will be described elsewhere (K. Warncke, in preparation). The reflection microwave probe [122] incorporates a folded half-wave resonator [123]. By using a two–pulse ($\pi/2$–$\tau$–$\pi/2$) microwave pulse sequence [124-126], ESE-EPR spectra were obtained. The spectra were used to determine the magnetic field values for maximum ESE amplitudes for the ESEEM experiments. ESEEM was collected with the three–pulse ($\pi/2$–$\tau$–$\pi/2$–$T$–$\pi/2$) microwave pulse sequence [124-126],

with microwave pulse–swapping and phase–cycling [127, 128]. The $\tau$ values were selected to eliminate envelope modulation from the multitude of weakly dipolar–coupled "matrix" [129] protons that surround the radicals [130, 131]. For the matrix $^{1}H$, the hyperfine frequencies, $\nu_{\alpha}$ and $\nu_{\beta}$, corresponding to the $\alpha$ ($m_s$=+1/2) and $\beta$ ($m_s$ = +½) electron spin manifolds, respectively, both approximate the free $^{1}H$ frequency, $\nu_{1H}$ (13.3 MHz at 312.5 mT). Setting $\tau = n/\nu_{1H}$, where $n$ is a positive integer, suppresses the matrix proton modulation [130, 131]. Cosine transformation of the time domain envelope modulation generates the ESEEM frequency spectra. Envelope modulation was processed prior to Fourier transformation by dead time reconstruction, correction of the 40-60 ns amplitude trough in the pulse-crossover segment centered at $\tau$, and correction for any ESE envelope decay by fitting and subtracting a polynomial curve. ESEEM that was acquired with deuterated and natural-abundance aminoethanol samples was divided in order to attenuate the modulation from commonly coupled nuclei [131, 132]. ESEEM data analysis and simulation are discussed in more detail in chapter 4.

## §3.3.2 EPR and ESEEM spectroscopy results and discussions

### *(S)-2-Aminopropanol-generated $Co^{II}$-radical pairs.*

Figure 3.4A shows the EPR spectrum of the cryotrapped $Co^{II}$-substrate radical pair catalytic intermediate in WT EAL, which is formed during steady-state turnover on (*S*)-2-aminopropanol [25]. The $Co^{II}$ intensity is most prominent in the region around 285 mT, which is near to the $g_{\perp}$=2.26 value of the $Co^{II}$ line in the EPR spectrum of magnetically-isolated cob(II)alamin [119]. The $Co^{II}$ features in the radical pair spectrum

are inhomogeneously broadened, relative to isolated cob(II)alamin, by the interaction with the substrate radical [119, 133, 134]. The substrate radical line shape extends from approximately 325.0 to 345.0 mT. The doublet splitting of the substrate radical line shape is caused by the interaction with the unpaired electron spin on $Co^{II}$ [119, 133, 134]. Simulations of the WT EPR spectrum are shown as the red curve in Figure 3.4A. The EPR simulation parameters for the electron-electron separation distance (*R*) and isotropic exchange interaction (*J*) are presented in Table 3.3. The best-fit value for the separation distance between $Co^{II}$ and the C1 radical center of the substrate radical is 11.2 Å, and the isotropic exchange coupling is -290 MHz (corresponding to anti-ferromagnetic coupling in the ground state), which are consistent with values previously reported [30, 135].

Figure 3.4C shows the EPR spectrum of the cryotrapped $Co^{II}$-radical pair formed during steady-state turnover of R160K EAL on (*S*)-2-aminopropanol. An excellent simulation of the EPR spectrum is achieved by using the best-fit parameters, *R*=13.3 Å and *J*=-79 MHz (Table 3.3). The significant value of *J*, and the absence of resolved $Co^{II}$ hyperfine and $^{14}N$ axial ligand superhyperfine features in the spectrum, show that $Co^{II}$ and the radical center are electronically coupled in the R160K mutant, although the weaker coupling leads to a narrowing of the $Co^{II}$ and radical line widths, relative to WT. The $Co^{II}$-radical separation is increased by 2.1 ±0.6 Å in R160K EAL, relative to WT.

Figure 3.4E shows that the $Co^{II}$ and radical features of the EPR spectrum of the cryotrapped $Co^{II}$-radical pair formed during steady-state turnover of R160A EAL on (*S*)-2-aminopropanol are further narrowed relative to R160K. The $Co^{II}$ line shape shows six of the eight features that arise from the octet splitting at $g_{||}$ that arises from the cobalt hyperfine interaction ($I_{Co}$=7/2), and each feature is further split into a triplet by the

superhyperfine interaction with the proximal $^{14}N$ ($I_N$=1) of the dimethylbenzimidazole axial ligand [119]. The presence of resolved cobalt hyperfine and proximal nitrogen superhyperfine features is characteristic of magnetically-isolated cob(II)alamin. This suggests that the $Co^{II}$ and radical species are separated by >14 Å.

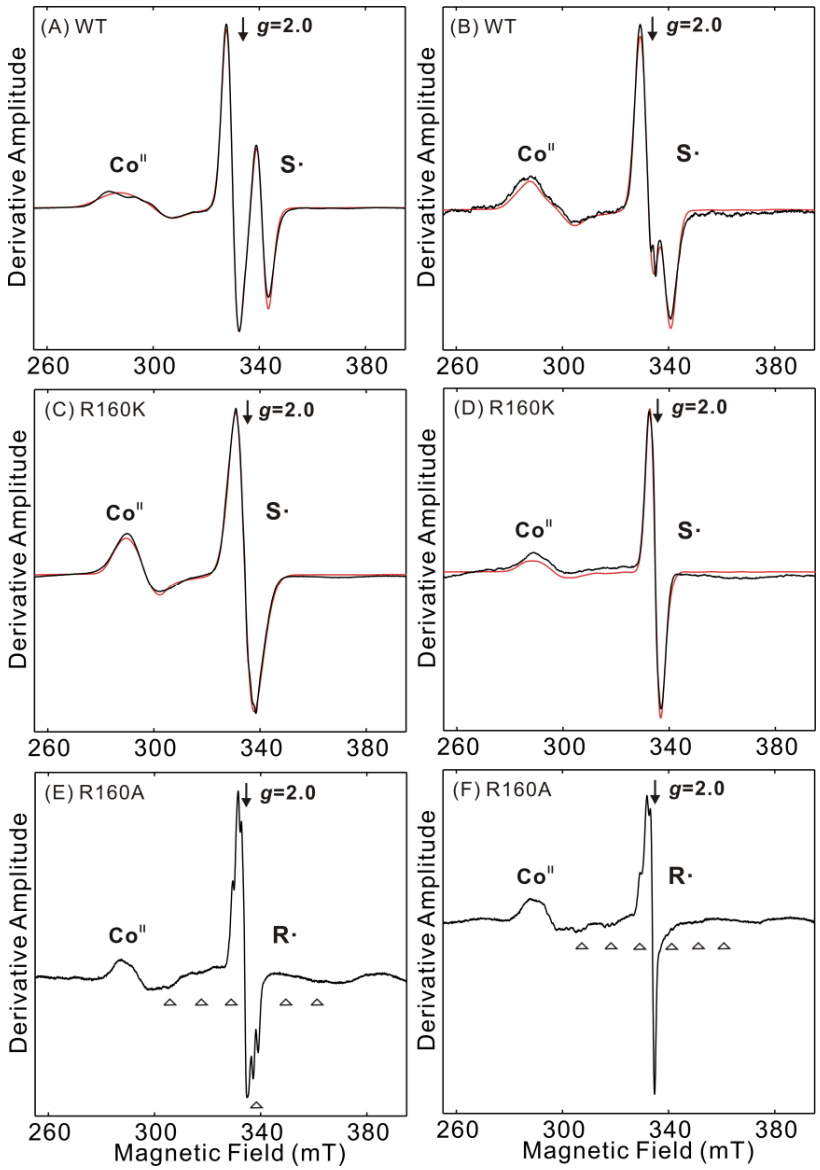


**Figure 3.4. Continuous-wave EPR spectra of the cryotrapped $Co^{II}$-substrate radical pair and $Co^{II}$/radical states formed in wild type, R160K and R160A ethanolamine ammonia-lyase, and overlaid EPR simulations (red) for wild type and R160K EAL.** The free electron resonance position at $g$=2.0 is indicated by downward arrows. The maximum amplitude of transitions associated with $Co^{II}$ at $g_{\perp}$, the substrate radical ($S^{\bullet}$), and unassigned radical ($R^{\bullet}$), are designated. In panels (E) and (F), the cobalt hyperfine coupling features at $g_{||}$ are indicated by triangles. (A) Wild type EAL, substrate (*S*)-2-aminopropanol. (B) Wild type EAL, substrate aminoethanol. (C) R160K EAL, substrate (*S*)-2-aminopropanol. (D) R160K EAL, substrate aminoethanol. (E) R160A EAL, substrate (*S*)-2-aminopropanol. (F) R160A EAL, substrate aminoethanol. *Experimental Conditions:* Microwave frequency, 9.348 GHz; temperature, 6 K; microwave power, 2 mW; magnetic field modulation, 1 mT; modulation frequency, 100 kHz; scan rate, 7.15 mT/s; time constant, 2.56 ms; number of averaged scans: (A)-(D), 32 scans; (E), 128 scans; (F), 64 scans. An average of 128 baseline scans has been subtracted from all spectra. Simulation parameters are listed in Table 3.3.

**Table 3.3 EPR simulation parameters for the $Co^{II}$-substrate radical pairs in wild type and R160K ethanolamine ammonia-lyase.** The table is divided into four parts, which correspond to the EAL protein and the substrate used to form the $Co^{II}$-substrate radical pair state. (A) Wild type, (*S*)-2-aminopropanol. (B) Wild type, aminoethanol. (C) R160K, (*S*)-2-aminopropanol. (D) R160K, aminoethanol.

A. Wild type, (*S*)-2-aminopropanol

| Parameter | Best Fit Value | Lower Bound | Upper Bound |
|---|---|---|---|
| *J* (MHz) | -291.2 | -294.2 | -288.3 |
| *R* (Å) | 11.2 | 11.0 | 11.7 |
| Cobalt $g_{\perp}$ | 2.2598 | 2.2505 | 2.2678 |
| Cobalt $g_{\parallel}$ | 2.0111 | 1.9775 | 2.0669 |
| Cobalt line-width $lw_{\perp}$ | 79.2638 | 44.7653 | 84.6319 |
| Cobalt line-width $lw_{\parallel}$ | 119.3089 | 48.7776 | N.A.* |
| Cobalt $A_{\perp}$ | 9.9477 | 1.5867 | 16.3812 |
| Cobalt $A_{\parallel}$ | 103.0064 | 93.7642 | 117.0826 |
| Radical $g_{\perp}$ | 2.0017 | 2.0014 | 2.002 |
| Radical $g_{\parallel}$ | 1.9986 | 1.9967 | 2.0004 |
| Radical line-width $lw_{\perp}$ | 41.0928 | 40.0273 | 42.2685 |
| Radical line-width $lw_{\parallel}$ | 75.4690 | 70.9772 | 79.5228 |

* Values greater than the best-fit value do not significantly change the spectra.

B. Wild type, aminoethanol

| Parameter | Best Fit Value | Lower Bound | Upper Bound |
|---|---|---|---|
| *J* (MHz) | -160.4 | -166.5 | -153.9 |
| *R* (Å) | 9.4 | 8.6 | 9.8 |
| Cobalt $g_{\perp}$ | 2.2598 | 2.2457 | 2.2751 |
| Cobalt $g_{\parallel}$ | 2.0119 | 1.9521 | 2.0571 |
| Cobalt line-width $lw_{\perp}$ | 62.4588 | 32.6088 | 112.8976 |
| Cobalt line-width $lw_{\parallel}$ | 74.1730 | N.A.* | 128.8730 |
| Cobalt $A_{\perp}$ | 7.5594 | 6.7797 | 19.5126 |
| Cobalt $A_{\parallel}$ | 102.9025 | 90.5803 | 112.4844 |
| Radical $g_{\perp}$ | 2.2598 | 2.2457 | 2.2751 |
| Radical $g_{\parallel}$ | 2.0119 | 1.9521 | 2.0571 |
| Radical line-width $lw_{\perp}$ | 50.7981 | 47.8905 | 52.4915 |
| Radical line-width $lw_{\parallel}$ | 55.8837 | 51.9765 | 65.7982 |

* Values less than the best-fit value do not significantly change the spectra.

**Table 3.3 (Continued)**

C. R160K, (*S*)-2-aminopropanol

| Parameter | Best Fit Value | Lower Bound | Upper Bound |
|---|---|---|---|
| *J* (MHz) | -79.4 | -83.2 | -74.9 |
| *R* (Å) | 13.3 | 12.3 | 14.4 |
| Cobalt $g_{\perp}$ | 2.2599 | 2.2583 | 2.2614 |
| Cobalt $g_{\parallel}$ | 1.9727 | 1.9619 | 1.9814 |
| Cobalt line-width $lw_{\perp}$ | 42.5447 | 36.5174 | 48.8348 |
| Cobalt line-width $lw_{\parallel}$ | 100.6546 | 82.1211 | 123.2203 |
| Cobalt $A_{\perp}$ | 9.4326 | 8.7590 | 10.1077 |
| Cobalt $A_{\parallel}$ | 104.3303 | 100.0610 | 105.0137 |
| Radical $g_{\perp}$ | 1.9973 | 1.9970 | 1.9976 |
| Radical $g_{\parallel}$ | 2.0100 | 2.0097 | 2.0103 |
| Radical line-width $lw_{\perp}$ | 42.5447 | 36.5174 | 48.8348 |
| Radical line-width $lw_{\parallel}$ | 100.6546 | 82.1211 | 123.2203 |

D. R160K, aminoethanol

| Parameter | Best Fit Value | Lower Bound | Upper Bound |
|---|---|---|---|
| *J* (MHz) | 37.7 | 21.5 | 48.1 |
| *R* (Å) | 11.5 | 10.0 | 11.7 |
| Cobalt $g_{\perp}$ | 2.2601 | 2.2392 | 2.2703 |
| Cobalt $g_{\parallel}$ | 2.0058 | 1.8544 | 2.0265 |
| Cobalt line-width $lw_{\perp}$ | 70.2261 | 44.9663 | 105.1751 |
| Cobalt line-width $lw_{\parallel}$ | 35.8106 | 13.7936 | 108.2217 |
| Cobalt $A_{\perp}$ | 6.9885 | 6.4942 | 13.8492 |
| Cobalt $A_{\parallel}$ | 103.0151 | 82.7961 | 109.4666 |
| Radical $g_{\perp}$ | 2.0037 | 2.0033 | 2.0041 |
| Radical $g_{\parallel}$ | 1.9935 | 1.9922 | 1.9949 |
| Radical line-width $lw_{\perp}$ | 36.9482 | 35.7802 | 38.1977 |
| Radical line-width $lw_{\parallel}$ | 48.3139 | 43.3140 | 54.2218 |

***Aminoethanol-generated $Co^{II}$-substrate radical pair states.***

Figure 3.5A shows the EPR spectra of the $Co^{II}$-substrate radical pair generated by using natural abundance-, 1-$^{13}$C- and 1,2-$^{13}C_2$-aminoethanols in WT EAL. The $^{13}$C-label replaces diamagnetic $^{12}$C with the $I$=1/2 $^{13}$C nucleus, and produces EPR line broadening when the unpaired electron spin is localized on the labeled carbon [136]. The EPR spectra of the cryotrapped $Co^{II}$-radical pairs formed during steady-state turnover of R160K EAL on natural abundance and $^{13}$C-labeled aminothanols are shown in Figure 3.5B. The line widths of the radicals in R160K EAL are narrowed relative to WT, but the increase in line width for the $^{13}$C-labeled radicals, and the identical linewidth for the 1-$^{13}$C- and 1,2-$^{13}C_2$-labelled radicals, follows the same pattern observed for WT [29]. Therefore, as in WT, the aminoethanol-generated radical in R160K is the C1-centered substrate radical.

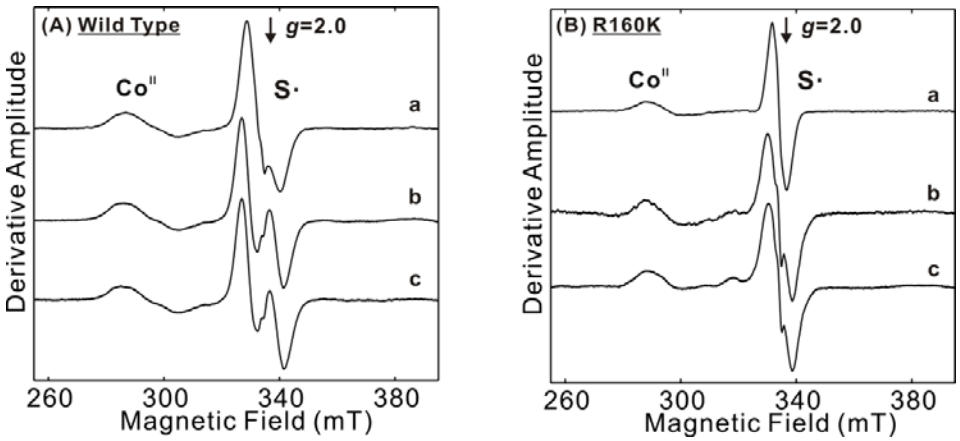


**Figure 3.5 Continuous-wave EPR spectra of the cryotrapped $Co^{II}$-substrate radical pair states formed in wild type and R160K EALduring reaction with natural abundance and $^{13}$C-labelled aminoethanols.** The arrow indicates $g$=2.0. (A) Wild type EAL. (a) natural abundance aminoethanol. (b) 1-$^{13}$C- aminoethanol. (c) 1,2-$^{13}C_2$- aminoethanol. (B) R160K EAL. (a) natural abundance aminoethanol. (b) 1-$^{13}$C- aminoethanol. (c) 1,2-$^{13}C_2$- aminoethanol. *Experimental Conditions:* Microwave frequency, 9.348 GHz; temperature, 6 K; microwave power, 2 mW; magnetic field modulation, 1 mT; modulation frequency, 100 kHz; scan rate, 7.15 mT/s; time constant, 2.56 ms; number of averaged scans: All spectra are averages of 64 scans, subtracted by the average of 128 baseline scans.

Figure 3.4B shows the EPR spectrum of the cryotrapped $Co^{II}$-substrate radical pair catalytic intermediate in WT EAL, which is formed during steady-state turnover on $^{2}H_{4}$-aminoethanol [29]. EPR simulation of the radical pair spectrum gives best-fit values of $R$=9.4 Å for the $Co^{II}$-C1 separation distance and $J$=-160 MHz (Table 3.3) for the $Co^{II}$-substrate radical intermediate in WT EAL, as previously reported [32, 38]. Figure 3.4D shows the EPR spectrum and overlaid simulation for the aminoethanol-generated $Co^{II}$-substrate radical in R160K EAL. The R160K spectrum simulation shows a best-fit $Co^{II}$-C1 separation distance of $R$=11.5 Å, and $J$=38 MHz (Table 3.3) Comparison with the $R$ value for WT EAL indicates that the R160K mutation causes the radical center to shift 2.1 ±1.2 Å away from $Co^{II}$.

The EPR spectrum of the cryotrapped paramagnetic state formed after reaction of R160A EAL with aminoethanol is shown in Figure 3.4F. As observed for the reaction of 2-aminopropanol with R160A, the EPR spectrum shows a narrow line radical and cobalt hyperfine and axial ligand nitrogen superhyperfine features that are characteristic of uncoupled cob(II)alamin [119]. The origin of the radical in R160A EAL radical was addressed by comparing the line shapes of the radical formed from $^{1}H_{4}$- and $^{2}H_{4}$-aminoethanol. The $^{2}H$ labels at C1 and C2 cause narrowing of the substrate radical line shape [25, 29], because the 6.51-fold smaller magnetic dipole moment of deuterium relative to protium causes a corresponding decrease in the hyperfine coupling constant. In addition, reaction of R160A EAL with $^{2}H_{4}$-aminoethanol may result in the incorporation of one or more $^{2}H$ into the C5'-methyl group hydrogen positions [137]. If the R160A radical resides on C1, C2, or C5', a narrowing of the line shape is expected. It is observed that substitution of deuterium on aminoethanol does not influence the EPR

line shape of the radical in R160A EAL (data not shown). The radical formed in R160A EAL is therefore not associated with the substrate or with the C5' center.

The EPR spectroscopic results for R160A EAL are consistent with the absence of detectable turnover in this mutant. The substitution of alanine at R160 causes a mechanism-based inactivation of the enzyme, prior to completion of the full catalytic cycle. The radicals formed upon reaction with both (*S*)-2-aminopropanol and aminoethanol in R160A appear to arise from a site (or multiple sites) on the protein, which is ≥14 Å from $Co^{II}$.

***Comparison of $Co^{II}$-substrate radical pair separation distances in WT and R160K EAL.***

The increase in $Co^{II}$-radical pair separation distance of 2.1 ±0.6 Å and 2.1 ±1.2 Å for the (*S*)-2-aminopropanol and aminoethanol-derived radical pair states, respectively, are the same within the uncertainty. The distance increase is comparable to the predicted ~1.6 Å shortening of the side chain, and therefore, of the corresponding positional shift in the positive charge center of residute 160 when lysine is substituted for arginine. As shown in Figure 2.8, the comparative model of EutB predicts that the guanidinium group of R160 is positioned directly over the substrate, on the side opposite $Co^{II}$ in cobalamin [5]. Therefore, the EPR results suggest that the substitution of the shorter lysine side chain at R160 results in an *en bloc* movement of C1 of the substrate towards the side chain terminus, which corresponds to movement away from $Co^{II}$.

### *$^{14}$N ESEEM of the Substrate Radical in WT and R160K EAL.*

The hyperfine coupling of the unpaired electron spin on C1 of the substrate radicals with a $^{14}$N nucleus on the protein was previously characterized by using $^{14}$N ESEEM spectroscopy [34, 35]. This coupling provides a sensitive probe of the radical-protein interaction.

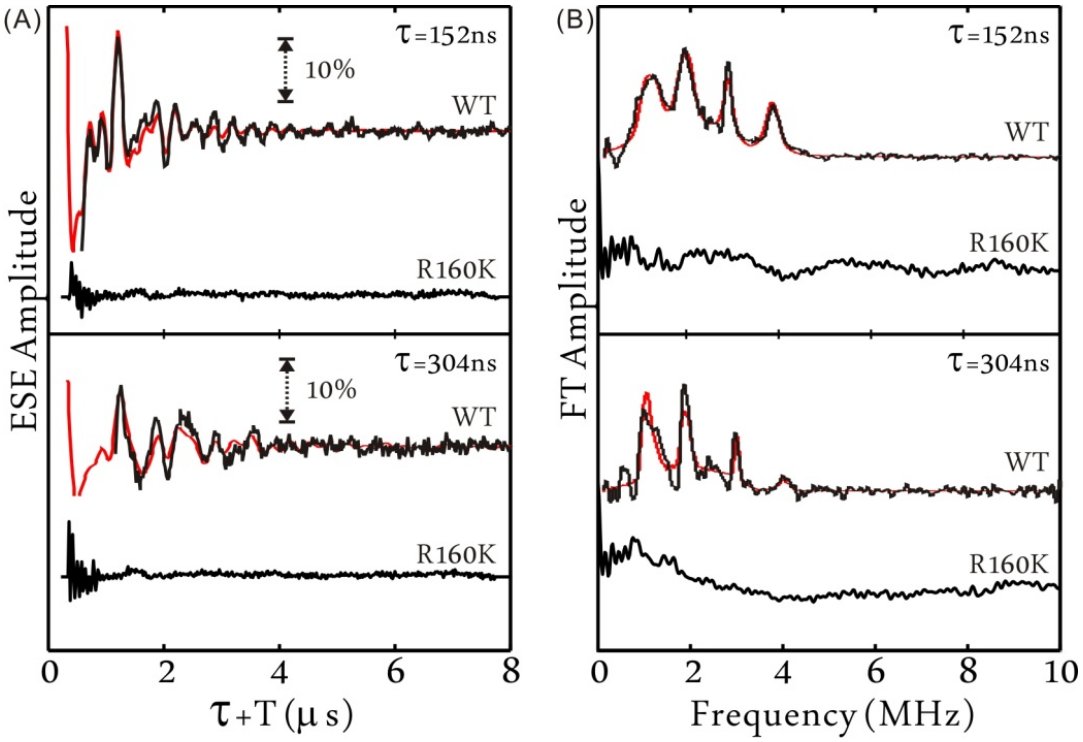


**Figure 3.6. Three-pulse ESEEM and Fourier transforms for the (*S*)-2-aminopropanol-derived substrate radical in the cryotrapped Co$^{II}$-substrate radical pair state in wild type and R160K ethanolamine ammonia-lyase, and overlaid $^{14}$N ESEEM simulations (red) for wild type.** (A) ESEEM waveforms for wild type and R160K EAL at $\tau$ values of 152 and 304 ns. The wild-type and R160K ESEEM is normalized to the unmodulated echo amplitude. The scale bar represents 10% of the total unmodulated ESE amplitude. (B) Fourier transformations of the corresponding ESEEM in panel (A). The frequency spectra are normalized to match the common noise level in each spectrum. *Experimental conditions:* temperature, 6 K; magnetic

field, 309.0 mT; microwave frequency 8.772 GHz; microwave power 20 W; τ values, 152 ns and 304 ns; initial τ+T value, 120 ns; T increment, 20 ns; $\pi/2$ pulse width, 20 ns; pulse repetition rate, 100 Hz; 128 repetitions averaged per point. *Simulation parameters*: $e^2qQ/h$=3.09 MHz, $\eta$=0.55, $r_{eff}$=3.42 Å, $A_{iso}$=0.93 MHz.

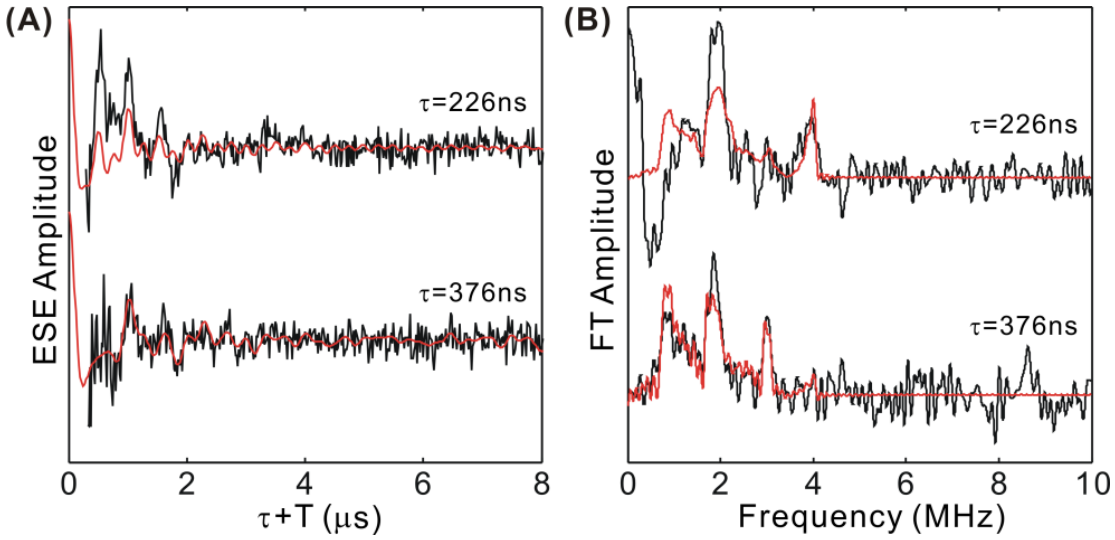


**Figure 3.7. Three-pulse ESEEM and Fourier transforms for the aminoethanol-derived substrate radical in the cryotrapped $Co^{II}$-substrate radical pair state in wild type ethanolamine ammonia-lyase, and overlaid $^{14}N$ ESEEM simulations (red).** *Experimental conditions*: temperature, 6 K; magnetic field, 312.0 mT; microwave frequency 8.7720 GHz; microwave power 20-35 W; initial τ+T value, 120 ns; T increment, 20 ns; π/2 pulse width, 20 ns; pulse repetition rate, 200 Hz; 2560 repetitions averaged per point. *Simulation parameters*: $e^2qQ/h$=3.19 MHz, $\eta$=0.51, $r_{eff}$=3.48 Å, $A_{iso}$=0.78 MHz.

Figure 3.6 shows the three-pulse ESEEM and Fourier transforms for the (*S*)-2-aminopropanol-generated $Co^{II}$-substrate radical pair in WT and R160K EAL, that were obtained for $\tau$ values of 152 and 304 ns. ESEEM simulation formulism and procedures are presented in details in Chapter 4. Comparable $^{14}N$ ESEEM is observed from the aminoethanol-generated $Co^{II}$-substrate radical pair (Figure 3.7), but at significantly lower signal-to-noise ratios, because of the lower fraction of the $Co^{II}$-substrate radical pair

intermediate that accumulates during steady-state turnover on aminoethanol [10-20% [29]], relative to (*S*)-2-aminopropanol [100% [110]] [35]. The ESEEM frequency spectra from WT EAL in Figure 3.6 show the characteristic pattern of four features observed previously [34, 35]. The 0.8-1.0, 1.7-2.0, and 2.9-3.0 MHz features were assigned to the "hyperfine-adulterated" [138] $\nu_0$, $\nu_-$, and $\nu_+$ nuclear quadrupole frequencies of $^{14}$N, and the feature at 4.0 MHz was assigned to the $\Delta m_I = \pm 2$ splitting in the $m_s$ manifold in which the hyperfine and nuclear Zeeman terms are additive [34, 35]. This distinctive pattern of features arises from the condition of "exact cancellation", in which the hyperfine and nuclear Zeeman contributions to the electron-$^{14}$N coupling cancel ($A/2 \approx \nu_N$, where $A$ is the hyperfine coupling constant and $\nu_N$ is the free $^{14}$N frequency) for one $m_s$ manifold [138, 139]. The $^{14}$N ESEEM simulations, shown as overlaid red lines in Figure 3.6, reveal a nuclear quadrupole coupling constant ($e^2qQ/h$) of 3.09 MHz and an electric field gradient asymmetry parameter ($\eta$) of 0.55, in agreement with previous simulations [35]. Comparable values of $e^2qQ/h$=3.19 MHz and $\eta$=0.51 are obtained for the aminoethanol substrate radical in WT EAL. In dramatic contrast to WT EAL, Figure 3.6 shows that there is no significant ESEEM signal from the $Co^{II}$-substrate radical in R160K EAL at $\tau$=152 and 304 ns.

The coupled $^{14}$N ESEEM nucleus was originally assigned to a peptide secondary amide nitrogen [34, 35], based on comparison of the $\nu_0$, $\nu_-$, and $\nu_+$ values [34] or correspondence of the simulated $e^2qQ/h$ and $\eta$ values [35] with nuclear quadrupole double resonance parameters reported for peptide nitrogen in di- and tri- peptides [140]. The loss of the $^{14}$N ESEEM in R160K EAL, which is predicted by the EutB model, suggests that the arginine guanidinium group harbors the coupled nitrogen nucleus.

There appear to be no reported nuclear quadrupole parameters for arginine or guanidinium for comparison. $^{14}$N ESEEM from the $Q_H$ semiquinone in cytochrome $bo_3$, which had been assigned to a peptide nitrogen [141], may have an origin from a hydrogen bond to arginine R71, as suggested by the X-ray crystallographic structure of cytochrome $bo_3$ [142]. The origin of the $^{14}$N coupling from arginine or other nitrogen containing side chains in the $Q_H$ binding site has been assessed [143]. The reported value of $e^2qQ/h$=3.72 MHz for cytochrome $bo_3$ is higher than the value of $e^2qQ/h$=3.09-3.19 for EAL. However, the uncertainty in –NH versus $–NH_2$ origin, and the charge distribution and bond orders in the R160 guanidinium group that arise from interactions in the protein, make R160 a candidate for the coupled $^{14}$N nucleus. We therefore suggest that the $^{14}$N ESEEM in WT EAL arises from coupling to a guanidinium nitrogen in the side chain of R160 from EutB. This interaction is absent in the R160K EAL, which is consistent with the prediction of the EutB model. The absence of any detectable $^{14}$N ESEEM in R160K EAL suggests that the sensitive condition of "exact cancellation" [138, 139] is not met, which is consistent with a displacement of the lysine ammonium nitrogen relative to the guanidinium nitrogen.

***$^2$H ESEEM of the aminoethanol-generated radical in WT and R160K EAL.***

The hyperfine interaction of the unpaired electron spin at C1 of the substrate radical with $^2$H nuclei in the C5'-methyl group provides a sensitive probe of the in situ reactant geometry at the substrate binding site [30, 36-38]. The $^2$H are incorporated into the C5'-methyl group by turnover of EAL on 1,1,2,2-$^2H_4$-aminoethanol, prior to cryotrapping [115]. The $^2$H-labeled aminoethanol also allows measurement of the

hyperfine coupling between the unpaired electron at C1 and the β-$^2$H positions on C2. Figure 3.8 shows representative three-pulse $^2$H/$^1$H quotient ESEEM for the $^2H_4$-aminoethanol-generated substrate radicals in WT and R160K EAL that was collected at the maximum echo amplitude ($g$=2.009). Division of the echo envelopes for $^2$H-labeled and natural abundance ($^1$H) samples to produce the $^2$H/$^1$H quotient ESEEM eliminates or strongly attenuates the contribution of common coupled nuclei [131, 132]. The $^2$H ESEEM and the corresponding Fourier transforms shown in Figure 3.8 for WT and R160K EAL show a high degree of similarity. The Fourier transforms in Figure 3.8 for $\tau$ values of 226 and 376 ns show the same features as previously reported for a different set of $\tau$ values [37]. The broad feature at 1.0-3.0 MHz arise from coupling to one short-distance (effective electron-nuclear separation distance, $r_{eff}$=2.1 Å) $^2$H nucleus and two $^2$H nuclei at longer separation distances ($r_{eff}$=2.4-2.6 Å). The feature at 4.5-6.0 MHz in ESEEM frequency spectrum arises from one of the two β-$^2$H hyperfine couplings ($^2H_{\beta b}$). Features from the second, more strongly coupled β-$^2$H ($^2H_{\beta b}$) are clearly observed in ESEEM experiments performed at higher resonant microwave frequency/magnetic field values (10.89 GHz/388.0 mT), but are within the noise level at 8.7-8.9 GHz/ 300.0-310.0 mT [37]. Therefore, the ESEEM results presented in Figure 3.8 show that the same complement of $^2$H are comparably coupled to the radical in WT and R160K EAL.

ESEEM simulations for the combination of $^2$H hyperfine couplings in WT and R160K EAL are shown as the overlaid red curves in Figure 3.8, and the best-fit parameter values are presented in Table 3.4. The $r_{eff}$ and $A_{iso}$ values for the three C5'-methyl $^2$H are conserved for WT and R160K EAL. The $A_{iso}$ values of the $^2H_{\beta a}$ coupling in WT and R160K are also comparable. The $A_{iso}$ value for $^2H_{\beta a}$ is related to the dihedral

angle (θ) between the C2-$^{2}H_{\beta a}$ bonding orbital and the *p*-orbital on C1 that contains the unpaired electron spin density, as follows [144]: $A_{iso}=\rho B_2\cos^2\theta$, where $\rho$ is the unpaired electron spin density on C2 and $B_2$=24.99 MHz for $^{2}H$. Therefore, the C1-C2 rotamer states in WT and R160K are comparable. Overall, the $^{2}H$ ESEEM results indicate that the geometry of the C2, C1 and C5'-methyl centers in WT EAL is conserved in the substrate binding region of the active site in R160K EAL.

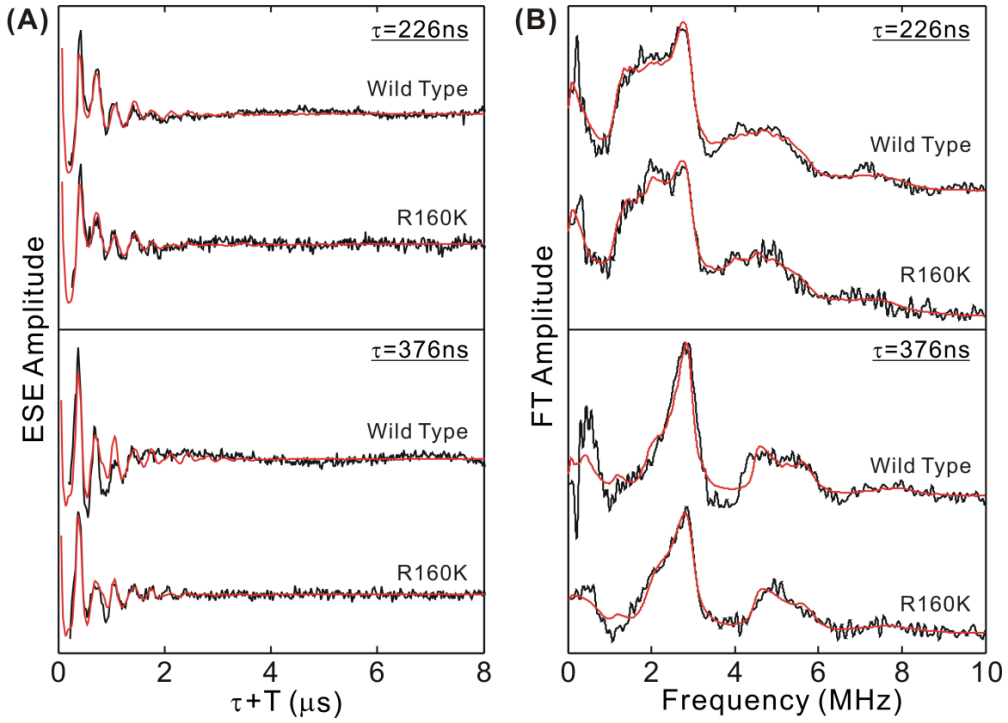


**Figure 3.8 Three-pulse $^{2}H/^{1}H$ quotient ESEEM and corresponding Fourier transforms for the aminoethanol-derived substrate radical in the cryotrapped $Co^{II}$-substrate radical pair state in wild type and R160K ethanolamine ammonia-lyase, and overlaid $^{2}H$ ESEEM simulations (red).** (A) ESEEM waveforms for wild type and R160K EAL at $\tau$ values of 226 and 376 ns. The wild-type and R160K ESEEM is normalized to the unmodulated echo amplitude. (B) Fourier transformations of the corresponding ESEEM in panel (A). The frequency spectra are normalized to match the common noise level in each spectrum. *Experimental conditions:* temperature, 6 K; magnetic field, 312.0 mT (wild type) and 312.8 (R160K) ; microwave frequency 8.7720 GHz; microwave power 20-35 W; τ values, 226 ns and 376 ns; initial τ+T value, 120 ns; T increment, 20 ns; $\pi/2$ pulse width, 20 ns; pulse repetition rate, 200 Hz; 1280 repetitions averaged per point. Simulation parameters are presented in Table 3.4.

**Table 3.4. ESEEM simulation parameters for the deuterium hyperfine coupling in wild type and R160K EALobtained by global optimization of the $^{2}H/^{1}H$ quotient ESEEM for $\tau$=226 and 376 ns conditions.**

| Enzyme | $^{2}H$ Coupling | Parameter | Best-fit Value | Lower Bound | Upper Bound |
|---|---|---|---|---|---|
| Wild Type | $^{2}H_{1}$ | $A_{iso}$ (MHz) | -0.38 | -0.13 | -0.40 |
| | | $r_{eff}$ (Å) | 2.08 | 2.07 | 2.14 |
| | $^{2}H_{2}$ | $A_{iso}$ (MHz) | -1.18 | -0.92 | -1.21 |
| | | $r_{eff}$ (Å) | 2.57 | 2.47 | 2.62 |
| | $^{2}H_{3}$ | $A_{iso}$ (MHz) | -1.15 | -0.99 | -1.20 |
| | | $r_{eff}$ (Å) | 2.43 | 2.28 | 2.46 |
| | $^{2}H_{\beta a}$ | $A_{iso}$ (MHz) | 5.33 | 4.39 | 5.37 |
| | | $r_{eff}$ (Å) | 2.72 | 2.28 | 2.79 |
| R160K | $^{2}H_{1}$ | $A_{iso}$ (MHz) | -0.35 | -0.06 | 0.55 |
| | | $r_{eff}$ (Å) | 2.10 | 2.07 | 2.20 |
| | $^{2}H_{2}$ | $A_{iso}$ (MHz) | -1.11 | -0.90 | -1.23 |
| | | $r_{eff}$ (Å) | 2.45 | 2.37 | 2.66 |
| | $^{2}H_{3}$ | $A_{iso}$ (MHz) | -1.11 | -0.66 | 1.31 |
| | | $r_{eff}$ (Å) | 2.45 | 2.28 | 2.73 |
| | $^{2}H_{\beta a}$ | $A_{iso}$ (MHz) | 5.33 | 5.04 | 5.68 |
| | | $r_{eff}$ (Å) | 2.72 | 2.45 | 3.14 |

## §3.4 Conclusions

The principal conclusions from the protein chemical, enzyme kinetic and EPR spectroscopic studies of the EutB R160 mutations, and their implications for factors that determine the structure and function of EAL, are depicted in Figure 3.9, and summarized as follows:

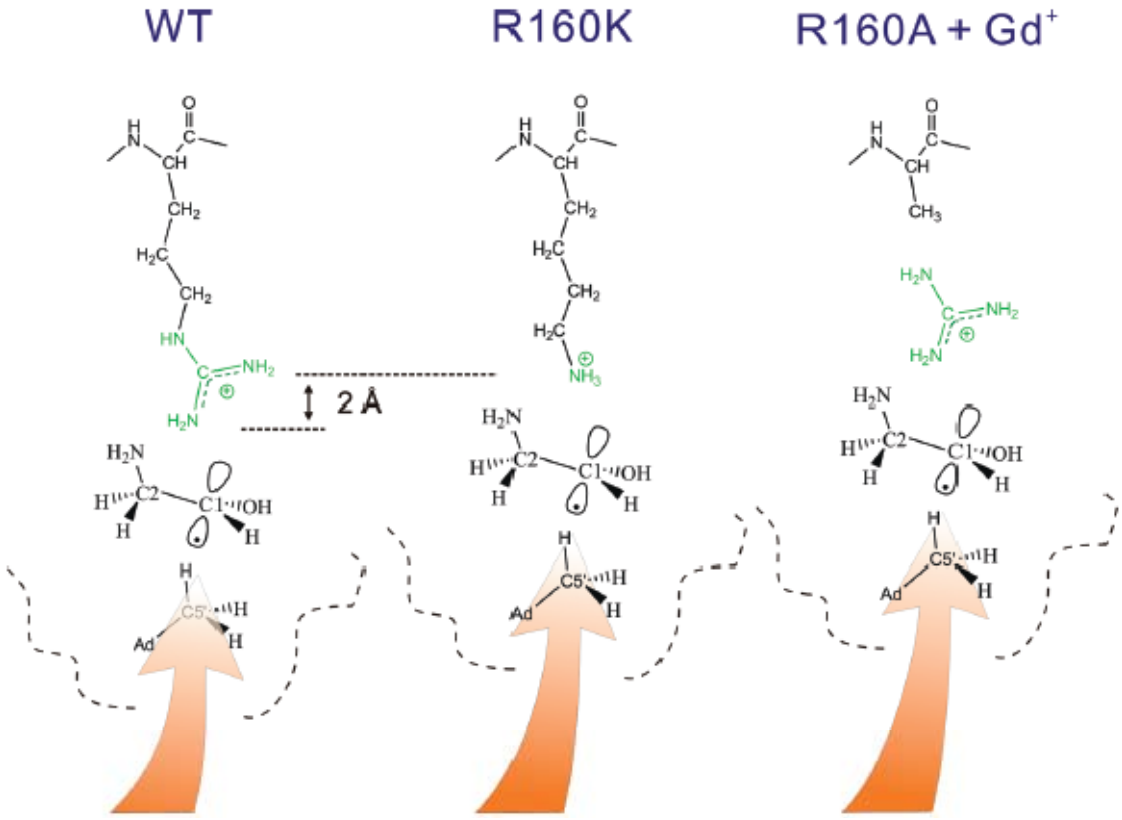


**Figure 3.9. Model for the substrate binding region of the active site in WT, R160K and R160A with bound guanidinium, and depiction of effects of mutations on structure.** The predicted 1.6 Å difference in side chain length between arginine and lysine, and the corresponding 2.1 Å displacement of the substrate and C5'-center, are shown. Guanidinium is predicted to bind in the cavity created by removal of the native *n*-propyl-guanidinium side chain in R160A EAL. The direction of the protein mediated force that maintains the C5' center and substrate at van der Waals contact is depicted by the arrow.

(1) The positive charge at the terminus of the R160 side chain of EutB is required to achieve the native EutB protein fold and the oligomeric structure of EAL. Mutations R160E and R160I do not assemble into an oligomer. R160A EAL assembles, but is unstable. The R160K mutation, which is predicted to maintain the positive charge at a similar position in the interior of EutB, leads to proper folding of EutB and assembly of a stable EAL oligomer.

(2) The positive charge at the terminus of the R160 side chain is critical for catalysis in EAL. The R160K mutation, which is predicted to maintain the positive charge at a similar position in the active site of EutB, affords turnover of aminoethanol with a relatively modest decrement of 180-fold in $k_{cat}/K_M$, relative to WT EAL. In addition to executing complete turnover, R160K EAL displays magnetically-coupled, $Co^{II}$-substrate radical pair states, which are a signature of functional intermediates in EAL [25, 29] and other coenzyme $B_{12}$-dependent enzymes [134]. In contrast, R160A EAL is inactive, and forms magnetically-isolated $Co^{II}$ and radical species. The positive charge of the guanidinium group therefore appears to be the primary feature of the R160 side chain that is most essential to catalysis, followed by the capability to form hydrogen bonds with the reactant species.

(3) The R160 side chain directly interacts with the substrate species through a hydrogen bond. This conclusion is based on the proposal that the $^{14}N$ nuclear quadrupole parameters ($e^2qQ/h$=3.09 MHz, $\eta$=0.5) from the ESEEM simulations correspond to an amide-like guanidinium nitrogen. The significant $A_{iso}$ value from the ESEEM simulations indicates a through-bond transmission of unpaired electron spin density to the 14N nucleus. A minimum through-bond pathway from the C1 spin density center to a

guanidinium nitrogen that is hydrogen bonded to the carbinol oxygen would involve two covalent bonds and one hydrogen bond. This is similar to the path length for the classical case of exact cancellation $^{14}N$ ESEEM from coupling to the remote nitrogen of imidazole in $Cu^{II}$-imidazole complexes [139]. The minimum path for guanidinium nitrogen hydrogen bonded to the substrate amine nitrogen is three covalent bonds and one hydrogen bond. This conclusion suggests that R160 has a role in both substrate binding and catalysis. Whether the guanidinium moiety is hydrogen bonded to the substrate hydroxyl, substrate amine, or both cannot be determined from the present data.

(4) The C1 electron spin density center on the (*S*)-2-aminopropanol- and aminoethanol-derived substrate radicals shifts away from $Co^{II}$ by distances of 2.1 ±0.6 Å and 2.1 ±1.2 Å, respectively, in R160K relative to WT EAL. This is equivalent to the calculated 1.6 Å reduction in the length of the residue 160 side chain when arginine is replaced by lysine. The displacement is predicted to lie approximately along the $Co^{II}$-C1 axis (the $g_{||}$ axis of $Co^{II}$) by the EutB model [5]. We therefore interpret this shift as indicating a strong Coulombic and hydrogen bonding interaction of the positive charge on R160 with one or both substrate heteroatoms, which draws the substrate "up" in the active site in the view shown in Figure 2.8. The $^{2}H$ ESEEM results show that the position of the 5'-methyl center of 5'-deoxyadenosine tracks with the displacement of the radical center, because geometry of the C1, C2 and C5' centers in R160K and WT EAL is conserved. Therefore, we propose that a force is applied to the C5'-methyl center, which maintains the native van der Waals contact with the substrate radical center. The force on the C5'-methyl center reveals a vectoral, mechanical coupling between the protein and the radical. The force maintains the association of the C5' center and the

substrate radical at van der Waals contact during the first hydrogen atom transfer, radical rearrangement, and second hydrogen atom transfer reactions. The force must be released following HT2, so that the C5' radical center can leave the substrate binding region and recombine with $Co^{II}$ to reform the Co-C bond.

(5) The comparative model of EutB provides a consistent interpretation of the results of the protein chemical, enzyme kinetic, and EPR and ESEEM spectroscopies, which provides strong support for the verity of the EutB model.

# Chapter 4

# Numerical simulation and nuclear parameter optimization of ESEEM with OPTESIM toolbox

## § 4.1 Motivation of OPTESIM toolbox

Electron spin echo envelope modulation (ESEEM) is an important technique of pulsed-electron paramagnetic resonance (EPR) spectroscopy that reveals the energies of electron-nuclear hyperfine (hf) interactions, and, for nuclei with $I \geq 1$, nuclear quadrupole interaction (nqi) energies [33, 145]. Nuclear state mixing and a bandwidth of the microwave pulse magnetic field ($B_1$) larger than the hf splitting afford favorable conditions for the detection of ESEEM from single or small numbers of electron-nuclear couplings. The modulation of the echo envelope is collected in the pulse time domain, and is cosine Fourier-transformed to create an ESEEM frequency spectrum.

In certain cases, qualitative information about the identity, number, and magnitude of the coupling of the coupled nuclear spin(s) can be gained from inspection of the modulation periodicities in the waveform or the line positions in the Fourier transform (FT). However, simulation of the ESEEM is required to extract the coupling parameters that give the desired detailed information about the nuclear and electronic structure of the molecular paramagnet, such as the hf tensor and electric field gradient (efg) tensor (nqi parameters). Different programs for the numerical simulation of ESEEM have been reported and used to interpret one-dimensional and two-dimensional experiments [30, 146-151]. Although the simulations yield, in many cases, a high level of agreement with experiment, the process of matching simulation and experiment remains both subjective and time-consuming, and is the bottleneck in the interpretation of ESEEM experiments. Here, we designed an ESEEM simulation software toolbox that incorporates automated fitting with an efficient hybrid optimization approach, and provides an assessment of the statistical significance of the simulation parameters.

The overall design goal of this tool box (denoted by OPTESIM, for optimization of ESEEM simulation) is to provide a modular, flexible set of routines for an automated, optimization-based simulation of diverse paramagnetic systems (arbitrary number of hf coupled nuclei with arbitrary nuclear spin; different classes of electron spins). For this purpose, a collection of functions for scripting is most appropriate. The MATLAB (Mathworks, Natick, MA, USA) programming environment is chosen for the development of OPTESIM, because of the following features: (1) MATLAB is generally available for academic use, (2) MATLAB is easier for further development, compared to a C++ or FORTRAN environment, and (3) a number of literature numerical and optimization routines are available in MATLAB. In addition, a practical consideration is that experiment control and data collection software can be developed in MATLAB, which allows a seamless flow from ESEEM data acquisition to end analysis. The core calculations of the nuclear eigenvalues and eigenvectors and the time evolution of the density matrix use direct diagonalization methods [130, 152]. The optimization is achieved by using the minimization of the least-squares residuals of the match of the simulated to the experimental ESEEM, and is generally carried out in the time-domain, where distortions of the FT owing to waveform truncation are absent. Algorithms for the search of the hf, nqi, and system-specific Euler angle parameter space are as follows: genetic algorithm [153], simulated annealing [154], Nelder-Mead simplex [155], Gauss-Newton method, and Levenberg-Marquardt method [156]. A statistical assessment of the uncertainty in a parameter, with respect to other parameters, is given by the calculated simultaneous confidence region for each parameter.

We apply OPTESIM to the simulation of a system, which had been previously simulated by “manual” simulation, and to one which is too complex to efficiently simulate by “manual” simulation. In each case, a two-tier approach of genetic algorithm (coarse grained search for basin around the global minimum) followed by simplex (fine grained search for the global minimum within the basin) optimization are used. The automated, optimization simulations converge on dramatically shorter timescales, relative to manual simulations.

## § 4.2 Simulation of ESEEM

### § 4.2.1 ESEEM Theory

***Spin Hamiltonian***

The stationary-state Hamiltonian for treatment of the interaction of an electron spin ($S$=1/2) with one nuclear spin ($I$) is formulated with a hf coupling term, a nuclear Zeeman term, and a nqi term (for $I \geq 1$), as follows:

$$\mathbf{H} = h\,\bar{S}\bullet\mathbf{A}\bullet\bar{I} - g_{\mathrm{N}}\beta_{\mathrm{N}}\,\bar{B}_0\bullet\bar{I} + \bar{I}'\bullet\mathbf{Q}\bullet\bar{I}' \qquad \text{[Eq. 4.1]}$$

where $\beta_{\mathrm{N}}$, $\bar{S}$, and $\bar{I}$ are the nuclear magneton, electron spin operator, and nuclear spin operator, respectively, $g_{\mathrm{N}}$ is the nuclear $g$-value, **A** is the hf coupling tensor, and **Q** is the nqi tensor [157]. The $I$ and $I'$ indicate that the eigenfunctions of **A** and **Q** are, in general, different. The hf tensor has the principal components, $\mathbf{A_p} = \begin{bmatrix} A_{xx} & A_{yy} & A_{zz} \end{bmatrix}$, and is composed of an isotropic part, $A_{\mathrm{iso}}$, and a dipolar part. In the point dipole approximation, the dipolar part is given by an axially symmetric dipolar tensor, $\begin{bmatrix} -A_{\mathrm{dip}} & -A_{\mathrm{dip}} & 2A_{\mathrm{dip}} \end{bmatrix}$, where $A_{\mathrm{dip}} = \left(\mu_0 / 2h\right) g_{\mathrm{e}}\beta_{\mathrm{e}} g_{\mathrm{N}}\beta_{\mathrm{N}} r_{\mathrm{eff}}^{-3}$, and $g_{\mathrm{e}}$, $\beta_{\mathrm{e}}$, and $r_{\mathrm{eff}}$ are the electron $g$-value, Bohr magneton, and an effective distance between the electron and the nuclear spins, respectively [157]. The hf tensor $\mathbf{A_p}$ is rotated to the molecular frame, which is defined by a reference principal axis system (PAS) that is provisionally related to the electron $g$ tensor PAS, by the following operation:

$$\mathbf{A} = \mathbf{R}_{\mathrm{A}}{}^{T}\mathbf{A}_{\mathbf{p}}\mathbf{R}_{\mathrm{A}} \qquad \text{[Eq. 4.2]}$$

where $\mathbf{R}_{\mathrm{A}}$ is a rotation matrix, which is defined by the Euler angles, $\begin{bmatrix}\alpha_{\mathrm{A}} & \beta_{\mathrm{A}} & \gamma_{\mathrm{A}}\end{bmatrix}$, to rotate from the electron *g* tensor PAS to the hf coupling tensor PAS.

The nqi tensor, **Q**, is defined by the nuclear quadrupole coupling constant, $e^2qQ/h$, and the efg asymmetry parameter, $\eta$, where $e$, $q$, and $Q$ are the elementary charge, the magnitude of the principal component of the efg tensor, and the nuclear quadrupole moment, respectively. In its PAS, the traceless nqi tensor $\mathbf{Q}_{\mathbf{p}} = \begin{bmatrix}Q_{\mathrm{xx}} & Q_{\mathrm{yy}} & Q_{\mathrm{zz}}\end{bmatrix}$ is related to $e^2qQ/h$ and $\eta$ by the following expressions:

$$Q_{\mathrm{zz}} = e^2qQ/(2I(2I-1)h) \qquad \text{[Eq. 4.3]}$$

$$\eta = (Q_{\mathrm{xx}} - Q_{\mathrm{yy}})/Q_{\mathrm{zz}} \qquad \text{[Eq. 4.4]}$$

where $|Q_{\mathrm{zz}}| \geq |Q_{\mathrm{yy}}| \geq |Q_{\mathrm{xx}}|$. In the toolbox, the orientation between the nqi tensor PAS and the hf tensor PAS is defined by the Euler angles $\begin{bmatrix}\alpha_{\mathrm{Q}} & \beta_{\mathrm{Q}} & \gamma_{\mathrm{Q}}\end{bmatrix}$, which consequently define a rotational matrix, $\mathbf{R}_{\mathrm{Q}}$, and nqi tensor **Q** in the molecular PAS (*g* tensor PAS) to be expressed as follows:

$$\mathbf{Q} = \mathbf{R}_{\mathrm{A}}{}^{\mathrm{T}}\mathbf{R}_{\mathrm{Q}}{}^{\mathrm{T}}\mathbf{Q}_{\mathrm{p}}\mathbf{R}_{\mathrm{Q}}\mathbf{R}_{\mathrm{A}} \qquad \text{[Eq. 4.5]}$$

This definition of **Q** by using a two-stage rotation allows an additional constraint on the mutual orientations of **A** and **Q** during numerical optimization, which adds to the

flexibility in the specification of the geometry model for multiple electron-nuclear interactions.

To summarize, the general coupled electron – single nucleus system for $I \geq 1$ is parameterized by using the following thirteen parameters: $g_N$, $I$, $A_{xx}$, $A_{yy}$, $A_{zz}$, $\alpha_A$, $\beta_A$, $\gamma_A$, $e^2qQ/h$, $\eta$, $\alpha_Q$, $\beta_Q$, and $\gamma_Q$ (for $I = 1/2$, the five nqi-associated parameters are dropped). In practice, the coupled nucleus is assigned based on characteristic spectral features or a priori knowledge of the system, so that $g_N$ and $I$ are fixed, which entails eleven adjustable parameters. In disordered, or powder, systems (no preferred orientation of the molecular, or *g*, PAS relative to $B_0$), $\alpha_A$, $\beta_A$, and $\gamma_A$ are not considered, and the number of adjustable parameters is reduced to eight. The point dipole assumption for electron-nuclear dipolar coupling reduces the number of adjustable parameters to seven. Additional nuclear couplings introduce three Euler angle parameters that describe the mutual orientations of the hf PAS, in addition to the individual nuclear coupling parameters, described above. In OPTESIM, all of the parameters can be subjected to the numerical optimization.

***ESEEM Calculation***

The time-domain ESEEM calculation incorporates the full density matrix diagonalization from the Mims density matrix treatment of ESEEM [130, 152]. Although the time evolution simulation of a pulsed magnetic resonance experiment is straightforward [150, 151], the explicit evaluation of the transition frequencies and intensities is computationally advantageous for optimization purposes, relative to methods that calculate the time evolution of the density matrix. Specifically, the

following adapted numerical procedure [33, 145, 158], which we have used previously in ESEEM simulations [30], is found to be most efficient for constructing the two-pulse and three-pulse modulation in OPTESIM.

The Hamiltonian is first partitioned into sub-matrix representations $\mathbf{H}_\alpha$ and $\mathbf{H}_\beta$ corresponding to $\alpha$ ($m_s$=1/2) and $\beta$ ($m_s$=-1/2) electron spin manifolds. $\mathbf{H}_\alpha$ and $\mathbf{H}_\beta$ are diagonalized separately with unitary matrices $\mathbf{M}_\alpha$ and $\mathbf{M}_\beta$ to obtain the eigenfrequencies, $\nu_\alpha$ and $\nu_\beta$, belonging to the $\alpha$ and $\beta$ manifolds, respectively. The Hamiltonian, $\mathbf{H}$, is transformed into $\mathbf{H}'$, which corresponds to the coupled representation, as follows:

$$\mathbf{H}' = \begin{bmatrix} 0 & \mathbf{M} \\ \mathbf{M}^\dagger & 0 \end{bmatrix} \qquad \text{[Eq. 4.6]}$$

where $\mathbf{M} = \mathbf{M}_\alpha{}^\dagger \mathbf{M}_\beta$.

The modulation frequencies and amplitudes are then calculated for specific types of experiments. For example, for the two-pulse echo envelope modulation (pulse sequence: $\pi/2 - \tau - \pi/2 - \tau$ –*detection*), the frequencies $\omega$ and amplitudes $A$ are as follows:

$$\omega_{lmjk} = 2\pi(\nu_{\beta l} + \nu_{\alpha m} - \nu_{\beta j} - \nu_{\alpha k})$$

$$A_{lmjk} = \mathrm{M}_{jk}{}^\dagger \mathrm{M}_{kl} \mathrm{M}_{lm}{}^\dagger \mathrm{M}_{mj} \qquad \text{[Eq. 4.7]}$$

The indices $i$, $m$, $j$, and $k$ allow permutation through all elements of $\mathbf{M}$ and the eigenvalues.

The echo modulation amplitude $\mathrm{E}(\tau)$ is formulated as follows [33, 145, 158]:

$$\mathrm{E}(\tau) = \sum_{l,m,j,k} A_{\mathrm{lmjk}} \mathrm{e}^{\mathrm{i}\omega_{\mathrm{lmjk}}\tau} \qquad \text{[Eq. 4.8]}$$

For the three-pulse echo modulation (pulse sequence: $\pi/2 - \tau - \pi/2 - T - \pi/2 - \tau -$ *detection*), the frequencies $\omega$ and amplitudes $A$ are separated into terms representing the $\alpha$ and $\beta$ manifolds, as follows:

$$\omega_{\alpha\mathrm{jk}} = 2\pi(\nu_{\alpha\mathrm{j}} - \nu_{\alpha\mathrm{k}})$$

$$\omega_{\beta\mathrm{ml}} = 2\pi(\nu_{\beta\mathrm{m}} - \nu_{\beta\mathrm{l}})$$

$$A_{\alpha\mathrm{jk}} = \sum_{m,l} \mathrm{M}_{\mathrm{jm}} \mathrm{M}^{\dagger}_{\mathrm{mk}} \mathrm{M}_{\mathrm{kl}} \mathrm{M}^{\dagger}_{\mathrm{lj}} \mathrm{e}^{\mathrm{i}(\omega_{\alpha\mathrm{jk}} + \omega_{\beta\mathrm{ml}})\tau}$$

$$A_{\beta\mathrm{ml}} = \sum_{k,j} \mathrm{M}_{\mathrm{lj}} \mathrm{M}^{\dagger}_{\mathrm{kl}} \mathrm{M}_{\mathrm{mk}} \mathrm{M}^{\dagger}_{\mathrm{jm}} \mathrm{e}^{\mathrm{i}(\omega_{\alpha\mathrm{jk}} + \omega_{\beta\mathrm{ml}})\tau} \qquad \text{[Eq. 4.9]}$$

The simulated echo modulation amplitude $\mathrm{E}(\tau, T)$ is formulated as follows [33, 145, 158]:

$$\mathrm{E}(\tau, T) = \sum_{j,k} (A_{\alpha\mathrm{jk}} \mathrm{e}^{\mathrm{i}\omega_{\alpha\mathrm{jk}}T} + A_{\beta\mathrm{ml}} \mathrm{e}^{\mathrm{i}\omega_{\beta\mathrm{ml}}T}) \qquad \text{[Eq. 4.10]}$$

For flexibility, users can easily substitute their own time-domain ESEEM calculations into OPTESIM as MATLAB functions. Therefore, experimental systems and pulse sequences other than those considered here [33, 145, 158] can be addressed by using the optimization and statistical assessment features of OPTESIM.

***Powder average for disordered systems: General and geometry-preserving treatments***

For a disordered, or powder, system, the ESEEM represents the average over all possible orientations of the external magnetic field vector $B_0$, with respect to the molecular PAS [33, 145, 157, 158]. The number of orientations of $B_0$ relative to the molecular PAS that can be evaluated is practically limited by the available computational power (time scale of the calculations) and the required accuracy. The orientation sampling vectors can be chosen as either a set of random unit vectors, or a set of evenly-distributed points on a unit sphere, or octant, if symmetry permits. Several methods have been used in the latter cases [159, 160]. In OPTESIM, a geodesic-sphere sequence generator evenly distributes vectors on a unit sphere. The generator subdivides each of the triangular faces of an icosahedron into four smaller triangles on a unit sphere in an iterative manner, yielding geodesic-sphere sequences at different levels of angular resolution. Users of OPTESIM also have the option to specify their own orientation vectors, as well as the weights of particular subsets of vectors. This approach is used for simulating orientation-selection experiments, in cases where the assumption of an isotropic *g*-tensor cannot be made [30, 161].

The overall envelope modulation when multiple nuclei are coupled to the electron is combined according to the product rule. Specifically, for the two-pulse sequence, the modulation is as follows [162],

$$\mathrm{E}_{\mathrm{tot}}(\tau) = \prod_{\mathrm{i}} \mathrm{E}_{\mathrm{i}}(\tau) \qquad \text{[Eq. 4.11]}$$

and for the three-pulse sequence, the product is taken separately over the $\alpha$ and $\beta$ manifolds, and the overall envelope modulation is combined, as follows [163],

$$\mathrm{E}_{\mathrm{tot}}(\tau,T)=\prod_{\mathrm{i}}\mathrm{E}_{\alpha\mathrm{i}}(\tau,T)+\prod_{\mathrm{i}}\mathrm{E}_{\beta\mathrm{i}}(\tau,T) \qquad \text{[Eq. 4.12]}$$

In the case of electron spin coupling with multiple nuclei, it is computationally expedient to perform the powder, or spherical, average of individual electron–nuclear interactions prior to combination of the individual $\mathrm{E}_{\mathrm{tot}}(\tau)$, or $\mathrm{E}_{\alpha,\mathrm{i}}(\tau,T)$ and $\mathrm{E}_{\beta,\mathrm{i}}(\tau,T)$, modulation terms. This method implicitly assumes that the hf PAS of all electron–nuclear interactions are coincident. However, coincidence of all hf PAS is not generally the case, and information about the mutual orientation of the nuclei, which is contained in the order≥2 modulation product terms, is lost with this treatment. In OPTESIM, the combination rule is specified at the level of a physical model of the geometry of electron–nuclear interactions, as defined by Euler angles that specify the mutual orientation(s) of the different hf PAS. Therefore, the ESEEM is calculated and combined at each powder average orientation, and subtle structural information arising from the geometry of the multiple hf PAS can be retrieved from the overall averaged spectrum.

***Optimization and statistical analysis of experimental parameters***

Optimization is achieved by fitting the simulated spectrum ( $\tilde{Y}$ ) to the experimental one $(Y)$ by using the minimization of the least-squares residuals, which are a function of the parameters ($\vartheta=[\theta_1,\theta_2,\cdots,\theta_{\mathrm{L}}]$) of the Hamiltonian. The goodness of least-squares fitting for one spectrum is defined as $Q^2=\sum_i(Y_{\mathrm{i}}-\tilde{Y}_{\mathrm{i}})^2$. For a global fitting

of multiple spectra with different experimental parameters, $Q^2$ for the individual spectrum is weighted by its own noise level. The global goodness of fitting is defined as:

$$Q^2 = \sum_k \sum_{i=1}^{n_k} (Y_{k,i} - \tilde{Y}_{k,i})^2 / {\sigma_k}^2 \qquad \text{[Eq. 4.13]}$$

where ${\sigma_k}^2$ and $n_k$ are the noise variance and number of points in spectrum $k$, respectively. We assume that the experimental envelope modulation is free of artifacts (for example microwave phase drift or "glitches" above the rms noise level), which is the case in our experiments [38, 39, 164], and that the noise variance of each point in the ESEEM waveform is identical (dead time points removed). In this normalized representation and with the two assumptions, the variance of the simulation error of all data points $(Y_{k,i})$ is estimated as follows:

$$S^2 = Q^2_{\min} / (N - L) \qquad \text{[Eq. 4.14]}$$

where $Q^2_{min}$ is the optimized $Q^2$ minimum at the global minimum, $N$ is the total number of experimental points, as given by $N = \sum_k n_k$, and $L$ is the number of independent parameters for optimization. $Q^2(\vartheta)$ can be expanded at $\hat{\vartheta}$, as:

$$Q^2(\vartheta) = Q^2_{\min} + \frac{1}{2} \sum_{i,j} \left( \frac{\partial^2 Q^2}{\partial \vartheta_i \partial \vartheta_j} \right)_{\vartheta = \hat{\vartheta}} \left(\vartheta_i - \hat{\vartheta}_i\right)\left(\vartheta_j - \hat{\vartheta}_j\right) + \cdots \qquad \text{[Eq. 4.15]}$$

where $\hat{\vartheta}$ are the optimal parameters when $Q^2\left(\hat{\vartheta}\right) = Q^2_{min}$. Assuming that the terms with order >2 are negligible, and that the variances of the measurement errors follow a normal distribution, then the term:

$$a = \frac{1}{2S^2}\sum_{i,j}\left(\frac{\partial^2 Q^2}{\partial\vartheta_i \partial\vartheta_j}\right)_{\vartheta=\hat{\vartheta}} \qquad \text{[Eq. 4.16]}$$

follows a $\chi^2$ distribution with a freedom of $L$ [165]. The simultaneous confidence region of parameters $\vartheta$ can be determined by the following expression:

$$Q^2\left(\vartheta\right) = Q^2_{\min} + aS^2 \qquad \text{[Eq. 4.17]}$$

The simultaneous confidence region gives an estimation of the uncertainty of a selected optimization parameter with respect to the other parameters. The probability of the simultaneous confidence region containing the true values of $\vartheta_i$ is determined by $\chi^2(L)$ and $a$, as a cumulative distribution of $\chi^2(L)$ from $-\infty$ to $a$. For example, when $L = 2$ and $a = 1^2$, $2^2$, or $3^2$, the simultaneous confidence probability is 39.3%, 86.5%, or 98.9%.

### § 4.2.2 OPTESIM toolbox

The OPTESIM software consists of twenty-eight MATLAB functions and eight Java classes that can be divided into the following four categories: experimental data filtering, numerical simulation, simulation parameter optimization, and distributed

computation framework. These categories are incorporated into four stand-alone modules that are integrated in OPTESIM, as described below. The modules may be individually substituted by the routines of other users, if desired. Table 4.1 shows a code fragment example of OPTESIM.

***Experimental data filtering***

Experimental data filtering allows extraction of experimental parameters from saved experimental data files and processing of the modulation amplitude waveform for simulation analysis. For two-pulse ESEEM, the extracted experimental parameters are experiment type, $\tau$ value, and $B_0$ field. For three-pulse ESEEM, the additional experimental parameter, $T$, is extracted. An example data filtering routine that is compatible with our experimental control software is provided with the package. The ESEEM waveform is then processed, as follows. If necessary, baseline decay of the echo envelope, owing to spin-lattice ($T_1$) or spin-spin ($T_2$) relaxation processes, is removed by using an empirical decay function. This is accomplished by fitting the waveform to a user specified monoexponential, biexponential, or stretched exponential function, and then subtracting the decay function from the experimental waveform. Any points recorded within the spectrometer deadtime are removed. The deadtime portion of the waveform is reconstructed as part of the optimization during the simulation. The average noise variance ($\sigma^2$) of the processed waveform (minus the deadtime fill points) is calculated.

**Table 4.1. A code fragment example of OPTESIM.**

```
%Load experimental spectrum
spectrum = filterRaw(filename, filter_options);

%Define a model of one deuterium and one proton with a product rule
optmodel = eseOptModel({'H2','H1'}, '1*2');

%Set the nulcear parameters for Nucleus 1
optmodel = changeOptModelValue(optmodel, 'eeqQ', 1, 0.2, 'eta', 1, 0.1);

%Set the nulcear parameters for Nucleus 2
optmodel = changeOptModelValue(optmodel, 'eeqQ', 2, 0.2, 'eta', 2, 0.1);

%Register variables for Optimization
%Parameter, nucleus number, index, min value, max value
optmodel = registerOptModelVariable(optmodel,...
   'Aiso', 1, 1, -10, 10, ...
   'r'   , 1, 2, 0, 10, ...
   'Aiso', 2, 3, -10, 10, ...
   'r'   , 2, 4, 0, 10);

%Set initial values for Aiso1, r1, Aiso2, r2, respectively
x = [6.5, 2.5, 1.5, 2.5];

%Use simplex methods for optimization
options = optimset('OutputFcn', @esePlot, 'Display', 'iter');
x_opted = fminsearch(@eseOptObjective, x, options);
```

### *Numerical Simulation*

Numerical simulations of ESEEM are performed for one or more electron-nuclear couplings with the identity of each nucleus supplied by the user, corresponding to the parameters: $g_N$ and $I$. This information can often be gleaned from a qualitative interpretation of the spectrum, or with the aid of multiple microwave frequency/$B_0$ experiments. In the case of unidentified nuclei or numbers of nuclei, the most likely models can be assessed from the statistical criteria in OPTESIM for separate simulations that incorporate different types and numbers of coupled nuclei. A model for the combination of the ESEEM from multiple nuclei is also specified. In the most simple

case of all nuclei coupled to the same electron spin, and a single nuclear isotope for each coupling, the envelope modulation for two-pulse and three-pulse experiments is combined according to the product rule expressions in Eq. 11 and Eq. 12, respectively. When the experimental ESEEM is known to arise from sub-populations of electron spins with different nuclear coupling(s), the "sum rule" [162, 166] for combination of the individual ESEEM is used, with a weighting factor for each population, to generate the total ESEEM. Practical cases where the sum rule and weighting are necessary are incomplete or heterogeneous isotopic substitution of a nuclear site, conformational isomers of the same paramagnetic molecule, and different paramagnetic molecules. OPTESIM allows users to specify the particular combination rule by using a numerical expression. For example, in the case of two-pulse ESEEM from a molecule with uniform isotope content at coupled nuclear site 1, and heterogenous isotopic content at coupled nuclear site 2, the combination of the individual modulation is specified by "$E_1$*[$C_{2,a}$*$E_{2,a}$+$C_{2,b}$*$E_{2,b}$]", where the $E_i$ are the envelope modulation and the $C_{i,a/b}$ are the normalized isotope substitution factors at site 2. The weighting can be included in the simulation as an adjustable parameter. In cases of three-pulse experiments, this combination is done by separately combining the modulation from the $\alpha$ and $\beta$ manifolds (Eq. 12).

The calculation of the powder average ESEEM is achieved by using the following steps: (1) The user defines the nuclear parameters for each coupled nucleus and the combination rule. (2) The Hamiltonians for each electron-nuclear coupling are calculated for one orientation of the external magnetic field. (3) The modulation envelopes for each nuclear coupling are calculated based on the Hamiltonians calculated in step 2. (4) The

modulation envelopes are combined to attain the overall spectrum for the one orientation. (5) Steps 2-4 are repeated until the spectra include averaging over all orientations of $B_0$ relative to the reference PAS.

***Optimization***

The simulated spectrum is matched to the filtered experimental spectrum after the following transformation,

$$\tilde{Y} = c_1\left(\hat{Y} + c_2\right)\mathrm{e}^{c_3 t} + c_4 \qquad \text{[Eq. 4.18]}$$

where $\tilde{Y}$ and $\hat{Y}$ are the transformed and original simulation envelope modulation, respectively. The four real constants $c_{1-4}$ are calculated by minimizing the squared residuals of the match between the transformed amplitude and the filtered experimental data. The $c_i$ are defined, as follows: $c_1$ is the scaling factor for the simulated waveform, $c_3$ compensates for modulation amplitude decay, and $c_2$ and $c_4$ are offset terms for adjusting the waveform prior to, or subsequent to, the decay operation. The scaling constants, $c_1$, $c_2$, and $c_4$, in Eq. 4.18 are used when the target ESEEM originates from a sample in which the contributions to the ESEEM from nuclear couplings, other than the target couplings of the simulation, are not known. The minimum squared residual is set as the objective function for optimization over the adjustable parameters. For one model of number and type of nuclear couplings and method of combination, users can specify multiple experimental spectra under different experimental conditions (for example, $\tau$ value, $B_0$ value for the same $g_e$ value) for a global optimization, with the fitting objective function defined as in Eq. 13. A standardized definition of the objective function in

OPTESIM allows users to take full advantage of available MATLAB optimization routines.

In our experience, owing to the large dimension of the parameter space and the rugged landscape of the objective function, gradient-based local optimization methods often fail to find the set of global optimal parameters because of the existence of many local minima, and non-gradient-based optimization methods require extensive computation power. Therefore, the computation speed becomes the bottleneck for the scale and accuracy of optimizations. One solution is to use faster C/C++ code for the core calculations that are crucial to computational speed. In our test, using a compiled C++ executable (MEX-files) for the modulation envelope calculation in MATLAB resulted in a decrease of up to 1.5-fold in the time of the overall simulation. Another solution is to distribute the computation of the powder averaging to the available computational resources on a local network.

***Distributed computing framework***

A Java-RMI based distributed computational framework is included in OPTESIM. This framework allows users to build a distributed system of ESEEM simulation on their own PC computer hardware resources. The orientation sampling vectors for the powder average are first divided on the local computer and sent to different remote PCs according to their computational speed. The results are then collected and averaged on the local PC. A linear increase in computation speed with respect to the number of PCs can be achieved with this approach.

## § 4.3 Applications of OPTESIM

### § 4.3.1 Global simulation of ESEEM from a single $^{14}$N nucleus in an enzyme radical reaction intermediate

In three-pulse ESEEM studies of the cryotrapped 2-aminopropanol-generated substrate radical catalytic intermediate in the active site of coenzyme $B_{12}$-dependent EAL from *Salmonella typhimurium*, a $^{14}$N nucleus was found to be coupled to the substrate radical [35, 167]. The narrow features in Fourier transforms of the three-pulse $^{14}$N ESEEM at 0.8-0.9, 1.7-2.0 and 2.9-3.0 MHz correspond approximately to the $\nu_0$, $\nu_-$ and $\nu_+$ nqi frequencies of $^{14}$N ($\alpha$ electron spin manifold for $A_{iso}>0$), and the broad feature positioned around 4 MHz is the $\Delta m_{I,\beta}=\pm 2$, or "double quantum", feature. This spectral pattern is characteristic of the "exact cancellation" condition, in which the hf coupling and nuclear Zeeman contributions approximately cancel in one electron spin manifold [139, 168]. The coupled $^{14}$N has been assigned to a nitrogen belonging to the guanidinium group of Arg160 in the EutB subunit of EAL [5, 164]. Arg160 makes hydrogen bond contact with the substrate in the active site [164]. Multi-frequency three-pulse ESEEM spectra obtained at $g$=2.02 for the substrate radical were analyzed by using OPTESIM for a global optimization over the $^{14}$N nuclear parameters. Figure 4.1 shows the global fitting in both time and frequency domains at two different $\tau$ values. Table 4.2 lists the optimal parameters found for the simulation and their simultaneous trusted regions at the confidence level of 0.99. The simulation with hybrid optimization converged to the parameters presented in Table 2 in approximately 1.5 hours.

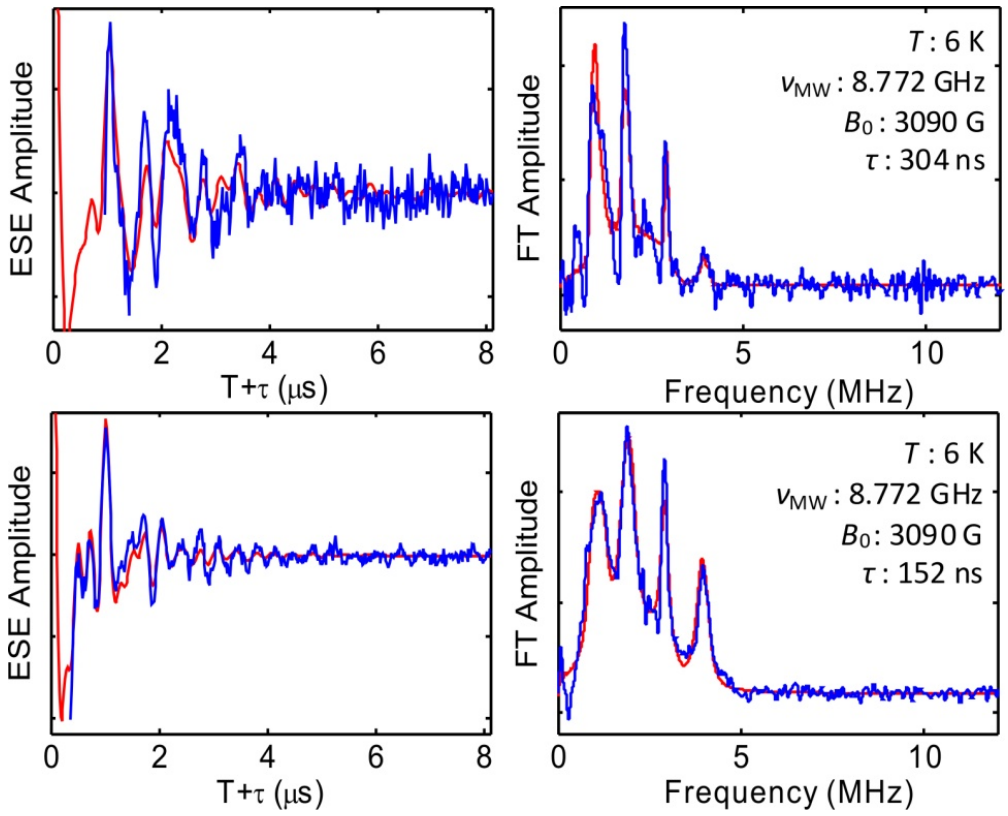


**Figure 4.1. Global simulation of ESEEM from a single $^{14}$N nucleus.** Figure is showing $^{14}$N ESEEM collected from the substrate radical intermediate in ethanolamine ammonia-lyase, and Fourier transform, and simulation (red line) representing a global optimization. Top: $\tau$=304 ns. Bottom: $\tau$=152 ns. Simulation parameters are listed in Table 2. *Conditions*: Microwave frequency, 8.772 GHz; magnetic field, 3090 Gauss; temperature, 6 K; microwave pulse width, 20 ns.

**Table 4.2. Simulation parameters for ESEEM from the coupled $^{14}$N in the substrate radical intermediate in ethanolamine ammoia-lyase.** Euler angles correspond to the relative orientation of the hf and nqi PAS.

| Nuclear Parameters | Values | Simultaneous Confidence (0.99) Region |
|---|---|---|
| $A_{iso}$ (MHz) | 0.933 | 0.910 – 0.958 |
| $r_{eff}$ (Å) | 3.90 | 3.63 – 4.34 |
| $e^2qQ/h$ (MHz) | 3.075 | 3.068 – 3.084 |
| $\eta$ | 0.579 | 0.568– 0.590 |
| $\alpha$ (deg) | 30.1 | 27.4 – 36.7 |
| $\beta$ (deg) | 60.1 | 55.5 – 64.9 |
| $\gamma$ (deg) | 14.7 | 6.9 – 39.0 |

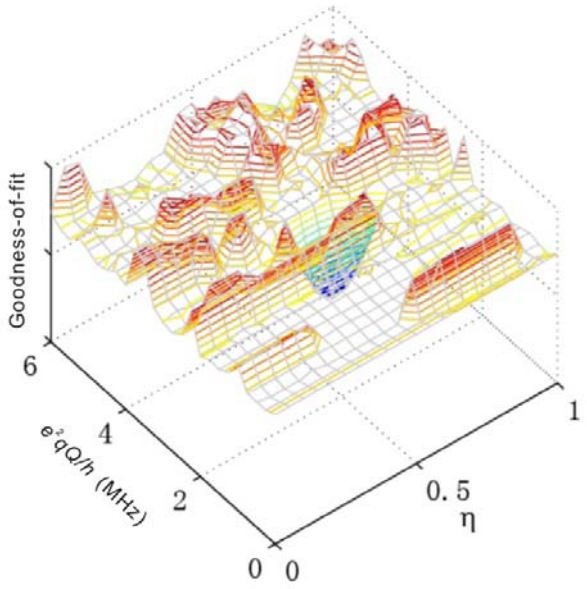


**Figure 4.2. Goodness-of-fit landscape for the parameters $e^2qQ/h$ and $\eta$.** Other simulation parameters are fixed to their optimal values, as listed in Table 4.2.

Figure 4.2 shows a goodness-of-fit landscape plot, which is constructed by varying the $e^2qQ/h$ and $\eta$ values from 0 to 6 MHz and 0 to 1, respectively. This plot represents a simpler optimization problem over $e^2qQ/h$ and $\eta$ with all other parameters fixed at their optimized values (Table 2). The roughness of the goodness-of-fit landscape prevents local optimization methods, such as conjugated gradient and Nelder-Mead simplex, from finding the global minimum with outlying starting points.

Figure 4.3A demonstrates the failure of the Nelder-Mead simplex method with random starting points, except for the initial parameter set that is close to the global minimum. In contrast, the non-local genetic algorithm converges to the global minimum within 20 generations of a population size of 50, as shown in

Figure **4.3**B. However, a larger scale genetic algorithm with full parameter optimization requires a larger population size and generation numbers. Our experience shows that an efficient optimization of 10 parameters requires a population size of about 200 for a thorough search of the parameter space. The algorithm termination condition can be set to a specific number of stale generations [169], and the best individuals will condense at the global minimum in approximately 50 generations. With a hybrid optimization approach, the best individual generated by the genetic algorithm serves as the starting point for further local optimization methods with fast convergence (for example, simplex).

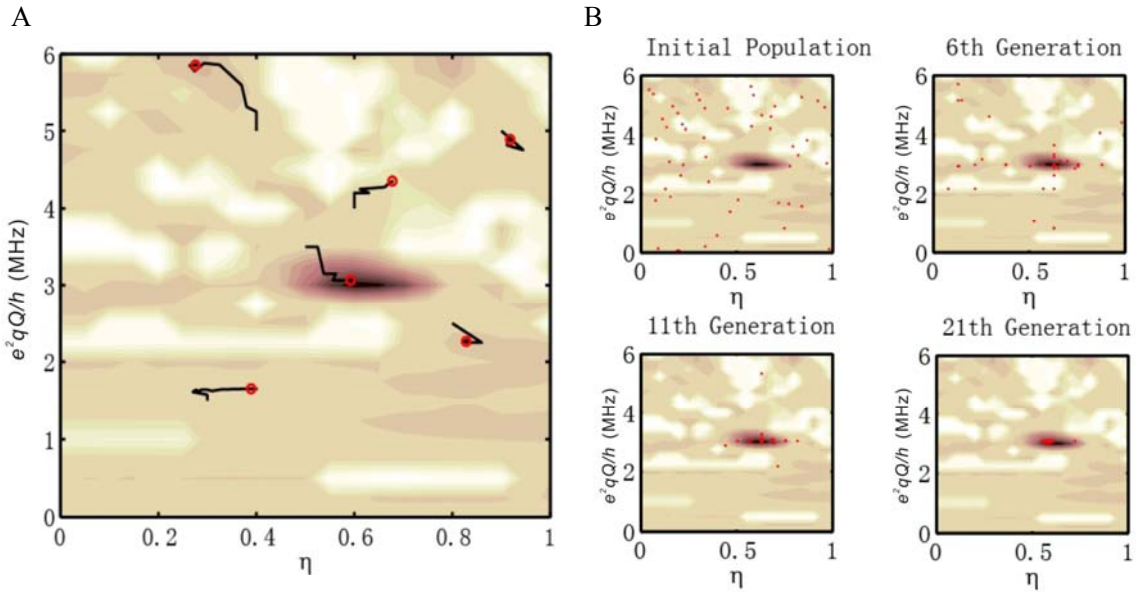


**Figure 4.3. Trajectories of the goodness-of-fit obtained by using simplex and genetic optimization algorithms. A.** Trajectories of the optimization for the Nelder-Mead simplex method on $e^2qQ/h$ and $\eta$. Red circles represent the ending points of the trajectories, which correspond to local minima. The simplex method only succeeds in finding the global minimum when the starting point is close. **B.** Optimization trajectory for the genetic algorithm. Red dots represent individuals in a population of 50.

The statistical assessment quantifies the importance of individual simulation parameters in reproducing the ESEEM. For discrete systems (single structure), a narrow confidence region indicates that a parameter contributes significantly to the ESEEM. For example, in the case of the $^{14}$N ESEEM in Figure 4.1, $e^2qQ/h$ and $\eta$ directly influence the frequency values of the narrow, large-amplidue $\nu_0$, $\nu_-$, and $\nu_+$ components, and $A_{iso}$ influences the center frequency value of the $\nu_{dq}$ component. These parameters therefore have relatively narrow simultaneous confidence regions (Table 4.2). The $r_{eff}$ parameter influences the width of the $\nu_{dq}$ feature, and therefore has a relatively wider confidence region. The Euler angles in Table 4.2 subtly modify the relative amplitudes of the $\nu_0$, $\nu_-$, and $\nu_+$, and $\nu_{dq}$ features, and have correspondingly wider confidence regions. The relatively narrow interval for Euler angle, $\beta$, in Table 4.2, and wider confidence regions for $\alpha$ and $\gamma$, is consistent with the choice of an axial dipolar (*versus* rhombic) hf tensor. If the influence of a parameter on the ESEEM is within the experimental noise, then the simultaneous confidence region would approach the full allowed range (for example, -180° to 180° for an angular parameter). The statistical assessment is useful for culling unnecessary parameters, which may reduce computation time.

### § 4.3.2 Mutual orientation of two $^{14}$N hf PAS in Cu(II)-bis-histamine complex

The Cu(II)-bis-histamine complex, Cu(II)(him)$_2$(NO$_3$)$_2$, is a model for bis-*trans* equatorial coordination of Cu(II) by imidazole in proteins. The structure of the complex in Cu(II)-doped Zn(II)(him)$_2$(NO$_3$)$_2$ crystal has been determined by X-ray crystallography [170], and superhyperfine (shf) coupling parameters for the remote $^{14}$N of imidazole have

been calculated by using experimentally constrained density functional theory [171]. The $Cu(II)(him)_2(NO_3)_2$ complex was synthesized as described [170], and a frozen 5 mM sample was prepared in 1:1 ethylene glycol/water glass in a 0.4 cm outer diameter EPR tube. Three-pulse ESEEM was collected at the $B_0$ value for maximum ESE amplitude at $g$=2.069 in this disordered sample. In the special case of this nominally octahedral Cu(II) complex, all orientations of the $g$-tensor can be assumed to contribute to the ESE at this $g_\perp$ position of the EPR spectrum [172]. Powder simulation of the ESEEM leads to determination of the mutual orientation of the two remote imidazole $^{14}N$ hyperfine PAS, based on reproduction of subtle spectral features, which exemplifies the efficiency of OPTESIM.

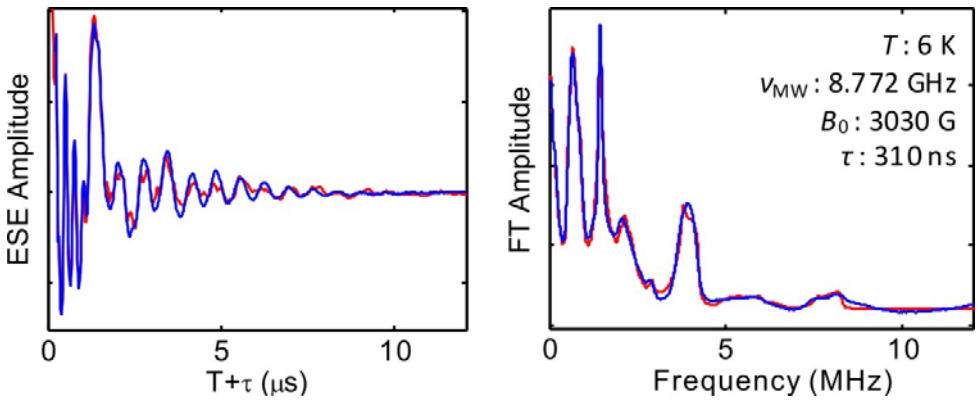


**Figure 4.4. ESEEM collected from the remote $^{14}N$ imidazole nuclei in the Cu(II)-*bis*-histamine complex and Fourier transform, and simulation (red).** The simulation parameters are listed in Table 3. *Conditions*: Microwave frequency, 8.772 GHz; magnetic field, 3030 Gauss; temperature, 6 K; microwave pulse width, 20 ns; $\tau$=310 ns.

Figure 4.4 shows the converged simulation of three-pulse ESEEM collected from the Cu(II)-bis-histamine complex. The simulation parameters are presented in Table 4.3. The simulation was achieved by using a sequential application of the genetic algorithm ($10^4$ evaluations) and Nelder-Mead simplex ($10^3$ iterations) algorithm. The simulation of the experimental ESEEM spectrum involved optimization on eleven shf and nqi parameters. Eight of the parameters, including the three principal components of the remote $^{14}$N hyperfine tensor, $e^2qQ/h$, $\eta$, and the three Euler angles that relate the shf and nqi PAS, are assumed to be common to the two $^{14}$N nuclei. A significant feature of the FT of the experimental ESEEM, which is enhanced at the selected $\tau$ value of 310 ns, is the spectral feature around 8 MHz that arises from combination of the $\Delta m_{I,\beta}=\pm 2$, or "double quantum" modulation from the two coupled $^{14}$N. The line shape of this $\Delta m_{I,\beta}=\pm 2$ combination line is sensitively dependent on the mutual orientation of the two remote imidazole $^{14}$N shf PAS. This was accounted for in the simulations by varying Euler angles that relate the two remote imidazole $^{14}$N shf PAS as part of the optimization. The values of these Euler angles cause extremely subtle changes in the overall ESEEM amplitude, and the parameter space has a large dimension. Therefore, manual fitting and gradient-based optimization methods are exceptionally time-consuming and impractical.

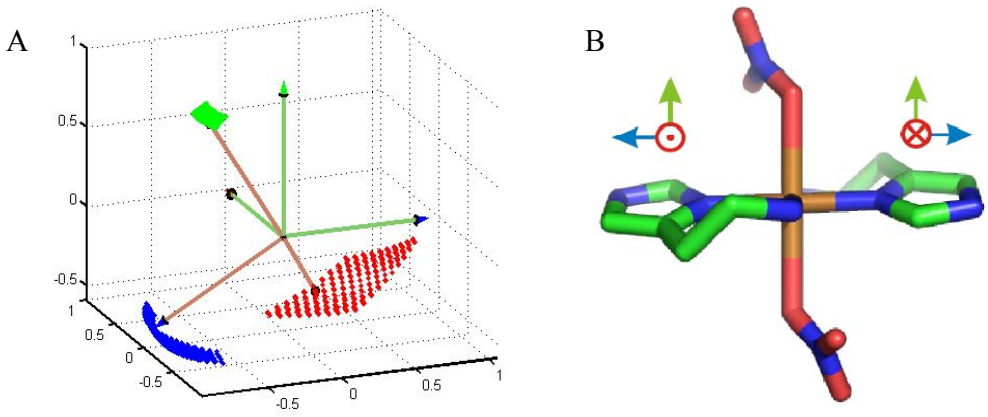


**Figure 4.5. Physical model for the mutual orientation of the imidazole remote $^{14}$N hf PAS in the Cu(II)-bis-histamine complex, and X-ray crystallographic structure of Cu(II)(him)$_2$(NO$_3$)$_2$.** **A.** Best-fit mutual orientation of the hf PAS of the imidazole remote $^{14}$N nuclei, obtained by using OPTESIM. The dotted surfaces represent the simultaneous confidence regions of the axes, at level 0.99. **B.** X-ray crystallographic structure of Cu(II)(him)$_2$(NO$_3$)$_2$.[170] Proposed Cartesian axes that correspond to the directions in (A) are shown.

Optimization using the "hybrid" approach with a population size of 500 converged in about 10 hours on the distributed network of fourteen Pentium 4 PCs. Table 4.3 displays the best fit values and simultaneous confidence intervals of the simulation parameters. The shf coupling tensor, nqi parameters, and Euler angles that relate the shf and nqi PAS are in good agreement with values reported for ESEEM-detected imidazole remote $^{14}$N coupling in other Cu(II)-imidazole complexes [173, 174]. The Euler angles for the relation of the two $^{14}$N PAS in Table 4.3, $\alpha$=177°, $\beta$=32.5°, and $\gamma$= -6.7°, indicate that the *x*- and *y*-axes are approximately opposed, and that the *z*-axes are approximately collinear, as depicted in Figure 4.5A. The relatively high symmetry of the simulated mutual orientation of the remote $^{14}$N shf tensors is consistent with the symmetry in the X-

ray crystallographic structure of the Cu(II)$(him)_2(NO_3)_2$ complex [170], which is shown in Figure 4.5B.

**Table 4.3. Simulation parameters for ESEEM from the remote $^{14}$N imidazole nuclei in Cu(II)-bis-histamine complex.** Euler angles, $\alpha$, $\beta$, and $\gamma$, correspond to the relative orientation of the hf and nqi PAS, and Euler angles $\alpha'$, $\beta'$, and $\gamma'$, correspond to the relative orientation of the two $^{14}$N hf PAS.

| Nuclear Parameters | Values | Simultaneous Confidence (0.99) Region |
|---|---|---|
| $A_{xx}$ (MHz) | 2.13 | 2.10 – 2.15 |
| $A_{yy}$ (MHz) | 1.81 | 1.77 – 1.84 |
| $A_{zz}$ (MHz) | 0.854 | 0.822 – 0.886 |
| $e^2qQ/h$ (MHz) | 1.51 | 1.50 – 1.52 |
| $\eta$ | 0.668 | 0.653 – 0.683 |
| $\alpha$ (deg) | 7.28 | -1.35 – 16.42 |
| $\beta$ (deg) | -9.34 | -11.12 – -7.33 |
| $\gamma$ (deg) | -9.41 | -24.02 – 6.01 |
| $\alpha'$ (deg) | 177.0 | 168.3 – 200.9 |
| $\beta'$ (deg) | 32.5 | 29.6 – 35.5 |
| $\gamma'$ (deg) | -6.7 | -17.9 – 16.6 |

# References


1. Bandarian, V. and G.H. Reed, *Ethanolamine Ammonia-Lyase*, in *Chemistry and Biochemistry of B12*, R. Banerjee, Editor. 1999, John Wiley and Sons: New York. p. 811-833.
2. Toraya, T., *Radical catalysis in coenzyme B-12-dependent isomerization (eliminating) reactions.* Chemical Reviews, 2003. **103**(6): p. 2095-2127.
3. Bradbeer, C., *Clostridial Fermentations of Choline and Ethanolamine .I. Preparation and Properties of Cell-Free Extracts.* Journal of Biological Chemistry, 1965. **240**(12): p. 4669-&.
4. Faust, L.R., et al., *Cloning, sequencing, and expression of the genes encoding the adenosylcobalamin-dependent ethanolamine ammonia-lyase of Salmonella typhimurium.* J Biol Chem, 1990. **265**(21): p. 12462-6.
5. Sun, L. and K. Warncke, *Comparative model of EutB from coenzyme B-12-dependent ethanolamine ammonia-lyase reveals a beta(8)alpha(8), TIM-barrel fold and radical catalytic site structural features.* Proteins-Structure Function and Bioinformatics, 2006. **64**(2): p. 308-319.
6. Faust, L.P. and B.M. Babior, *Overexpression, purification, and some properties of the AdoCbl-dependent ethanolamine ammonia-lyase from Salmonella typhimurium.* Arch Biochem Biophys, 1992. **294**(1): p. 50-4.
7. Buckel, W. and B.T. Golding, *Glutamate and 2-methyleneglutarate mutase: from microbial curiosities to paradigms for coenzyme B12-dependent enzymes.* Chem. Soc. Rev., 1996. **25**: p. 329-338.
8. Banerjee, R., *Chemistry and Biochemistry of B12*. 1999, New York: John Wiley and Sons.
9. Brown, K., *Chemistry and Enzymology of Vitamin B12.* Chem. Rev., 2005. **105**: p. 2075-2149.
10. Barker, H.A., et al., *Glutamate Mutase System - Assays + Properties.* Journal of Biological Chemistry, 1964. **239**(10): p. 3260-&.
11. Moore, B.S., et al., *On the Stereospecificity of the Coenzyme B-12-Dependent Isobutyryl-Coa Mutase Reaction.* Journal of the American Chemical Society, 1995. **117**(45): p. 11285-11291.
12. Pierik, A.J., et al., *Rotation of the exo-methylene Group of (R)-3-methylitaconate catalyzed by coenzyme B-12-dependent 2-methyleneglutarate mutase from Eubacterium barkeri.* Journal of the American Chemical Society, 2002. **124**(47): p. 14039-14048.
13. Banerjee, R. and S. Chowdhury, *Methylmalonyl-CoA Mutase*, in *Chemistry and Biochemistry of B12*, R. Banerjee, Editor. 1999, Wiley: New York. p. 707-730.
14. Marsh, E.N. and S.E. Harding, *Methylmalonyl-Coa Mutase from Propionibacterium-Shermanii - Characterization of the Cobalamin-Inhibited Form and Subunit Cofactor Interactions Studied by Analytical Ultracentrifugation.* Biochemical Journal, 1993. **290**: p. 551-555.
15. Toraya, T., *Radical Catalysis in Coenzyme B12-Dependent Isomerization (Eliminating) Reactions.* Chem. Rev., 2003. **103**: p. 2095-2127.

16. Shibata, N., et al., *A new mode of B12 binding and the direct participation of a potassium ion in enyzme catalysis: X-ray structure of diol dehydratase.* Structure, 1999. **7**: p. 997-1008.
17. Masuda, J., et al., *How a protein generates a catalytic radical from coenzyme B12: X-ray structure of a diol dehydratase-adeninylpentylcobalamin complex.* Structure, 2000. **8**: p. 775-788.
18. Yaminishi, M., et al., *The Crystal Structure of Coenzyme $B_{12}$-Dependent Glycerol Dehydratase in Complex with Cobalamin and Propane-1,2-Diol.* Eur. J. Biochem., 2002. **269**: p. 4484-4494.
19. Liao, D.-I., et al., *Crystal Structure of Substrate Free Form of Glycerol Dehydratase.* J. Inorg. Biochem., 2003. **93**: p. 84-91.
20. Stubbe, J. and W. van der Donk, *Protein radicals in enzyme catalysis.* Chem. Rev., 1998. **98**: p. 705-762.
21. Frey, P. and C.H. Chang, *Aminomutases*, in *Chemistry and Biochemistry of B12*, R. Banerjee, Editor. 1999, Wiley: New York. p. 835-858.
22. Chirpich, T.P., et al., *Lysine 2,3-Aminomutase - Purification and Properties of a Pyridoxal Phosphate and S-Adenosylmethionine-Activated Enzyme.* Journal of Biological Chemistry, 1970. **245**(7): p. 1778-&.
23. Wolthers, K.R., S.E.J. Rigby, and N.S. Scrutton, *Mechanism of Radical-based Catalysis in the Reaction Catalyzed by Adenosylcobalamin-dependent Ornithine 4,5-Aminomutase.* J. Biol. Chem., 2008. **283**(50): p. 34615-34625.
24. Kratky, C., et al., *Accurate Structural Data Demystify B-12 - High-Resolution Solid-State Structure of Aquocobalamin Perchlorate and Structure-Analysis of the Aquocobalamin Ion in Solution.* Journal of the American Chemical Society, 1995. **117**(16): p. 4654-4670.
25. Babior, B.M., et al., *The mechanism of action of ethanolamine ammonia-lyase, a B-12-dependent enzyme. The participation of paramagnetic species in the catalytic deamination of 2-aminopropanol.* J Biol Chem, 1974. **249**(14): p. 4537-44.
26. Retey, J., et al., *The stereochemistry of the reaction catalyzed by ethanolamine ammonia-lyase, an adenosylcobalamin-dependent enzyme.* J. Biol. Chem., 1974. **249**: p. 6359-6360.
27. Frey, P.A., *Importance of Organic Radicals in Enzymatic Cleavage of Unactivated C-H Bonds.* Chemical Reviews, 1990. **90**(7): p. 1343-1357.
28. Licht, S.S., C.C. Lawrence, and J. Stubbe, *Class II ribonucleotide reductases catalyze carbon-cobalt bond reformation on every turnover.* Journal of the American Chemical Society, 1999. **121**(33): p. 7463-7468.
29. Warncke, K., J.C. Schmidt, and S.C. Ke, *Identification of a rearranged-substrate, product radical intermediate and the contribution of a product radical trap in vitamin B-12 coenzyme-dependent ethanolamine deaminase catalysis.* Journal of the American Chemical Society, 1999. **121**(45): p. 10522-10528.
30. Canfield, J.M. and K. Warncke, *Geometry of reactant centers in the Co-II-substrate radical pair state of coenzyme B-12-dependent ethanolamine deaminase determined by using orientation-selection-ESEEM spectroscopy.* Journal of Physical Chemistry B, 2002. **106**(34): p. 8831-8841.

31. Bandarian, V. and G.H. Reed, *Analysis of the Electron Paramagnetic Resonance Spectrum of a Radical Intermediate in the Coenyzme B12-Dependent Ethanolamine Ammonia-Lyase Catalyzed Reaction of S-2-Aminopropanol.* Biochemistry, 2002. **41**: p. 8580-8588.
32. Ke, S.C., *Spin-spin interaction in ethanolamine deaminase.* Biochimica Et Biophysica Acta-General Subjects, 2003. **1620**(1-3): p. 267-272.
33. Schweiger, A. and G. Jeschke, *Principles of pulse electron paramagnetic resonance*. 2001, Oxford, UK ; New York: Oxford University Press. xxvi, 578 p.
34. Tan, S.L., et al., *Electron spin echo studies of the composition of the paramagnetic intermediate formed during the deamination of propanolamine by ethanolamine ammonia lyase, an AdoCbl-dependent enzyme.* J. Biol. Chem., 1986. **261**: p. 3483-3485.
35. Ke, S.C. and K. Warncke, *Interactions of substrate and product radicals with Co-II in cobalamin and with the active site in ethanolamine deaminase, characterized by ESE-EPR and N-14 ESEEM spectroscopies.* Journal of the American Chemical Society, 1999. **121**(43): p. 9922-9927.
36. Warncke, K. and A.S. Utada, *Interaction of the substrate radical and the 5'-deoxyadenosine-5'-methyl group in vitamin B(12) coenzyme-dependent ethanolamine deaminase.* J Am Chem Soc, 2001. **123**(35): p. 8564-72.
37. Warncke, K., *Characterization of the product radical structure in the Co(II)-product radical pair state of coenzyme B12-dependent ethanolamine deaminase by using three-pulse 2H ESEEM spectroscopy.* Biochemistry, 2005. **44**(9): p. 3184-93.
38. Canfield, J.M. and K. Warncke, *Active site reactant center geometry in the Co-II-product radical pair state of coenzyme B-12-dependent ethanolamine deaminase determined by using orientation-selection electron spin-echo envelope modulation spectroscopy.* Journal of Physical Chemistry B, 2005. **109**(7): p. 3053-3064.
39. Warncke, K., *Characterization of the product radical structure in the Co-II-product radical pair state of coenzyme B-12-dependent ethanolamine deaminase by using three-pulse H-2 ESEEM spectroscopy.* Biochemistry, 2005. **44**(9): p. 3184-3193.
40. Feher, G., *Observation of Nuclear Magnetic Resonances Via the Electron Spin Resonance Line.* Physical Review, 1956. **103**(3): p. 834-835.
41. Mancia, F., et al., *How coenzyme B12 radicals are generated: the crystal structure of methylmalonyl-coenzyme A mutase at 2 Å resolution.* Structure, 1996. **4**: p. 339-350.
42. Mancia, F. and P.R. Evans, *Conformational changes on substrate binding to methylmalonyl-CoA mutase and new insights into the free radical mechanism.* Structure, 1998. **6**: p. 711-720.
43. Mancia, F., G.A. Smith, and P.R. Evans, *Crystal structure of substrate complexes of methylmalonyl-CoA mutase.* Biochemistry, 1999. **38**: p. 7999-8005.
44. Reitzer, R., et al., *Glutamate mutase from Clostridium cochlearium: the structure of a coenzyme B12-dependent enzyme provides new mechanistic insights.* Structure, 1999. **7**: p. 891-902.

45. Gruber, K., R. Reitzer, and C. Kratky, *Radical Shuttling in a Protein: Ribose Pseudorotation Controls Alkyl-Radical Transfer in the Coenzyme $B_{12}$ Dependent Enzyme Glutamate Mutase.* Angew. Chem. Int. Ed. Engl., 2001. **40**: p. 3377-3380.
46. Sintchak, M.D., et al., *The Crystal Structure of Class II Ribonucleotide Reductase Reveals How an Allosterically Regulated Monomer Mimics a Dimer.* Nature Struct. Biol., 2002. **9**: p. 293-300.
47. Gerlt, J.A. and F.M. Raushel, *Evolution of function in $(\beta/\alpha)_8$-barrel enzymes.* Curr. Op. Chem. Biol., 2003. **7**(252-264).
48. Nicolet, Y. and C.L. Drennan, *AdoMet radical proteins-from structure to evolution-alignment of divergent protein sequences reveals strong secondary structure element conservation.* Nucleic Acids Res., 2004. **32**(13): p. 4015-4025.
49. Sophia, H.J., et al., *Radical SAM, a novel protein superfamily linking unresolved steps in familiar biosynthetic pathways with radical mechanisms: functional characterization using new analysis and information visualization methods.* Nucleic Acids Res., 2001. **29**: p. 1097-1106.
50. Frey, P., *Radical mechanisms of enzymatic catalysis.* Ann. Rev. Biochem., 2001. **70**: p. 121-148.
51. Jarrett, J.T., *The generation of 5'-deoxyadenosyl radicals by adenosylmethionine-dependent radical enzymes.* Curr. Opin. Chem. Biol., 2003. **7**: p. 174-182.
52. Chance, M.R. and e. al., *Structural genomics: A pipeline for providing structures for the biologist.* Prot. Sci., 2002. **11**: p. 723-738.
53. Faust, L.P., et al., *AdoCbl-dependent ethanolamine amino-lyase from Salmonella typhimurium.* J. Biol. Chem., 1990. **265**: p. 12462-12466.
54. Faust, L.P. and B.M. Babior, *Overexpression, purification and some properties of AdoCbl-dependent ethanolamine amino-lyase from Salmonella typhimurium.* Arch. Biochem. Biophys., 1992. **294**: p. 50-54.
55. Genomics, J.C.f.S., *Crystal structure of ethanolamine ammonia-lyase heavy chain (YP_013784.1) from Listeria monocytogenes 4b F2365 at 2.15 Å resolution,.* 2007.
56. Marti-Renom, M.A., et al., *Comparative protein structure modeling of genes and genomes.* Annual Review of Biophysics and Biomolecular Structure, 2000. **29**: p. 291-325.
57. Barker, W.C., et al., *The PIR-International Protein Sequence Database.* Nucleic Acids Research, 1999. **27**(1): p. 39-43.
58. Sali, A. and T.L. Blundell, *Comparative modelling by satisfaction of spatial constraints.* J. Mol. Biol., 1993. **234**: p. 779-815.
59. Laskowski, R.A., et al., *PROCHECK: A program to check the stereochemical quality of protein structures.* J. Appl. Cryst., 1993. **26**: p. 283-291.
60. Bowie JU, L.R., Eisenberg D, *A method to identify protein sequences that fold into a known three-dimensional structure.* Science, 1991. **253**(5016): p. 164-70.
61. Sippl, M.J., *Recognition of Errors in Three-Dimensional Structures of Proteins.* Proteins, 1993(17): p. 355-362.
62. Colovos C, Y.T., *Verification of protein structures: patterns of nonbonded atomic interactions.* Protein Sci, 1993. **2**(9): p. 1511-9.
63. Altscul, S.F., et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res., 1997. **25**: p. 3389-3402.

64. *Enzyme database - BRENDA url: http://www.brenda.uni-koeln.de/*. 1987 [cited 2004; Available from: http://www.brenda.uni-koeln.de/
65. Boeckmann, B., et al., *The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003*. Nucleic Acids Res., 2003. **31**: p. 354-370.
66. Warncke, K. and A.S. Utada, *Interaction of the Substrate Radical and the 5'-Deoxyadenosine-5'-Methyl Group in Vitamin B12 Coenzyme-Dependent Ethanolamine Deaminase*. J. Am. Chem. Soc., 2001. **123**: p. 8564-8572.
67. Canfield, J.M. and K. Warncke, *Geometry of Reactant Centers in the CoII-Substrate Radical Pair State of Coenzyme B12-Dependent Ethanolamine Deaminase Determined by using Orientation-Selection-ESEEM Spectroscopy*. J. Phys. Chem. B, 2002. **106**: p. 8831-8841.
68. Canfield, J.M. and K. Warncke, *Active Site Reactant Center Geometry in the CoII-Product Radical Pair State of Coenzyme B12-Dependent Ethanolamine Deaminase Determined by Using Orientation-Selection-ESEEM Spectroscopy*. J. Phys. Chem. B, 2005. **109**: p. 3053-3046.
69. Warncke, K., *Characterization of the Product Radical Structure in the CoII-Product Radical Pair State of Coenzyme B12-Dependent Ethanolamine Deaminase by using Three-Pulse 2H ESEEM Spectroscopy*. Biochemistry, 2005. **44**: p. 3184-3193.
70. Kurowski, M.A. and J.M. Bujnicki, *GeneSilico Protein Structure Prediction Meta-Server*. Nucleic Acids Res., 2003. **31**(13): p. 3305-3307.
71. Kelley, L.A., C.M. McCallum, and M.J. Sternberg, *Enhanced genome annotation using structural profiles in the program 3D-PSSM*. J. Mol. Biol., 2000. **299**: p. 501-522.
72. Rychlewski, L., et al., *Comparison of sequence profiles. Strategies for structural predictions using sequence information*. Protein Sci., 2000. **9**: p. 232-241.
73. Shi, J., T.L. Blundell, and K. Mizuguchi, *Fugue: Sequence-structure homology recognition using environmnent-specific substitution tables and structure-dependent gap penalties*. J. Mol. Biol., 2001. **310**: p. 243-257.
74. Jones, D.T., *GenTHREADER: An Efficient and Reliable Protein Fold Recognition Method*. J. Mol. Biol., 1999. **287**: p. 797-815.
75. Wallner, B., H. Fang, and A. Elofsson, *Automatic consensus based fold recognition using Pcons, ProQ and Pmodeller*. Proteins, 2003. **53**: p. 534-541.
76. Karplus, K., *HIdden Markov Models for Detecting Remote Protein Homologies*. Bioinformatics, 1998. **14**(10): p. 846-856.
77. Karplus, K., et al., *Combining local-structure, fold-recognition, and new-fold methods for protein structure prediction*. Proteins, 2003. **53**: p. 491-496.
78. Berkovitch, F., et al., *Crystal structure of biotin synthase, an S-adenosylmethionine-dependent radical enzyme*. Science, 2004. **303**: p. 76-79.
79. Layer, G., et al., *Crystal structure of coproporphyrinogen III oxidase reveals cofactor geometry of radical SAM enzymes*. EMBO J., 2003. **22**(6214-6224).
80. Chenna, R., Sugawara, Hideaki, Koike,Tadashi, Lopez, Rodrigo, Gibson, Toby J, Higgins, Desmond G, Thompson, Julie D., *Multiple sequence alignment with the Clustal series of programs*. Nucleic Acids Res, 2003. **31**(13): p. 3497-500.
81. Bateman, A., et al., *The Pfam Protein Families Database*. Nucleic Acids Res., 2004. **32**: p. D138-D141

82. Chenna, R., et al., *Multiple Sequence Alignment with the Clustal Series of Programs.* Nucleic Acids Res., 2003. **31**(13): p. 3497-3500.
83. Hollaway, M.R., et al., *The number of functional sites per molecule of the adenosylcobalamin-dependent enzyme, ethanolamine deaminase, as determined by a kinetic method.* Eur. J. Biochem., 1980. **111**: p. 177-188.
84. Bandarian, V. and G.H. Reed, *Hydrazine cation radical in the active site of ethanolamine ammonia-lyase: mechanism-based inactivation by hydroxyethylhydrazine.* Biochemistry, 1999. **38**: p. 12394-12402.
85. Lesk, A.M., C.-I. Branden, and C. Chothia, *Structural principles of α/β barrel proteins: The packing of the interior of the sheet.* Proteins, 1989. **5**: p. 139-148.
86. Gromiha, M.M., et al., *Locating stabilizing residues in (alpha/beta)8 barrel proteins based on hydrophobicity, long-range interactions, and sequence conservation.* Proteins, 2004. **55**(2): p. 316-329.
87. Sippl, M.J., J. Mol. Biol., 1990. **213**: p. 859-883.
88. Cheng, B., A. Nayeem, and H.A. Scheraga, J. Comp. Chem., 1996. **17**: p. 1453-1480.
89. MacKerell, A.D., Jr., et al., J. Phys. Chem. B, 1998. **102**: p. 3586-3616.
90. Luthy R, B.J., Eisenberg D, *Assessment of protein models with three-dimensional profiles.* Nature, 1992. **356**(6364): p. 83-5.
91. Sun, L. and K. Warncke, *Comparative model of EutB from coenzyme B12-dependent ethanolamine ammonia-lyase reveals a b8a8, TIM-barrel fold and radical catalytic site structural features.* Proteins: Structure, Function, and Bioinformatics, 2006. **64**(2): p. 308-319.
92. Holm, L. and J. Park, *DaliLite workbench for protein structure comparison.* Bioinformatics, 2000. **16**(6): p. 566-567.
93. Kamachi, T., T. Toraya, and K. Yoshizawa, *Catalytic Roles of Active-Site Amino Acid Residues of Coenzyme B12-Dependent Diol Dehydratase: Protonation State of Histidine and Pull Effect of Glutamate.* J. Am. Chem. Soc., 2004. **126**: p. 16207-16216.
94. Smith, D.M., B.T. Golding, and L. Radom, *Towards a consistent mechanism for diol dehydratase catalyzed reactions: An application of the partial-proton-transfer concept.* J. Am. Chem. Soc., 1999. **121**: p. 5700-5704.
95. Smith, D.M., B.T. Golding, and L. Radom, *Understanding the mechanism of B12-dependent diol dehydratase: A synergistic retro-push-pull proposal.* J. Am. Chem. Soc., 2001. **123**: p. 1664-1675.
96. Wetmore, S.D., et al., *Understanding the Mechanism of Action of B12-Dependent Ethanolamine Ammonia-Lyase: Synergistic Interactions at Play.* J. Am. Chem. Soc., 2002. **124**: p. 14054-14065.
97. Semialjac, M. and H. Schwartz, *Computational exploration of rearrangements related to the vitamin B12-dependent ethanolamine ammonia lyase catalyzed transformation.* J. Am. Chem. Soc., 2002. **124**: p. 8974-8983.
98. Semialjac, M. and H. Schwartz, *Computational study on mechanistic details of the aminoethanol rearrangement catalyzed by the vitamin B12-dependent ethanolamine ammonia lyase: His and Asp/Glu acting simultaneously as catalytic auxiliaries.* J. Org. Chem., 2003. **68**: p. 6967-6983.

99. Sandala, G.M., D.M. Smith, and L. Radom, *Divergent mechanisms of suicide inactivation for ethanolamine ammonia-lyase.* J. Am. Chem. Soc., 2005. **127**: p. 8856-8864.
100. Xia, L., D.P. Ballou, and E.N.G. Marsh, *Role of Arg100 in the Active Site of Adenosylcobalamin-Dependent Glutamate Mutase.* Biochemistry, 2004. **43**: p. 3238-3245.
101. Babior, B.M., et al., *The mechanism of action of ethanolamine ammonia lyase, a B12-dependent enzyme: The participation of paramagnetic species in the catalytic deamination of 2-aminopropanol.* J. Biol. Chem., 1974. **249**: p. 4537-4544.
102. Mancia, F., G.A. Smith, and P.R. Evans, *Crystal structure of substrate complexes of methylmalonyl-CoA mutase.* Biochemistry, 1999. **38**(25): p. 7999-8005.
103. Reitzer, R., et al., *Glutamate mutase from Clostridium cochlearium: the structure of a coenzyme B-12-dependent enzyme provides new mechanistic insights.* Structure, 1999. **7**(8): p. 891-902.
104. Xia, L., D.P. Ballou, and E.N.G. Marsh, *Role of Arg100 in the active site of adenosylcobalamin-dependent glutamate mutase.* Biochemistry, 2004. **43**(11): p. 3238-3245.
105. Harkins, T.T. and C.B. Grissom, *The Magnetic-Field Dependent Step in Bit Ethanolamine Ammonia-Lyase Is Radical-Pair Recombination.* Journal of the American Chemical Society, 1995. **117**(1): p. 566-567.
106. Ornstein, L. and B.J. Davis, *Disc Electrophoresis 1.Background and Theory.* Ann NY Acad Sci, 1964. **121**: p. 321-349.
107. Laemmli, U.K., *Cleavage of structural proteins during the assembly of the head of bacteriophage T4.* Nature, 1970. **227**(5259): p. 680-5.
108. Kim, J., J. Mao, and M.R. Gunner, *Are acidic and basic groups in buried proteins predicted to be ionized?* J. Mol. Biol., 2005. **348**: p. 1283-1298.
109. Kaplan, B.H. and E.R. Stadtman, *Ethanolamine deaminase, a cobamide coenzyme-dependent enzyme. I. Purification, assay, and properties of the enzyme.* J Biol Chem, 1968. **243**(8): p. 1787-93.
110. Holloway, M.R., et al., *A spectrophotometric rapid kinetic study of reactions catalyzed by coenzyme B12-dependent ethanolamine ammonia-lyase.* Eur. J. Biochem., 1978. **82**: p. 143-154.
111. Cornish-Bowden, A., *Fundamentals of enzyme kinetics. Butterworths: Boston*. 1981, Boston: Butterworths.
112. Fersht, A., *Structure and Mechanism is Protein Science*. 1999, New York: W. H. Freeman and Company.
113. LoBrutto, R., et al., *5'-Deoxyadenosine contacts the substrate radical intermediate in the active site of ethanolamine ammonia-lyase: 2H and 13C electron nuclear double resonance studies.* Biochemistry, 2001. **40**(1): p. 9-14.
114. Hollaway, M.R., et al., *The number of functional active sites per molecule of the adenosylcobalamin-dependent enzyme, ethanolamine ammonia-lyase, as determined by a kinetic method.* Eur J Biochem, 1980. **111**(1): p. 177-88.
115. Bandarian, V. and G.H. Reed, *Hydrazine cation radical in the active site of ethanolamine ammonia-lyase: mechanism-based inactivation by hydroxyethylhydrazine.* Biochemistry, 1999. **38**(38): p. 12394-402.

116. Ke, S.C., et al., *Identification of dimethylbenzimidazole axial coordination and characterization of (14)N superhyperfine and nuclear quadrupole coupling in Cob(II)alamin bound to ethanolamine deaminase in a catalytically-engaged substrate radical-Cobalt(II) biradical state.* Biochemistry, 1999. **38**(39): p. 12681-9.
117. Eaton, S.S., et al., *Metal Nitroxyl Interactions .29. Electron-Paramagnetic-Res Studies of Spin-Labeled Copper-Complexes in Frozen Solution.* Journal of Magnetic Resonance, 1983. **52**(3): p. 435-449.
118. Boas, J.F., et al., *Interpretation of Electron-Spin Resonance-Spectra Due to Some B12-Dependent Enzyme-Reactions.* Journal of the Chemical Society-Faraday Transactions Ii, 1978. **74**: p. 417-431.
119. Pilbrow, J.R., *EPR of B12-dependent enzyme reactions and related systems*, in *B12, Volume 1*, D. Dolphin, Editor. 1982, Wiley: New York. p. 431-462.
120. Press, W.H., et al., *Nemerical Recipes in C: The Art of Scientific Computing, 2nd ed.* 1992, New York: Cambridge University Press.
121. Lagarias, J.C., et al., *Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions.* SIAM Journal on Optimization, 1998. **9**(1): p. 112-147.
122. Britt, R.D. and M.P. Klein, *A Versatile Loop Gap Resonator Probe for Low-Temperature Electron Spin-Echo Studies.* Journal of Magnetic Resonance, 1987. **74**(3): p. 535-540.
123. Lin, C.P., M.K. Bowman, and J.R. Norris, *A Folded Half-Wave Resonator for Electron-Spin-Resonance Spectroscopy.* Journal of Magnetic Resonance, 1985. **65**(2): p. 369-374.
124. Dikanov, S.A. and Y.D. Tsvetkov, *Electron Spin Echo Envolop Modulation (ESEEM) spectroscopy*. 1992, Boca Raton, FL: CRC Press.
125. Deligiannakis, Y., M. Louloudi, and N. Hadjiliadis, *Electron spin echo envelope modulation (ESEEM) spectroscopy as a tool to investigate the coordination environment of metal centers.* Coordination Chemistry Reviews, 2000. **204**: p. 1-112.
126. Jeschke, G. and A. Schweiger, *Principles of Pulse Electron Paramagnetic Resonance*. 2001, New York: Oxford University Press.
127. Fauth, J.M., et al., *Elimination of Unwanted Echoes and Reduction of Dead Time in 3-Pulse Electron Spin-Echo Spectroscopy.* Journal of Magnetic Resonance, 1986. **66**(1): p. 74-85.
128. Fauth, J.M., A. Schweiger, and R.R. Ernst, *Recovery of Broad Hyperfine Lines in Electron Spin-Echo Envelope Modulation Spectroscopy of Disordered-Systems.* Journal of Magnetic Resonance, 1989. **81**(2): p. 262-274.
129. Hyde, J.S., G.H. Rist, and L.E. Eriksson, *Endor of Methyl Matrix and Alpha Protons in Amorphous and Polycrystalline Matrices.* Journal of Physical Chemistry, 1968. **72**(12): p. 4269-&.
130. Mims, W.B., *Envelope Modulation in Spin-Echo Experiments.* Physical Review B, 1972. **5**(7): p. 2409-&.
131. Mims, W.B. and J. Peisach, *Advanced EPR: Appications in Biology and Chemistry*, A.J. Hoff, Editor. 1989, Elsevier: New York. p. 1-57.

132. Mims, W.B., J.L. Davis, and J. Peisach, *The Exchange of Hydrogen Ions and of Water Molecules near the Active Site of Cytochrome c.* J. Magn. Reson., 1990. **86**: p. 273-292.
133. Boas, J.F., et al., *Interpretation of Electron Spin Resonance Spectra Due to Some B12-dependent Enzyme Reactions.* J. Chem. Soc. Faraday II, 1978. **74**: p. 417-431.
134. Gerfen, G.J., *EPR spectroscopy of B12-dependent enzymes". In: Chemistry and Biochemistry of B12*, in *Chemistry and Biochemistry of B12*, R. Banerjee, Editor. 1999, John Wiley and Sons: New York. p. 165-195.
135. Bandarian, V. and G.H. Reed, *Analysis of the electron paramagnetic resonance spectrum of a radical intermediate in the coenzyme B(12)-dependent ethanolamine ammonia-lyase catalyzed reaction of S-2-aminopropanol.* Biochemistry, 2002. **41**(27): p. 8580-8.
136. Wertz, J.E. and J.R. Bolton, *Electron Spin Resonance*. 1986, New York: Chapman and Hall.
137. Bandarian, V. and G.H. Reed, *Isotope effects in the transient phases of the reaction catalyzed by ethanolamine ammonia-lyase: Determination of the number of exchangeable hydrogens in the enzyme-cofactor complex.* Biochemistry, 2000. **39**(39): p. 12069-12075.
138. Flanagan, H.L. and D.J. Singel, *Analysis of 14N ESEEM Patterns of Randomly Oriented Solids.* J. Chem. Phys., 1987. **87**: p. 5606-5616.
139. Mims, W.B. and J. Peisach, *Nuclear Modulation Effect in Electron-Spin Echoes for Complexes of Cu2+ and Imidazole with N-14 and N-15.* Journal of Chemical Physics, 1978. **69**(11): p. 4921-4930.
140. Edmonds, D.T., *Nuclear Quadrupole Double Resonance.* Physics Letters C, 1977. **29**(4): p. 233-290.
141. Grimaldi, S., et al., *QH- ubisemiquinone radical in the bo3-type ubiquinol oxidase studied by pulsed electron paramagnetic resonance and hyperfine sublevel correlation spectroscopy.* Biochemistry, 2001. **40**: p. 1037-1043.
142. Grimaldi, S., et al., *Asymmetric Binding of the High-Affinity QH- Ubisemiquinone in Quinol Oxidase (bo3) from Escherichia coli Studied by Multifrequency Electron Paramagnetic Resonance Spectroscopy.* Biochemistry, 2003. **42**: p. 5632-5639.
143. Yap, L.L., et al., *Characterization of mutants that change the hydrogen bonding of the semiquinone radical at the QH site of the cytochrome b03 from Eschericia coli.* J. Biol. Chem., 2007. **282**(12): p. 8777-8785.
144. Heller, C. and H.M. McConnell, *Electron-nuclear interactions through hyperconjugation.* J. Chem. Phys., 1960. **32**: p. 1535-1539.
145. Dikanov, S.A. and Y.D. Tsvetkov, *Electron spin echo envelope modulation (ESEEM) spectroscopy*. 1992, Boca Raton: CRC Press. 412 p.
146. Keijzers, C.P., et al., *Magres - a General Program for Electron-Spin-Resonance, Endor and Eseem.* Journal of the Chemical Society-Faraday Transactions I, 1987. **83**: p. 3493-3503.
147. Poppl, A., et al., *A study of the coordination geometry of the tetrapyridine copper(II) complex in NaY and MCM-41 by two-dimensional electron spin resonance HYSCORE spectroscopy.* Journal of Physical Chemistry B, 1998. **102**(40): p. 7752-7763.

148. Poppl, A., et al., *Coordination geometry of the copper-pyridine complex in frozen solution as studied by proton and deuterium two-dimensional hyperfine sublevel correlation electron spin resonance spectroscopy.* Journal of Physical Chemistry A, 1998. **102**(20): p. 3599-3606.
149. Poppl, A. and L. Kevan, *A practical strategy for determination of proton hyperfine interaction parameters in paramagnetic transition metal ion complexes by two-dimensional HYSCORE electron spin resonance spectroscopy in disordered systems.* Journal of Physical Chemistry, 1996. **100**(9): p. 3387-3394.
150. Madi, Z.L., S. Van Doorslaer, and A. Schweiger, *Numerical simulation of one- and two-dimensional ESEEM experiments.* Journal of Magnetic Resonance, 2002. **154**(2): p. 181-191.
151. Shane, J.J., L.P. Liesum, and A. Schweiger, *Efficient simulation of ESEEM spectra using gamma.* Journal of Magnetic Resonance, 1998. **134**(1): p. 72-75.
152. Mims, W.B., *Amplitudes of Superhyperfine Frequencies Displayed in Electron-Spin Echo Envelope.* Physical Review B, 1972. **6**(9): p. 3543-&.
153. Goldberg, D.E., *Genetic algorithms in search, optimization, and machine learning*. 1989, Reading, Mass.: Addison-Wesley Pub. Co. xiii, 412 p.
154. Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi, *Optimization by Simulated Annealing.* Science, 1983. **220**(4598): p. 671-680.
155. Nelder, J.A. and R. Mead, *A Simplex-Method for Function Minimization.* Computer Journal, 1965. **7**(4): p. 308-313.
156. Levenberg, K., *A Method for the Solution of Certain Problems in Least Squares.* Quart. Appl. Math., 1944. **2**: p. 164-168.
157. Wertz, J.E. and J.R. Bolton, *Electron spin resonance; elementary theory and practical applications*. 1972, New York: McGraw-Hill. xiv, 497 p.
158. Stoll, S. and A. Schweiger, *Rapid construction of solid-state magnetic resonance powder spectra from frequencies and amplitudes as applied to ESEEM.* Journal of Magnetic Resonance, 2003. **163**(2): p. 248-256.
159. Hanson, G.R., et al., *XSophe-Sophe-XeprView (R). A computer simulation software suite (v. 1.1.3) for the analysis of continuous wave EPR spectra.* Journal of Inorganic Biochemistry, 2004. **98**(5): p. 903-916.
160. Stoll, S. and A. Schweiger, *EasySpin, a comprehensive software package for spectral simulation and analysis in EPR.* Journal of Magnetic Resonance, 2006. **178**(1): p. 42-55.
161. Hoffman, B.M., J. Martinsen, and R.A. Venters, *General-Theory of Polycrystalline Endor Patterns - G and Hyperfine Tensors of Arbitrary Symmetry and Relative Orientation.* Journal of Magnetic Resonance, 1984. **59**(1): p. 110-123.
162. Rowan, L.G., E.L. Hahn, and W.B. Mims, *Electron-Spin-Echo Envelope Modulation.* Physical Review, 1965. **137**(1A): p. A61-A71.
163. Dikanov, S.A., A.A. Shubin, and V.N. Parmon, *Modulation effects in the electron spin echo resulting from hyperfine interaction with a nucleus of an arbitrary spin.* J. Magn. Reson., 1981. **42**: p. 474-487.
164. Sun, L., et al., *Critical role of arginine 160 of the EutB protein subunit for active site structure and radical catalysis in coenzyme B-12-dependent ethanolamine ammonia-lyase.* Biochemistry, 2008. **47**(20): p. 5523-5535.

165. Zhu, Y.S., *Probability and Statistics in Experimental Physics*. 2nd ed. 2006, Beijing: Science Press.
166. Warncke, K. and J. McCracken, *Analysis of static distributions in hydrogen hyperfine interactions in randomly oriented radicals in the solid state by using 2H electron spin echo envelope modulation spectroscopy: conformational dispersion of b-2H coupling in the model tyrosyl radical.* J. Chem. Phys., 1995. **103**(16): p. 6829-40.
167. Tan, S.L., et al., *Electron Spin-Echo Studies of the Composition of the Paramagnetic Intermediate Formed during the Deamination of Propanolamine by Ethanolamine Ammonia-Lyase, and Adocbl-Dependent Enzyme.* Journal of Biological Chemistry, 1986. **261**(8): p. 3483-3485.
168. Flanagan, H.L. and D.J. Singel, *Analysis of N-14 Eseem Patterns of Randomly Oriented Solids.* Journal of Chemical Physics, 1987. **87**(10): p. 5606-5616.
169. LATIF ABUHIJLEH, A., C. WOODS, and I.Y. AHMED, *Synthesis and molecular structure of monomeric copper(II) acetates with 2-methylimidazole and 1,2-dimethylimidazole.* Inorganica chimica acta 1991. **190**: p. 11-17.
170. Zhang, C., et al., *Self-assembly in transition metal complexes: structural characterization of a copper histamine nitrate [Cu(II)(him)2(NO3)2].* Journal of Chemical Crystallography, 1999. **29**(11): p. 1153-1155.
171. Baute, D., et al., *Carboxylate binding in copper histidine complexes in solution and in zeolite Y: X- and W-band pulsed EPR/ENDOR combined with DFT calculations.* Journal of the American Chemical Society, 2004. **126**(37): p. 11733-11745.
172. Weltner, W., *Magnetic atoms and molecules*. 1989, New York: Dover Publications. 436 p.
173. Colaneri, M.J. and J. Peisach, *An Electron Spin-Echo Envelope Modulation Study of Cu(Ii)-Doped Single-Crystals of L-Histidine Hydrochloride Monohydrate.* Journal of the American Chemical Society, 1992. **114**(13): p. 5335-5341.
174. Colaneri, M.J. and J. Peisach, *A Single-Crystal Epr and Eseem Analysis of Cu(Ii)-Doped Bis(L-Histidinato)Cadmium Dihydrate.* Journal of the American Chemical Society, 1995. **117**(23): p. 6308-6315.

# Appendices

## Appendix A. Protocol of cell growth and harvest

1. Before Start, check supplies, including:
    a) Pre-made 2xYT cluture with 75 μg/ml ampicillin
    b) Pre-made LB agar plate with 100 μg/ml ampicillin
    c) About 350 g 2xYT powder
    d) 1.1 g ampicillin
    e) 3.1 g IPTG

2. Preparing inoculums
    a) Day 1. Grow some cells overnight in a culture tube with 4 ml 2xYT culture (with 75 μg/ml ampicillin).
        - Transfer 4 ml 2xYT into a culture tube.
        - Inoculate 2xYT from a glycerol stock in the -80 °C freezer. Use a sterilized pipette tip to touch the surface of the glycerol stock and put it in the culture tube. Do it quick so that the glycerol stock doesn't melt.
        - Flame the tube and its cap before putting in to the incubator.
        - Set the incubator to 37 °C, 225 rpm.
        - Make sure the cap of the culture tube is at the second position, i.e. not fully closed.
        - Important: Use sterile techniques in those steps.

    b) Day 2. Put the culture tube in 4 °C fridge in the morning. Streak the cells on a plate in the late afternoon, and let it grow overnight.
        - Make sure the incubator shaking is turned off.

3. Preparing fermenter
    a) Day 3. Move the plate into 4 °C fridge. Wrap it with para-film and check for colonies.
    b) Preparing fermenter:
        - Assemble the sampler attachment; make sure you put on clamps on the two small tubing.
        - Put on the 6 screws for the head plate. Hand tight them and make sure the head plate and the vessel are in perfect contact.
        - Put on the condenser if it is not on. Check the o-ring, and apply more grease if needed.
        - Attaching the harvest tubing to the harvest port. Fill the vessel with distilled water, wait for 10 minutes, and let it drain.

- Put on a clamp on the harvest tubing and make sure it is close fully. Refill the vessel with 9.5 liter distilled water.
- Weigh 295 g (31 g/L x 9.5 L) 2xYT powder. Pour it in through the inoculation port (the big port) carefully using a funnel.
- Put on the agitator, and connect it to the main controller. Set agitation to 200 to help the 2xYT power dissolve. Remove the agitator after the powder is fully dissolved.
- Put on filters on the condenser and air intake port.
- Put on short tubings on the foam trap port and the air-intake filter. Clamp them tightly.
- Wrap all filters with aluminum foil. Make sure the condenser port is NOT covered.
- Put the $O_2$ probe into the head plate.
- Put in the pH probe after calibrating the pH probe with 7.0 and 4.0 standards. If the pH probe has an unreasonably high reading, check its connection, and restart the console if needed. Important: use extra care when putting in the pH probe.
- Recheck all the clamps.
- Recheck all the screws and ports on the head plate and make sure they are all tight.
- Remove the white plastic tubing connector from the harvest tubing. (since it melts during autoclaving)
- Put on protection caps on the central shaft and pH probe. (Black and red caps, respectively)

c) Prepare 2 flasks with 500 ml 2xYT broth. Put on alumimum foil caps.
d) Autoclave the fermenter and 2 flasks for 25-minutes in the liquid cycle
e) After autoclaving, check for spills, and damages on the tubing. And carefully:
    - Connect the cooling water to the condenser and the internal cooling coil of the vessel. Connect their corresponding drains. Turn on the main water valve.
    - Put in temperature probe. The temperature should be reading high now. Set temperature to 37 °C in the main controller, and control to “Auto”. The internal cooling water should be running automatically.
    - Put on the agitator and set it to 200 rpm to help with cooling.
    - Make about 11 ml 0.1 g/ml ampicillin solution. After the liguids cools down to less than 40 °C, add 9.5 ml to the fermenter vessel, and 0.375 ml to each of the 500 ml flasks. Important: use sterile techniques.

- Put on heat blanket.
- Turn off agitator. Connect the $O_2$ and pH cables. Remove all the foils.

4. Cell Growing
    a) Late in Day 3: inoculate one colony and two colonies to the flasks. Put them in the incubator at 37 °C and 255 rpm. Let the cell grow for about 8 hours.
    b) On Day 4:
        - Measure and record $OD_{600}$ of the 2 flasks' culture; choose the one has OD closer to 0.6 for inoculation.
        - Put on the air tubing. Turn on the main air valve slowly and carefully.
        - Calibrate $O_2$ probe. First disconnect $O_2$ probe temporarily, set the zero. Then reconnect the cable, set the air flow to more than 5 liter/minutes, and agitation back to 200 rpm.
        - Prepare a sterile funnel by washing with ethanol. Clean all the ethanol with Kim-wipe. Inoculate the 500-ml culture through the inoculation port.
        - Set $dO_2$ to 30% in the main controller, its control to "auto" and cascade to "agit". Now in the main controller display, the agitation control should change from "auto" to "$dO_2$"
        - Watch it often while growing. Make sure the pH is within 6.7 and 7.2 range. If not, adjust with HCl or 10% NaOH accordingly.
        - Let it grow until $OD_{600}$ reaches 0.8, induce with 3.1 g IPTG through the inoculation port carefully.
        - Let it grow for another 4 hours after induction.

5. Cell Harvesting
    a) Measure and record OD before harvesting.
    b) Turn off air and water.
    c) In the main controller, set everything to "off".
    d) Put about 5 2-liter flasks on ice. Pour the culture broth in through the harvest tubing. Cover them with para-film. Remove and clean the pH probe and put it in 3 M KCl.
    e) Change the rotor for RC5B centrifuge, and rotor code to 10.
    f) Centrifuge culture broth in batches in 250 ml bottles. You can either centrifugre at 3800 rpm for 15 minutes or 5000 rpm for 10 minutes. Dispose the supernatant. Dissolve the cell pellet with minimum pH 7.5 40 mM KPi, and trask to a flask on ice.

g) Centrifuge the cell pellet with KPi in the flask. Make sure you use a balace. Weigh and record the new pallet. Dissolve with minimum KPi and transfer to a labeled 50 ml centrifuge tube. Flash freeze and put it in the 80 °C freezer.
h) Refill fermenter vessel with water and soap. Leave it overnight.
i) Make sure to clean to rotor, the centrifuge and everything else. Use ethanol to kill ampicillin-resistant bacteria.

6. Fermenter Cleaning
    a) Detach agitator.
    b) Remove the temperature probe.
    c) Remove the 6 screws on the head plate.
    d) Detach the heat blanket.
    e) Detach and wash tubing and filters.
    f) Detach and wash the sampler assembly.
    g) Detach $O_2$ probe.
    h) Detach the condenser.
    i) Drain all the water through the harvest port.
    j) Pull the head plate off. Clean the head plate carefully so that you don't bend anything.
    k) Check ports and o-rings on the head plate. Apply more grease if necessary.
    l) Carefully clean the vessel.
    m) Put the vessel back. Apply more grease on the rim if necessary.
    n) Put the head plate back.
    o) Cleanup everything else.

## Appendix B. Protocol of EAL site-directed mutagenesis

1. Methylation of target plasmid
    a. Reaction
        i. 1.5 µl purified plasmid following Qiagen™ qickspin miniprep kit procedures
        ii. 4.8 µl methylation buffer
        iii. 4.8 µl 10x SAM
        iv. 3.0 µl DNA methylase
        v. 30.9 $H_20$
    b. Incubate at 37 °C for one hour.

2. Introduce mutations with PCR
    a. Reaction
        i. 36.0 µl $H_2O$
        ii. 5.0 µl 10x PCR buffer
        iii. 1.5 µl 10 mM dNTP
        iv. 1.0 µl 50 mM $MgSO_4$
        v. 1.5 µl 10 µM Primer A
        vi. 1.5 µl 10 µM Primer B
        vii. 3.0 µl methylated DNA from methylation reaction
        viii. 0.5 µl TAQ polymerase
    b. Run thermalcycler for the following cyle:
        i. 94 °C 2 minutes
        ii. 94 °C for 30 seconds
        iii. 55 °C for 30 seconds
        iv. 68 °C for 12 minutes
        v. repeat ii-iv for 20 cyles
        vi. 68 °C for 10 minutes
        vii. hold at 4 °C

3. Transformation
    a. Warm up SOC medium to 37 °C
    b. Thaw competent cells for 5~7 minutes on ice
    c. Add 2 µl PCR reaction to cell vial and mix by tapping
    d. Cover in ice for 7~10 minutes
    e. Place in water bath at 42 °C for 30 seconds
    f. Place in ice for 1 minutes
    g. Add 200 µl SOC medium
    h. Incubate vial at 37 °C and 255rpm for one hour

i. Take 125 µl form vial and spread on 2xYT plate with 100 µg/ml ampicillin
j. Incubate at 37 °C overnight
k. Select single conolies for sequencing

## Appendix C. Source code of OPTESIM toolbox

The OPTESIM toolbox is maintained and documented on website www.physics.emory.edu/optesim with explaination and tutorials at the time when this dissertation is prepared. This appendix includes source code of all MATLAB functions in an alphabetic order.

### changeOptModelValue.m

```
0001 function opt_model = changeOptModelValue( opt_model, varargin )
0002 % CHANGEOPTMODELVALUE - Change a parameter value in the
optimization
0003 %                        model structure
0004 %   Authur: Li Sun (lsun3@emory.edu)
0005 %   Last Update: 04/24/2008
0006 %
0007 %   opt_model = changeOptModelValue( opt_model, field_name, index,
value )
0008 %   INPUT Parameters:
0009 %         opt_model : Optimization model structure to update
0010 %        field_name : Parameter to change
0011 %                     Values can be:
0012 %                        'I', 'g', 'Aiso', 'r', 'AEuler1',
'AEuler2',
0013 %                        'AEuler3', 'Axx', 'Ayy', 'Azz',
0014 %                        'eeqQ', 'eta', 'QEuler1', 'QEuler2',
'QEuler3'
0015 %                     for Nulear parameters, or
0016 %                        'factor'
0017 %                     for additive nodes factors.
0018 %             index : Index of nuclues for Nulear parameters, or
0019 %                     index of nodes for factor.
0020 %             value : value to change to.
0021 %
0022 %
0023 %   OUTPUT Parameter:
0024 %         opt_model : Updated model structure for optimization.
0025 %
0026 %   SEE ALSO registerOptModelVariable, eseOptModel
0027
0028 k = size(varargin,2)/3;
0029
0030 for i=1:k
0031     field_name = varargin{(i-1)*3+1};
0032     index = varargin{(i-1)*3+2};
0033     value = varargin{(i-1)*3+3};
0034
0035     if strcmp(field_name,'factor')
0036         opt_model.nodes.factors{index}.value = value;
0037     else
0038         opt_model.nuclei{index}.(field_name).value = value;
0039     end
```

```
0040 end
```

**cnstrctHamiltonian.m**

```
0001 function H = cnstrctHamiltonian( nu, B_value, B_vector )
0002 %CNSTRCTHAMILTONIAN - Construct Hamiltonian from Nucleus structure
and a
0003 %field vector.
0004 %   H = cnstrctHamiltonian( nu, B_value, B_vector )
0005 %   INPUT parameters:
0006 %       nu      : Nucleus structure
0007 %       B_value : External field in Tesla
0008 %       B_vector: External field directional vector
0009 %   OUTPUT parameters:
0010 %       H       : Constructed and properly transformed Hamiltonian
0011 %                 matrix
0012 %                   H{1} is Hamiltonian in alpha manifold
0013 %                   H{2} is Hamiltonian in beta manifold
0014 %   SEE ALSO simulateTDSignal
0015
0016 % Update Li Sun, 07/09/08
0017
0018 %dinfining constants
0019 be=9.27401*10^(-24);%[JT^-1]
0020 bn=5.05078*10^(-27);%[JT^-1]
0021 hbar=1.054572*10^(-34);%[JS]
0022 h=6.62607*10^(-34);%[JS]
0023 sqrt2=sqrt(2);
0024 mhz=10^6*2*pi; % [(radians/sec)/MHz], MHz to radians/sec
0025
0026 gnbn=nu.g*bn; %[J/T]
0027 gnbnB=gnbn*B_value; %[J]
0028 wI=2*pi/h*gnbnB; %[radians/sec]
0029
0030 Idim = 2*nu.I+1;
0031
0032 %Setup Spin matrices Ix, Iy, Iz
0033 %Ix, Iy, Iz = Ixyz(:,:,1), Ixyz(:,:,2), Ixyz(:,:,3)
0034 Ixyz=zeros(Idim, Idim, 3);
0035 for mrow=-nu.I:nu.I;
0036     row= -mrow+nu.I+1;
0037     % row=1 to 2*I+1 while mrow=I to -I
0038     % row's are usually horizontal in matrices
0039     %   and are the 1st index
0040     Ixyz(row,row,3)=mrow;
0041     if(mrow<nu.I)
0042         mcol=mrow+1;
0043         col= -mcol+nu.I+1;
0044         % col=1 to 2*I+1 while mcol=I to -I
0045         % col's are usually vertical in matrices
0046         %   and are the 2nd index
0047         val=sqrt(nu.I*(nu.I+1)-mrow*mcol)/2.0;
0048         Ixyz(row,col,1)=val;
0049         Ixyz(col,row,1)=val; % Ix is symmetric
0050         Ixyz(row,col,2)=i*val;
```

```
0051         Ixyz(col,row,2)= -i*val; % Iy is anti-symmetric
0052     end; % end of if
0053 end; % next mrow
0054
0055 %Setup hyperfine coupling tensor
0056 A_raw = diag([nu.Axx, nu.Ayy, nu.Azz]*mhz); % [radians/sec]
0057
0058 %Setup quadrupole interaction parameter tensor
0059 K = nu.eeqQ/(4*nu.I*(2*nu.I-1))*mhz;  % [radians/sec]
0060 Q_raw = diag([nu.eta-1, -nu.eta-1, 2]*K); % [radians/sec]
0061
0062 %Rotate A_raw and Q_raw into molecular frame
0063 A_EulerMatrix = getEulerMatrix(nu.AEuler);
0064 Q_EulerMatrix = getEulerMatrix(nu.QEuler);
0065
0066 A_mol = inv(A_EulerMatrix)*A_raw*A_EulerMatrix;
0067 Q_mol =
inv(A_EulerMatrix)*inv(Q_EulerMatrix)*Q_raw*Q_EulerMatrix*A_EulerMatrix;
0068
0069
0070 % Add up Hamiltonian
0071 H1 = zeros(Idim, Idim);
0072 % No quadrupole interaction for I<1
0073 if nu.I >= 1
0074     for row=1:3;
0075         for col=1:3;
0076             if(Q_mol(row,col)~=0.0)
0077                 H1=H1 +
hbar*Ixyz(:,:,row)*Q_mol(row,col)*Ixyz(:,:,col);
0078             end;
0079         end;
0080     end;
0081 end
0082
0083 ms_alpha = 1/2;
0084 ms_beta = -1/2;
0085
0086 agn_alpha=hbar*ms_alpha*A_mol;
0087 agn_beta=hbar*ms_beta*A_mol;
0088
0089 H_alpha = H1;
0090 H_beta = H1;
0091
0092 for row=1:3;
0093     if(B_vector(row)~=0)
0094         H_alpha=H_alpha-gnbnB*B_vector(row)*Ixyz(:,:,row);
0095         H_beta =H_beta -gnbnB*B_vector(row)*Ixyz(:,:,row);
0096         for col=1:3;
0097
H_alpha=H_alpha+agn_alpha(row,col)*B_vector(row)*Ixyz(:,:,col);
0098             H_beta =H_beta +agn_beta(row,col)
*B_vector(row)*Ixyz(:,:,col);
0099         end;
0100     end;
0101 end;
0102
0103 H = {H_alpha, H_beta};
```

```
0104
0105
0106 function R = getEulerMatrix( angles_in )
0107 %GETEULERMATRIX - get EulEr Rotational Matrix from Euler angles
0108 %   R = getEulerMatrix( angles )
0109 angles = angles_in/180*pi;
0110
0111 CG = cos(angles(3));
0112 CB = cos(angles(2));
0113 CA = cos(angles(1));
0114 SG = sin(angles(3));
0115 SB = sin(angles(2));
0116 SA = sin(angles(1));
0117
0118 R = zeros(3,3);
0119
0120 R(1,1)=CG*CB*CA-SG*SA;
0121 R(1,2)=CG*CB*SA+SG*CA;
0122 R(1,3)=-CG*SB;
0123 R(2,1)=-SG*CB*CA-CG*SA;
0124 R(2,2)=-SG*CB*SA+CG*CA;
0125 R(2,3)=SG*SB;
0126 R(3,1)=SB*CA;
0127 R(3,2)=SB*SA;
0128 R(3,3)=CB;
```

### constructModel.m

```
0001 function model = constructModel( x, opt_model )
0002 % CONSTRUCTMODEL - construct a system model for opt_model and
current
0003 %                  optimization vector x
0004 %   model = constructModel( x, opt_model )
0005 %   INPUT parameters:
0006 %               x : optimization paramter vector
0007 %       opt_model : Optimization Model
0008 %   OUTPUT parameters:
0009 %           model : Constructed system model with the correct
value in x
0010
0011 opt_nuclei = opt_model.nuclei;
0012 model_nuclei = cell(size(opt_nuclei));
0013
0014 for i=1:length(opt_nuclei)
0015     model_nuclei{i} = constructNucleus( x, opt_nuclei{i} );
0016 end
0017 model.nuclei = model_nuclei;
0018
0019 model.nodes = constructNodes( x, opt_model.nodes );
0020
0021
0022
0023 function nucleus = constructNucleus( x, opt_nucleus )
0024
0025 if strcmp(opt_nucleus.type, 'Aiso')
```

```
0026     nu_fields = {'I', 'g', 'Aiso', 'r', 'AEuler1', 'AEuler2',
'AEuler3',...
0027         'eeqQ', 'eta', 'QEuler1', 'QEuler2', 'QEuler3'};
0028     nu_idx = [1 2 3 4 8 9 10 11 12 13 14 15];
0029
0030 else
0031     nu_fields = {'I', 'g', 'Axx', 'Ayy', 'Azz', 'AEuler1',
'AEuler2', ...
0032          'AEuler3', 'eeqQ', 'eta', 'QEuler1', 'QEuler2', 'QEuler3'};
0033     nu_idx = [1 2 5 6 7 8 9 10 11 12 13 14 15];
0034 end
0035
0036 nucleus = struct;
0037 nucleus_vector = zeros(size(nu_idx));
0038
0039 for i=1:length(nu_fields)
0040     opt_option = opt_nucleus.(nu_fields{i});
0041     if opt_option.fix == 1
0042         nucleus_vector(i) = opt_option.value;
0043     else
0044         nucleus_vector(i) = x(opt_option.idx);
0045     end
0046 end
0047
0048 nucleus = setnucleus(nucleus_vector, nu_idx, nucleus);
0049
0050 function nodes = constructNodes( x, opt_nodes )
0051
0052 nodes = opt_nodes.value;
0053
0054 % This is a confusing routine.
0055 % Basically, what is doing here is to set factor values for
children of
0056 % additive nodes.
0057 % However, the last sibling should be 1 minus the sum of other
siblings.
0058 % Children of multiplicative nodes should always have a factor of
1,
0059 % therefore its opt_nodes.factors structure is ignored.
0060
0061 for i=1:length(opt_nodes.factors)
0062     factor_options = opt_nodes.factors{i};
0063     if factor_options.fix == 0
0064         if nodes(i,3) ~= 0
0065             if nodes(2, nodes(i, 3)) == 1
0066                 sibling_nodes = find(nodes(:,3) == nodes(i,3));
0067                 if i == max(sibling_nodes)
0068                     nodes(i,4) = 1-sum(sibling_nodes(1:end-1));
0069                 else
0070                     nodes(i,4) = x(factor_options.idx);
0071                 end
0072             end
0073         end
0074     end
0075 end
```

### eseOptCftInterval.m

```
0001 function xbounds = eseOptCftInterval( x, fval, level)
0002 % ESEOPTCFTINTERVAL - returns the simutanous confident interval
for optimal
0003 %                     parameter x
0004 %
0005 %   xbounds = eseOptCftInterval( x, fval, level)
0006 %   INPUT parameters:
0007 %               x : optimal paramter vector
0008 %            fval : optimal fitting score
0009 %           level : confident level (0.99, 0.95 etc.)
0010 %   OUTPUT parameters:
0011 %         xbounds : lower and upper bounds for vector x
0012
0013 global global_spectra
0014
0015 total_points = 0;
0016 for i=1:length(global_spectra)
0017     total_points =
total_points+length(global_spectra{i}.exp_data.y);
0018 end
0019
0020 total_freedom = total_points-length(x);
0021
0022 sigma2 = fval/total_freedom;
0023
0024 a = sigma2*chi2inv(level, length(x));
0025
0026 xbounds = zeros(2, length(x));
0027
0028 disp(sprintf('a for confident level %.3f is %.2f', level, a));
0029
0030 for i=1:length(x)
0031     options = optimset('TolX', abs(x(i)*1e-3));
0032     guess = abs(x(i)*0.05);
0033     xbounds(1, i) = fzero(@(v)(objectiveup(v, x, i)-(fval+a)),
x(i)+guess, options);
0034     xbounds(2, i) = fzero(@(v)(objectivedown(v, x, i)-(fval+a)),
x(i)-guess, options);
0035     disp(sprintf('Interval for idx: %d (%.6f) is %.6f ~ %.6f', i,
x(i),  xbounds(2, i), xbounds(1, i)));
0036 end
0037
0038 end
0039
0040
0041 function r = objectiveup(v, x, i)
0042 if v>x(i)
0043     x(i) = v;
0044 end
0045 r = eseOptObjective(x);
0046 end
0047
0048 function r = objectivedown(v, x, i)
0049 if v<x(i)
```

```
0050     x(i) = v;
0051 end
0052 r = eseOptObjective(x);
0053 end
```

### eseOptModel.m

```
0001 function opt_model = eseOptModel( nuclei_type, exp )
0002 % ESEOPTMODEL - Create a new optimization model structure
0003 %   Authur: Li Sun (lsun3@emory.edu)
0004 %   Last Update: 04/24/2008
0005 %
0006 %   opt_model = eseOptModel( nuclei_type, exp )
0007 %   INPUT Parameters:
0008 %       nuclei_type : A cell structure of nuclei types, valid
values are
0009 %                     'N14', 'N15', 'H2', 'H1', default is 'H1'.
0010 %                     I and g are set for those types of nuclei.
0011 %              exp : Expression for node structure, example,
0012 %                     '(1+2)*3/4'
0013 %                     '1+2+3/4+5'
0014 %                    Numbers in the expression corespond to the
nuclei
0015 %                    indices in nuclei_type.
0016 %   Note: Nuclei optimization model constructed are in 'Aiso' and
'r'
0017 %         representation. To change to tensor representation,
0018 %         change opt_model.nuclei{i}.type to 'tensor'
0019 %
0020 %   OUTPUT Parameter:
0021 %         opt_model : A model structure for optimization.
0022 %                     Parameter values can be changed with
changeOptModelValue
0023 %                     Optimization varaibles can be registered
with
0024 %                     registerOptModelVariable
0025 %
0026 %   SEE ALSO changeOptModelValue, registerOptModelVariable
0027
0028 nuclei = cell(size(nuclei_type));
0029
0030 for i=1:length(nuclei_type)
0031     nuclei{i} = constructOptNucleus( nuclei_type{i} );
0032 end
0033
0034 opt_model.nuclei = nuclei;
0035
0036 opt_model.nodes = struct;
0037 opt_model.nodes.value = parseNodes( exp );
0038 opt_model.nodes.factors =
constructNodesFactors( opt_model.nodes.value );
0039 opt_model.variables={};
0040
0041 function factors = constructNodesFactors( nodes )
0042 factors = cell(1,size(nodes,1));
```

```
0043 for i = 1:size(nodes,1)
0044     factors{i} = optStruct( 1, nodes(i, 4) );
0045 end
0046
0047 function nucleus = constructOptNucleus( nu_type )
0048 nucleus = struct;
0049 if strcmp(nu_type,'N14')
0050     nucleus.I = optStruct( 1, 1 );
0051     nucleus.g = optStruct( 1, 0.4038 );
0052 elseif strcmp(nu_type, 'N15')
0053     nucleus.I = optStruct( 1, 1/2 );
0054     nucleus.g = optStruct( 1, -0.5664 );
0055 elseif strcmp(nu_type, 'H2')
0056     nucleus.I = optStruct( 1, 1 );
0057     nucleus.g = optStruct( 1, 0.8574 );
0058 else %'H1'
0059     nucleus.I = optStruct( 1, 1/2 );
0060     nucleus.g = optStruct( 1, 5.5857 );
0061 end
0062 nucleus.Aiso = optStruct( 1, 0 );
0063 nucleus.r = optStruct( 1, 2.5 );
0064 nucleus.Axx = optStruct( 1, 0 );
0065 nucleus.Ayy = optStruct( 1, 0 );
0066 nucleus.Azz = optStruct( 1, 0 );
0067 nucleus.AEuler1 = optStruct( 1, 0 );
0068 nucleus.AEuler2 = optStruct( 1, 0 );
0069 nucleus.AEuler3 = optStruct( 1, 0 );
0070 nucleus.eeqQ = optStruct( 1, 0 );
0071 nucleus.eta = optStruct( 1, 0.5 );
0072 nucleus.QEuler1 = optStruct( 1, 0 );
0073 nucleus.QEuler2 = optStruct( 1, 0);
0074 nucleus.QEuler3 = optStruct( 1, 0 );
0075 nucleus.type = 'Aiso';
0076
0077 function nodes = parseNodes( exp )
0078 % parseNodes - parse exp string into nodes matrix
0079
0080 splits = regexp(exp, '/','split');
0081
0082 if length(splits) > 1
0083     numerator_str = splits{1};
0084     denominator_str = splits{2};
0085 else
0086     numerator_str = splits{1};
0087     denominator_str = ' ';
0088 end
0089
0090 nodes = [0 0 0 1; 0 0 0 1];
0091 nodes = generateNodesforStr(nodes, numerator_str, 1, 0, 1);
0092 nodes = generateNodesforStr(nodes, denominator_str, 2, 0, 1);
0093
0094 function nodes = generateNodesforStr(nodes, str, nodes_idx,
parent_idx, factor )
0095
0096 [parts, operator] = splits_str( str );
0097
0098 if strcmp(parts{1},' ')
```

```
0099     parts={};
0100 end
0101
0102 switch length(parts)
0103     case 0
0104         nodes(nodes_idx,:) = [-1 -1 0 factor];
0105     case 1
0106         nodes(nodes_idx,:) = [str2num(str) -1 parent_idx factor];
0107     otherwise
0108         switch operator
0109             case '+'
0110                 mode_num = 1;
0111                 children_factor = 1/length(parts);
0112             case '*'
0113                 mode_num = 0;
0114                 children_factor = 1;
0115         end
0116         nodes(nodes_idx,:) = [0, mode_num, parent_idx, factor];
0117         for i = 1:length(parts)
0118             nodes = generateNodesforStr(nodes, parts{i},
size(nodes,1)+1, nodes_idx, children_factor);
0119         end
0120
0121 end
0122
0123 function [parts, operator] = splits_str( str )
0124
0125 if strcmp(str,' ')
0126     parts = {' '};
0127     operator = ' ';
0128     return;
0129 end
0130
0131 parenthese_level = 0;
0132 operator = ' ';
0133 for i=1:length(str)
0134     if parenthese_level == 0
0135         switch str(i)
0136             case '+'
0137                 operator = '+';
0138                 break;
0139             case '*'
0140                 operator = '*';
0141             case '('
0142                 parenthese_level = parenthese_level+1;
0143         end
0144     else
0145        switch str(i)
0146             case '('
0147                 parenthese_level = parenthese_level+1;
0148             case ')'
0149                 parenthese_level = parenthese_level-1;
0150        end
0151     end
0152 end
0153
0154 parts = {};
```

```
0155 parenthese_level = 0;
0156 new_part = '';
0157
0158 for i=1:length(str)
0159     p = str(i);
0160     if parenthese_level ==0
0161         switch str(i)
0162             case operator
0163                 if new_part(1) == '(' && new_part(end) == ')'
0164                     new_part = new_part(2:end-1);
0165                 end
0166                 parts = {parts{:} new_part};
0167                 new_part = '';
0168             case '('
0169                 parenthese_level = parenthese_level+1;
0170                 new_part(end+1) = str(i);
0171             otherwise
0172                 new_part(end+1) = str(i);
0173         end
0174     else
0175         new_part(end+1) = str(i);
0176         switch str(i)
0177             case '('
0178                 parenthese_level = parenthese_level+1;
0179             case ')'
0180                 parenthese_level = parenthese_level-1;
0181         end
0182     end
0183
0184 end
0185 if new_part(1) == '(' && new_part(end) == ')'
0186     new_part = new_part(2:end-1);
0187 end
0188 parts = {parts{:} new_part};
0189
0190 function opt_struct = optStruct( fix, value )
0191 opt_struct.fix = fix;
0192 opt_struct.value = value;
```

### eseOptObjective.m

```
0001 function r = eseOptObjective( x )
0002 %ESEOPTOBJECTIVE - Objective function for Optimization
0003 %  r = eseOptObjective( x )
0004 %  INPUT Parameter:
0005 %     x : Vector of registered variable values.
0006 %  OUTPUT Parameter:
0007 %     r : Goodness of fit.
0008 %
0009 %  Note: Global variables: global_opt_model global_spectra
global_bvecs
0010 %        and global_ese_routine have to be set before calling this
function
0011
```

```
0012 global global_opt_model global_spectra global_bvecs
global_ese_routine
0013
0014 model = constructModel( x, global_opt_model );
0015
0016 Q_squares = zeros(length(global_spectra),1);
0017
0018 for i = 1:length(Q_squares)
0019     Ev = global_ese_routine( model, global_spectra{i}.exp_param,
global_bvecs );
0020     Q_squares(i) = matchSimulation( Ev, i );
0021 end
0022
0023 r = sum(Q_squares) + boundPenalty( x );
0024
0025 end
0026
0027 function Q_square = matchSimulation( Ev, index )
0028 global global_spectra
0029
0030 exp_param = global_spectra{index}.exp_param;
0031
0032 sim_y_all = Ev';
0033
0034 sim_y_data = sim_y_all((exp_param.deadtime_dpts):end);
0035
0036
0037 %this is matching algrithm
0038 mean_sim_y_data = mean(sim_y_data);
0039
0040 A = [ones(size(sim_y_data)) sim_y_data-
mean_sim_y_data]\global_spectra{index}.exp_data.y;
0041 t_all = (1:length(sim_y_all))';
0042 t = (exp_param.deadtime_dpts:length(sim_y_all))';
0043
0044 [p feval] = fminsearch(@(x)(match_goodness(x, t,
global_spectra{index}.exp_data.y, sim_y_data)), [A(1) A(2) 0
mean_sim_y_data], optimset('Display','none'));
0045
0046
0047 global_spectra{index}.exp_data.sim_y_matched_data = (sim_y_data-
p(4)).*exp(p(3)*t)*p(2)+p(1);
0048 global_spectra{index}.exp_data.sim_y_matched_all = (sim_y_all-
p(4)).*exp(p(3)*t_all)*p(2)+p(1);
0049
0050 Q_square = feval/global_spectra{index}.exp_data.var;
0051 end
0052
0053 function r = match_goodness ( x, t, exp_data, sim_data )
0054
0055
0056 transformed_sim_data = (sim_data-x(4)).*exp(x(3)*t)*x(2)+x(1);
0057
0058 r = sum((exp_data-transformed_sim_data).^2);
0059 end
0060
0061
```

```
0062 function penalty = boundPenalty( x )
0063 global global_opt_model
0064
0065 scaling_factor = 1e-3;
0066
0067 penalty = 0;
0068 for i=1:length(x)
0069     if global_opt_model.bounds(1,i)>x(i)
0070         penalty = penalty + exp( (global_opt_model.bounds(1,i)-
x(i)) / (abs(x(i)) * scaling_factor) );
0071     end
0072     if global_opt_model.bounds(2,i)<x(i)
0073         penalty = penalty + exp( (x(i)-
global_opt_model.bounds(2,i)) / (abs(x(i)) * scaling_factor) );
0074     end
0075 end
0076 end
```

### esePlot.m

```
0001 function stop = esePlot(x, optimValues, state)
0002 % ESEPLOT - ploting function to be suplied to the optimization
routines
0003
0004 global global_plot_options
0005 stop = false;
0006     switch state
0007         case 'init'
0008             esePlotInitFigures(x);
0009             global_plot_options.last_x = x;
0010             global_plot_options.saved_x = {x};
0011             global_plot_options.fval = optimValues.fval;
0012         case 'iter'
0013             global_plot_options.fval = [global_plot_options.fval,
optimValues.fval];
0014             esePlotFigures();
0015             esePlotDisplayInfo(x);
0016             global_plot_options.last_x = x;
0017             global_plot_options.saved_x{end+1} = x;
0018         case 'done'
0019
0020         otherwise
0021     end
0022 end
```

### esePlotDisplayInfo.m

```
0001 function esePlotDisplayInfo(opt_vector)
0002 % ESEPLOTDISPLAYINFO - Helper function of esePlot
0003 % Author Li Sun (lsun3@emory.edu)
0004 % Last Update 04/25/2008
0005
0006 global global_opt_model global_plot_options
0007
```

```
0008 figure(global_plot_options.info_handle);
0009
0010 for i=1:length(global_opt_model.variables)
0011     vth = global_plot_options.variable_text_handles{i};
0012     x_index = global_opt_model.variables{i}.x_index;
0013
0014     bounds = global_opt_model.bounds(:,x_index);
0015
0016     if opt_vector(x_index)>bounds(1)
0017         set(vth.hLoBound,'ForeGroundColor',[0.165 0.384 0.275]);
0018     else
0019         set(vth.hLoBound,'ForeGroundColor',[1.0 0.2 0.2]);
0020     end
0021
0022     if opt_vector(x_index)<bounds(2)
0023         set(vth.hHiBound,'ForeGroundColor',[0.165 0.384 0.275]);
0024     else
0025         set(vth.hHiBound,'ForeGroundColor',[1.0 0.2 0.2]);
0026     end
0027
0028     set(vth.hvalue, 'String', num2str(opt_vector(x_index)));
0029
0030     change = opt_vector(x_index)-
global_plot_options.last_x(x_index);
0031
0032     set(vth.hchange, 'String', sprintf('%.3g',change));
0033
0034     if change>0
0035         set(vth.hchange,'ForeGroundColor',[1.0 0.2 0.2]);
0036     elseif change<0
0037         set(vth.hchange,'ForeGroundColor',[0.165 0.384 0.275]);
0038     else
0039         set(vth.hchange,'ForeGroundColor',[0 0 0]);
0040     end
0041 end
0042
0043 figure(global_plot_options.fval_handle);
0044 semilogy(global_plot_options.fval,'o-');
0045 xlim([1,max(10, length(global_plot_options.fval))]);
0046 xlabel('Iteration');
0047 ylabel('Goodness of Fit');
0048 title(num2str(global_plot_options.fval(end)));
0049 end
```

### esePlotFigures.m

```
0001 function esePlotFigures()
0002 %ESEPLOTFIGURES - Helper function of esePlot
0003 % Author Li Sun (lsun3@emory.edu)
0004 % Last Update 04/25/2008
0005
0006 global global_spectra global_plot_options
0007 figure(global_plot_options.figure_handle);
0008
0009 n = length(global_spectra);
```

```
0010 for i=1:n
0011     %plot time domain signal
0012     subplot(n,2,2*(i-1)+1);
0013 
0014     spectrum = global_spectra{i};
0015 
0016     exp_data = spectrum.exp_data;
0017     exp_param = spectrum.exp_param;
0018 
0019      sim_x_all = 0:exp_param.Tinc:(exp_param.dpts-1)*exp_param.Tinc;
0020     hold off
0021     plot(sim_x_all*1e6, exp_data.sim_y_matched_all, 'r-');
0022     hold on
0023     plot(exp_data.x*1e6, exp_data.y,'b-');
0024     xlim([0 max(sim_x_all)*1e6]);
0025     yspan =  max(exp_data.y)-min(exp_data.y);
0026     ylim([min(exp_data.y)-0.05*yspan max(exp_data.y)+0.05*yspan]);
0027     ylabel('ESE Amplitude');
0028     if i==n
0029         xlabel('Time (\mus)');
0030     end
0031 
0032     %plot frequency domain signal
0033     subplot(n,2,2*(i-1)+2);
0034 
0035     freq_incr = 1/(2^14*exp_param.Tinc);
0036 
0037     ft_x = (0:(2^14-1))*freq_incr;
0038 
0039     sim_ft = real(fft(exp_data.sim_y_matched_all, 2^14));
0040 
0041     reconstructed_y =
[exp_data.sim_y_matched_all(1:exp_param.deadtime_dpts,1);
exp_data.y(2:end,1)];
0042     exp_ft = real(fft(reconstructed_y, 2^14));
0043 
0044     hold off
0045     plot(ft_x/1e6,sim_ft, 'r-');
0046     hold on
0047     plot(ft_x/1e6,exp_ft,'b-');
0048     ylabel('FT Amplitude');
0049 
0050     xlim([0,12]);
0051     ylim([-0.05*max(exp_ft)+min(exp_ft),1.05*max(exp_ft)]);
0052     if i==n
0053         xlabel('Frequency (MHz)');
0054     end
0055 
0056 end
0057 end
```

### esePlotInitFigures.m

```
0001 function esePlotInitFigures( opt_vector )
0002 %ESEPLOTINITFIGURES - Helper function of ESEPLOT
```

```
0003 % Author Li Sun (lsun3@emory.edu)
0004 % Last Update 04/25/2008
0005
0006 global global_spectra global_plot_options global_opt_model
0007
0008 global_plot_options.figure_handle = figure;
0009 n = length(global_spectra);
0010 set(gcf,'position',[450, 925-225*n, 800, 225*n])
0011 global_plot_options.info_handle = figure;
0012
0013 h_all = (length(global_opt_model.variables)+1)*20+35;
0014
0015 set(gcf,'position',[100, 950-h_all, 340, h_all]);
0016
0017 global_plot_options.variable_text_handles =
cell(length(global_opt_model),1);
0018
0019 y = (length(global_opt_model.variables)+1)*20;
0020 uicontrol(gcf,'Style','text', 'String','Param',...
0021     'Position',[15 y 50 15],'FontWeight','bold');
0022
0023 uicontrol(gcf,'Style','text', 'String','Index',...
0024     'Position',[65 y 50 15],'FontWeight','bold');
0025
0026 uicontrol(gcf,'Style','text', 'String','Low',...
0027     'Position',[115 y 50 15],'FontWeight','bold');
0028
0029 uicontrol(gcf,'Style','text', 'String','High',...
0030     'Position',[165 y 50 15],'FontWeight','bold');
0031
0032 uicontrol(gcf,'Style','text', 'String','Value',...
0033     'Position',[215 y 50 15],'FontWeight','bold');
0034
0035 uicontrol(gcf,'Style','text', 'String','Change',...
0036     'Position',[265 y 60 15],'FontWeight','bold');
0037
0038 for i=1:length(global_opt_model.variables)
0039     y = (length(global_opt_model.variables)+1)*20-i*20;
0040     x = 15;
0041     w = 50;
0042     h = 15;
0043
0044     uicontrol(gcf,'Style','text',
'String',global_opt_model.variables{i}.field_name,...
0045         'Position',[x y w h]);
0046
0047     uicontrol(gcf,'Style','text',
'String',num2str(global_opt_model.variables{i}.index),...
0048         'Position',[x+w y w h]);
0049
0050     variable_text_handle.hLoBound = uicontrol(gcf,'Style','text',
'String',...
0051
num2str(global_opt_model.bounds(1,global_opt_model.variables{i}.x_index
)),...
0052         'Position',[x+w*2 y w h]);
0053
```

```
0054     variable_text_handle.hHiBound = uicontrol(gcf,'Style','text',
'String',...
0055
num2str(global_opt_model.bounds(2,global_opt_model.variables{i}.x_index
)),...
0056         'Position',[x+w*3 y w h]);
0057
0058     variable_text_handle.hvalue = uicontrol(gcf,'Style','text',
'String',...
0059
num2str(opt_vector(global_opt_model.variables{i}.x_index)),...
0060         'Position',[x+w*4 y w h]);
0061
0062     variable_text_handle.hchange = uicontrol(gcf,'Style','text',
'String',...
0063         'N/A', 'Position',[x+w*5 y w+10 h]);
0064
0065     global_plot_options.variable_text_handles{i} =
variable_text_handle;
0066 end
0067
0068 global_plot_options.fval_handle = figure;
0069 set(gcf,'position',[100, 720-h_all, 340, 150]);
0070
0071 end
```

**esePlotga.m**

```
0001 function [state,options,optchanged] = esePlotga(options, state,
flag, interval)
0002 % ESEPLOT - ploting function to be suplied to the optimization ga
routine
0003
0004
0005 global global_plot_options
0006
0007 optchanged = false;
0008
0009 [min_score idx ] = min(state.Score);
0010 x = state.Population(idx,:);
0011
0012 switch flag
0013     case 'init'
0014         esePlotInitFigures(x);
0015         global_plot_options.last_x = x;
0016         global_plot_options.saved_x = {state.Population};
0017         global_plot_options.fval = min_score;
0018     case 'iter'
0019         global_plot_options.fval = [global_plot_options.fval,
min_score];
0020         esePlotFigures();
0021         esePlotDisplayInfo(x);
0022         global_plot_options.last_x = x;
0023         global_plot_options.saved_x{end+1} = state.Population;
0024     case 'done'
0025         disp('Exiting population is');
0026         disp(state.Population);
```

```
0027     otherwise
0028 end
0029
0030 end
```

### eseReport.m

```
0001 function eseReport(x, f_val, x_bound)
0002 % ESEREPORT - Prints an optimization report
0003 %   eseReport(x, f_val, x_bound)
0004 %   INPUT Parameters:
0005 %               x : optimal paramter vector
0006 %            fval : optimal fitting score
0007 %         x_bound : eseOptCftInterval output
0008
0009 global global_opt_model global_spectra
0010
0011 sc_size = get(0,'ScreenSize');
0012
0013 figure;
0014 section_top= min([970,
(220+length(global_spectra)*20+ceil((length(global_opt_model.variables)
-1)/3)*20+...
0015    length(global_opt_model.nuclei)*20)+
length(global_spectra)*260])
0016
0017
0018 set(gcf,'position',[10, sc_size(4)-section_top-60, 765,
section_top],'resize','off', 'Color',[1 1 1]);
0019 drawText(['ESEEM Simulation Report (', date, ')'], [1,
section_top-20, 765, 20], 12, 'bold', 'center');
0020
0021
0022 % nuclear parameters
0023 drawText(' Nuclear Parameters', [1, section_top-40, 765, 19], 10,
'bold', 'left');
0024
0025 nu_fields = {'I', 'g', 'Aiso', 'r', 'Axx', 'Ayy', 'Azz', 'AEuler1',
'AEuler2', 'AEuler3',...
0026         'eeqQ', 'eta', 'QEuler1', 'QEuler2', 'QEuler3'};
0027
0028 drawText('Nu #', [1, section_top-60, 45, 20], 8, 'normal',
'center');
0029 drawText('Type', [46, section_top-60, 45, 20], 8, 'normal',
'center');
0030
0031 for i=1:length(nu_fields)
0032     drawText(nu_fields{i}, [(1+i)*45, section_top-60, 45, 20], 8,
'normal', 'center');
0033 end
0034
0035
0036 for i=1:length(global_opt_model.nuclei)
0037     nu = global_opt_model.nuclei{i};
```

```
0038     drawText(num2str(i), [1, section_top-60-i*20, 45, 20], 8,
'normal', 'center');
0039     drawText(nu.type, [46, section_top-60-i*20, 45, 20], 8,
'normal', 'center');
0040     for j=1:length(nu_fields)
0041         if nu.(nu_fields{j}).fix == 1
0042             drawText(sprintf('%5.4f', nu.(nu_fields{j}).value),
[(1+j)*45, section_top-60-i*20, 45, 20], 8, 'normal', 'center');
0043         else
0044             drawText(sprintf('%5.4f', x(nu.(nu_fields{j}).idx)),
[(1+j)*45, section_top-60-i*20, 45, 20], 8, 'bold', 'center');
0045         end
0046     end
0047 end
0048
0049 % spectra
0050 section_top = section_top-60-length(global_opt_model.nuclei)*20;
0051
0052 drawText(' Experimental Spectra', [1, section_top-20, 540, 19], 10,
'bold', 'left');
0053
0054
0055 drawText('Spec #', [1, section_top-40, 45, 20], 8, 'normal',
'center');
0056 drawText('type', [46, section_top-40, 75, 20], 8, 'normal',
'center');
0057 drawText('File name', [121, section_top-40, 150, 20], 8, 'normal',
'center');
0058 drawText('tau', [271, section_top-40, 45, 20], 8, 'normal',
'center');
0059 drawText('T0', [316, section_top-40, 45, 20], 8, 'normal',
'center');
0060 drawText('Tinc', [361, section_top-40, 45, 20], 8, 'normal',
'center');
0061 drawText('dpts', [406, section_top-40, 45, 20], 8, 'normal',
'center');
0062 drawText('dt dpts', [451, section_top-40, 45, 20], 8, 'normal',
'center');
0063 drawText('B field', [496, section_top-40, 45, 20], 8, 'normal',
'center');
0064
0065 for i=1:length(global_spectra)
0066     spec_param = global_spectra{i}.exp_param;
0067     drawText(num2str(i), [1, section_top-40-i*20, 45, 20], 8,
'normal', 'center');
0068      drawText(spec_param.type, [46, section_top-40-i*20, 75, 20], 8,
'normal', 'center');
0069     drawText(spec_param.filename, [121, section_top-40-i*20, 150,
20], 8, 'normal', 'center');
0070     drawText(sprintf('%d',spec_param.tau*1e9), [271, section_top-
40-i*20, 45, 20], 8, 'normal', 'center');
0071     drawText(sprintf('%d',spec_param.T0*1e9), [316, section_top-
40-i*20, 45, 20], 8, 'normal', 'center');
0072     drawText(sprintf('%d',spec_param.Tinc*1e9), [361, section_top-
40-i*20, 45, 20], 8, 'normal', 'center');
0073     drawText(sprintf('%d',spec_param.dpts), [406, section_top-40-
i*20, 45, 20], 8, 'normal', 'center');
```

```
0074     drawText(num2str(spec_param.deadtime_dpts), [451, section_top-
40-i*20, 45, 20], 8, 'normal', 'center');
0075     drawText(sprintf('%5.4f',spec_param.B), [496, section_top-40-
i*20, 45, 20], 8, 'normal', 'center');
0076 end
0077
0078 %optimization parameters
0079 section_top = section_top-40-length(global_spectra)*20;
0080 drawText([' Optimization Parameters ', '(Minimum Q-squared = ',
num2str(f_val), ')'], [1, section_top-20, 677, 19], 10, 'bold', 'left');
0081 for k = 0:2
0082     drawText('Param', [1+k*226, section_top-40, 45, 20], 8, 'bold',
'center');
0083     drawText('Index', [46+k*226, section_top-40, 45, 20], 8,
'normal', 'center');
0084     drawText('Value', [91+k*226, section_top-40, 45, 20], 8,
'normal', 'center');
0085     drawText('Low', [136+k*226, section_top-40, 45, 20], 8,
'normal', 'center');
0086     drawText('High', [181+k*226, section_top-40, 45, 20], 8,
'normal', 'center');
0087 end
0088 for i=1:length(global_opt_model.variables)
0089     var = global_opt_model.variables{i};
0090     column = mod(i-1,3);
0091     row = floor((i-1)/3)+1;
0092     drawText(var.field_name, [1+column*226, section_top-40-20*row,
45, 20], 8, 'bold', 'center');
0093     drawText(var.index, [46+column*226, section_top-40-20*row, 45,
20], 8, 'normal', 'center');
0094     drawText(x(var.x_index), [91+column*226, section_top-40-20*row,
45, 20], 8, 'normal', 'center');
0095     drawText(x_bound(2,var.x_index), [136+column*226, section_top-
40-20*row, 45, 20], 8, 'normal', 'center');
0096     drawText(x_bound(1,var.x_index), [181+column*226, section_top-
40-20*row, 45, 20], 8, 'normal', 'center');
0097 end
0098
0099 %Spectrum Plots
0100 section_top = section_top-60-
ceil((length(global_opt_model.variables)-1)/3)*20;
0101 drawText(' Simulation Plots', [1, section_top-20, 677, 19], 10,
'bold', 'left');
0102 page = 1;
0103 for i=1:length(global_spectra)
0104     if (section_top-260)<20
0105         drawText(['Page ', num2str(page)], [1, section_top-60, 760,
20], 10, 'bold', 'Center');
0106         page = page+1;
0107         figure('Position',[10+30*page, sc_size(4)-970-60, 765,
970], 'resize','off', 'Color',[1 1 1]);
0108         drawText(['ESEEM Simulation Report (', date, ') -
Continued '], [1, 930, 765, 20], 12, 'bold', 'center');
0109         section_top = 950;
0110     end
0111
0112     drawPlot(i, section_top);
```

```
0113     section_top=section_top-260;
0114 end
0115 drawText(['Page ', num2str(page)], [1, section_top-60, 760, 20],
10, 'bold', 'Center');
0116 end
0117
0118
0119
0120 function drawText(text, pos, size, weight, align)
0121 uicontrol(gcf,'Style','text', 'String',text,...
0122     'Position',pos,'FontSize', size, 'FontWeight',weight,
'HorizontalAlignment', align, 'BackgroundColor',[1 1 1]);
0123 end
0124
0125
0126 function drawPlot(index, section_top)
0127 global global_spectra
0128 axes('unit', 'pixel','position', [50, section_top-260, 325, 220]);
0129
0130 spectrum = global_spectra{index};
0131
0132 exp_data = spectrum.exp_data;
0133 exp_param = spectrum.exp_param;
0134
0135 sim_x_all = 0:exp_param.Tinc:(exp_param.dpts-1)*exp_param.Tinc;
0136 hold off
0137 plot(sim_x_all*1e6, exp_data.sim_y_matched_all, 'r-');
0138 hold on
0139 plot(exp_data.x*1e6, exp_data.y,'b-');
0140 xlim([0 max(sim_x_all)*1e6]);
0141 yspan =  max(exp_data.y)-min(exp_data.y);
0142 ylim([min(exp_data.y)-0.05*yspan max(exp_data.y)+0.05*yspan]);
0143 ylabel('ESE Amplitude');
0144 xlabel('Time (\mus)');
0145
0146
0147 axes('unit', 'pixel','position', [425, section_top-260, 325, 220]);
0148
0149 freq_incr = 1/(2^14*exp_param.Tinc);
0150
0151 ft_x = (0:(2^14-1))*freq_incr;
0152
0153 sim_ft = real(fft(exp_data.sim_y_matched_all, 2^14));
0154
0155 reconstructed_y =
[exp_data.sim_y_matched_all(1:exp_param.deadtime_dpts,1);
exp_data.y(2:end,1)];
0156 exp_ft = real(fft(reconstructed_y, 2^14));
0157
0158 hold off
0159 plot(ft_x/1e6,sim_ft, 'r-');
0160 hold on
0161 plot(ft_x/1e6,exp_ft,'b-');
0162 ylabel('FT Amplitude');
0163
0164 xlim([0,12]);
0165 ylim([-0.05*max(exp_ft)+min(exp_ft),1.05*max(exp_ft)]);
```

```
0166 xlabel('Frequency (MHz)');
0167 legend('Simulation', exp_param.filename)
0168
0169
0170 end
```

### eseSumSpectrum.m

```
0001 function Ev = eseSumSpectrum( model, exp_param, bvecs, weights )
0002 % ESESUMSPECTRUM - Take the sum of ese spectrum under series of B
vectors
0003 %   Authur: Li Sun (lsun3@emory.edu)
0004 %
0005 % Ev = eseSumSpectrum( model, exp_param, bvecs, weights )
0006 %   INPUT Parameters:
0007 %           model : System model
0008 %       exp_param : experimental parameters
0009 %           bvecs : B vectors
0010 %         weights : weights for the B vectors
0011 %   OUTPUT Parameter:
0012 %              Ev : Summed echo amplitude
0013
0014 Ev_total = zeros(1, exp_param.dpts);
0015
0016 if exist('weights','var')
0017     for i = 1:size(bvecs,1)
0018         exp_param.B_vec = bvecs(i,:);
0019         Ev_orient = simulateSystemESE( model, exp_param );
0020         Ev_total = Ev_total + Ev_orient(3, :).*weights(i);
0021     end
0022 else
0023     for i = 1:size(bvecs,1)
0024         exp_param.B_vec = bvecs(i,:);
0025         Ev_orient = simulateSystemESE( model, exp_param );
0026         Ev_total = Ev_total + Ev_orient(3, :);
0027     end
0028 end
0029
0030 Ev = Ev_total;
```

### eseSumSpectrumFull.m

```
0001 function Ev = eseSumSpectrumFull( model, exp_param, bvecs,
weights )
0002 % ESESUMSPECTRUMFULL - Returns the sum of the full ese spectrum
under series
0003 %                       of B vectors
0004 %   Authur: Li Sun (lsun3@emory.edu)
0005 %
0006 % Ev = eseSumSpectrum( model, exp_param, bvecs, weights )
0007 %   INPUT Parameters:
0008 %           model : System model
0009 %       exp_param : experimental parameters
0010 %           bvecs : B vectors
```

```
0011 %        weights : weights for the B vectors
0012 %   OUTPUT Parameter:
0013 %        Ev(1,:) : Summed echo amplitude in alpha manifold
0014 %        Ev(2,:) : Summed echo amplitude in alpha manifold
0015 %        Ev(3,:) : Summed echo amplitude
0016
0017 Ev_total = zeros(3, exp_param.dpts);
0018
0019 if exist('weights','var')
0020     for i = 1:size(bvecs,1)
0021         exp_param.B_vec = bvecs(i,:);
0022         Ev_orient = simulateSystemESE( model, exp_param );
0023         Ev_total = Ev_total + Ev_orient.*weights(i);
0024     end
0025 else
0026     for i = 1:size(bvecs,1)
0027         exp_param.B_vec = bvecs(i,:);
0028         Ev_orient = simulateSystemESE( model, exp_param );
0029         Ev_total = Ev_total + Ev_orient;
0030     end
0031 end
0032
0033 Ev = Ev_total;
```

### filterRaw.m

```
0001 function spectrum = filterRaw( filename, B, options )
0002 %FILTERRAW - Filter raw experimental data file and, populate
exp_data and
0003 %            exp_param for further analysis
0004 %   spectrum = filterRaw( filename, data_range, filter_range )
0005 %   INPUT Parameters:
0006 %        filename : Raw experimental data file name acquired by
PEPR suite
0007 %               B : B Field in Tesla, which is not available in
saved
0008 %                   experiment raw file
0009 %         options : Filter options, with the following fields:
0010 %                   data_range - Data range for exp_data
exportation
0011 %                   filter_range - Data range for decay removal
filter
0012 %                   var_range - Data range for variance estimation
0013 %   OUTPUT Parameters:
0014 %        spectrum : A structure with the following fields
0015 %        exp_data - A struncture of filtered experimental data,
with the
0016 %                   following fields,
0017 %                   x - time axis in data_range
0018 %                   y - signal axis in data_range
0019 %                   x_raw - untruncated time axis
0020 %                   y_raw - untruncated signal data with delay
removal
0021 %                   var - signal variance estimated at var_range
```

```
0022 %       exp_param - A struncture of experimental parameter, with
the
0023 %                   following fields,
0024 %                   type - experiment type
0025 %                           '2-pulse ESEEM' or,
0026 %                           '3-pulse ESEEM'
0027 %                   tau  - tau initial value in seconds
0028 %                   Tinc - T or tau increment in seconds
0029 %                   T0   - T initial value for 3-pulse experiment
in
0030 %                          seconds
0031 %                   dpts - number of data points
0032 %  SEE ALSO simulateSystemESE, simulateNucluesESE,
simulateTDSignal
0033
0034 load(filename,'experiment_param');
0035
0036 if strcmp(experiment_param.experiment_type, '3p Pulse ESEEM')
0037     [exp_data, exp_param] = filterRaw3P( experiment_param,
options );
0038 elseif strcmp(experiment_param.experiment_type, '2p Pulse ESEEM')
0039     [exp_data, exp_param] = filterRaw2P( experiment_param,
options );
0040 end
0041
0042 exp_param.B = B;
0043 exp_param.filename = filename;
0044
0045 spectrum.exp_data = exp_data;
0046 spectrum.exp_param = exp_param;
0047
0048
0049 function exp_data = filterRawGeneric( x, y, options )
0050 % setup filter
0051 decay_filter = fittype('exp2');
0052
0053 % setup options here
0054 fit_options = fitoptions(decay_filter);
0055
0056 decay_model = fit(x(options.filter_range), y(options.filter_range),
decay_filter, fit_options);
0057
0058 exp_data.x = x(options.data_range)*1e-9; %in seconds
0059 exp_data.x_raw = x*1e-9;
0060
0061 factor = max(y);
0062 exp_data.y = (y(options.data_range)-feval(decay_model,
x(options.data_range)))/factor;
0063 exp_data.y_raw = (y-feval(decay_model, x))/factor;
0064
0065 exp_data.var = sum(((y(options.var_range)-feval(decay_model,
x(options.var_range)))/factor).^2)/length(options.var_range);
0066
0067
0068 function [exp_data, exp_param] = filterRaw3P( experiment_param,
options )
0069 % populate exp_data and exp_param for 3-pulse eseem
```

```
0070
0071 T0 = experiment_param.T_init;
0072 T_incr = experiment_param.T_incr;
0073 tau = experiment_param.tau;
0074 dpts = experiment_param.itens+round((T0+tau)/T_incr);
0075
0076 x = ((T0+tau):T_incr:T0+tau+T_incr*(experiment_param.itens-1))';
0077 y = experiment_param.phase_cycle_amplitude';
0078
0079 exp_data = filterRawGeneric(x, y, options);
0080
0081
0082 exp_param.T0 = T0*1e-9; %in seconds
0083 exp_param.Tinc = T_incr*1e-9; % in seconds
0084 exp_param.tau = tau*1e-9; % in seconds
0085 exp_param.dpts = dpts;
0086 exp_param.type = '3-pulse ESEEM';
0087 exp_param.deadtime_dpts =
round((T0+tau)/T_incr)+min(options.data_range);
0088
0089 function [exp_data, exp_param] = filterRaw2P( experiment_param,
options )
0090 % populate exp_data and exp_param for 2-pulse eseem
0091
0092 tau0 = experiment_param.tau_init;
0093 tau_incr = experiment_param.tau_incr;
0094 dpts = experiment_param.itens+round(tau/tau_incr);
0095
0096 x = (tau0:tau_incr:tau0+tau_incr*(experiment_param.itens-1))';
0097 y = experiment_param.amplitude';
0098
0099 exp_data = filterRawGeneric(x, y, options);
0100
0101 exp_param.T0 = 0;
0102 exp_param.tau = tau*1e-9; %in seconds
0103 exp_param.Tinc = tau_incr*1e-9; %in seconds
0104 exp_param.filter_options = options;
0105 exp_param.dpts = dpts;
0106 exp_param.type = '2-pulse ESEEM';
0107 exp_param.deadtime_dpts =
round(tau/tau_incr)+min(options.data_range);
```

### gensphrser.m

```
0001 function S = gensphrser( N, flag )
0002 %GenSphrSer - Generator of spherical geodesic series for shperical
averaging.
0003 %   S = GenSphrSer( N ) return the Nth level spherical geodesic
series S.
0004 %       Starting for the verteces of Octahedron/Icosahedron as
level 1.
0005 %       Subdividng the trianglal faces generates higher levels of
0006 %       shperical series.
0007
0008
```

```
0009 % octahedron
0010 % % level 1 series
0011 % p = [1 0 0;
0012 %     0 1 0;
0013 %     0 0 1;
0014 %     -1 0 0;
0015 %     0 -1 0;
0016 %     0 0 -1];
0017 %
0018 % % level 1 faces
0019 % f = [1 2 3;
0020 %     1 2 6;
0021 %     1 3 5;
0022 %     1 5 6;
0023 %     4 5 6;
0024 %     3 4 5;
0025 %     2 3 4;
0026 %     2 4 6];
0027
0028
0029 % icosahedron
0030 phi = (1+sqrt(5))/2;
0031
0032 p_unnormed = [0, +1, +phi;
0033     0, -1, +phi;
0034     0, +1, -phi;
0035     0, -1, -phi;
0036     +1, +phi, 0;
0037     -1, +phi, 0;
0038     +1, -phi, 0;
0039     -1, -phi, 0;
0040     +phi, 0, +1;
0041     +phi, 0, -1;
0042     -phi, 0, +1;
0043     -phi, 0, -1];
0044
0045 p = normalized(p_unnormed);
0046
0047 f = [1 2 11;
0048     2 8 11;
0049     2 7 8;
0050     4 7 8;
0051     4 7 10;
0052     3 4 10;
0053     3 5 10;
0054     3 5 6;
0055     1 5 6;
0056     1 6 11;
0057     1 5 9;
0058     1 2 9;
0059     2 7 9;
0060     7 9 10;
0061     5 9 10;
0062     3 4 12;
0063     3 6 12;
0064     6 11 12;
0065     8 11 12;
```

```
0066     4 8 12];
0067
0068 for i = 1:N
0069     [p,f] = geodesic(p,f);
0070 end
0071
0072
0073 % if exist('flag')
0074 %     figure;
0075 %     patch('Vertices',p,'Faces',f,...
0076 %       'FaceVertexCData',hsv(length(f)),'FaceColor','flat')
0077 %     daspect([1 1 1]);
0078 % end
0079
0080 S = p;
0081
0082
0083 function vn = normalized( v )
0084 %vn = normalized( v ) Returns a normalized row vector
0085 vn = v./sqrt(diag(v*v')*ones(1, size(v,2)));
0086
0087 function [pg, fg] = geodesic(p, f)
0088 %subdivide trianglar faces and create more points
0089 p_length = length(p);
0090 for i=1:length(f)
0091     p = [p; normalized(p(f(i,1),:)+p(f(i,2),:));
normalized(p(f(i,1),:)+p(f(i,3),:));
normalized(p(f(i,2),:)+p(f(i,3),:))];
0092 end
0093
0094 %create more faces
0095 fg = [];
0096 [pg, m, n] = unique(p,'rows');
0097
0098 for i=1:length(f)
0099     idx = p_length+(i-1)*3;
0100     fg = [fg;
0101         n(f(i,1)), n(idx+1), n(idx+2);
0102         n(idx+1), n(f(i,2)), n(idx+3);
0103         n(idx+2), n(idx+3), n(f(i,3));
0104         n(idx+1), n(idx+2), n(idx+3)];
0105 end
```

### matchSpectrum.m

```
0001 function [a,b,c,d] = matchSpectrum(simx, simy, expx, expy)
0002
0003 x0= [1,0,0,0];
0004 x = fminsearch(@(x)goodness(x,simx, simy, expx, expy), x0);
0005
0006 a = x(1);
0007 b = x(2);
0008 c = x(3);
0009 d = x(4);
0010 end
```

```
0011
0012
0013 function q = goodness(x, simx, simy, expx, expy)
0014
0015 a = x(1);
0016 b = x(2);
0017 c = x(3);
0018 d = x(4);
0019
0020 %y_m = a*(y-b)*exp(c*x)+d
0021
0022 simy_m = a.*(simy(simx)-b).*exp(c*simx)+d;
0023
0024 q = sum((expy(expx)-simy_m).^2);
0025
0026 end
```

### mwArrayConvert.m

```
0001 function out = mwArrayConvert( in )
0002 % MWARRATCONVERT - Converts Matlab native data structures into
java
0003 %                   counterparts
0004 %   Authur: Li Sun (lsun3@emory.edu)
0005 %
0006 % out = mwArrayConvert( in )
0007 %   INPUT Parameters:
0008 %              in : Matlab variable
0009 %   OUTPUT Parameter:
0010 %             out : Converted Matlab variable
0011
0012 import com.mathworks.toolbox.javabuilder.*
0013
0014 classstr = class(in);
0015
0016 switch classstr
0017     case 'double'
0018         out = MWNumericArray(in);
0019     case 'char'
0020         out = MWCharArray(in);
0021     case 'cell'
0022         [m, n] = size(in);
0023         out = MWCellArray(m, n);
0024         for i = 1:m
0025             for j = 1:n
0026                 out.set([i,j], mwArrayConvert(in{i,j}));
0027             end
0028         end
0029     case 'struct'
0030         [m, n] = size(in);
0031         fields = fieldnames(in);
0032         out = MWStructArray(m, n, fields);
0033         for i=1:m
0034             for j=1:n
0035                 for k=1:length(fields)
```

```
0036                     out.set(fields{k}, [i,j],
mwArrayConvert(getfield(in,{i,j}, fields{k})));
0037                 end
0038             end
0039         end
0040 end
```

### optStruct.m

```
0001 function opt_struct = optStruct( fix, value )
0002 opt_struct.fix = fix;
0003 opt_struct.value = value;
```

### parseNodes.m

```
0001 function nodes = parseNodes( exp )
0002 % parseNodes - parse exp string into nodes matrix
0003
0004 splits = regexp(exp, '/','split');
0005
0006 if length(splits) > 1
0007     numerator_str = splits{1};
0008     denominator_str = splits{2};
0009 else
0010     numerator_str = splits{1};
0011     denominator_str = ' ';
0012 end
0013
0014 nodes = [0 0 0 1; 0 0 0 1];
0015 nodes = generateNodesforStr(nodes, numerator_str, 1, 0, 1);
0016 nodes = generateNodesforStr(nodes, denominator_str, 2, 0, 1);
0017
0018 function nodes = generateNodesforStr(nodes, str, nodes_idx,
parent_idx, factor )
0019
0020 [parts, operator] = splits_str( str );
0021
0022 if strcmp(parts{1},' ')
0023     parts={};
0024 end
0025
0026 switch length(parts)
0027     case 0
0028         nodes(nodes_idx,:) = [-1 -1 0 factor];
0029     case 1
0030         nodes(nodes_idx,:) = [str2num(str) -1 parent_idx factor];
0031     otherwise
0032         switch operator
0033             case '+'
0034                 mode_num = 1;
0035                 children_factor = 1/length(parts);
0036             case '*'
0037                 mode_num = 0;
0038                 children_factor = 1;
```

```
0039         end
0040         nodes(nodes_idx,:) = [0, mode_num, parent_idx, factor];
0041         for i = 1:length(parts)
0042             nodes = generateNodesforStr(nodes, parts{i},
size(nodes,1)+1, nodes_idx, children_factor);
0043         end
0044
0045 end
0046
0047 function [parts, operator] = splits_str( str )
0048
0049 if strcmp(str,' ')
0050     parts = {' '};
0051     operator = ' ';
0052     return;
0053 end
0054
0055 parenthese_level = 0;
0056 operator = ' ';
0057 for i=1:length(str)
0058     if parenthese_level == 0
0059         switch str(i)
0060             case '+'
0061                 operator = '+';
0062                 break;
0063             case '*'
0064                 operator = '*';
0065             case '('
0066                 parenthese_level = parenthese_level+1;
0067         end
0068     else
0069        switch str(i)
0070             case '('
0071                 parenthese_level = parenthese_level+1;
0072             case ')'
0073                 parenthese_level = parenthese_level-1;
0074        end
0075     end
0076 end
0077
0078 parts = {};
0079 parenthese_level = 0;
0080 new_part = '';
0081
0082 for i=1:length(str)
0083     p = str(i);
0084     if parenthese_level ==0
0085         switch str(i)
0086             case operator
0087                 if new_part(1) == '(' && new_part(end) == ')'
0088                     new_part = new_part(2:end-1);
0089                 end
0090                 parts = {parts{:} new_part};
0091                 new_part = '';
0092             case '('
0093                 parenthese_level = parenthese_level+1;
0094                 new_part(end+1) = str(i);
```

```
0095             otherwise
0096                 new_part(end+1) = str(i);
0097         end
0098     else
0099         new_part(end+1) = str(i);
0100         switch str(i)
0101             case '('
0102                 parenthese_level = parenthese_level+1;
0103             case ')'
0104                 parenthese_level = parenthese_level-1;
0105         end
0106     end
0107
0108 end
0109 if new_part(1) == '(' && new_part(end) == ')'
0110     new_part = new_part(2:end-1);
0111 end
0112 parts = {parts{:} new_part};
```

### powderAverager.m

```
0001 function Ev = powderAverager ( model, exp_param,  bvec )
0002 % POWDERAVERAGER - Returns the powder averaged specturm
0003 %   Authur: Li Sun (lsun3@emory.edu)
0004 %
0005 % Ev = powderAverager ( model, exp_param, average_level )
0006 %   INPUT Parameters:
0007 %             model : System model
0008 %         exp_param : experimental parameters
0009 %              bvec : B vectors
0010 %   OUTPUT Parameter:
0011 %               Ev : Averaged echo amplitude
0012
0013
0014 Ev_total = zeros(1,exp_param.dpts);
0015
0016 for i = 1:length(bvec)
0017     exp_param.B_vec = bvec(i,:);
0018     Ev_orient = simulateSystemESE( model, exp_param );
0019     Ev_total = Ev_total + Ev_orient(3, :);
0020 end
0021
0022 Ev = Ev_total./size(bvec,1);
```

### powderAveragerDistributed.m

```
0001 function Ev = powderAveragerDistributed ( model, exp_param, bvec )
0002 % POWDERAVERAGERDISTRIBUTED - Distributed version of system
sprecta powder
0003 %                              averager
0004 %   Authur: Li Sun (lsun3@emory.edu)
0005 %   Last Update: 04/25/2008
0006 %
0007 % Ev = powderAveragerDistributed ( model, exp_param, bvec )
```

```
0008 %   INPUT Parameters:
0009 %             model : System model, which should have a nuclei
field and a
0010 %                     nodes field.
0011 %         exp_param : Expreimental parameters for simulation
0012 %              bvec : nomalized n x 3 directional vector of B
field
0013 %   OUTPUT Parameter:
0014 %               Ev : Sum of spectrum of B field directions
0015 %
0016 %   Note: Global variable DistEseManager has to be set before
calling this
0017 %         funtion.
0018
0019 global DistEseManager
0020
0021 Ev_total = zeros(1,exp_param.dpts);
0022
0023
0024 nodeServerNames = DistEseManager.listServers;
0025 serverNumber = nodeServerNames.size;
0026
0027 cp = DistEseManager.getServerCompetence;
0028 lt = DistEseManager.getServerLatency;
0029
0030 w = size(bvec,1);
0031 nodeforservers = round(cp.*(w+sum(lt))/(sum(cp))-lt);
0032
0033
0034
0035 nodeServer = {};
0036 response_token = {};
0037
0038
0039 model_mw = mwArrayConvert(model);
0040 exp_param_mw = mwArrayConvert(exp_param);
0041
0042
0043 lvecindex = 1;
0044 for i=0:serverNumber-1
0045     nodeServer{i+1} =
java.rmi.Naming.lookup(nodeServerNames.get(i));
0046     if i~=serverNumber-1
0047         bvec_mw =
mwArrayConvert(bvec(lvecindex:sum(nodeforservers(1:i+1)),:));
0048     else
0049         bvec_mw = mwArrayConvert(bvec(lvecindex:end,:));
0050     end
0051
0052     lvecindex = lvecindex+nodeforservers(i+1);
0053     response_token{i+1} =
nodeServer{i+1}.asEseSumSpectrum( model_mw, exp_param_mw, bvec_mw );
0054
com.mathworks.toolbox.javabuilder.MWArray.disposeArray(bvec_mw);
0055
0056 end
0057
```

```
0058 for i=0:serverNumber-1
0059     response = nodeServer{i+1}.retrieve(response_token{i+1});
0060     Ev_total = Ev_total+response(1);
0061     nodeServer{i+1}.disposeResult(response_token{i+1});
0062 end
0063
0064 com.mathworks.toolbox.javabuilder.MWArray.disposeArray(model_mw);
0065
com.mathworks.toolbox.javabuilder.MWArray.disposeArray(exp_param_mw);
0066
0067
0068 Ev = Ev_total./size(bvec,1);
```

### registerOptModelVariable.m

```
0001 function opt_model = registerOptModelVariable ( opt_model,
varargin )
0002 % REGISTERMODELVARIABLE - Register a variable in the optimization
model structure
0003 %   Authur: Li Sun (lsun3@emory.edu)
0004 %   Last Update: 04/24/2008
0005 %
0006 %   opt_model = registerOptModelVariable ( opt_model,  field_name,
index,
0007 %   x_index, lo_bound, hi_bound )
0008 %   INPUT Parameters:
0009 %         opt_model : Optimization model structure to register to
0010 %        field_name : Parameter to register as an optimization
variable
0011 %                     Values can be:
0012 %                        'I', 'g', 'Aiso', 'r', 'AEuler1',
'AEuler2',
0013 %                        'AEuler3', 'Axx', 'Ayy', 'Azz',
0014 %                        'eeqQ', 'eta', 'QEuler1', 'QEuler2',
'QEuler3'
0015 %                     for Nulear parameters, or
0016 %                        'factor'
0017 %                     for additive nodes factors.
0018 %             index : Index of nuclues for Nulear parameters, or
0019 %                     index of nodes for factor.
0020 %           x_index : index in the input vector for use
0021 %          lo_bound : Lower bound of the variable
0022 %          hi_bound : Higher bound of the variable
0023 %
0024 %   Note: x_index can be the same for two variables, lower bound
and higher
0025 %         bound will be the last registered value.
0026 %         registerOptModelVariable does not change the value for
the parameter.
0027 %
0028 %   OUTPUT Parameter:
0029 %         opt_model : Updated model structure for optimization.
0030 %
0031 %   SEE ALSO changeOptModelValue, eseOptModel
0032 k = size(varargin,2)/5;
```

```
0033
0034 for i=1:k
0035     field_name = varargin{(i-1)*5+1};
0036     index = varargin{(i-1)*5+2};
0037     x_index = varargin{(i-1)*5+3};
0038     lo_bound = varargin{(i-1)*5+4};
0039     hi_bound = varargin{(i-1)*5+5};
0040
0041     if strcmp(field_name,'factor')
0042         opt_model.nodes.factors{index}.fix = 0;
0043         opt_model.nodes.factors{index}.idx = x_index;
0044     else
0045         opt_model.nuclei{index}.(field_name).fix = 0;
0046         opt_model.nuclei{index}.(field_name).idx = x_index;
0047     end
0048
0049     opt_model.bounds(:, x_index) = [lo_bound; hi_bound];
0050
0051     variable.index = index;
0052     variable.field_name = field_name;
0053     variable.x_index = x_index;
0054     opt_model.variables{end+1} = variable;
0055 end
```

### setnucleus.m

```
0001 function nu = setnucleus( values, varin2, nu_in )
0002 %SETNUCLEUS - set nucleus structure
0003 %   INPUT parameters:
0004 %     nu = setnucleus( values, param_vector, nu_in )
0005 %    param_vector : Index vector of paramters for change.
0006 %          values : Value vector of paramters for caange.
0007 %           nu_in : Input Nucleus paramter structure for change.
0008 %     nu = setnucleus( values, type )
0009 %          values : Index vector of paramters
0010 %            type : 'Aiso' for Aiso and r
0011 %                   'tensor' for Axx, Ayy, Azz
0012 %   OUTPUT parameter:
0013 %        nu - Nucleus paramter structure, which has the following
fields
0014 %               I : Half Integral Nuclear Spin.
0015 %                   Dimensionless. Index: 1.
0016 %                   N14 - 1
0017 %                   N15 - 1/2
0018 %                   H2  - 1
0019 %                   H1  - 1/2
0020 %               g : Nuclear G-value.
0021 %                   Dimensionless. Index: 2.
0022 %                   N14 - 0.4038
0023 %                   N15 - -0.5664
0024 %                   H2  - 0.8574
0025 %                   H1  - 5.5857
0026 %            Aiso : Isotropic hyperfine cupling.
0027 %                   Unit: MHz. Index: 3.
0028 %                r : Distance between unpaired electron and nucleus.
```

```
0029 %                    Unit: Angstrom. Index: 4.
0030 %              Axx : Hyperfine coupling tensor xx part.
0031 %                    Unit: MHz. Index: 5.
0032 %              Ayy : Hyperfine coupling tensor yy part.
0033 %                    Unit: MHz. Index: 6.
0034 %              Azz : Hyperfine coupling tensor zz part.
0035 %                    Unit: MHz. Index: 7.
0036 %                    (Note : [Asio, r] and [Axx, Ayy, Azz] are
converted to
0037 %                     their counterpart automatically.)
0038 %           AEuler : Euler angles of hyperfine coupling tensor.
0039 %                    Unit: Rad. Index 8, 9, 10.
0040 %             eeqQ : Qudrapole interaction paramter.
0041 %                    Unit: MHz. Index 11.
0042 %              eta : Anisotropic factor of qudrapole interaction
tensor.
0043 %                    Dimensionless. 0~1. Index: 12.
0044 %           QEuler : Euler angles of hyperfine coupling tensor.
0045 %                    Unit: Rad. Index 13, 14, 15.
0046 %   EXAMPLE:
0047 %      nu = setnucleus([1, 5.5], [1,2], nu_in);
0048 %        will result:
0049 %           nu.I = 1
0050 %           nu.g = 5.5
0051 %
0052 %   See also GETNUCLEUS.
0053
0054 if nargin == 3
0055     nu = nu_in;
0056     param_vector = varin2;
0057 end
0058
0059 if nargin == 2
0060     if strcmp(varin2, 'Aiso')
0061         param_vector = [1:4, 8:15];
0062     elseif strcmp(varin2, 'tensor')
0063         param_vector = [1,2, 5:15];
0064     else
0065         error(strcat('Invalid type ', varin2));
0066     end
0067 end
0068
0069 Aiso_flag = 0;
0070 A_tensor_flag = 0;
0071
0072 for i = 1:length(param_vector)
0073     switch (param_vector(i))
0074         case 1
0075             nu.I = values(i);
0076         case 2
0077             nu.g = values(i);
0078         case 3
0079             nu.Aiso = values(i);
0080             Aiso_flag = 1;
0081         case 4
0082             nu.r = values(i);
0083             Aiso_flag = 1;
```

```
0084          case 5
0085              nu.Axx = values(i);
0086              A_tensor_flag = 1;
0087          case 6
0088              nu.Ayy = values(i);
0089              A_tensor_flag = 1;
0090          case 7
0091              nu.Azz = values(i);
0092              A_tensor_flag = 1;
0093          case 8
0094              nu.AEuler(1) = values(i);
0095          case 9
0096              nu.AEuler(2) = values(i);
0097          case 10
0098              nu.AEuler(3) = values(i);
0099          case 11
0100              nu.eeqQ = values(i);
0101          case 12
0102              nu.eta = values(i);
0103          case 13
0104              nu.QEuler(1) = values(i);
0105          case 14
0106              nu.QEuler(2) = values(i);
0107          case 15
0108              nu.QEuler(3) = values(i);
0109      end
0110  end
0111
0112  %Adip = (mu_null/(4*Pi*h))*ge*beta_e*gn*beta_n * r^3
0113  %factor = (mu_null/(4*Pi*h))*ge*beta_e*beta_n
0114  factor = 14.1387332;
0115
0116  %[Axx Ayy Azz] = [Aiso-Adip, Aiso_Adip, Asio+2Adip]
0117  m = [1 1 1; -1 -1 2];
0118
0119  if Aiso_flag == 1
0120      Adip = factor*nu.g/nu.r^3;
0121      A = [nu.Aiso, Adip] * m;
0122      nu.Axx = A(1);
0123      nu.Ayy = A(2);
0124      nu.Azz = A(3);
0125  elseif A_tensor_flag == 1
0126      A_temp = [nu.Axx, nu.Ayy, nu.Azz] / m;
0127      nu.Aiso = A_temp(1);
0128      nu.r = (factor*nu.g/A_temp(2))^(1/3);
0129  end
```

### simulateNucluesESE.m

```
0001 function Ev = simulateNucluesESE( nu, exp_param )
0002 %SIMULATENUCLUESESE simulates ESE spectrum for a single nuclues
0003 %   Ev = simulateNucluesESE( nu, exp_param )
0004 %   INPUT Parameters:
0005 %               nu : nuclues structure
0006 %        exp_param : experimental parameters
```

```
0007 %   OUTPUT Parameters:
0008 %          Ev(1,:) : ESE signal in alpha manifold
0009 %          Ev(2,:) : ESE signal in beta manifold
0010 %          Ev(3,:) : Average of Ev(1,:) and Ev(2,:)
0011 %SEE ALSO simulateTDSignal, simulateSystemESE
0012
0013 H = cnstrctHamiltonian(nu, exp_param.B, exp_param.B_vec);
0014
0015 Ev = simulateTDSignal(exp_param, H);
```

### simulateSystemESE.m

```
0001 function Ev = simulateSystemESE( model, exp_param )
0002 %%SIMULATESYSTEMESE simulates ESE spectrum for a system model
0003 %   Ev = simulateSystemESE( model, exp_param )
0004 %   INPUT Parameters:
0005 %            model : system model, should have the following
fields:
0006 %                        model.nodes : n by 4 array of
0007 %                      [nuclues_number mode parent_node factor]
0008 %                        nuclues_number : indices in model.nuclues
0009 %                                  mode : -1 for nuclues node
0010 %                                          0 for muliplication
node
0011 %                                         +1 for addition node
0012 %                           parent_node : parent node
0013 %                                factor : a factor for addition
node
0014 %                      model.nuclues : cell of structures of
nuclues
0015 %                                      structures
0016 %        exp_param : experimental parameters, in addition to the
specified
0017 %                    in simulateTDSignal.m, exp_param should have:
0018 %                        exp_param.B : B field in T
0019 %                    exp_param.B_vec : B directional vector
0020 %   OUTPUT Parameters:
0021 %          Ev(1,:) : ESE signal in alpha manifold
0022 %          Ev(2,:) : ESE signal in beta manifold
0023 %          Ev(3,:) : Average of Ev(1,:) and Ev(2,:)
0024 %SEE ALSO simulateTDSignal, simulateNucluesESE
0025
0026
0027 global global_nodes_Ev globle_model globle_exp_param
0028
0029 globle_model = model;
0030 globle_exp_param = exp_param;
0031
0032 global_nodes_Ev = cell(1,length(model.nodes));
0033
0034 simulateSpectrumForNode(1);
0035
0036 if model.nodes(2,1) == -1
0037     Ev = global_nodes_Ev{1};
0038 else
```

```
0039     simulateSpectrumForNode(2);
0040     Ev = combineSpectra([1, 2], 2);
0041 end
0042
0043 clear global global_nodes_Ev globle_model globle_exp_param
0044
0045 function simulateSpectrumForNode(node_number)
0046 % This is a recursive funtion that generate spectra for each node
0047 global  global_nodes_Ev globle_model globle_exp_param
0048
0049 op_mode = globle_model.nodes(node_number, 2);
0050
0051 if op_mode == -1
0052     global_nodes_Ev{node_number} =
simulateNucluesESE(globle_model.nuclei{globle_model.nodes(node_number,1
)}, globle_exp_param);
0053 else
0054     childrenNodes = find(globle_model.nodes(:,3)==node_number);
0055     for i = 1:length(childrenNodes)
0056        simulateSpectrumForNode(childrenNodes(i));
0057     end
0058     global_nodes_Ev{node_number} = combineSpectra(childrenNodes,
op_mode);
0059 end
0060
0061 function Ev = combineSpectra(nodes, op)
0062 % op can be '*' : 0
0063 %           '+' : 1
0064 %           '/' : 2
0065 global global_nodes_Ev globle_model
0066
0067 switch op
0068     case 0
0069         Ev_alpha = global_nodes_Ev{nodes(1)}(1,:);
0070         Ev_beta = global_nodes_Ev{nodes(1)}(2,:);
0071
0072         for i = 2:length(nodes)
0073             Ev_alpha = Ev_alpha.*global_nodes_Ev{nodes(i)}(1,:);
0074             Ev_beta = Ev_beta.*global_nodes_Ev{nodes(i)}(2,:);
0075         end
0076
0077     case 1
0078         factor = globle_model.nodes(nodes(1), 4);
0079         Ev_alpha = global_nodes_Ev{nodes(1)}(1,:)*factor;
0080         Ev_beta = global_nodes_Ev{nodes(1)}(2,:)*factor;
0081
0082         for i = 2:length(nodes)
0083             factor = globle_model.nodes(nodes(i), 4);
0084             Ev_alpha = Ev_alpha +
global_nodes_Ev{nodes(i)}(1,:)*factor;
0085             Ev_beta = Ev_beta +
global_nodes_Ev{nodes(i)}(2,:)*factor;
0086         end
0087
0088     case 2
0089         Ev_alpha = global_nodes_Ev{nodes(1)}(1,:);
0090         Ev_beta = global_nodes_Ev{nodes(1)}(2,:);
```

```
0091
0092         for i = 2:length(nodes)
0093             Ev_alpha = Ev_alpha./global_nodes_Ev{nodes(i)}(1,:);
0094             Ev_beta = Ev_beta./global_nodes_Ev{nodes(i)}(2,:);
0095         end
0096 end
0097
0098 Ev = [ Ev_alpha; Ev_beta; (Ev_alpha+Ev_beta)/2 ];
```

### simulateTDSignal.m

```
0001 function Ev = simulateTDSignal( exp_param, H )
0002 %SIMULATETDSIGNAL - Simulate time domain signal from Hamitonial H
and
0003 %                   experimental parameter exp_param
0004 %   Ev = simulateTDSignal( exp_param, H )
0005 %   INPUT parameters:
0006 %       H         : H{1} Hamitonial in alpha manifold
0007 %                   H{2} Hamitonial in beta manifold
0008 %       exp_param : A struncture of experimental parameter, with
the
0009 %                   following fields,
0010 %                   type - experiment type
0011 %                          '2-pulse ESEEM' or,
0012 %                          '3-pulse ESEEM'
0013 %                   tau  - tau initial value in seconds
0014 %                   Tinc - T or tau increment in seconds
0015 %                   T0   - T initial value for 3-pulse experiment
in
0016 %                          seconds
0017 %                   dpts - number of data points
0018 %   OUTPUT Parameters:
0019 %          Ev(1,:) : ESE signal in alpha manifold
0020 %          Ev(2,:) : ESE signal in beta manifold
0021 %          Ev(3,:) : Average of Ev(1,:) and Ev(2,:)
0022 %  SEE ALSO simulateSystemESE, simulateNucluesESE
0023
0024 [Ma, Va] = getFreqsfromH(H{1});
0025
0026 [Mb, Vb] = getFreqsfromH(H{2});
0027
0028 M = Ma'*Mb;
0029
0030 Ev = zeros(exp_param.dpts,1);
0031
0032 if strcmp(exp_param.type, '3-pulse ESEEM')
0033     Ev = generate3PSignal( exp_param, M, diag(Va), diag(Vb) );
0034 end
0035
0036
0037 function [M, V] = getFreqsfromH( H )
0038
0039 hbar=1.054572*10^(-34); %JS
```

```
0040
0041 [evecs, evals] = eig( H );
0042
0043 vals = real(diag(evals));
0044 [vals, indx] = sort(vals, 'descend');
0045
0046
0047 V = zeros(size(H));
0048 M = zeros(size(H));
0049
0050 for row = 1:size(H,1)
0051     V(row, row) = vals(row)/hbar;
0052     M(:, row) = evecs(:, indx(row));
0053 end
0054
0055
0056 function Ev = generate3PSignal( exp_param, M, Va, Vb )
0057 %generate time domain signal for 3-pulse experiment
0058
0059 N = length(M);
0060 N2 = N*N;
0061 N3 = N2*N;
0062 N4 = N3*N;
0063 Mp = M';
0064
0065 tau = exp_param.tau;
0066
0067 An=zeros(2*N4,1); % make a col vector
0068 wn=zeros(2*N4,1); % make a col vector
0069 t1n=zeros(2*N4,1); % make a col vector
0070
0071 vN=1:N;
0072 mdim=0;
0073
0074 for m=vN % m
0075     kmdim=mdim;
0076     for k=vN % k
0077         if(M(k,m)~=0)
0078             klmdim=kmdim;
0079             for l=vN % l
0080                 if(M(k,l)~=0)
0081                     wbml=Vb(m)-Vb(l);
0082                     eiwbml=exp(i*wbml*tau);
0083                     Amkl=Mp(m,k)*M(k,l);
0084                     for j=vN % j
0085                         if(M(j,m)~=0)
0086                             if(M(j,l)~=0)
0087                                 n=2*j-1+klmdim;
0088                                 wajk=Va(j)-Va(k);
0089                                 efac=exp(i*wajk*tau)*eiwbml;
0090                                 %   sum_kljm Mkl M*km Mjm M*jl
exp[- i wakj (tau+T) - i wblm tau] (G1) Term1 n->m
0091                                 % + sum_lkmj M*kl Mkm M*jm Mjl
exp[- i wakj tau - i wblm (tau+T)] (G2) Term2 n->m
0092                                 An(n)=M(j,m)*Amkl*Mp(l,j);
0093                                 An(n+1)=An(n)';
0094                                 An(n)=An(n)*efac;
```

```
0095                                    An(n+1)=An(n+1)*efac;
0096                                    wn(n)=wajk;
0097                                    wn(n+1)=wbml;
0098                                    t1n(n)=1;
0099                                    t1n(n+1)=0;
0100                                    % n   is Term1 and has t1n=1
0101                                    % n+1 is Term2 and has t1n=0
0102                                end;
0103                            end;
0104                        end; %j
0105                    end;
0106                    klmdim=klmdim+2*N;
0107                end; % l
0108            end;
0109            kmdim=kmdim+2*N2;
0110        end; % k
0111        mdim=mdim+2*N3;
0112 end; % m
0113
0114
0115 idx_a = find(An~=0 & t1n==1);
0116 idx_b = find(An~=0 & t1n==0);
0117
0118 Anf_a = An(idx_a);
0119 wnf_a = wn(idx_a);
0120
0121 Anf_b = An(idx_b);
0122 wnf_b = wn(idx_b);
0123
0124 [Aa, wa, DCa] = combineAmplitude(Anf_a, wnf_a);
0125 [Ab, wb, DCb] = combineAmplitude(Anf_b, wnf_b);
0126
0127 %T = (0:(exp_param.dpts-1))*exp_param.Tinc+exp_param.T0; %not
correct
0128 T = (0:exp_param.Tinc:(exp_param.dpts-1)*exp_param.Tinc)-
exp_param.tau;
0129
0130 EvA = real(Aa).'*cos(wa*T)-imag(Aa).'*sin(wa*T)+DCa;
0131 EvB = real(Ab).'*cos(wb*T)-imag(Ab).'*sin(wb*T)+DCb;
0132
0133 Ev = [EvA; EvB; (EvA+EvB)/2];
0134
0135 function [A, w, dc] = combineAmplitude( An, wn )
0136 dc = sum(real(An(wn == 0)));
0137
0138
0139 An_nonDC = An(wn ~= 0);
0140 wn_nonDC = wn(wn ~= 0);
0141
0142
0143 negative_freq = find(wn_nonDC<0);
0144
0145 An_nonDC_posFreq = An_nonDC;
0146 An_nonDC_posFreq(negative_freq) =
conj(An_nonDC_posFreq(negative_freq));
0147
0148 wn_nonDC_posFreq = wn_nonDC;
```

```
0149 wn_nonDC_posFreq(negative_freq) = -wn_nonDC_posFreq(negative_freq);
0150
0151 [wnfx, dummy, idx] = unique(wn_nonDC_posFreq);
0152
0153 Anfx = zeros(size(wnfx));
0154 for i = 1:length(wnfx)
0155     Anfx(i) = sum(An_nonDC_posFreq(find(idx == i)));
0156 end;
0157
0158 A = Anfx(Anfx~=0);
0159 w = wnfx(Anfx~=0);
```