Statistical Study of the Decay Lifetimes of the Photo-excited DNA Nucleobase Adenine

by

Zhuofei Hou

(Under the direction of Jaxk Reeves)

Abstract

To provide physicists, Nicholas L. Evans and Susanne Ullrich, at the University of Georgia (UGA) with the statistical support to determine the decay lifetimes of the photo-excited DNA nucleobase Adenine, which is one of the four basic DNA nucleobases, we developed and applied a set of statistical analysis methods, including background signal analysis, inverse-variance-weighted Gaussian fitting, Gaussian-weighted summation, and non-linear regressions with the *Single-*τ model in the long-lived and short-lived channels of TRPES experiments of Adenine, through applying SAS procedures and C++ computer programming. For one TRPES data set, we obtained the decay lifetimes $\tau_1 = 884\,fs$ for the long-lived channel and $\tau_2 = 67\,fs$ for the the short-lived channel, both of which are in a good agreement with experimental results of $\tau_1 = (880 \pm 50)\,fs$ and $\tau_2 = (70 \pm 30)\,fs$ [1]. With another TRPES data set, we validated our methods, obtaining estimates of $\tau_1 = 917\,fs$ and $\tau_2 = 91\,fs$.

Index words: UV photostability; photo-excited DNA nucleobase; Adenine; decay lifetime; time constant; TRPES experiments; electronic relaxation pathways; background signal subtraction; inverse-variance-weighted Gaussian fitting; Gaussian weighted fitting; non-linear regression

Statistical Study of the Decay Lifetimes of the Photo-excited DNA Nucleobase Adenine

by

Zhuofei Hou

B.S., Nankai University, Tianjin, China, 1998

A Thesis Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Master of Science

Athens, Georgia

2010

Statistical Study of the Decay Lifetimes of the Photo-excited DNA Nucleobase Adenine

by

Zhuofei Hou

Approved:

Major Professor: Jaxk Reeves

Committee: Abhyuday Mandal
Susanne Ullrich

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2010

Dedication

This paper is dedicated to my wife Lijun, my daughter Sophia, and my parents.

Acknowledgments

I would first thank my major advisor, Prof. Jaxk Reeves, for providing insightful advice with constant encouragement, strong support, and great patience throughout my entire research and academic years. I have learned to analyze problems thoroughly and pay attention to details. I remember his words so well: "We should always guarantee that good-looking results are not the results of wrong reasoning". This statement impressed me a lot.

I would also like to thank both the other members of my advisory committee, Profs. Abhyuday Mandal and Susanne Ullrich, for their insights and perspective as specialists in their fields and for spending time serving as my committee members. Prof. Abhyuday Mandal gave me a lot of encouragement and help in my academic endeavors in Statistics study, especially in his Statistics class STAT 6800. Prof. Susanne Ullrich provided this project with all experimental designs and data. She also helped me a lot to understand the background knowledge and principles in Physics needed for this project.

I should also thank my church family who gave me and my family tremendous help and encouragement through years.

I thank my parents and my parents-in-law for being supportive all the time. Finally, in a very inadequate manner, I want to thank my wife and my dear daughter for standing with me as a family.

Table of Contents

Appendix

List of Figures

List of Tables

Chapter 1

Introduction

Electronic relaxation pathways in photo-excited DNA nucleobases have received much theoretical and experimental attention due to their underlying importance to the ultraviolet (UV) photostability of these biomolecules [1]. Physicists, Nicholas L. Evans and Susanne Ullrich, from the Department of Physics and Astronomy at the University of Georgia are interested in the UV photostability of one of those biomolecules: Adenine, one of the four basic DNA nucleobases. This molecule fights against destructive photochemical processes through the so-called ultra-fast deactivation pathways. These physicists designed experiments to study how these deactivation pathways work, but they need statistical help in analyzing the large data sets they have obtained from their experiments. In their experiments, they use a pump pulse to excite an electronic state in the bio-molecules, followed by a probe pulse for ionization. They detect the electrons emitted and measure the flight time of electrons from the ionization region to the detector. This flight time is labeled as TOF or $Time\ of\ Flight$. They set the probes to occur at different intervals relative to the pulse pump, calling this interval the pump-probe delay time, which is labeled as $Delay$ or $DelayTime$. From their experiments, they obtained so-called femtosecond ($1 femtosecond = 1 fs = 10^{-15} seconds$) time-resolved photoelectron spectra (TRPES) of the DNA nucleobase Adenine, which are two-dimensional data (TRPES data) as a function of pump-probe delay time and time of flight. They want to analyze the non-linear relationship between the pump-probe delay time and the intensity signals.

Physicists are expecting to fit the data as a bi-exponential decay, i.e. an exponential decay with two decay lifetimes. Through this fitting, they wish to find the decay time constants

in a decaying relationship between the pump-probe delay time and the intensity signals in two relaxation pathways, i.e. the so-called long-lived channel (LLC) and short-lived channel (SLC) of TRPES experiments. These decay time constants are actually the decay lifetimes of the photo-excited states of Adenine that they are investigating. Determining these lifetimes and decay associated photoelectron spectra are the overall goals of their experiments.

In this work, to study the decay lifetimes of the photo-excited states of Adenine, we developed and applied a set of statistical methods of analysis, including background signal analysis, inverse-variance-weighted Gaussian fitting, Gaussian-weighted summation, and multivariate linear/non-linear regression analysis. These analyses were conducted using statistical analysis software SAS, computational software Mathematica, and computer programming in the C++ programming language.

The remainder of this document is as follows. Experimental and theoretical backgrounds are presented in Chapter 2. Statistical analysis methods are described in Chapter 3. In Chapter 4, statistical analysis results for the decay lifetimes of the photo-excited states of Adenine are reported. Finally, the thesis conclusions are presented in Chapter 5. The appendices A, B, and C contain fully annotated versions of all the software that has been developed.

Chapter 2

Background

The work in this thesis is an extension of the work previously accomplished on the statistical consulting project "Decay Times for Genetic Bases" [2], which was provided as a project for the course of Supervised Statistical Consulting (STAT 8000) in the Summer of 2007 at the Department of Statistics at the University of Georgia. As our consulting clients, two physicists, Nicholas L. Evans and Susanne Ullrich, from the Department of Physics and Astronomy at the University of Georgia provided the initial data for this project. In that project, we performed basic statistical investigations, from which we obtained some preliminary results on the decay lifetimes of the photo-excited states of Adenine and in the light of which we began the study under discussion in this thesis. In Section 2.3, a brief review of our previous work can be found.

In an effort to measure the decay lifetimes of the photo-excited states of Adenine more precisely and completely, instead of the original data set (*Adenine Canada*) and *ab initio* methods we used previously in 2007, two new TRPES data sets from the latest TRPES experiments and newly developed statistical analysis methods are applied in this work. A detailed explantation of the new TRPES data sets (*Adenine 803153* and *Adenine 8030651*) can be found in Section 2.1. In Section 2.2, we provide an introduction to the theoretical background to predict the decay lifetimes of the photo-excited states of Adenine. An introduction to the statistical analysis methods used in the current study can be found in Chapter 3. To further illustrate the improvements we performed in the current study, two major drawbacks of the previous study are discussed in Section 2.4.

2.1 Experimental Design

2.1.1 TRPES Experiments

In this work, we analyze two data sets obtained from TRPES experiments of Adenine. Physicists [1] at UGA set 151 different pump-probe delay time values, from $-1565fs$ to $5935fs$ in $50fs$ steps for one data set, and from $-1242fs$ to $6258fs$ in $50fs$ steps for the second data set. For the convenience of study, these 151 different pump-probe delay time values are labeled as 151 indices, which are called as $Delay$ or $DelayTime$, running from 0 to 150 in steps of 1. Each step of $Delay$ represents $50fs$. For each pump-probe delay time, physicists measured the intensity signals at 4000 electron kinetic energy bins running nonlinearly from $0eV$ to $4.00eV$. Those 4000 electron energy bins are also labeled as 4000 indices, which are called as TOF or $TimeofFlight$, ranging from 0 to 3999 in steps of 1. A summary of these physical and indexed quantities in TRPES experiments can be found in Table 2.1 in Section 2.1.2.

According to [1], at the photo-excited states of Adenine, the relationship between the electron kinetic energy (E) and TOF is shown in Eqn. 2.1 as follows

$$E = \frac{IR}{(t - t_0)^2} - E_0, \tag{2.1}$$

where t is the real time of flight of photoelectrons in TRPES experiments with $t = TOF$ (in units of $10^{-9}s$). The quantities IR, t_0 and E_0 are quantities determined by the geometry of the experiments, with the values of $IR = 9.2329 \times 10^{-13}eV \cdot s^2$, $t_0 = 9.8341 \times 10^{-9}s$ and $E_0 = 3.9569 \times 10^{-1}eV$.

The relationship between the electronic binding energy (EBE) and the electron kinetic energy (E) is given in Eqn. 2.2 as follows

$$EBE = E_{total} - E, \tag{2.2}$$

where EBE is the electronic binding energy in the unit of eV. E is the physical quantity measured in TRPES experiments. $E_{total} = 11.09eV$ is the total photon energy, which is

the sum of the pump and probe energies used to first excite and then ionize the Adenine molecules. Combining Eqn. 2.1 and Eqn. 2.2, we show the relationship between t, i.e. the real time of flight of excited electrons, and EBE, i.e. the total energy to ionize Adenine into cationic states as follows

$$t = t_0 + \sqrt{\frac{IR}{E_{total} + E_0 - EBE}} \tag{2.3}$$

Thus, the relationship between TOF and EBE is given by the following equation:

$$TOF = 9.8341 + \frac{960.8798}{\sqrt{11.4857 - EBE}} \tag{2.4}$$

In summary, the experiment measures electron TOF but we convert it to electron kinetic energy (E) or electronic binding energy (EBE) in photoelectron spectrum through Eqn. 2.1 and Eqn. 2.2. In this thesis, we refer to our two-dimensional data set as being in $TOF \times Delay$ scale.

2.1.2 TRPES Data Sets

The data sets of Adenine from TRPES experiments were labeled by physicists [Evans, et al.] in 2008 at UGA with serial numbers of 803153 and 8030651. Thus, we call those data sets as *Adenine 803153* and *Adenine 8030651*. A detailed summary of the main characteristics of *Adenine 803153* and *Adenine 8030651* can be found in Table 2.1. Explanation of the quantities *FWHM* (*Full Width at Half Maximum*) and σ (experimentally measurable standard deviation) in Table 2.1 can be found in Section 2.1.3, and an explanation of the scaling factor in Table 2.1 can be found in Section 3.1.1.

The physicists scanned through 151 different pump-probe delay times 10 times back and forth, which are recorded as 10 sweeps labeled as sweep index s running from 1 to 10. From each sweep, the physicists created 3 data files, which are named as $D_s.dat$, $E_s.dat$ and $P_s.dat$, $s \in [1, 10]$, each containing a two-dimensional $4000(TOF) \times 151(Delay)$ data matrix. The file of $D_s.dat$ contains raw intensity signals without noise subtraction, while the files $E_s.dat$ and $P_s.dat$ contain independent background signal signals from two channels. Thus,

Table 2.1: Summary of data characteristics of *Adenine 803153* and *Adenine 8030651* experiments

	Adenine 803153	*Adenine 8030651*
Pump-probe delay time	$[-1565fs, 5935fs]$	$[-1242fs, 6258fs]$
Pump-probe delay time step	$50fs$	$50fs$
Delay index	$[0, 150]$	$[0, 150]$
Electron energy bin	$[0eV, 4.00eV]$	$[0eV, 4.00eV]$
Electron energy bin step	non-linear	non-linear
TOF index	$[0, 3999]$	$[0, 3999]$
Sweeps	10	10
FWHM of pump-probe delay time	$256fs$	$255fs$
σ of pump-probe delay time	$108.713fs$	$108.289fs$
σ in *Delay* index unit	2.17426	2.16577
Suggested scaling factor	5.00	2.00
Year of experiment	2008	2008
Location of experiment	UGA	UGA

Table 2.2: Summary of the data files for *Adenine 803153* and *Adenine 8030651* experiments

Name of Data Files	**Sweep Index**	**File Contents**
$D_s.dat$	$s \in [1, 10]$	Raw intensity signals
$E_s.dat$		Background signal from channel 1
$P_s.dat$		Background signal from channel 2

each experiment in sum has 30 data files, i.e. 10 files of raw intensity signals and 20 files of noise signals. A summary of the data files for the *Adenine 803153* and *Adenine 8030651* experiments can be found in Table 2.2.

Since the intensity signals contained in $D_s.dat$, $i \in [1, 10]$ are raw intensity signals, in order to extract the net intensity signals, we need to perform a proper background signal subtraction from those raw intensity signals before we proceed with performing statistical analyses on *Adenine 803153* and *Adenine 8030651*. A detailed explanation of the background signal subtraction technique can be found in Section 3.1.

2.1.3 Standard Deviation σ of *Delay*

Another physical quantity we should know for TRPES experiments is the experimental error, i.e. the experimentally measurable standard deviation (SD) σ due to the instrumental time resolution (i.e. $130fs$ pulse duration used in the experiment). Physicists generally use *FWHM*, which means *Full Width at Half Maximum*, to represent the imprecision in their experiments. With assumption of normality, the relationship between *FWHM* and standard deviation σ for any normal curve is, as shown in Fig. 2.1, given by

$$FWHM = 2.35482\sigma. \tag{2.5}$$

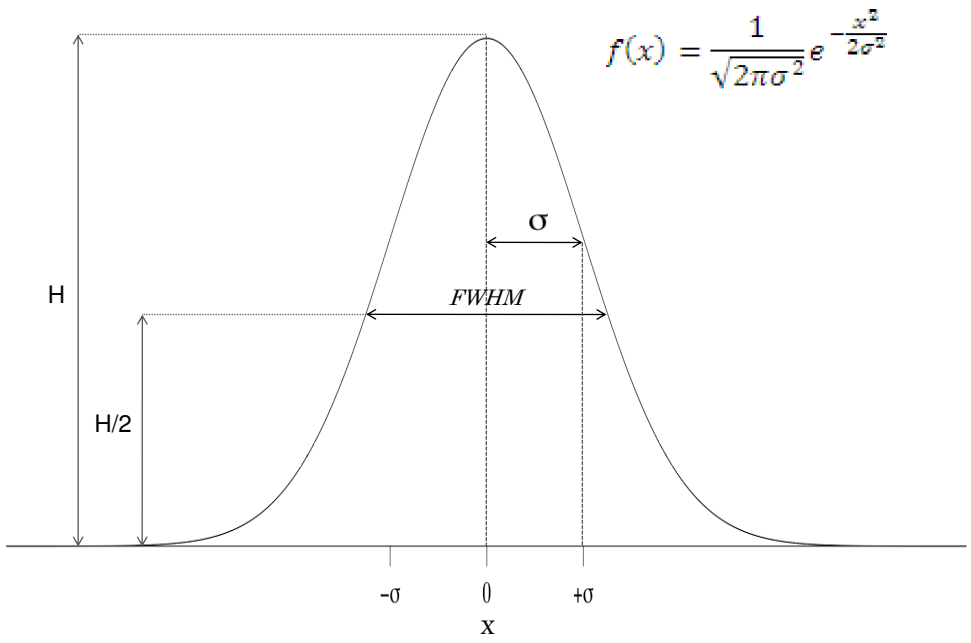

Figure 2.1: Illustration of relationship between *FWHM* and σ for a normal distribution.

For *Adenine 803153*, as one of experimental parameters, $FWHM = 256fs$. Thus, we obtained $\sigma = 108.713fs$, which is the experimentally measurable standard deviation of pump-probe delay time for *Adenine 803153*. In the unit of *Delay*, $\sigma = \frac{108.713fs}{50fs} = 2.17426$. For the other data set *Adenine 8030651*, $FWHM = 255fs$. Thus, we obtained $\sigma = 108.289fs$, or in the unit of *Delay*, $\sigma = \frac{108.713fs}{50fs} = 2.16577$ for the *Adenine 8030651* data set.

2.2 Theoretical Background

Physicists predict that the relationship between the pump-probe delay time and the intensity signals is bi-exponential decay, especially for Adenine, which should be modeled by the convolution of a Gaussian error function and the intensity signals.

When the decay is a *single-τ* decay, which means there is only one exponential decay and only one decay lifetime involved in the decaying process in the channel or relaxation pathway

of TRPES experiments, the total intensity signals can be expressed theoretically as follows

$$I(t) = A * \exp(\frac{\sigma^2}{2\tau^2} - \frac{t}{\tau}) * \{1 - \text{erf}(\frac{\sigma/\tau - t/\sigma}{\sqrt{2}})\}, \tag{2.6}$$

where t is the pump-probe delay time. σ is the experimentally measurable standard deviation due to imprecision in detector time resolution (In the unit of *Delay*, $\sigma = 2.17426$ for *Adenine 803153* and $\sigma = 2.16577$ for *Adenine 8030651*), and A and τ are the amplitude and decay lifetime parameters we wish to estimate via this fitting. In this thesis, we call this model the *Single-*τ model. The Gaussian error function used above has the following relationship to the well-known normal distribution function

$$\text{erf(z)} = \frac{2}{\sqrt{\pi}} \int_0^z \text{e}^{-t^2}\, \text{d}t = 2 * \{\text{CDF}('normal', \sqrt{2}z, 0, 1) - 0.5\}, \tag{2.7}$$

where $\text{CDF}('normal', \sqrt{2}z, 0, 1)$ is the SAS (Statistical Analysis Software) expression of the standard normal cumulative distribution function with $\sqrt{2}z$ as the upper limit of integral.

2.3 Brief Review of Previous Study in STAT 8000

Of course, the mathematical model in Eqn. 2.6 cannot exactly describe real-world physical phenomena. However, it can be a good starting point from which to build up an understanding of more complicated decay models with convolutions. In the work (Summer 2007) on the statistical consulting project "Decay Times for Genetic Bases"in STAT 8000 [2], we attempted to develop such a decay model by combining two *Single-*τ exponential decay models into a single convoluted *Two-*τ model. From this model, we obtained some preliminary results on the decay lifetimes τ_1 and τ_2 of Adenine.

The data set we analyzed in 2007 was obtained from earlier experiments conducted by Prof. Ullrich in 2005 in Canada. Thus, we called this data set as *Adenine Canada*. In those experiments, the physicists [Ullrich, et al.] set 53 different values of pump-probe delay time, from $-350fs$ to $2250fs$ in $50fs$ steps, and 50 electron energy bins, ranging from $0.25eV$ to $4.00eV$ in about $0.077eV$ steps. They scanned through the 53 different pump-probe delays 14

Table 2.3: Summary of data characteristics of *Adenine Canada* experiment

	Adenine Canada
Pump-probe delay time	$[-350fs, 2250fs]$
Pump-probe delay time step	$50fs$
Delay index	$[0, 52]$
Electron energy bin	$[0.25eV, 4.00eV]$
Electron energy bin step	$0.077eV$
TOF index	$[0, 49]$
Sweeps	14
FWHM of pump-probe delay time	$150fs$
σ of pump-probe delay time	$63.699fs$
σ in *Delay* index unit	1.27398
Suggested scaling factor	5.30
Year of experiment	2005
Location of experiment	Canada

times back and forth, and at each pump-probe delay time they recorded 5000 shots of pump-prob intensity signals. Thus each entry of the two-dimensional $50(TOF) \times 53(Delay)$ data matrix was the average of 70,000 intensity measurements. A summary of the characteristics of the data set *Adenine Canada* can be found in Table 2.3.

Before the data set was sent to us for analysis, the physicists performed some background signal subtractions. At each pump-probe delay time, they recorded a total of 500 shots of pump only and 500 probe only signals. Since these signals were independent of pump-probe delay time, the physicists simply summed them and then scaled them back by the scaling factor of 5.3 to achieve the same averaging effect as for the pump-probe intensity signals $(500 \times 53 \times 14/5.3 = 70,000)$. These values were subtracted from the raw intensity signals to

produce the background-subtracted intensity signals which were provided to us for the 2007 STAT 8000 analysis.

The physicists [Ullrich, et al.] also reported the *Full Width at Half Maximum* of this TRPES experiment to be $FWHM = 150fs$. From Eqn. 2.5, the experimentally measurable standard deviation σ of the pump-probe delay time was thus $\sigma = 63.699fs$, or in the unit of $Delay$, $\sigma = \frac{63.699fs}{50fs} = 1.27398$ for the *Adenine Canada* data set.

2.3.1 *Shift* Methods Selection

In the data set of two-dimensional $50 \times 53 = 2650$ delay-energy bin intensity values, there were some negative values which were due to the noise subtraction method utilized. Since true intensities can't be negative, we considered applying the following five adjustments, or five *shift* methods, to these negative data points before we began the analysis:

Method 1: We did nothing to the data.

Method 2: We added 0.07 to each data point of the $50(TOF) \times 53(Delay)$ data matrix in order to change some negative values to positive ones.

Method 3: A value of 0.10 was added to each data point of the $50(TOF) \times 53(Delay)$ data matrix in order to make the total (summed over TOF levels) intensity signals positive, although some individual data points could still be negative.

Method 4: An even larger value of 0.20 was added to each data point of the $50(TOF) \times 53(Delay)$ data matrix so that all individual negative data points became positive.

Method 5: We simply assigned zero to all the negative intensities.

To select the best *shift* method, we tried non-linear fitting, which is described by the *Single-τ* model in Eqn. 2.6, on the data after being shifted. Since the TRPES date set contains a two-dimensional data matrix, before we could do the non-linear fitting one-dimensionally,

we need to remove the dimension of TOF. To do that, we simply performed a direct summation over all TOF levels at each pump-probe delay time. To illustrate this procedure clearly, three steps are listed as follows

Step 1: *shift* methods were applied.

Step 2: A direct non-weighted summation over all TOF levels at each pump-probe delay time ($Delay$) was applied to make the data one-dimensional.

Step 3: The non-linear fitting described by the *Single-*τ model was applied.

With $\sigma = 63.699fs$ in Eqn. 2.6, we used maximum-likelihood methods to obtain estimates of the parameters A and τ to best fit the data. Table 2.4 displays the maximum likelihood estimations (MLE's) of A and τ for each of the five *shift* methods under the *Single-*τ model of Eqn. 2.6. The last column, SSE, gives the sum of squared error for these fits. Both the MLE's for A and τ , as well as the SSE, were calculated from PROC NLIN, a non-linear fitting algorithm in the SAS package.

Minimization of SSE is the optimal solution when errors at each point (the 53 pump-probe delay times) are normally distributed with a constant standard deviation. Normality itself is probably not a particularly crucial assumption here, given that each observation is really an average over 70000 trials. However, assuming a constant standard deviation is more problematic. The 'best' solution would be to minimize the weighted sum of squared errors, where the weights are inversely proportional to the error variance for each pump-probe delay time. Since we did not know these variances, did not have the raw data (i.e. the 70000 measurements which had been averaged to yield the provided intensities) and had been assured by the physicists that the typical error at each delay time was of roughly the same order of magnitude, we used minimization of SSE as the fit criteria in the 2007 study. But we are not sure that this is really correct. For this study, we found *shift* method 5 to yield the smallest error, as shown in Table 2.4. To be concise in this review section, we show in Fig. 2.2 the fitting results from *shift* method 5 only.

Table 2.4: Summary of the MLE's of A and τ for each of the five *shift* methods under the *Single-*τ model for Adenine

***Shift* Method Number**	***Shift***	$\hat{A}$	$\hat{\tau}$	SSE
1	$neg + .00$	13.31	299.3	300.6
2	$neg + .07$	12.40	408.2	134.7
3	$neg + .10$	11.88	482.9	159.9
4	$neg + .20$	9.92	987.0	473.2
5	$neg \rightarrow .00$	11.85	459.2	120.0

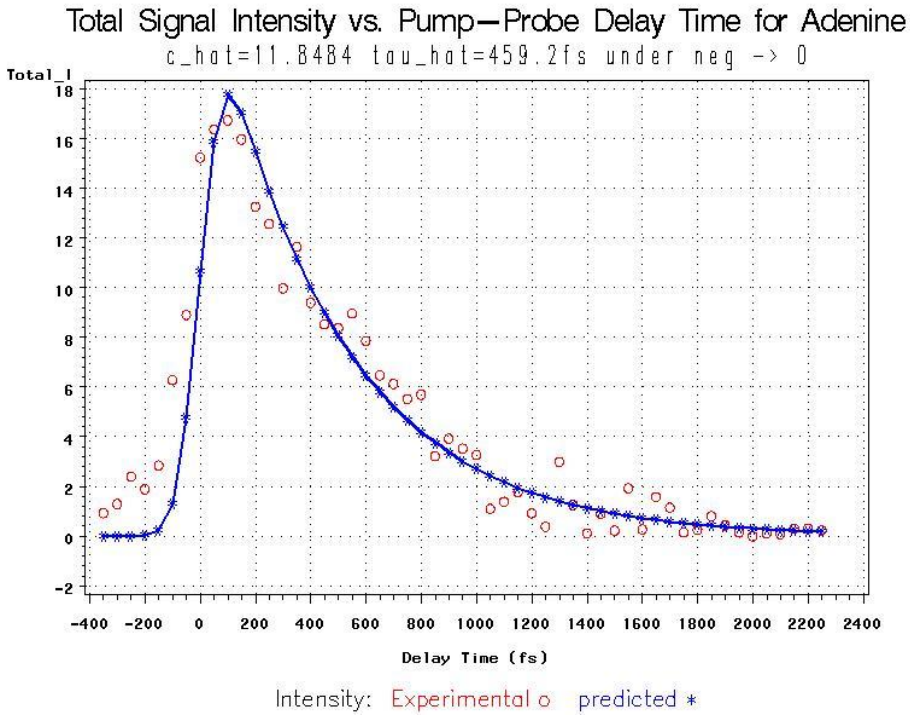

Figure 2.2: Results of non-linear regression with the *Single-*τ model with *shift* method 5 applied to the *Adenine Canada* data set.

2.3.2 Fitting with *Single-τ* Model

Although we knew that the decay is bi-exponential for Adenine, we started fitting data by using the *Single-τ* model described in Eqn. 2.6 since it is a relatively simple model from which we hoped to get some preliminary results. With $\sigma = 63.699fs$, according to the *Single-τ* model fitting, we obtained the predicted values of A, τ and SSE as the following: $\hat{A} = 11.85$, $\hat{\tau} = 459.2$, and $SSE = 120.0$. From Fig. 2.2, we can see that the fit is fairly good when the delay time is positive and large. However, when the delay time goes to zero or negative, the fit is not as good as in the right-hand tail. The reason that this simple *Single-τ* model could give such a decent fitting at all is due to the fact that one of two decays of the bi-exponential model is the dominant one, so that what we obtain here by the *Single-τ* model is a good approximation of this dominant decay.

2.3.3 Fitting with *Two-τ* Model

As we stated previously, for Adenine, the physical model is that the first state A is achieved by laser pulse excitation, then decays to a second state B, and the second state decays to the ground state. So the total intensity of signals could be described by the following mathematical expression:

$$I(t) = A_1 * X_1(t) + A_2 * X_2(t), \tag{2.8}$$

where $I(t)$ is the total observed intensity of signals, $X_1(t)$ and $X_2(t)$ are the particle population in excited state A and B, respectively, and A_1 and A_2 are the unit intensity given off at state A and state B, respectively. In this formulation, only $I(t)$ is observed, so both A_1 and A_2 as well as the functions $X_1(t)$ and $X_2(t)$ would need to be estimated. To do this, we first considered the simplest situation in which σ is zero, i.e. there is no convolution caused by pump and probe pulses. In this situation, we can write down the time differential equations of $X_1(t)$ and $X_2(t)$ as follows

$$\frac{dX_1(t)}{dt} = -r_1 * X_1(t), \tag{2.9}$$

$$\frac{dX_2(t)}{dt} = -r_2 * X_2(t) + r_1 * X_1(t), \tag{2.10}$$

where $r_1 = \frac{1}{\tau_1}$, $r_2 = \frac{1}{\tau_2}$. These equations merely state that $X_1(t)$ decays to $X_2(t)$ at a rate (r_1) proportional to the amount of $X_1(t)$ present and that $X_2(t)$ is created by the input from decaying X_1 particles, but decreases at a rate (r_2) proportional to amount of $X_2(t)$ present. As stated, the equations have an infinite number of solutions, so we impose the standard initial conditions $X_1(0) = 1$ and $X_2(0) = 0$; At time $t \sim 0$, all particles are in excited state 0. (This would be exactly true if there were no convolution, which is the assumption behind Eqn. 2.9 and Eqn. 2.10). Ignoring convolution caused by pump and probe pulses, we found the analytical solutions to these differential equations as follows

$$X_1(t) = \exp(-r_1 * t), \tag{2.11}$$

$$X_2(t) = \frac{r_1}{r_1 - r_2}\{\exp(-r_2 * t) - \exp(-r_1 * t)\}. \tag{2.12}$$

If we consider the convolution caused by pump and probe pulses, the analytical solution of $X_1(t)$ will simply be changed to the form:

$$\begin{aligned} X_1(t) &= \exp(\frac{\sigma^2}{2\tau_1^2} - \frac{t}{\tau_1}) * 2\{1 - erf(\frac{\sigma/\tau_1 - t/\sigma}{\sqrt{2}})\} \\ &= \exp(\frac{1}{2}\sigma^2 * r_1^2 - r_1 * t) * 2\{1 - erf(\frac{\sigma * r_1 - t/\sigma}{\sqrt{2}})\}. \end{aligned} \tag{2.13}$$

Thus, in order to find $X_2(t)$, we would need to solve the following differential equation:

$$\frac{dX_2(t)}{dt} = -r_2 * X_2(t) + r_1 * \exp(\frac{1}{2}\sigma^2 * r_1^2 - r_1 * t) * 2\{1 - erf(\frac{\sigma * r_1 - t/\sigma}{\sqrt{2}})\}. \tag{2.14}$$

It is not easy to find the analytical solution to the above equation, so we turned to construct an approximate *Two-*τ model, in which we added a convolution term to $X_2(t)$ independently. The following is the equation for this constructed *Two-*τ model:

$$\begin{aligned} I(t) &= A_1 * \exp(-r_1 * t) * 2\{1 - erf(\frac{\sigma * r_1 - t/\sigma}{\sqrt{2}})\} + \\ &\quad A_2 * \frac{r_1}{r_1 - r_2}\{\exp(-r_2 * t) - \exp(-r_1 * t)\} * 2\{1 - erf(\frac{\sigma * r_2 - t/\sigma}{\sqrt{2}})\}. \end{aligned} \tag{2.15}$$

The fitting results are shown in Fig. 2.3. For this constructed approximate *Two-*τ model,

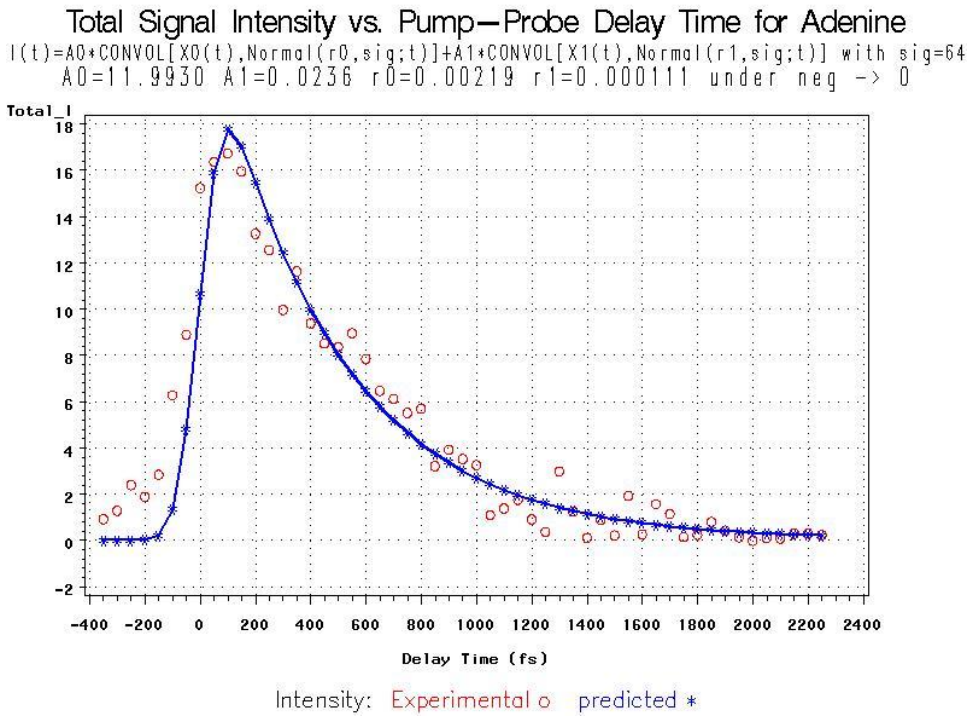

Figure 2.3: Results of non-linear regression with the constructed *Two-*τ model with *shift* method 5.

the fit looks fairly good, especially when the delay time is positive and large. However when the delay time goes to zero or negative, the fit is not as good as the fit when the delay time is positive and large. The fitting curve looks very similar to the fitting curve obtained from the preliminary fitting with *Single-*τ model, introduced in Section 2.3.1. From this model, we obtained the following parameter estimates: $\hat{A}_1 = 11.9930$, $\hat{A}_2 = 0.0236$, $\hat{\tau_1} = 1/0.00219fs = 456.62fs$, $\hat{\tau_2} = 1/0.00011fs = 900.90fs$, $SSE = 120.0$. Compared with the values of lifetimes previously suggested by physicists: $\tau_1 = 100fs$, $\tau_2 = 1100fs$, the convergence of predicted τ_2 to the suggested τ_2 is better than the convergence of predicted τ_1 to the suggested τ_1. The parameter estimates under the two models are shown in Table 2.5. Our reason for the very close agreement is because A_2 is so small relative to A_1 that the *Two-*τ model is almost a *Single-*τ model.

Table 2.5: Summary of the τs of Adenine with the *Single-*τ and constructed approximate *Two-*τ model

Molecule	**Model**	$\sigma(fs)$	$\tau_1(fs)$	$\tau_2(fs)$	A_1	A_2	SSE
Adenine	*Single-*τ	63.699	459.2		11.85		120.07
	Two-τ		456.6	900.0	11.99	0.02	120.01

2.3.4 Conclusions on Previous Study

For *Adenine Canada*, the *Single-*τ model is meaningful because the fitting with it tells us that one of two decays is a dominant decay in the bi-exponential model. The approximate *Two-*τ model gives us a good fitting to the data, especially when the delay time is positive and large. In Table 2.5, a short summary of our previous study on the decay lifetimes of Adenine is given. As we discuss in Section 2.4, there are two major drawbacks in our previous study, so these results may not be correct.

Now, we have access to more refined data. In the remainder of this thesis, we will show our further study on the decay lifetimes of Adenine. As discussed in Section 2.1 and 2.4, to remedy the drawbacks in our previous study and improve the precision in the statistical analysis, we used two new TRPES data sets (*Adenine 803153* and *Adenine 8030651*) plus new statistical analysis methods in this follow-up study. In addition, we followed new understanding of physics background to improve our estimation techniques.

2.4 Drawbacks of Previous Study in STAT 8000

We now realize that there were two major drawbacks in our previous study. These two drawbacks should be eliminated from our current study in order to improve the precision of statistical analyses. These two limitations are discussed below:

1: No proper background signal subtraction Before the data set was sent to us for analysis, the background signal subtraction has been performed directly by physicists. Many negative signals were found in data. Instead of proposing a proper background signal subtraction, we tried several crude *Shift* methods to remove those negative signals simply by changing them to be zero, or by shifting whole data surface upwards with different magnitudes to make those negative signals to be positive. Due to the limited time to complete the STAT 8000 project, we didn't perform a detailed statistical study on the effects of background signal subtraction. We lack a proper statistical study on background signal for this study.

2: Crude summation over time-of-flight bins The data set contains a two-dimensional data matrix. Before we could perform the non-linear regressions one-dimensionally with the pump-probe delay time ($Delay$) as the predictor variable, we removed the dimension of electron energy by doing a summation over all electron energy bins (TOF) at each $Delay$. In our previous study, we performed a direct non-weighted summation over TOF, which could be too crude. We merely summed the intensities over TOF (after the intuitive correcting for negative values, as discussed in Section 2.3.1) to obtain the final $I(t)'s$ which were our response variables in these analyses. This is not the best way to do things, but with the data set we had and our understanding of the physical system in 2007, it was the best which we could do then. We suspected that we might be able to obtain more physically interpretable models if we weighted the intensities from different electron energy bins differently, but we were unsure what the correct weighting should be at that time.

In this current study, we eliminated the drawbacks discussed above from our analysis. In Section 3.1 and 3.2, respective detailed explanations of the background signal subtraction procedure and the improvements in the summation over TOF procedure can be found.

In addition to those improvements, we also found a new way to extract the decay lifetimes τ_1 and τ_2 of Adenine. We focused on two relaxation pathways or channels in TRPES exper-

iments, i.e. the long-lived channel (LLC) and short-lived channel (SLC). We know from the physicists [Evans, et al.] that, in the long-lived channel, signals are mainly dominated with the long-lived signals with the decay lifetime τ_1, which could be extracted directly by fitting with the *Single-*τ model described in Eqn. 2.6. In the short-lived channel, the situation becomes much more complicated. Not only can we find the signals from the short-lived channel, but we can also find the signals from the long-lived channel there. Those signals are overlapped in the short-lived channel. This overlapping is found to be strong, both experimentally and statistically, which causes a heavy superposition of intensity signals in the short-lived channel. In order to extract the decay lifetime τ_2 of the short-lived channel, we must decompose those overlapped signals into two components, one of which is the signal purely from the short-lived channel, and the other of which is the remnant signal from the long-lived channel. With this decomposition,the signals from the short-lived channel are purified and can be fitted again to extract the decay lifetime τ_2 with the *Single-*τ model described in Eqn. 2.6. In Section 3.3, a detailed description of how we accomplished the decomposition and fitting in the short-lived channel can be found.

Chapter 3

Statistical Analysis Methods

In this chapter, we focus on the new statistical analysis methods we developed since 2007. In Section 3.1, we will explain how we accomplished the background signal subtraction for *Adenine 803153*. In Section 3.2, we explain how we improved the procedure for summation over electron energy bins at each pump-probe delay time. By the procedures introduced in Section 3.1 and 3.2, we ameliorated the drawbacks discussed in Section 2.4, of our previous study. In the last section of this chapter, Section 3.3, we discuss the new procedure we followed to extract the decay lifetimes of Adenine from the perspectives of physics and statistics.

3.1 Background Signal Subtraction

3.1.1 The Scaling Factor k

As noted in Section 2.1.2, the data set *Adenine 803153* contains 30 data files: 10 files names as $D_i.dat$, $i \in [1, 10]$, 10 files named as $E_i.dat$, $i \in [1, 10]$ and 10 files named as $P_i.dat$, $i \in [1, 10]$. Each data file contains a two-dimensional $4000(TOF) \times 151(Delay)$ data matrix. The D files contain raw intensity signals without background signal subtraction. The E and P files contain background signal signals to be subtracted. Since background signal signals are experimentally independent of intensity signals, physicists [Evans, et al.] proposed the background signal subtraction as follows

$$\begin{aligned} R &= D - k * BG \\ &= D - k * (E + P), \end{aligned} \tag{3.1}$$

where R represents the resultant intensity signals after background signal subtraction. D represents the raw intensity signal before background signal subtraction. $E+P$ represent the total background signals to be subtracted from intensity signals. k, with an initial suggested value of 5.00, is the scaling factor used to scale noise signals back to achieve the same averaging effect as for the intensity signals.

Of course, we could use the suggested value of $k = 5.00$ in Eqn. 3.1, but to determine the background signal subtraction more precisely, we need to determine the scaling factor k more precisely, which actually is the goal of this part of the work.

3.1.2 Methods for Determining Scaling Factor k

To do a very precise estimate on k, we redefine Eqn. 3.1 on each data point of two-dimensional $4000(TOF) \times 151(Delay)$ data matrix as follows

$$\begin{aligned} R_{sij} &= D_{sij} - k * BG_{sij} \\ &= D_{sij} - k * (E_{sij} + P_{sij}), \end{aligned} \tag{3.2}$$

where s is the *index of Sweep*, $s \in [1, 10]$,

i is the *index of TOF (electron energy bin)*, $i \in [0, 3999]$,

j is the *index of Delay (Pump-probe delay time)*, $j \in [0, 150]$.

In Section 2.1.1 and 2.1.2, detailed explanations of these indices can be found.

First, we average over sweeps to obtain $R_{\cdot ij}(k)$ as a function of k as follows

$$\overline{R}_{\cdot ij}(k) = \overline{D}_{\cdot ij} - k * \overline{BG}_{\cdot ij}, \tag{3.3}$$

where $\overline{D}_{\cdot ij} = \dfrac{1}{10} \sum_s D_{sij}$,

$\overline{BG}_{\cdot ij} = \dfrac{1}{10} \sum_s BG_{sij} = \dfrac{1}{10} \sum_s (E_{sij} + P_{sij})$.

The optimal value of k can be determined from the region in which there is no real activity, so that $\overline{R}_{\cdot ij}(k)$ behaves approximately like a zero mean and constant standard deviation process. However, determining where those regions are requires some thought. To do this,

Table 3.1: Summary of the strategies of grid division for k determination

$pixelTOF \times pixelDelay$	160×30	80×15	40×3	1×1 **(raw data)**
♯ of rows in cell	160	80	40	1
♯ of columns in cell	30	15	3	1
♯ of rows in grid	25	50	100	4000
♯ of columns in grid	5	10	50	151
♯ of cells (C)	125	500	5000	604000
♯ of pts in cell (N)	4800	1200	120	1

we divided the entire $4000(TOF) \times 151(Delay)$ grid into sub-grids ("cells"), each of size of $pixelTOF \times pixelDelay$. In each $cell(i', j')$, we have

$$\begin{cases} Index\ of\ cell\ in\ TOF: & i' = 0 \sim INT(4000/pixelTOF) - 1, \\ Index\ of\ cell\ in\ Delay: & j' = 0 \sim INT(151/pixelDelay) - 1. \end{cases}$$

Several strategies of grid division that we considered are given in Table 3.1. The second strategy with $pixelTOF \times pixelDelay = 80 \times 15$ is the strategy ultimately utilized in this study. The latter two strategies (40×3 and 1×1) have so few observations per cell and so many different cells that the data appears very noisy - in statistical terms, they appear to under-smooth the data. The first strategy (160×30) over-smoothes the data, using only 125 cells, and not allowing the data to display true variation due to $Delay$ and TOF. The 80×15 strategy which we have decided to utilize seems to strike a reasonable balance between smoothing and signal detection.

Within each of the C cells (each containing N data points), the mean of $\overline{R}_{\cdot ij}(k)$'s can be found as follows

$$\frac{\sum_{i \in i'\ j \in j'} \overline{R}_{\cdot ij}(k)}{N_{i'j'}} = \frac{\sum_{i \in i'\ j \in j'} \overline{D}_{\cdot ij}}{N_{i'j'}} - k * \frac{\sum_{i \in i'\ j \in j'} \overline{BG}_{\cdot ij}}{N_{i'j'}}, \tag{3.4}$$

where $N_{i'j'} = pixelTOF \times pixelDelay$ is the sample size of data points in $cell(i', j')$. We denote the terms in Eqn. 3.4 as follows

$$\hat{\mu}_R(i', j', k) = \overline{R}(i', j', k) = \overline{D}(i', j') - k * \overline{BG}(i', j'). \tag{3.5}$$

The sample standard deviation of $\overline{R}_{\cdot ij}(k)$ within each cell can be found in Eqn. 3.6 as follows

$$\hat{\sigma}_R(i', j', k) = \sqrt{\frac{\sum\limits_{i \in i' \; j \in j'} \left(\overline{R}_{\cdot ij}(k) - \overline{R}(i', j', k)\right)^2}{N_{i'j'} - 1}}. \tag{3.6}$$

One can use the quantities in Eqn. 3.5 and Eqn. 3.6 to perform a one-sample *t-test* (with $N_{i'j'} - 1$ degree freedom (df)) of the *null hypothesis* that the true mean of $\overline{R}_{\cdot ij}(k)$, i.e. $\hat{\mu}_R(i', j', k)$, is zero. The *t-statistics*, which follows a standard normal distribution $\mathcal{N}(0, 1)$ under the *null hypothesis*, is defined for each cell as follows

$$t(i', j', k) = \frac{\hat{\mu}_R(i', j', k) - 0}{\hat{\sigma}_R(i', j', k)/\sqrt{N_{i'j'}}}. \tag{3.7}$$

If every region where this were tested were such that there were no real activity, then it would be easy to determine the optimal value of k such that this hypothesis is satisfied. However, there are many regions in the $TOF \times Delay$ region for which there is significant activity beyond the baseline level. In these regions, even after subtracting the background signal level, the residual contains real signals. So, before we can estimate the optimal value of k, we must first identify the true no-signal regions.

In Section 4.1.1.2, with the strategy of grid division using the cell size of $pixelTOF \times pixelDelay = 80 \times 15$, we show how we applied the quantities defined in Eqn. 3.5 and Eqn. 3.6 to locate the individual no-signal cells. With the same strategy of grid division, in Section 4.1.1.3 and Section 4.1.1.4, we show the results demonstrating how we applied the one-sample *t-test* with the *t-statistics* described in Eqn. 3.7 to determine the no-signal regions, and then find the optimal estimated value of the scaling factor k, which is notated as k^* in this thesis.

3.2 Gaussian-Weighted Summation over TOF

In order to remedy another drawback caused by the way we calculated the summation over TOF in our previous study, we must find a more precise way to do this before we proceed to further fitting on $Delay$ in order to extract the decay lifetimes of Adenine. In 2007, we suspected that we might be able to obtain more physically interpretable models if we weighted the intensities from different TOF differently, but we were unsure what the correct weighting should be at that time. Now, we have data that allow us to do this.

As known from physicists, for a fixed level of $Delay$, the intensities over TOF should have approximately a Gaussian distribution. With this in mind, instead of staying with a direct equal-weighted summation over TOF as we performed in our previous study, we set out to consider a weighted summation, or a Gaussian-weighted summation, over TOF. To do this Gaussian-weighted summation, we need to know beforehand the Gaussian probability density function of data for a fixed level of TOF.

3.2.1 Gaussian Probability Density Function for a Fixed Level of TOF

Since TRPES data theoretically have a Gaussian distribution for a fixed level of TOF, to find the Gaussian probability density function, we could simply perform a Gaussian fitting to our data. However, the real world doesn't always behave as theories predict. With a detailed checking, we found there exists a skewed pattern in our data , which deforms the distribution of data to a minor extent from a perfectly symmetrical Gaussian distribution. To fix this skew, instead of performing a simple Gaussian fitting, we applied an inverse-variance-weighted Gaussian fitting, from which we obtained the probability density function of the skewed Gaussian distribution for a fixed level of TOF. Before we present details, the general outline of our procedure is presented as follows

Step 1: We summed all data points over $Delay$s at each level of TOF, thus obtaining one-dimensional data in TOF.

Step 2: We divided the data from Step 1 into small groups in TOF scale. For each group, we calculated the sample standard deviation σ of all data points in the group.

Step 3: We performed a linear regression on σ's obtained from Step 2 with TOF as the predictor variable. Thus, we obtained a function showing the dependence of σ on TOF level.

Step 4: We applied an inverse-variance-weighted Gaussian fitting on the data averaged from Step 1 with $\sqrt{TOF}$ as the predictor variable and $\frac{1}{\hat{\sigma}^2}$ as the weighting factor. Thus, we obtained the Gaussian probability density function of data.

The first step is to sum all data points over $Delay$s at each level of TOF. Thus, we obtain one-dimensional data as a function of TOF as follows

$$\overline{R}_{\cdot i\cdot}(k^*) = \sum_{j \in DelayRange} \overline{R}_{\cdot ij}(k^*), \tag{3.8}$$

where k^* is the optimal estimated value of scaling factor determined by the procedure of background signal subtraction, which is described in Section 3.1.2. $\overline{R}_{\cdot ij}(k)$ is the quantity defined in Eqn. 3.3. $DelayRange$ with $N_{DelayRange}$ data points inside is the range of $Delay$ over which the summation is performed. $\overline{R}_{\cdot i\cdot}(k^*)$ is the result of summation, which is a function of i, i.e. TOF, $i \in [300, 1500]$.

The second step is to divide the data defined in Eqn. 3.8 into small groups with 25 data points (with 25 i's) per group. Each group is assigned an index of g. For each group, we calculated the sample standard deviation σ_g of all data points in the group as follows

$$\sigma_g = \sqrt{\frac{\sum_{i \in g} \left(\overline{R}_{\cdot i\cdot}(k^*) - \overline{R}_g(k^*)\right)}{N_g - 1}}, \tag{3.9}$$

where $\overline{R}_{\cdot i\cdot}(k^*)$ is defined in Eqn. 3.8. $\overline{R}_g(k^*)$ is the overall average of 25 data points in group g. $N_g = 25$ is the sample size of data points in group g.

In the next step, to find the function describing the dependence of σ_g with respect to TOF, we propose the following linear regression model, through applying the SAS procedure

PROC REG, to do a fitting on σ_g's with TOF as the predictor variable.

$$\sigma_g = \beta_0 + \beta_1 * X + \beta_2 * X^2 + \beta_3 * X^3, \tag{3.10}$$

where X is a linearly transformed predictor variable of TOF with $X = \frac{TOF+12.5-1000}{100}$. β_0, β_1, β_2, and β_3 are the parameters to be estimated in the fitting.

Lastly, using the non-linear model given in Eqn. 3.11, we fit the data averaged from the first step, i.e. $\overline{R}_{\cdot i \cdot}(k^*)/N_{DelayRange}$, where $i \in TOFFittingRange$, with an inverse-variance-weighted Gaussian fitting with $\sqrt{TOF}$ as the predictor variable and $\frac{1}{\hat{\sigma_g}^2}$ as the weighting factor, through applying the SAS procedure PROC NLIN. $TOFFittingRange$ is the index range of TOF over which we performed the fitting.

$$f(\sqrt{TOF}) = \frac{C}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(\sqrt{TOF}-\mu)^2}{2\sigma^2}), \tag{3.11}$$

where C, μ and σ are the parameters to be determined in the fitting. $f(\sqrt{TOF})$ is the Gaussian probability density function of data as a function of $\sqrt{TOF}$.

3.2.2 Gaussian-Weighted Summation over TOF

With the Gaussian probability density function $f(x)$ described in Eqn. 3.11, we define the weighting factor for Gaussian-weighted summation over TOF as follows

$$W(TOF) = \int_{\sqrt{TOF-0.5}}^{\sqrt{TOF+0.5}} f(x) * dx, \tag{3.12}$$

where $W(TOF)$ is the weighting factor at each TOF. Note that TOF is an indexed quantity running from 0 to 3999 in steps of 1. For mathematical convenience, it is also equivalently represented by the index of i, $i \in [0, 3999]$. Therefore, $W(TOF)$ in Eqn. 3.12 can be also written as $W(i)$ or $W(TOF_i)$. We prefer the form of $W(i)$ in this thesis. In Section 2.1.1 and 3.1.2, detailed explanations of TOF and its index i can be found .

With the weighting factor $W(i)$ defined in Eqn. 3.12, we define the Gaussian-weighted summation over TOF as follows

$$\overline{R}_{\cdot\cdot j}(k^*) = \sum_{i \in TOFSumRange} \overline{R}_{\cdot ij}(k^*) * W(i), \tag{3.13}$$

where k^* is the optimal estimated value of scaling factor. $\overline{R}_{\cdot ij}(k^*)$ is the quantity defined in Eqn. 3.3. $TOFSumRange$ is the index range of TOF over which we performed the summation. $\overline{R}_{\cdot\cdot j}(k^*)$ is the result of Gaussian-weighted summation, which is a function of j, i.e. $Delay$.

In Section 4.1.2, we show the results related to Gaussian-weighted summation described in Section 3.2.

To extract the decay lifetimes of Adenine, we further take the quantity defined in Eqn. 3.13 into the *Single-τ* non-linear regressions with $Delay$ as the predictor variable. This part of the work will be introduced in the next section.

3.3 Non-linear Regression with the *Single-τ* Model on $Delay$

As we mentioned in Section 2.4, we followed a new way to extract the decay lifetimes of Adenine τ_1 and τ_2 in this study. Instead of trying to construct *Two-τ* decay model by convoluting two *Single-τ* decay models as we attempted in the previous study, we set out to perform fittings in two relaxation pathways or channels in TRPES experiments, i.e. the long-lived channel (LLC) and short-lived channel (SLC), separately with *Single-tau* models.

In the long-lived channel, signals are mainly dominated by the long-lived signals with the decay lifetime τ_1, which could be extracted directly by fitting with the *Single-τ* model described in Eqn. 2.6. However, in the short-lived channel, the situation is much more complicated. The signals from the short-lived channel overlap with the signals from the long-lived channel. We found both experimentally and statistically that this overlapping is very strong, enough to cause a heavy superposition of intensity signals in the short-lived channel. Thus, to extract the decay lifetimes τ_1 of LLC and τ_2 of SLC, we must follow two steps before we could perform fittings in those two channels. First, we need to identify in $Delay$ where the long-lived channel is and where the short-lived channel is; Secondly, to extract τ_2 of SLC, we need to conduct an efficient decomposition of those overlapped signals in the short-lived

channel into two components, one of which is the signal purely from the short-lived channel, and the other of which is the remnant signal from the long-lived channel.

In Section 3.3.1, we discuss how to define the long-lived and short-lived channels in the ranges of *Delay*. In Section 3.3.2, we show how to decompose signals in the short-lived channel.

3.3.1 Ranges of *Delay* for the long-lived and short-lived Channels

Table 2.1 in Section 2.1.1 shows both physical quantities and indexed quantities for the recent TRPES experiments. The physical range of pump-probe delay time for *Adenine 803153* is $[-1565fs, 5935fs]$ and for *Adenine 8030651* is $[-1242fs, 6258fs]$, in steps of $50fs$. The corresponding indexed range of *Delay* is $[0, 150]$ in steps of 1, and each step of *Delay* represents $50fs$ in the real physical scale. Since there exists a one-to-one correspondence between physical range and indexed range, we can calculate the zero point of indexed range for *Adenine 803153* as follow

$$Delay_0 = \frac{1565fs}{50fs} = 31.3, \tag{3.14}$$

where $Delay_0$ is the zero point of the indexed range of *Delay* for *Adenine 803153*. Similarly, for *Adenine 8030651*, the zero point of indexed range is $Delay_0 = \frac{1242fs}{50fs} = 24.84$.

For *Adenine 803153*, as shown in [1], to obtain purely exponential decay ranges for non-linear regressions with the *Single-*τ model, we chose the range of pump-probe delay time of $[100fs, 400fs]$ for the short-lived channel, which corresponds to the range of *Delay* of $[33, 39]$. For the long-lived channel, we chose the range of pump-probe delay time of $[400fs, 5885fs]$, which corresponds to the range of *Delay* of $[40, 149]$. We should emphasize that, we dropped $Delay = 150$ in our analyses since we found $Delay = 150$ to be an outlier point.

For *Adenine 8030651*, to meet the convergence criterion of non-linear fitting, we chose the range of pump-probe delay time of $[10fs, 310fs]$ for the short-lived channel, which corresponds to the range of *Delay* of $[25, 31]$. We also chose the range of pump-probe delay

Table 3.2: Summary of the chosen ranges of *Delay* for *SLC* and *LLC*

Data Set	**Channel**	**Physical Range**	*Delay* **Range**
Adenine 803153	*SLC*	$[100fs, 400fs]$	$[33, 39]$
	LLC	$[400fs, 5885fs]$	$[40, 149]$
Adenine 8030651	*SLC*	$[10fs, 310fs]$	$[25, 31]$
	LLC	$[310fs, 6208fs]$	$[32, 149]$

time of $[310fs, 6208fs]$ for the long-lived channel, which corresponds to the range of *Delay* of $[32, 149]$.

Table 3.2 is a summary of the chosen ranges of *Delay* for the long-lived and short-lived channels for *Adenine 803153* and *Adenine 8030651*. Fig. 3.1 is an illustration of those ranges for *Adenine 803153*.

3.3.2 Decomposition of Signals in the short-lived Channel

As mentioned in the beginning of this section, to extract the lifetime τ_2 of SLC, we need to decompose those strongly overlapped signals in the short-lived channel into two components, one of which is the signal purely from the short-lived channel, and the other of which is the remnant signal from the long-lived channel. To find those pure signals from the short-lived channel, we performed a linear subtraction of signals as shown as follows

Step 1: Fitting in LLC In LLC, we first performed a non-linear regression with the *Single-*τ model described in Eqn. 2.6 with *Delay* as the predictor variable. Thus, we obtained the lifetime τ_1 of LLC and the fitting function describing the intensity signals as an exponential decay function of *Delay* in LLC.

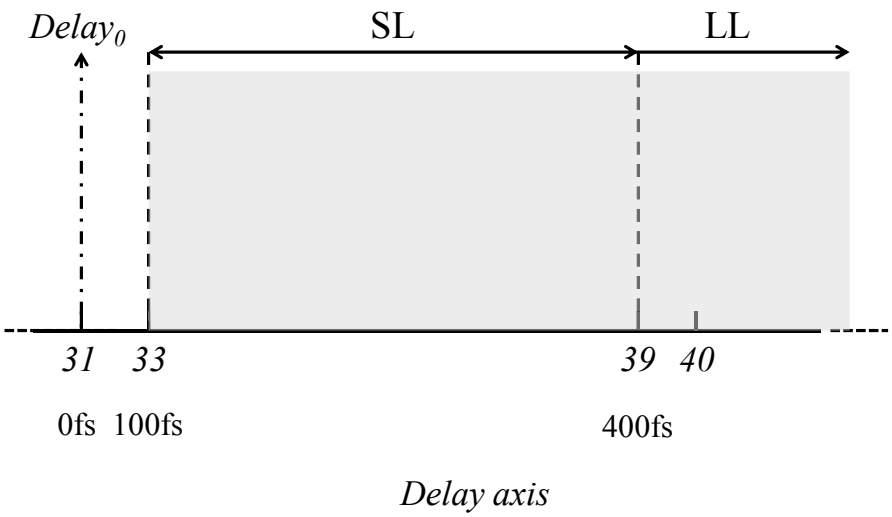

Figure 3.1: Illustration of the chosen ranges of *Delay* for *SLC* and *LLC* for *Adenine 803153*.

Step 2: Signal Subtraction in SLC In *SLC*, we subtracted the *LLC* signals, which are given by the fitting function obtained in Step 1, from the total signals. Thus, we obtained the resultant intensity signals as the pure signals from *SLC*.

Step 3: Fitting in SLC In *SLC*, we performed a non-linear regression with the *Single-*τ model again on the resultant signals, which we obtained from Step 2. Thus, we obtained the lifetime τ_2 of *SLC* and the fitting function describing the intensity signals as an exponential decay function of *Delay* in *SLC*.

To do the non-linear regressions, we applied the SAS procedure PROC NLIN.

In Section 4.1.3, we will show the results related to the "*LLC*" and "*SLC*" *Single-τ* non-linear regressions described in this Section 3.3.

Chapter 4

Statistical Analysis Results

In this chapter in Section 4.1 and Section 4.2, we present the statistical analysis results of data sets *Adenine 803153* and *Adenine 8030651*, respectively.

4.1 Statistical Analysis Results for *Adenine 803153*

Section 4.1.1 shows the results of background signal analyses, including the optimal estimated value of the scaling factor k for the background signal subtraction. The results of Gaussian-weighted summation over TOF can be found in Section 4.1.2. Section 4.1.3 is dedicated to show the results of non-linear regressions with the *Single-*τ model in LLC and SLC, by which we obtained the decay lifetimes of Adenine τ_1 (LLC) and τ_2 (SLC).

4.1.1 Background Signal Subtraction

4.1.1.1 Visualization on Background Signal

Fig. 4.1 shows a two-dimensional visualization of the background signal of *Adenine 803153*. Each colorful block in Fig. 4.1 represents a cell, notated as $cell(i', j')$, where $i' \in [0, 49]$ and $j' \in [0, 9]$, in the 50×10 divided grid with the cell size of $pixelTOF \times pixelDelay = 80 \times 15$. Different colors in the color scale give the background signal intensity values, i.e. $\overline{BG}(i', j')$, correspondingly. The x-axis and y-axis, labeled respectively as "*grid_TOF*" and "*grid_Delay*" in the plot, give the values of i' and j', i.e. the values of the indices of cell in TOF and $Delay$, respectively. In Section 3.1.2, an explanation of how we did the grid division on the original two-dimensional $4000(TOF) \times 151(Delay)$ data matrix to get $cell(i', j')$ and $\overline{BG}(i', j')$ can be found .

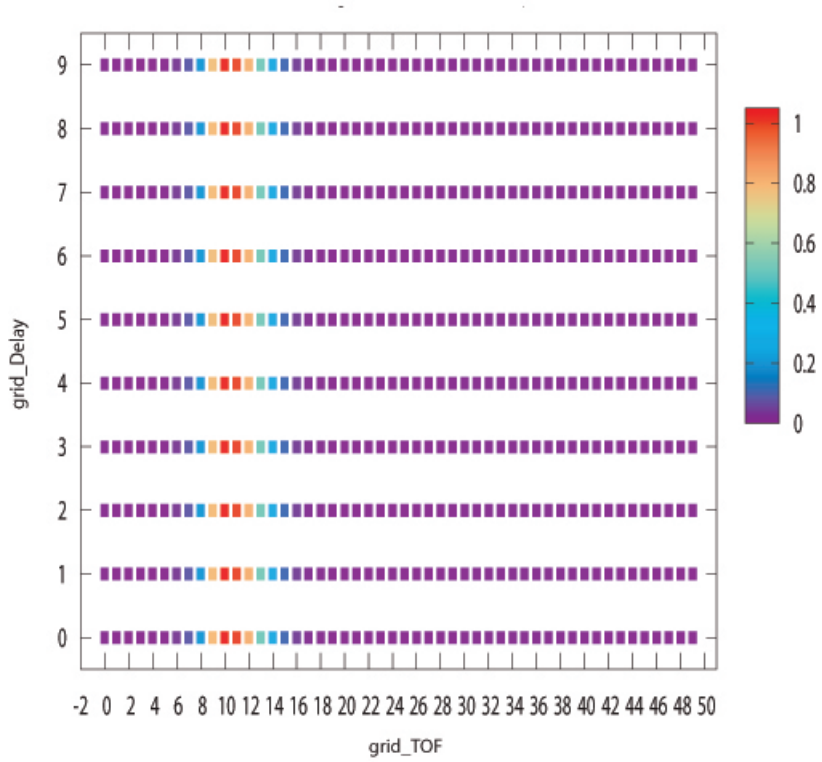

Figure 4.1: 2D visualization of the background signal of *Adenine 803153*.

From Fig. 4.1, we can see that, as a function of i' and j', the background signal intensity in each $cell(i', j')$, i.e. $\overline{BG}(i', j')$, does not behave as a random function, which is independent of i' and j', in the whole range of the plot. The observation is that $\overline{BG}(i', j')$ behaves as a function of i' only. Thus, $\overline{BG}(i', j')$ can be validly written as $\overline{BG}(i')$. The relatively large intensities all locate in the range of i' of $[9, 12]$, which corresponds to the range of TOF of $[720, 960]$. For each j', $\overline{BG}(i', j')$ approaches its approximate maximum value of 1.0 as i' approaches 10, i.e. as TOF approaches 800.

4.1.1.2 Determination of Individual No-Signal Cells

As we discussed at the end of Section 3.1.2, before we can estimate the optimal value of the scaling factor k, we must identify the no-signal regions. To identify the no-signal regions, we first set out to determine the individual no-signal cells.

We selected six representative individual cells (labeled from A to F) on the 50×10 divided grid with the cell size of $pixelTOF \times pixelDelay = 80 \times 15$. In each $cell(i', j')$, we have 1200 $\overline{D}_{\cdot ij}$ s, 1200 $\overline{BG}_{\cdot ij}$ s and 1200 $\overline{R}_{\cdot ij}(k)$ s, where $\overline{R}_{\cdot ij}(k) = \overline{D}_{\cdot ij} - k * \overline{BG}_{\cdot ij}$ and $i \in i'$; $j \in j'$. For each selected cell, three quantities were calculated as a function of k. Two of them are defined in Eqn. 3.5 and Eqn. 3.6, i.e. $\hat{\mu}_R(i', j', k)$ and $\hat{\sigma}_R(i', j', k)$. The other one is the median of $\overline{R}_{\cdot ij}(k)$ in $cell(i', j')$, notated as $\widetilde{\mu}_R(i', j', k)$. The strategy of grid division and definitions of $\hat{\mu}_R(i', j', k)$ and $\hat{\sigma}_R(i', j', k)$ can be found in Section 3.1.2.

Fig 4.2 is an illustration on the locations of those six individual cells. The x-axis and y-axis give the values of i' and j', i.e. the values of the indices of cell in TOF and $Delay$, respectively. The z-axis, labeled as "$aveR$" in the plot, gives 9 times of the value of $\hat{\mu}_R(i', j', k)$ in each $cell(i', j')$ with $k = 5.02$. The factor of 9 is due to the rescaling which is done by the plotting software. We also projected the magnitude of $\hat{\mu}_R(i', j', k)$ onto the $xy-plane$. Thus, we obtained a two-dimensional intensity plot, in which different colors show the relative intensity values of $\hat{\mu}_R(i', j', k)$. From Fig. 4.2, we found around the location of Cell F, there exists a hump running along i'-direction in the range of $i' \in [\sim 25, 49]$ with $j' \simeq 1.0$. This hump can not be explained by TRPES experiments. Thus, we called this hump as "weird unexplained hump" in this study.

A summary of results from those six individual cells on $\hat{\mu}_R(i', j', k)$, $\hat{\sigma}_R(i', j', k)$, and $\widetilde{\mu}_R(i', j', k)$ as a function of k can be found in Table 4.1. The 1st column of Table 4.1 gives the labels of six cells from A to F and the 2nd column gives brief descriptions of locations of six cells on the 50×10 divided grid. Column 3 and 4 give i' and j', respectively. Column 5 gives four representative values of k, of which $k = 5.00$ is the suggested value given by

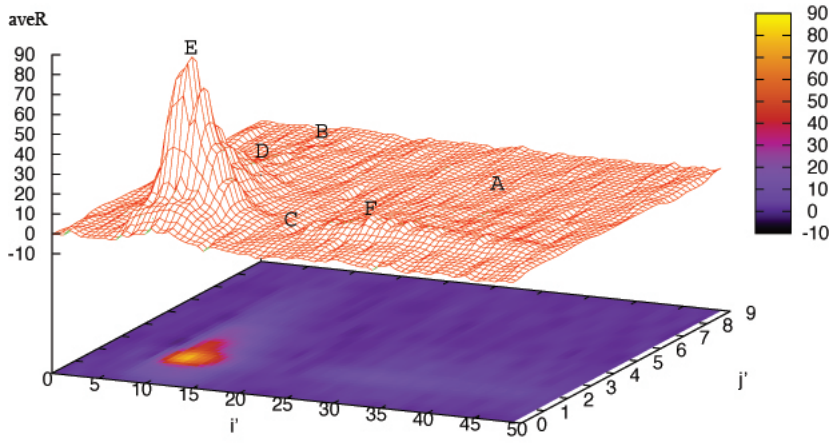

Figure 4.2: Illustration of the locations of six representative individual cells A to F.

physicists [Evans, et al.]. Column 6 to Column 8 give the three quantity values we calculated for each cell with different k's.

With a general understanding on the no-signal behavior in mind, we expect a no-signal cell to fulfill the following selection criteria:

1. $\hat{\mu}_R(i', j', k)$ of cell is expected to be a small value which is close to zero.

2. $\hat{\sigma}_R(i', j', k)$ of cell is expected to be a relatively small value.

3. $\widetilde{\mu}_R(i', j', k)$ of cell is expected to be close to $\hat{\mu}_R(i', j', k)$.

By checking the values of $\hat{\mu}_R(i', j', k)$ in Column 6, we at first excluded Cell D, Cell E and Cell F from the family of no-signal cells. Also considering Cell C locates in the middle of the main peak and the "weird unexplained hump"area, we also excluded Cell C. Next, comparing values of $\hat{\mu}_R(i', j', k)$, $\hat{\sigma}_R(i', j', k)$, and $\widetilde{\mu}_R(i', j', k)$ of Cell A to those of Cell B, we clearly see that Cell A performs better than Cell B. For the purpose of careful examination, we also included Cell B in our next investigation to determine the no-signal regions, which are expanded around the individual no-signal cells, i.e. Cell A and Cell B.

Table 4.1: Summary of the results of determination of individual no-signal cells

Cell	**Location**	i'	j'	k	$\hat{\mu}_R(i', j', k)$	$\hat{\sigma}_R(i', j', k)$	$\widetilde{\mu}_R(i', j', k)$
A	Baseline area	34	6	5.00	0.01	0.364	0.13
				5.02	0.01	0.365	0.11
				5.04	0.01	0.366	0.10
				5.06	0.00	0.368	0.09
B	End of main hump	10	8	5.00	0.08	3.990	0.12
				5.02	-0.03	4.000	0.03
				5.04	-0.13	4.017	-0.12
				5.06	-0.23	4.030	-0.21
C	Gap between E and F	22	2	5.00	0.00	0.196	0.04
				5.02	0.00	0.197	0.02
				5.04	0.00	0.197	0.01
				5.06	0.00	0.198	-0.01
D	Moderate peak of main hump	10	5	5.00	0.80	3.957	0.90
				5.02	0.69	3.970	0.82
				5.04	0.59	3.983	0.74
				5.06	0.49	3.996	0.58
E	Highest peak of main hump	10	2	5.00	10.90	4.640	11.21
				5.02	10.80	4.651	11.09
				5.04	10.70	4.663	11.02
				5.06	10.59	4.675	10.91
F	Weird unexplained hump area	30	2	5.00	0.10	0.325	0.28
				5.02	0.10	0.326	0.24
				5.04	0.09	0.327	0.22
				5.06	0.09	0.328	0.21

4.1.1.3 Determination of No-Signal Regions

To determine the no-signal regions, two expansions of cells, i.e. Expansion A and Expansion B, around Cell A and Cell B, respectively, are tried on the 50×10 divided grid. For Expansion A, the range of index of cell in TOF, i.e. i', is $i' \in [18, 49]$, and the range of index of cell in $Delay$, i.e. j', is $j' \in [4, 9]$. Thus, Expansion A has 192 cells. For Expansion B, $i' \in [7, 15]$, and $j' \in [8, 9]$. Thus, Expansion B has 18 cells. Fig. 4.3 is an illustration of the locations of Expansion A and B.

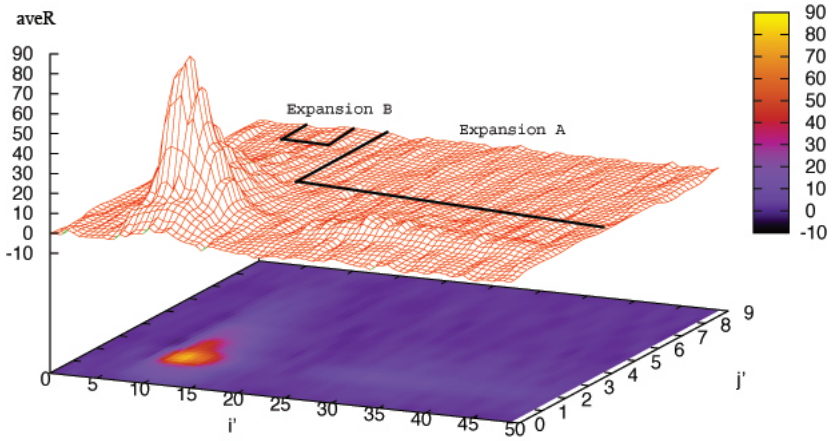

Figure 4.3: Illustration of the locations of Expansion A and B.

Distributions of *t-statistics*, defined in Eqn. 3.7, with different k's in Expansion A (with 192 *t-statistics* from 192 cells) and Expansion B (with 18 *t-statistics* from 18 cells) were calculated. The results are shown in Table 4.2 and Table 4.3, respectively. In each table, we have 13 columns. The 1st column gives six values of k in small steps of 0.01. Column 2 and 3 give the means and SDs of *t-statistics* in the expansion. Column 4 to column 12 give 9 percentiles of *t-statistics*. Column 13, titled as "Diff", gives the difference between the mean and median (50%-*percentile*) of *t-statistics*.

As we discussed in Section 3.1.2, in the one-sample *t-test*, *t-statistics* follows a standard normal distribution $\mathcal{N}(0, 1)$ under the *null hypothesis* that the true mean of $\overline{R}_{ij}(k)$, i.e.

Table 4.2: Summary of the results of distributions of 192 *t-statistics* in Expansion A

k	**Mean**	**SD**	1%	5%	10%	25%	50%	75%	90%	95%	99%	**Diff**
5.00	0.179	1.025	-2.518	-1.382	-0.954	-0.505	0.050	0.816	1.433	1.840	2.640	0.129
5.01	0.148	1.024	-2.550	-1.418	-0.975	-0.534	0.018	0.780	1.391	1.811	2.618	0.130
5.02	0.116	1.022	-2.582	-1.457	-0.996	-0.563	-0.014	0.743	1.348	1.781	2.596	0.130
5.03	0.085	1.021	-2.613	-1.495	-1.017	-0.593	-0.046	0.707	1.549	1.752	2.574	0.131
5.04	0.053	1.020	-2.645	-1.533	-1.038	-0.622	-0.073	0.671	1.288	1.723	2.552	0.126
5.05	0.022	1.018	-2.676	-1.571	-1.080	-0.653	-0.101	0.635	1.260	1.694	2.524	0.123

Table 4.3: Summary of the results of distributions of 18 *t-statistics* in Expansion B

k	**Mean**	**SD**	1%	5%	10%	25%	50%	75%	90%	95%	99%	**Diff**
5.00	0.211	1.214	-2.771	-1.632	-1.197	-0.613	0.072	0.910	1.634	2.054	2.947	0.148
5.01	0.179	1.212	-2.811	-1.677	-1.229	-0.661	0.035	0.869	1.591	2.012	2.893	0.145
5.02	0.138	1.209	-2.853	-1.715	-1.272	-0.717	-0.004	0.830	1.550	1.971	2.850	0.143
5.03	0.107	1.206	-2.895	-1.753	-1.315	-0.758	-0.051	0.788	1.509	1.929	2.807	0.143
5.04	0.068	1.204	-2.938	-1.795	-1.357	-0.797	-0.095	0.747	1.468	1.885	2.760	0.139
5.05	0.033	1.201	-2.980	-1.833	-1.399	-0.835	-0.134	0.709	1.425	1.839	2.715	0.132

$\hat{\mu}_R(i', j', k)$, is zero. With this consideration in mind, we checked the normality of data in each table and obtained observations as follows

1. With the same k, the means and SDs from Expansion B are apparently larger than those from Expansion A. Expansion A has better means and SDs for normality.

2. In two tables, the means and SDs continually decrease as k increases. In Expansion A, the means drop from 0.179 to 0.022 as k increases from 5.00 to 5.05, and meanwhile, the SDs drop from 1.025 to 1.018.

3. In Expansion A, as $k = 5.02$, distribution of *t-statistics* approaches nearly symmetric distribution with 50%-*percentile* close to zero for normality.

From the observation listed above, we concluded that Expansion A is the no-signal region.

4.1.1.4 Determination of Optimal Scaling Factor k

As we discussed in Section 4.1.1.3, as $k = 5.02$ in the no-signal region, i.e. Expansion A, distribution of *t-statistics* approaches nearly symmetric distribution with 50%-*percentile* close to zero for normality. Thus, we chose 5.02 as the optimal estimated value of the scaling factor k, i.e. $k^* = 5.02$.

Fig. 4.4 shows a three-dimensional data visualization from the software *Mathematica* for *Adenine 803153* with the strategy of grid division with the cell size of $pixelTOF \times pixelDelay = 80 \times 15$ and $k^* = 5.02$. The x-axis and y-axis, labeled respectively as "$TOF(i')$"and "$Delay(j')$", run along the directions of indices of cell in TOF and $Delay$, i.e. i' and j', respectively, and z-axis, labeled as "R ", gives the intensity of signals in each $cell(i', j')$, i.e. $\hat{\mu}_R(i', j', k^*)$.

Fig. 4.5 is a two-dimensional intensity plot for *Adenine 803153* with the strategy of grid division with the cell size of $pixelTOF \times pixelDelay = 80 \times 15$ and $k^* = 5.02$. The x-axis and y-axis run along the directions of indices of cell in TOF and $Delay$, i.e. i' and j', respectively. The brightness in graph represents the intensity of signals in each $cell(i', j')$, i.e. $\hat{\mu}_R(i', j', k^*)$.

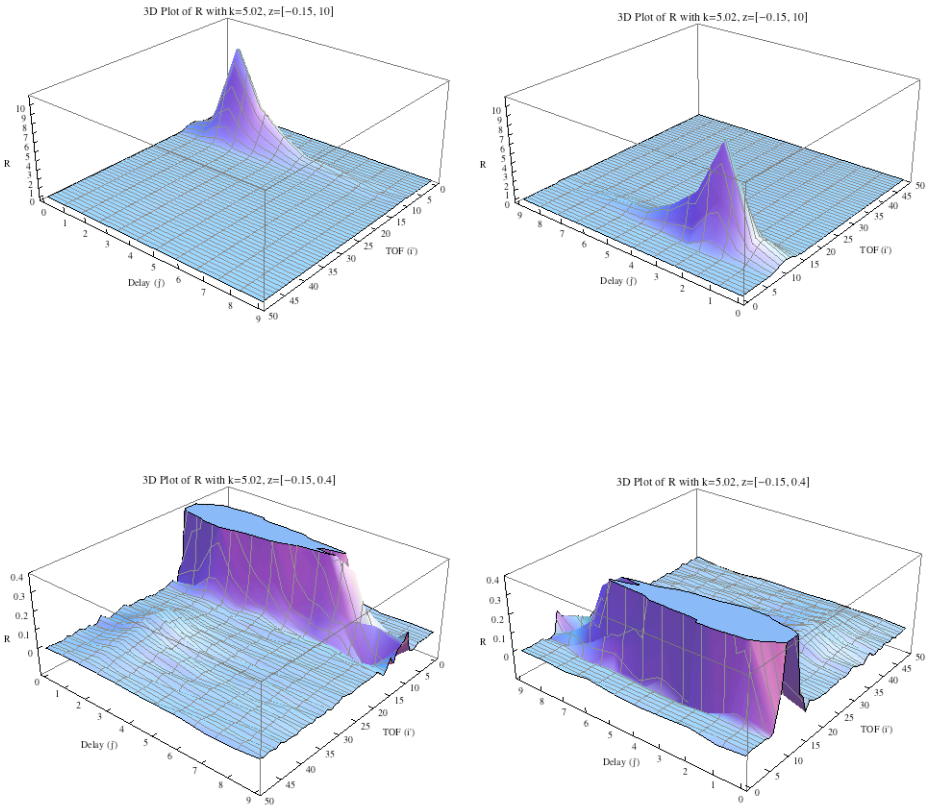

Figure 4.4: 3D visualization of *Adenine 803153*.

The brighter is the stronger. Two cut-offs of intensity were applied. One cut-off is at the main peak intensity and the other one is at the middle of main peak intensity. The left plot is with the cut-off at the main peak intensity and the right one is with the cut-off at the middle of main peak intensity.

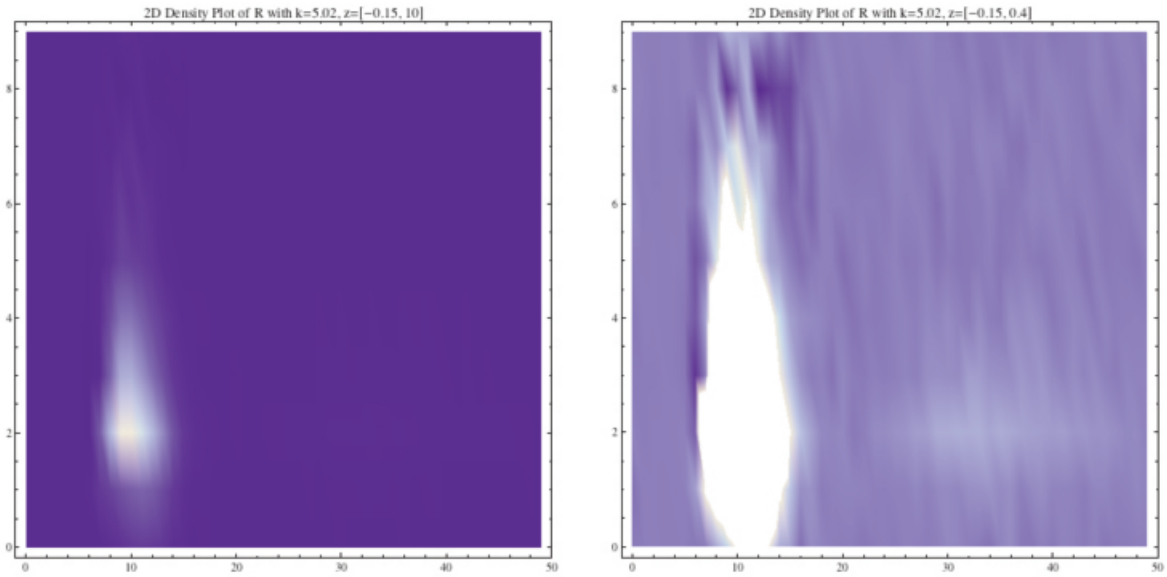

Figure 4.5: 2D intensity plot of *Adenine 803153*.

4.1.2 Gaussian-Weighted Summation over TOF

4.1.2.1 Gaussian Probability Density Function for a Fixed Level of TOF

As we discussed in Section 3.2.1, to obtain the probability density function of the skewed Gaussian distribution for a fixed level of TOF, we applied an inverse-variance-weighted Gaussian fitting.

Considering the difference between the long-lived channel (SLC) and short-lived channel (LLC), we applied two $DelayRange$s i.e. $j \in [31, 35]$ and $j \in [36, 149]$, when we calculated the quantity $\overline{R}_{\cdot i \cdot}(k^*)$ defined in Eqn. 3.8. Due to the limitation of the space, the results of $\overline{R}_{\cdot i \cdot}(k^*)$ are not presented here.

With the two $DelayRange$s introduced above, we calculated the quantities $\overline{R}_g(k^*)$ and σ_g, which are defined in Eqn. 3.9. A segment of results is shown in Table 4.4. Definitions of the

quantities in Column 2, 3, 5, and 6, i.e. g, N_g, $\overline{R}_g(k^*)$ and σ_g, can be found in Section 3.2.1. Column 4, titled as "TOF_0", gives the starting point of TOF for each group.

To find the weighting function describing the dependence of σ_g with respect to TOF, through applying the SAS procedure PROC REG, we performed a linear regression with the model defined in Eqn. 3.10 on σ_g in Table 4.4. The summary of SAS output is shown in Table 4.5. The quantity X, defined in Eqn. 3.10, is a linearly transformed predictor variable of TOF as of $X = \frac{TOF+12.5-1000}{100}$.

With the parameters β_0, β_1, β_2, and β_3 estimated by SAS, we obtained two SD functions describing the dependence of σ_g with respect to X, i.e. TOF, with two $DelayRange$s respectively, as follows

$$\hat{\sigma_g} = 7.6360 - 1.1403 * X - 0.3474 * X^2 + 0.0481 * X^3,\ DelayRange = [31, 35], \quad (4.1)$$

$$\hat{\sigma_g} = 36.3184 - 6.3575 * X - 1.9786 * X^2 + 0.3141 * X^3,\ DelayRange = [36, 149]. \quad (4.2)$$

With the weighting factor of $\frac{1}{\hat{\sigma_g}^2}$ given by Eqn. 4.1 and Eqn. 4.2, to obtain the probability density function of the skewed Gaussian distribution for a fixed level of TOF, we applied an inverse-variance-weighted Gaussian fitting, which is defined in Eqn. 3.11, on $\overline{R}_{\cdot i\cdot}(k^*)/N_{DelayRange}$, where $i \in TOFFittingRange$, through applying the SAS procedure PROC NLIN. The results are shown in Table 4.6. Column 2 shows $TOFFittingRange$. Column 3 gives two weighting functions given by Eqn. 4.1 and 4.2. Column 4 to Column 6 show respectively the fitting results of the parameters C, μ, and σ, which are defined in Eqn. 3.11. Thus, we obtained the probability density functions of the skewed Gaussian distribution for a fixed level of TOF, with two $DelayRange$s, respectively, as follows

$$f(\sqrt{TOF}) = \frac{1}{\sqrt{12.1032 * \pi}} \exp(-\frac{(\sqrt{TOF} - 28.69)^2}{12.1032}),\ DelayRange = [31, 35], \quad (4.3)$$

$$f(\sqrt{TOF}) = \frac{1}{\sqrt{8.5698 * \pi}} \exp(-\frac{(\sqrt{TOF} - 28.79)^2}{8.5698}),\ DelayRange = [36, 149]. \quad (4.4)$$

Fig. 4.6 and Fig. 4.7 show the fitting results from inverse-variance-weighted Gaussian fitting, with two $DelayRange$s, respectively. The x-axis, labeled as "$TOF(i)$", gives TOF.

Table 4.4: Summary of the results of $\overline{R}_g(k^*)$ and σ_g with two *DelayRange*s

DelayRange	g	N_g	TOF_0	$\overline{R}_g(k^*)$	σ_g
$j \in [31, 35]$	0	25	300	0.0080	0.0277
	1		325	0.0120	0.0332
	2		350	0.0080	0.0277
	3		375	0.0440	0.0583
	4		400	0.0960	0.0889
	5		425	0.0395	0.2671
	6		450	0.1978	0.5101
	...		...	...	...
	41		1325	0.8037	1.5671
	42		1350	0.2311	1.3746
	43		1375	0.7518	1.2954
	44		1400	0.3852	0.8601
	45		1425	-0.0947	0.9800
	46		1450	0.2035	0.7313
	47		1475	0.2035	0.8698
$j \in [36, 149]$	0	25	300	-0.1010	0.3253
	1		325	-0.0206	0.3106
	2		350	0.0034	0.2677
	3		375	0.0146	0.4668
	4		400	-0.1814	1.0491
	5		425	0.1618	1.4003
	6		450	-0.0138	2.2913
	...		...	...	...
	41		1325	2.2382	8.6666
	42		1350	0.6402	6.2052
	43		1375	-0.1481	5.1101
	44		1400	0.5288	4.8350
	45		1425	-0.7138	3.2393
	46		1450	2.2392	3.7579
	47		1475	-0.1200	3.4950

Table 4.5: Summary of the SAS output of PROC REG on linear regression model to obtain the weighting function for inverse-variance-weighted Gaussian fitting

$DelayRange$	Parameter	DF	Parameter Estimate	Standard Error	t Value	$Pr > \|t\|$
$j \in [31, 35]$	β_0	1	7.63598	0.27697	27.57	<.0001
	β_1	1	-1.14028	0.17513	-6.51	<.0001
	β_2	1	-0.34740	0.03200	-10.86	<.0001
	β_3	1	0.04811	0.01336	3.60	0.0011
$j \in [36, 149]$	β_0	1	36.31838	1.14491	31.72	<.0001
	β_1	1	-6.35752	0.80558	-7.89	<.0001
	β_2	1	-1.97855	0.22355	-8.85	<.0001
	β_3	1	0.31407	0.09259	3.39	0.0022

Table 4.6: Summary of the results of inverse-variance-weighted Gaussian fitting to obtain the probability density function of the skewed Gaussian distribution for a fixed level of TOF

$DelayRange$	$TOFFittingRange$	Weighting Function ($X = \frac{TOF+12.5-1000}{100}$)	Fitting Parameters		
			C	μ	σ
$j \in [31, 35]$	$i \in [525, 1400]$	$SD = 7.6360 - 1.1403 * X - 0.3474 * X^2 + 0.0481 * X^3$	81.85	28.69	2.46
$j \in [36, 149]$	$i \in [575, 1300]$	$SD = 36.3184 - 6.3575 * X - 1.9786 * X^2 + 0.3141 * X^3$	10.45	28.79	2.07

The y-axis, labeled as "$SweepAveR$", gives the magnitude of $\overline{R}_{\cdot i\cdot}(k^*)/N_{DelayRange}$, where $i \in TOFFittingRange$. The red open dots in the plot are the data to be fitted and the blue curve is the fitting line. To illustrate the improvements we obtained from the inverse-variance-weighted Gaussian fitting, in Fig. 4.8 and Fig. 4.9, we also show the fitting results from a unweighted Gaussian fitting on the same data, with two $DelayRange$s, respectively. We can see that we improved the fit in the left-hand tail with the inverse-variance-weighted Gaussian fitting.

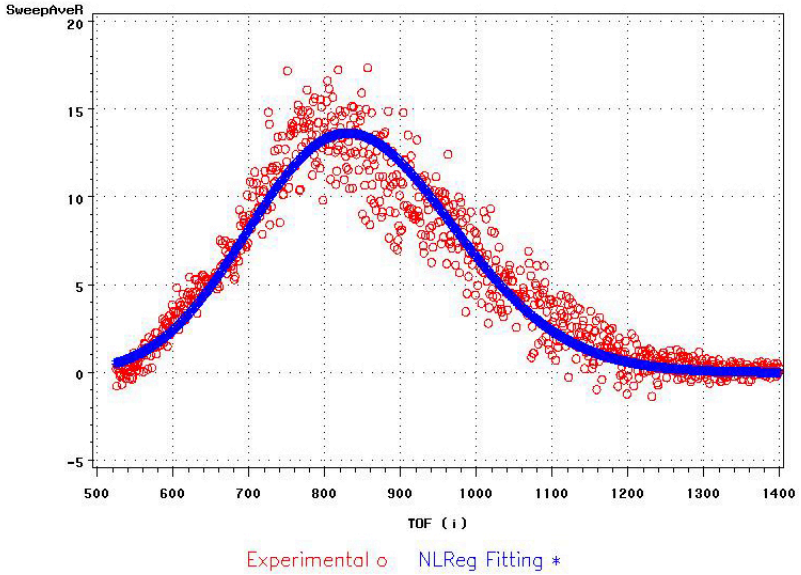

Figure 4.6: Results of inverse-variance-weighted Gaussian fitting with the $DelayRange$ of $j \in [31, 35]$.

4.1.2.2 Gaussian-Weighted Summation over TOF

With the probability density function of the skewed Gaussian distribution for a fixed level of TOF, given by Eqn. 4.3 and Eqn. 4.4, we calculated the Gaussian-weighted summation over TOF as a function of $Delay$, i.e. $\overline{R}_{\cdot\cdot j}(k^*)$, which is defined in Eqn. 3.13. The $TOFSumRange$ for summation is $[0, 3999]$. Fig. 4.10 shows the results of $\overline{R}_{\cdot\cdot j}(k^*)$, where $j \in [0, 149]$. The x-axis, labeled as "$Delay\,Time\,(j)$", gives $Delay$, running from 0 to 149. The y-axis, labeled as "$aveR\_WTTOF$", gives the magnitude of $\overline{R}_{\cdot\cdot j}(k^*)$.

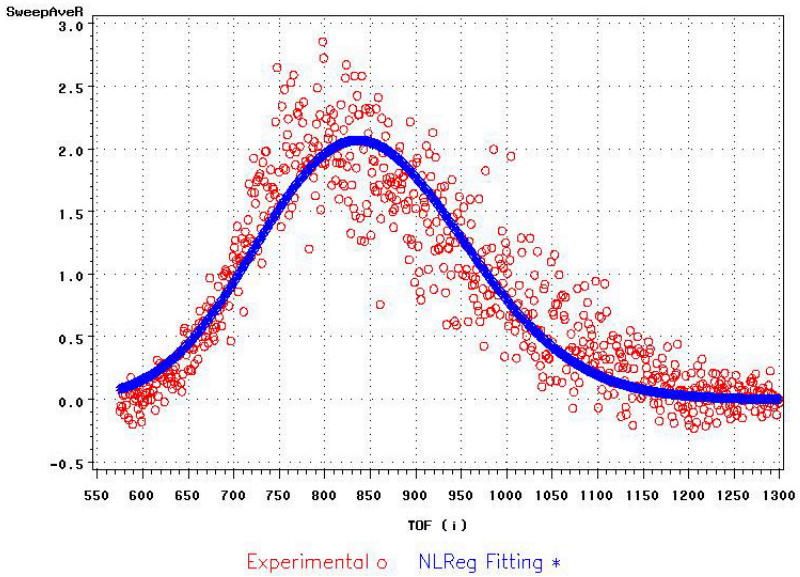

Figure 4.7: Results of inverse-variance-weighted Gaussian fitting with the *DelayRange* of $j \in [36, 149]$.

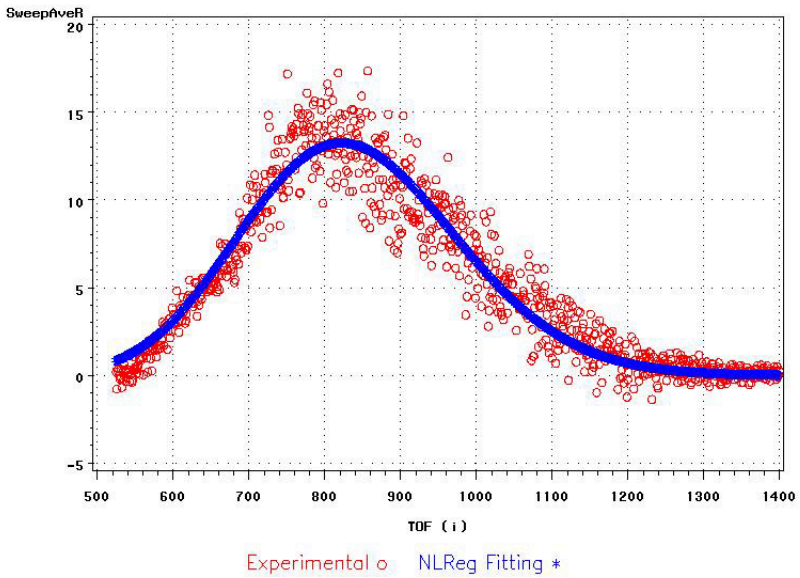

Figure 4.8: Results of unweighted Gaussian fitting with the *DelayRange* of $j \in [31, 35]$.

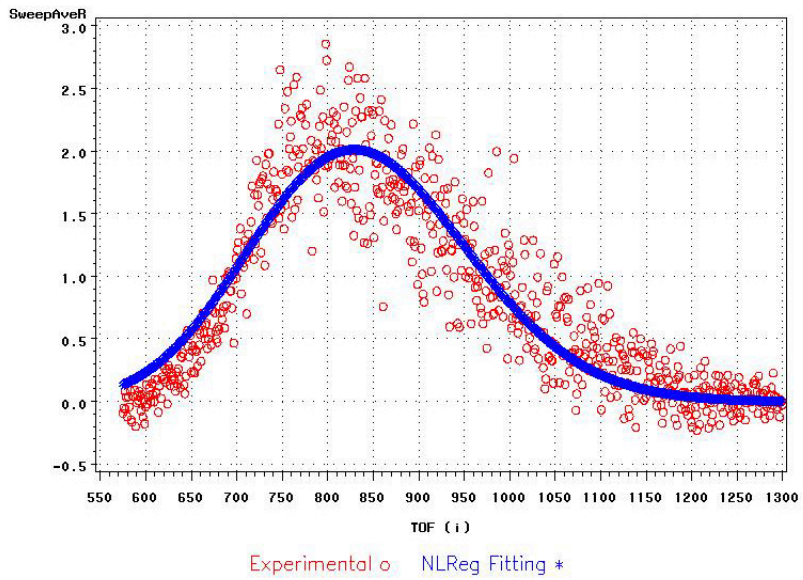

Figure 4.9: Results of unweighted Gaussian fitting with the *DelayRange* of $j \in [36, 149]$.

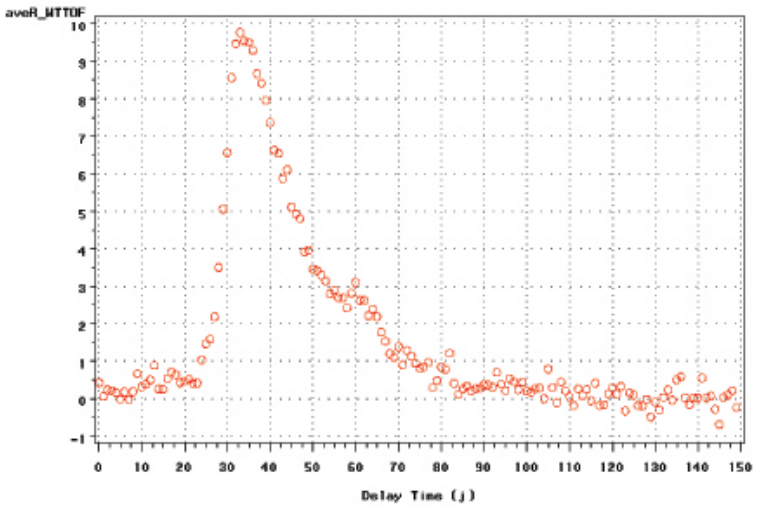

Figure 4.10: Results of Gaussian-weighted summation over TOF as a function of *Delay*.

With $\overline{R}_{..j}(k^*)$, where $j \in [0, 149]$, as we obtained in this section, we proceed to conduct the non-linear regressions with the *Single-*τ model on *Delay* in the long-lived (LLC) and short-lived (SLC) channels, as discussed in Section 3.3.

4.1.3 Non-linear Regression with the *Single-τ* Model on *Delay*

4.1.3.1 Non-linear Regression with the *Single-τ* Model on *Delay* in *LLC*

As discussed in Section 3.3.2, to obtain the lifetime τ_1 and the fitting function describing the intensity signals as an exponential decay function of *Delay* in *LLC*, we performed a non-linear regression in *LLC* with the *Single-τ* model shown in Eqn. 2.6 with *Delay* as the predictor variable.

Fig. 4.11 shows the fitting results. The x-axis, labeled as "$Delay\,Time\,(j)$", gives *Delay*, running from 40 to 149, i.e. the range of *Delay* for *LLC* which we chose in Section 3.3.1. The y-axis, labeled as "$aveR\_WTTOF$", gives the magnitude of $\overline{R}_{..j}(k^*)$, defined in Eqn. 3.13, where $j \in [40, 149]$. The red open dots in the plot are the data to be fitted and the blue curve is the fitting line. We can see that the fitting in *LLC* is fairly good over the entire fitting region.

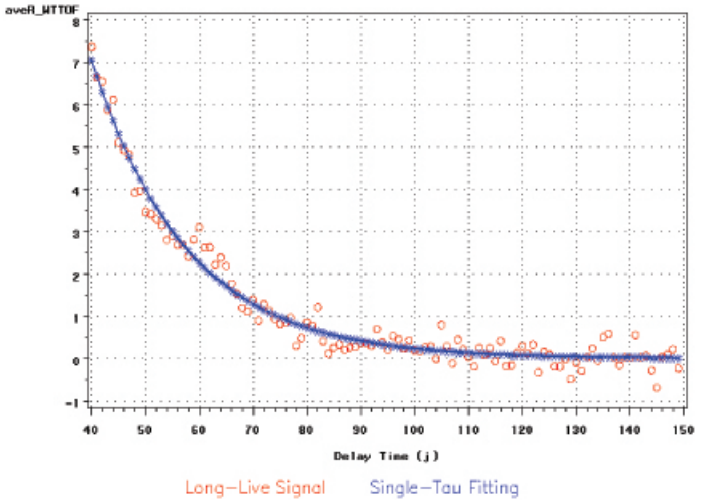

Figure 4.11: Results of non-linear regression with the *Single-τ* model in *LLC*.

A summary of results can be found in Table 4.7. Column 1 gives the range of *Delay* for *LLC*. Column 2 and 3 give respectively the best estimates of the parameters A_1 and τ_1 of *LLC*. Column 3 gives the regression standard error.

Table 4.7: Summary of the results of non-linear regression with the *Single-τ* model in *LLC*

LLC Delay Range	$\hat{A}_1$	$\hat{\tau}_1$	$\hat{\sigma}_e$
$j \in [40, 149]$	5.7229	17.6749	0.2874

Since we know that $1\,Delay\ unit = 50\,fs$, our best estimate of the lifetime τ_1 of *LLC* is

$$\tau_1 = 17.6749 \times 50\,fs = 884\,fs. \tag{4.5}$$

To obtain the fitting function describing the intensity signals as an exponential decay function of *Delay* in *LLC*, we insert $\hat{A}$ and $\hat{\tau}_1$ from Table 4.7 into Eqn. 2.6 as follows

$$\begin{aligned} I(t) &= \hat{A}_1 * \exp(\frac{\sigma^2}{2\hat{\tau}_1^2} - \frac{t}{\hat{\tau}_1}) * \{1 - \mathrm{erf}(\frac{\sigma/\hat{\tau}_1 - t/\sigma}{\sqrt{2}})\}, \\ &= 5.7229 * \exp(\frac{2.17426^2}{2 \times 17.6749^2} - \frac{t}{17.6749}) * \{1 - \mathrm{erf}(\frac{2.17426/17.6749 - t/2.17426}{\sqrt{2}})\}, \\ &= 5.7229 * \exp(0.0076 - 0.0566t) * \{1 - \mathrm{erf}(0.0870 - 0.3252t)\}, \end{aligned} \tag{4.6}$$

where $\sigma = 2.17426$ given in Section 2.1.3. $I(t)$, where $t \in [40, 149]$, is the *Single-τ*-fitted form of $\overline{R}_{..j}(k^*)$, where $j \in [40, 149]$.

4.1.3.2 Non-linear Regression with the *Single-τ* Model on *Delay* in *SLC*

As discussed in Section 3.3.2, to find the lifetime τ_2 and the fitting function describing the intensity signals as an exponential decay function of *Delay* in *SLC*, we first decomposed the overlapped signals in *SLC* by subtracting the *LLC* signals, which are given by Eqn. 4.6, from the total overlapped signals, i.e. $\overline{R}_{..j}(k^*)$, in *SLC* with $j \in [33, 39]$. Next, with the resultant signals obtained from the above decomposition, we performed a non-linear regression on those resultant signals with the *Single-τ* model again in *SLC*.

Fig. 4.12 shows the fitting results. The x-axis, labeled as "*Delay Time* (j)", gives *Delay*, running from 33 to 39, i.e. the range of *Delay* for *SLC* which we chose in Section 3.3.1.

Table 4.8: Summary of the results of non-linear regression with the *Single-τ* model in *SLC*

SLC Delay Range	$\hat{A}_2$	$\hat{\tau}_2$	$\hat{\sigma}_e$
$j \in [33, 39]$	3.2138	1.5626	0.3619

The y-axis, labeled as "*aveR_WTTOF*", gives the magnitude of the resultant data from the subtraction of $(\overline{R}_{..j}(k^*) - I(j))$, where $j \in [33, 39]$ and $I(j)$ is given by Eqn. 4.6. The red open dots in the plot are the data to be fitted and the blue curve is the fitting line.

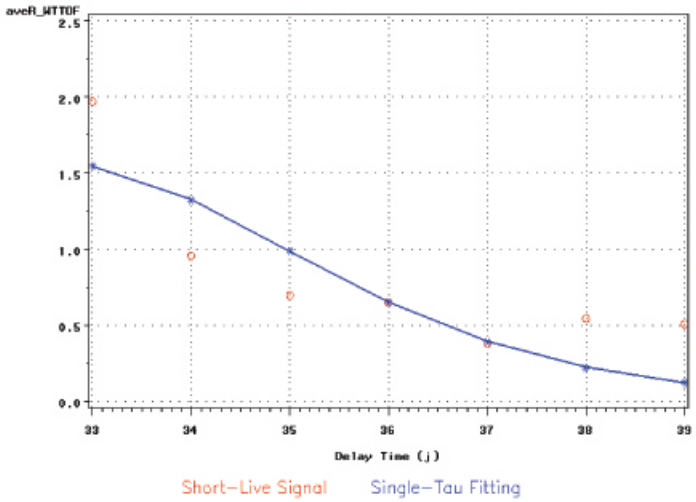

Figure 4.12: Results of non-linear regression with the *Single-τ* model in *SLC*.

A summary of results can be found in Table 4.8. Column 1 gives the range of *Delay* for *SLC*. Columns 2 and 3 give, respectively, the best estimates of the parameters A_2 and τ_2 of *SLC*. Column 3 gives the regression standard error.

Since we know that $1\,Delay\ unit = 50\,fs$, our best estimate of the lifetime τ_2 of *SLC* is

$$\tau_2 = 1.5626 \times 50\,fs = 78\,fs. \tag{4.7}$$

To obtain the fitting function describing the intensity signals as an exponential decay function of *Delay* in SLC, we insert $\hat{A}$ and $\hat{\tau}_2$ from Table 4.8 into Eqn. 2.6 as follows

$$\begin{aligned} I(t) &= \hat{A}_2 * \exp(\frac{\sigma^2}{2\hat{\tau}_2^2} - \frac{t}{\hat{\tau}_2}) * \{1 - \text{erf}(\frac{\sigma/\hat{\tau}_2 - t/\sigma}{\sqrt{2}})\}, \\ &= 3.2138 * \exp(\frac{2.17426^2}{2 \times 1.5626^2} - \frac{t}{1.5626}) * \{1 - \text{erf}(\frac{2.17426/1.5626 - t/2.17426}{\sqrt{2}})\}, \\ &= 3.2138 * \exp(0.9681 - 0.6400t) * \{1 - \text{erf}(0.9839 - 0.3252t)\}, \end{aligned} \tag{4.8}$$

where $\sigma = 2.17426$, which is given by Eqn. 2.5. $I(t)$, where $t \in [33, 39]$, is the *Single-τ*-fitted form of $\overline{R}_{..j}(k^*)$, where $j \in [33, 39]$.

4.1.3.3 Joint Non-linear Regression with the *Single-τ* Model on *Delay* in SLC

To estimate the parameters A_2 and τ_2 of SLC more precisely, we applied a joint non-linear regression in SLC with a joint non-linear regression model which is linearly constructed by two *Single-τ* models as follows

$$\begin{aligned} I(t) &= A_1 * \exp(\frac{\sigma^2}{2\tau_1^2} - \frac{t}{\tau_1}) * \{1 - \text{erf}(\frac{\sigma/\tau_1 - t/\sigma}{\sqrt{2}})\} + \\ &\quad A_2 * \exp(\frac{\sigma^2}{2\tau_2^2} - \frac{t}{\tau_2}) * \{1 - \text{erf}(\frac{\sigma/\tau_2 - t/\sigma}{\sqrt{2}})\}. \end{aligned} \tag{4.9}$$

The first part of Eqn 4.9 is the contribution from LLC and the second part is the contribution from SLC.

From the fittings we showed in Section 4.1.3.1 and Section 4.1.3.2, we obtained four best estimates of parameters, i.e. $\hat{A}_1$ and $\hat{\tau}_1$ of LLC, $\hat{A}_2$ and $\hat{\tau}_2$ of SLC, which are given in Table 4.7 and Table 4.8, respectively. We took those best estimates as the starting values for the parameter estimation procedure of PROC NLIN.

Fig. 4.13 shows the fitting results. The x-axis, labeled as "$Delay\,Time\,(j)$", gives *Delay*, running from 34 to 39. The y-axis, labeled as "$aveR\_WTTOF$", gives the magnitude of $\overline{R}_{..j}(k^*)$, where $j \in [34, 39]$. The red open dots in the plot are the data to be fitted and the blue curve is the fitting line. We note that because of the intrinsic high sensitivity of the

Table 4.9: Summary of the results of joint non-linear regression in SLC

$SLC\ Delay\ Range$	$\hat{A}_2$	$\hat{\tau}_2$	$\hat{\sigma}_e$
$j \in [34, 39]$	3.2964	1.3369	0.1095

fitting in SLC, we dropped $Delay = 33$ in our fitting since we failed to obtain a convergent fitting on $Delay = 33$, i.e. the lower boundary point of SLC.

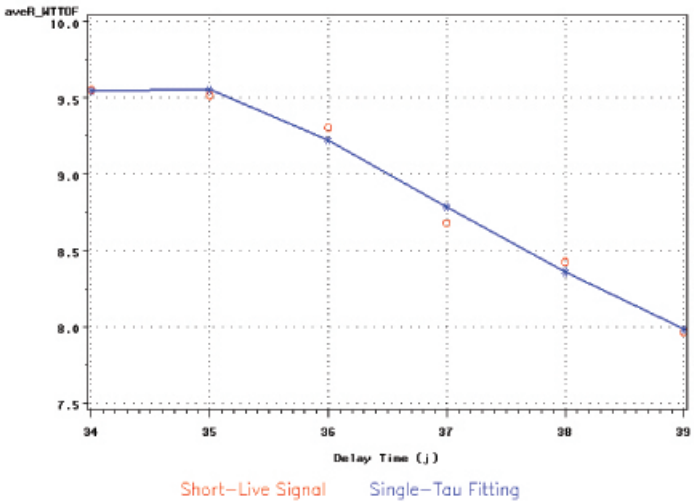

Figure 4.13: Results of joint non-linear regression in SLC.

A summary of results can be found in Table 4.9. Column 1 gives the fitting range of $Delay$ for SLC. Column 2 and 3 give respectively the best estimates of the parameters A_2 and τ_2 of SLC. Column 3 gives the regression standard error.

Since we know that $1\,Delay = 50\,fs$, therefore our best estimate of the lifetime τ_2 of SLC is

$$\tau_2 = 1.3369 \times 50\,fs = 67\,fs. \tag{4.10}$$

Table 4.10: Summary of the results of non-linear regressions on *Delay* for *Adenine 803153*

Channel	Fitting Range of *Delay*	Fitting Model	$\hat{A}$	$\hat{\tau}$	$\hat{\sigma}_e$
LLC	$j \in [40, 149]$	*Single-*τ	5.7229	17.6749 (884 fs)	0.2874
SLC	$j \in [33, 39]$	*Single-*τ	3.2138	1.5626 (78 fs)	0.3619
	$j \in [34, 39]$	*Joint Single-*τs	3.2964	1.3369 (67 fs)	0.1095

4.1.3.4 Summary of Non-linear Regression with the *Single-*τ Model on *Delay*

In Table 4.10, we give a summary of the results of non-linear regressions on *Delay* for *Adenine 803153*. Comparing our results to the experimental results listed in "TABLE 1: Excited-State Decay Lifetimes of Adenine Extracted from TRPES Spectra"in [1], we conclude that our results ($\tau_1 = 884\,fs$ and $\tau_2 = 67\,fs$) agree with the experimental results of the case of (3d) with $\tau_1 = (880 \pm 50)\,fs$ and $\tau_2 = (70 \pm 30)\,fs$, under the excitation condition of $excitation\ wavelength/energy = 251.3\,nm/4.93\,eV$.

4.2 Statistical Analysis Results for *Adenine 8030651*

To validate the statistical analysis methods we developed, we applied the methods of Gaussian-weighted summation over TOF and non-linear regressions in LLC and SLC, as we did on *Adenine 803153*, to the other new TRPES data set, *Adenine 8030651*. In this section, we show this validation. Section 4.2.1 gives a brief summary of the differences between the data set *Adenine 8030651* and *Adenine 803153*. Section 4.2.2 shows some main results of *Adenine 8030651*.

4.2.1 Brief Summary of Differences between *Adenine 8030651* and *Adenine 803153*

In this section, we give a brief summary of the differences between the data set of *Adenine 803153* and *Adenine 8030651*. A detailed summary of the main characteristics of *Adenine 803153* and *Adenine 8030651* can be found in Table 2.1 in Section 2.1.2.

The first difference is the suggested value of scaling factor k. The suggested value of k for *Adenine 8030651*is 2.00. To be time efficient, we used the suggested value of $k = 2.00$ as the optimal value of scaling factor for *Adenine 8030651*. A detailed explanation of the scaling factor k of can be found in Section 3.1.1.

As we showed in Section 2.1.3, the second difference is the *FWHM* (*Full Width at Half Maximum*) of TRPES experiments. For *Adenine 8030651*, $FWHM = 255fs$. Thus, the experimentally measurable standard deviation σ of the pump-probe delay for *Adenine 8030651* has a value of 2.16577.

The last difference is discussed in Section 3.3.1, i.e. the starting and ending points of the range of pump-probe delay time of TRPES experiments. For *Adenine 8030651*, the range of pump-probe delay time is $[1242fs, 6258fs]$. Thus, the zero point of *Delay* is calculated as $Delay_0 = \frac{1242fs}{50fs} = 24.84$. To meet the convergence criterion of non-linear fitting, the range of *Delay* for *LLC* and *SLC* for *Adenine 8030651* were chosen as $[25, 31]$ and $[32, 149]$, respectively.

4.2.2 Non-linear Regression with the *Single-τ* Model on *Delay*

In this section, we show the results from the non-linear regressions in *LLC* and *SLC* for *Adenine 8030651* with the *Single-τ* model.

Fig. 4.14 shows the fitting results in *LLC*. The x-axis, labeled as "$Delay\,Time\,(j)$", gives *Delay*, running from 32 to 149, i.e. the range of *Delay* for *LLC*. The y-axis, labeled as "$aveR\_WTTOF$", gives the magnitude of $\overline{R}_{..j}(k^*)$, where $j \in [32, 149]$. The red open dots

in the plot are the data to be fitted and the blue curve is the fitting line. We can see that, the fitting in LLC is fairly good overall the entire fitting region.

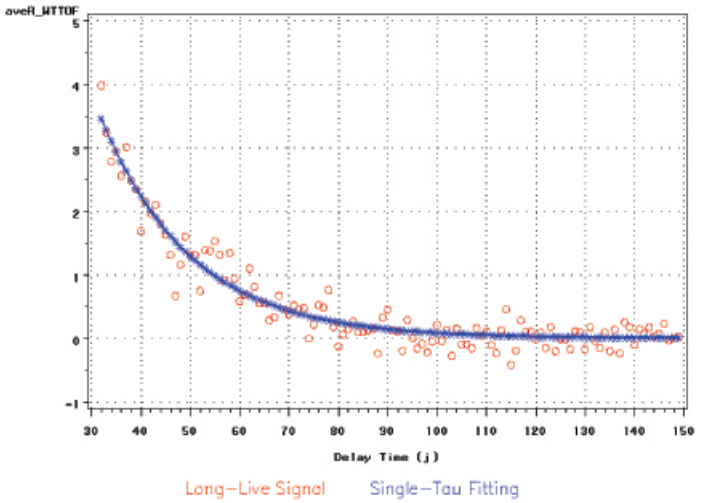

Figure 4.14: Results of non-linear regression with the *Single-*τ model in LLC.

Fig. 4.15 shows the fitting results in the SLC region. The x-axis, labeled as "$Delay\,Time\,(j)$", gives $Delay$, running from 25 to 31, i.e. the range of $Delay$ for SLC. The y-axis, labeled as "$aveR\_WTTOF$", gives the magnitude of the resultant data from the signal subtraction in SLC. The red open dots in the plot are the data to be fitted and the blue curve is the fitting curve.

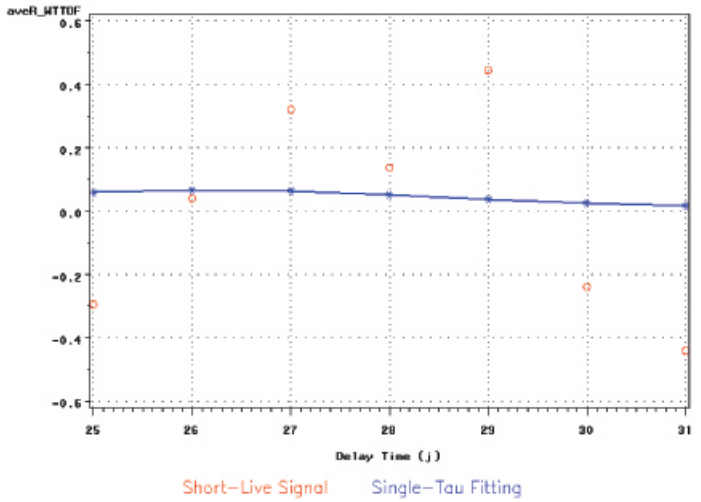

Figure 4.15: Results of non-linear regression with the *Single-*τ model in SLC.

Table 4.11: Summary of the results of non-linear regressions on *Delay* for *Adenine 8030651*

Channel	Fitting Range of *Delay*	Fitting Model	$\hat{A}$	$\hat{\tau}$	$\hat{\sigma}_e$
LLC	$j \in [32, 149]$	*Single-*τ	2.5420	18.3407 (917 fs)	0.2304
SLC	$j \in [25, 31]$	*Single-*τ	0.1191	1.8188 (91 fs)	0.3585

In Table 4.11, we give a summary of the results of non-linear regressions on *Delay* for *Adenine 8030651*. Comparing our regression results to the experimental results listed in "TABLE 1: Excited-State Decay Lifetimes of Adenine Extracted from TRPES Spectra" in [1], within the estimated experimental error range, we found that our results ($\tau_1 = 917\, fs$ and $\tau_2 = 91\, fs$) agree with the experimental results of the case of (3d) with $\tau_1 = (880 \pm 50)\, fs$ and $\tau_2 = (70 \pm 30)\, fs$, under the excitation condition of $excitation\ wavelength/energy = 251.3\, nm/4.93\, eV$, and (2d) with $\tau_1 = (938 \pm 50)\, fs$ and $\tau_2 = (71 \pm 30)\, fs$, under the excitation condition of $excitation\ wavelength/energy = 259.9\, nm/4.77\, eV$.

Chapter 5

Conclusions

We developed statistical analysis methods for the study on the decay lifetimes of photo-excited DNA nucleobase Adenine. We applied them on the new TRPES data set *Adenine 803153* and obtained a good agreement between the results from statistical analyses and those from physicists' [Evans, [1]] experiments. To validate those methods, we applied them on the other new TRPES data set *Adenine 8030651*. Within the estimated experimental error range, we also obtained an agreement between statistical analyses and physical experiments.

With the TRPES data set *Adenine 803153*, we performed an extensive study on the background signal subtraction, from which we determined a proper background signal subtraction with an optimal estimated value of scaling factor k, i.e. $k^* = 5.02$. We also improved the procedure for the summation over TOF. We applied a Gaussian-weighted summation over TOF. To find the probability density function of the skewed Gaussian distribution for a fixed level of TOF, we applied an inverse-variance-weighted Gaussian fitting with two $DelayRange$s, i.e. $j \in [31, 35]$ and $j \in [36, 149]$. We also followed a new physics understanding to extract the decay lifetimes τ_1 and τ_2 of *Adenine 803153*. We focused on the non-linear regressions with the *Single-*τ model in the long-lived (LLC) and short-lived (SLC) channels. We chose the LLC in the range of *Delay* of $[40, 149]$ and the SLC in the range of *Delay* of $[33, 39]$. To purify the signals in SLC and extract the decay lifetime τ_2, we performed a signal decomposition by a linear subtraction of signals with the total intensity signals and the fitting signals from LLC. Through applying the *Single-*τ non-linear regressions separately in LLC and SLC, we obtained the decay lifetime $\tau_1 = 17.6749(884\, fs)$

of LLC and the decay lifetime $\tau_2 = 1.5626(78\,fs)$ of SLC. We also applied a joint non-linear regression in SLC with a joint non-linear regression model which is linearly constructed by two *Single-τ* models. Through applying this joint non-linear regression in SLC, we improved our estimate on τ_2 with $\tau_2 = 1.3369(67\,fs)$ in SLC. From those regression results, we concluded that our results agree with the experimental results of the case of (3d) in Ref. [1] with $\tau_1 = (880 \pm 50)\,fs$ and $\tau_2 = (70 \pm 30)\,fs$, under the excitation condition of $excitation\ wavelength/energy = 251.3\,nm/4.93\,eV$.

To validate the statistical analysis methods, we repeated the methods of Gaussian-weighted summation over TOF and non-linear regressions in LLC and SLC on *Adenine 8030651*. Since *Adenine 8030651* has different starting and ending points of the range of pump-probe delay time of TRPES experiments, we have different zero point $Delay_0$. For *Adenine 8030651*, we chose the LLC in the range of $Delay$ of $[32, 149]$ and the SLC in the range of $Delay$ of $[25, 31]$. Through applying the *Single-τ* non-linear regressions separately in LLC and SLC, we obtained the decay lifetime $\tau_1 = 18.3407(917\,fs)$ of LLC and the decay lifetime $\tau_2 = 1.8188(91\,fs)$ of SLC. Within the estimated experimental error range, those regression results agree with the experimental results of the case of (3d) with $\tau_1 = (880 \pm 50)\,fs$ and $\tau_2 = (70 \pm 30)\,fs$, under the excitation condition of $excitation\ wavelength/energy = 251.3\,nm/4.93\,eV$, and (2d) with $\tau_1 = (938 \pm 50)\,fs$ and $\tau_2 = (71 \pm 30)\,fs$, under the excitation condition of $excitation\ wavelength/energy = 259.9\,nm/4.77\,eV$.

Bibliography

[1] N. L. Evans and S. Ullrich, *J. Phys. Chem. A 114(42):11225-30* (2010)

[2] Z. Hou, L. Pang and J. Reeves, *Report on STAT 8000 (Supervised Statistical Consulting) class project "Decay Times for Genetic Bases"* (2007)

[3] D. Wackerly, W. Mendenhall III and R. L. Scheaffer, *Mathematical Statistics with Applications (6th Ed.).* (Duxbury Press, Thomson Higher Education, Belmont, 2001).

Appendix A

The SAS Codes

A.1 Linear Regression with the Model Given by Eqn. 3.10 for *Adenine 803153*

```
options ps=100 ls=78 pageno=1 formdlim='*';

/* Input Data */

data Mtg0711.Delay31_39_TOF525_1400;
  infile "C:\Documents and Settings\ZHUOFEI HOU\My Documents\STAT_MSProject
          \dataSet_new\Meeting_100711
          \sumupDelay31-39_TOF300-1500-Step25_MeanSD.txt";
  input Group N TOF0 Mean SD;
  if(TOF0>=525 && TOF0<=1400);
  X = (TOF0+12.5-1000)/100;
  Q = X**2;
  C = X**3;
run;

data Mtg0711.Delay40_149_TOF575_1300;
  infile "C:\Documents and Settings\ZHUOFEI HOU\My Documents\STAT_MSProject
          \dataSet_new\Meeting_100711
          \sumupDelay40-149_TOF300-1500-Step25_MeanSD.txt";
  input Group N TOF0 Mean SD;
  if(TOF0>=575 && TOF0<=1300);
  X = (TOF0+12.5-1000)/100;
  Q = X**2;
  C = X**3;
run;

/* Regressions */

title2 "Fitting on Delay31_39_TOF525_1400";
proc reg data=Mtg0711.Delay31_39_TOF525_1400;
  model SD = X Q C / P R;
run;
```

```
title2 "Fitting on Delay40_149_TOF575_1300";
proc reg data=Mtg0711.Delay40_149_TOF575_1300;
  model SD = X Q C / P R;
run;

/* Quit */

quit;
```

A.2 Inverse-variance-weighted Gaussian Fitting with the Model Given by Eqn. 3.11 for *Adenine 803153*

```
options ps=100 ls=78 pageno=1 formdlim='*';

/* Input Data */

data Mtg0806.SumupDelay40_149_TOF0_3999;
  infile "C:\Documents and Settings\ZHUOFEI HOU\My Documents\STAT_MSProject
            \dataSet_new\Meeting_100806
            \curveAlongi_SumupDelay40-149_TOF0-3999.txt";
  input TOF sqrtTOF EBE aveR1;
  aveR = aveR1;
  X = (TOF+12.5-1000)/100;
  Q = X**2;
  C = X**3;
  SD = 0.32383-0.05579*X-0.01817*Q+0.00281*C;
  W = 1.0/SD**2;
run;

data Mtg0806.SumupDelay40_149_TOF0_3999_sym;
  set Mtg0806.SumupDelay40_149_TOF0_3999;
  where TOF>=575 && TOF<=1300;
run;

data Mtg0806.SumupDelay31_39_TOF0_3999;
  infile "C:\Documents and Settings\ZHUOFEI HOU\My Documents\STAT_MSProject
            \dataSet_new\Meeting_100806
            \curveAlongi_SumupDelay31-39_TOF0-3999.txt";
  input TOF sqrtTOF EBE aveR1;
  aveR = aveR1;
  X = (TOF+12.5-1000)/100;
```

```
  Q = X**2;
  C = X**3;
  SD = 1.13763-0.20597*X-0.04952*Q+0.00920*C;
  W = 1.0/SD**2;
run;

data Mtg0806.SumupDelay31_39_TOF0_3999_sym;
  set Mtg0806.SumupDelay31_39_TOF0_3999;
  where TOF>=525 && TOF<=1400;
run;

/* Inverse-variance-weighted Gaussian Fitting */

%let PI = 3.1415926;

title "Weighted NLIN on SumupDelay40_149_TOF0_3999_sym";
proc nlin data=Mtg0806.SumupDelay40_149_TOF0_3999_sym method=marquardt;
  parms c=100 u=30 s=2;
  _weight_ = W;
  model aveR=c*(1/sqrt(2*&PI*s*s))*exp(-1*(sqrtTOF-u)*(sqrtTOF-u)/(2*s*s));
  output out=nlinout_sym1 predicted=pred residual=r
         l95m=l95mean u95m=u95mean l95=l95ind u95=u95ind;
run;

title "Weighted NLIN on SumupDelay31_39_TOF0_3999_sym";
proc nlin data=Mtg0806.SumupDelay31_39_TOF0_3999_sym method=marquardt;
  parms c=100 u=30 s=2;
  _weight_ = W;
  model aveR=c*(1/sqrt(2*&PI*s*s))*exp(-1*(sqrtTOF-u)*(sqrtTOF-u)/(2*s*s));
  output out=nlinout_sym2 predicted=pred residual=r
         l95m=l95mean u95m=u95mean l95=l95ind u95=u95ind;
run;

/* Plotting */

goptions reset=all;
symbol1 v=circle cv=red h=1;
symbol2 v=: cv=blue h=1 i=join l=1 w=2 ci=blue;
footnote1 h=1.5 c=red   f=simplex 'Experimental o'
          h=1.5 c=blue  f=simplex '    NLReg Fitting *';
title1 h=1.8 'SweepAveR vs. TOF for Adenine 803153';

axis1 label=('SweepAveR') length=30 order=(-.5 to 3 by .5 )
      minor=(number=4);
axis2 label=('TOF (i)') length=80 order=(550 to 1300 by 50 )
```

```
      minor=(number=4);
title2 h=1.5 "Delay Time Sum-Range=40-149, TOF Range=575-1300, k=5.02";
proc gplot data=nlinout_sym1 UNIFORM;
  plot aveR*TOF pred*TOF / overlay vaxis=axis1 haxis=axis2 frame grid;
run;

axis1 label=('SweepAveR') length=30 order=(-5 to 20 by 5 )
      minor=(number=4);
axis2 label=('TOF (i)') length=80 order=(500 to 1400 by 100 )
      minor=(number=4);
title2 h=1.5 "Delay Time Sum-Range=31-39, TOF Range=525-1400, k=5.02";
proc gplot data=nlinout_sym2;
  plot aveR*TOF pred*TOF / overlay vaxis=axis1 haxis=axis2 frame grid;
run;

/* Quit */

quit;
```

A.3 Non-linear Regression with the *Single-τ* Model Given by Eqn. 2.6 in *LLC* for *Adenine 803153*

```
options ps=100 ls=78 pageno=1 formdlim='*';

/* Input Data */

data Mtg0911.Delay0_149_WTSumTOF;
  infile "C:\Documents and Settings\ZHUOFEI HOU\My Documents\STAT_MSProject
            \dataSet_new\Meeting_100911
            \curveAlongj_Delay0-149_WTSumTOF.dat";
  input delay aveR;
run;

data Mtg0911.Delay40_149_WTSumTOF;
set Mtg0911.Delay0_149_WTSumTOF;
where delay >= 40;
delay1 = delay - 31.3;
run;

/* Non-linear Fitting in LLC */

%let sig = 2.17426;
```

```
title "NLIN on Delay40_149_WTSumTOF";
proc nlin data=Mtg0911.Delay40_149_WTSumTOF method=marquardt;
  parms c=30 tau=14;
  model aveR=c*exp(&sig*&sig/(2*tau*tau)-delay1/tau)*
  (1-2*(cdf('normal',&sig/tau-delay1/&sig,0,1)-cdf('normal',0,0,1)));
  output out=nlinout predicted=pred residual=r l95m=l95mean u95m=u95mean
                                               l95=l95ind u95=u95ind;
run;

/* Plotting */

goptions reset=all;
axis1 label=('aveR_WTTOF') length=30 order=(-1 to 8 by 1)
      minor=(number=1);
axis2 label=('Delay Time (j)') length=80;
symbol1 v=circle cv=red h=1;
symbol2 v=: cv=blue h=1 i=join l=1 w=2 ci=blue;
footnote1 h=1.5 c=red   f=simplex 'Long-Live Signal'
          h=1.5 c=blue  f=simplex '       Single-Tau Fitting';
title1 h=1.8 'aveR_WTTOF vs. Pump-Probe Delay Time
              for Long-Live Chanel of Adenine';
title2 h=1.5 "Delay Time Range=40-149, k=5.02, 80315";
proc gplot data=nlinout;
  plot aveR*delay pred*delay / overlay vaxis=axis1 haxis=axis2 frame grid;
run;

/* Plotting Whole Range */

goptions reset=all;
axis1 label=('aveR_WTTOF') length=30 order=(-1 to 10 by 1)
      minor=(number=1);
axis2 label=('Delay Time (j)') length=80;
symbol1 v=circle cv=red h=1;
/*footnote1 h=1.5 c=red   f=simplex 'Total Intensity Signal';*/
title1 h=1.8 'aveR_WTTOF vs. Pump-Probe Delay Time of Adenine';
title2 h=1.5 "Delay Time Range=0-149, k=5.02, 80315";
proc gplot data=Mtg0911.Delay0_149_WTSumTOF;
  plot aveR*delay  / overlay vaxis=axis1 haxis=axis2 frame grid;
run;

/* Quit */

quit;
```

A.4 Non-linear Regression with the *Single-τ* Model Given by Eqn. 2.6 in *SLC* for *Adenine 803153*

```
options ps=100 ls=78 pageno=1 formdlim='*';

/* Input Data */

data Mtg0911.Delay0_149_WTSumTOF;
  infile "C:\Documents and Settings\ZHUOFEI HOU\My Documents\STAT_MSProject
           \dataSet_new\Meeting_100911
           \curveAlongj_Delay0-149_WTSumTOF_BK1.dat";
  input delay aveR;
run;

data Mtg0911.Delay33_39_WTSumTOF;
set Mtg0911.Delay0_149_WTSumTOF;
where delay >=33 && delay <= 39;
delay1 = delay - 31.3;
run;

/* Linear Subtraction in SLC */

%let sig = 2.17426;
%let C1 =  5.7229;
%let tau1 =  17.6749;

data Mtg0911.Delay33_39_LLSubtracted;
set Mtg0911.Delay33_39_WTSumTOF;
LL = &C1*exp(&sig*&sig/(2*&tau1*&tau1)-delay1/&tau1)*
     (1-2*(cdf('normal',&sig/&tau1-delay1/&sig,0,1)-cdf('normal',0,0,1)));
SL = aveR - LL;
run;

/* nonlinear fitting in separately SL*/

title "NLIN on Delay33_39_LLSubtracted";
proc nlin data=Mtg0911.Delay33_39_LLSubtracted method=marquardt;
  parms c=5, tau=2;
  model SL = c*exp(&sig*&sig/(2*tau*tau)-delay1/tau)*
  (1-2*(cdf('normal',&sig/tau-delay1/&sig,0,1)-cdf('normal',0,0,1)));
  output out=nlinout1 predicted=pred residual=r l95m=l95mean u95m=u95mean
                                                l95=l95ind u95=u95ind;
run;
```

```
/* Plotting */

goptions reset=all;
axis1 label=('aveR_WTTOF') length=30 order=(0 to 2.5 by 0.5)
      minor=(number=1);
axis2 label=('Delay Time (j)') length=80;
symbol1 v=circle cv=red h=1;
symbol2 v=: cv=blue h=1 i=join l=1 w=2 ci=blue;
footnote1 h=1.5 c=red   f=simplex 'Short-Live Signal'
          h=1.5 c=blue  f=simplex '       Single-Tau Fitting';
title1 h=1.8 'aveR_WTTOF vs. Pump-Probe Delay Time
              for Short-Live Chanel of Adenine';
title2 h=1.5 "Delay Time Range=33-39, k=5.02, 80315";
proc gplot data=nlinout1;
  plot SL*delay pred*delay / overlay vaxis=axis1 haxis=axis2 frame grid;
run;

/* Quit */

quit;
```

Appendix B

The C++ Codes

B.1 Data File Creation for SAS Input Procedures in Appendices A.1 and A.2

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <string>
#include <vector>
#include <map>
#include <cmath>

int main()
{
 std::ifstream ifs_P[10], ifs_E[10], ifs_D[10];
 std::string   inFileName;

 // Input P E D
 std::cout<<"\nReading 10 P, E and D files......"<<std::endl;
 for(unsigned int k=0; k<10; k++) {
   std::ostringstream oss_P, oss_E, oss_D;
   oss_P<<"P"<<k+1<<".dat";
   oss_E<<"E"<<k+1<<".dat";
   oss_D<<"D"<<k+1<<".dat";
   ifs_P[k].open(oss_P.str().c_str(),std::ios::in);
   ifs_E[k].open(oss_E.str().c_str(),std::ios::in);
   ifs_D[k].open(oss_D.str().c_str(),std::ios::in);
 }

 std::cout<<"\nProgram Running......\n"<<std::endl;

 // Initialization
 double** sweepAveBG = new double*[4000];
 double** sweepAveD  = new double*[4000];
 for(unsigned int i=0; i<4000; i++) {
```

```
  sweepAveBG[i] = new double[151];
  sweepAveD[i]  = new double[151];
}
for(unsigned int i=0; i<4000; i++) {
  for(int j=0; j<151; j++) {
    sweepAveBG[i][j] = 0.;
    sweepAveD[i][j]  = 0.;
  }
}

double*** BG = new double**[10];
double*** D  = new double**[10];
for(unsigned int k=0; k<10; k++) {
  BG[k] = new double*[4000];
  D[k]  = new double*[4000];
}
for(unsigned int k=0; k<10; k++) {
  for(unsigned int i=0; i<4000; i++) {
    BG[k][i] = new double[151];
    D[k][i]  = new double[151];
  }
}
for(unsigned int k=0; k<10; k++) {
  for(unsigned int i=0; i<4000; i++) {
    for(int j=0; j<151; j++) {
      BG[k][i][j] = 0.;
      D[k][i][j]  = 0.;
    }
  }
}

// Calculate sweepAveBG, sweepAveD, sweepAveR, BG and D
int p, e, d;
for(unsigned int k=0; k<10; k++) {
  for(unsigned int i=0; i<4000; i++) {
    for(unsigned int j=0; j<151; j++) {
      ifs_P[k]>>p;
      ifs_E[k]>>e;
      ifs_D[k]>>d;
      sweepAveBG[i][j] += (p+e)*0.1;
      sweepAveD[i][j]  += d*0.1;
      BG[k][i][j] = (p+e);
      D[k][i][j]  = d;
    }
  }
```

```
}

const double K = 5.02;

double sweepAveR[4000][151];
for(unsigned int i=0; i<4000; i++) {
  for(unsigned int j=0; j<151; j++) {
    sweepAveR[i][j] = sweepAveD[i][j]-K*sweepAveBG[i][j];
  }
}

// Curve along Delay_SumupTOF0-3999
// Curve along TOF_SumupDelay31-35 and 36-149
std::ostringstream oss1;
std::ofstream ofs1;
oss1<<"curveAlongj_Delay31-149_SumupTOF"<<0<<"-"<<3999<<".dat";
ofs1.open(oss1.str().c_str(),std::ios::out);
unsigned int startingDelay = 31,
             endingDelay = 149;
for(unsigned int j=startingDelay; j<=endingDelay; j++) {
  double sum = 0.;
  for(unsigned int i=0; i<4000; i++) {
    sum += sweepAveR[i][j];
  }
  ofs1<<std::setw(12)<<std::left<<j-31<<std::setw(15)
                     <<std::left<<std::setprecision(4)<<std::fixed<<sum
                     <<std::endl;
}

std::ostringstream oss2, oss3;
std::ofstream ofs2, ofs3;
oss2<<"curveAlongi_SumupDelay31-39_TOF"<<0<<"-"<<3999<<".dat";
oss3<<"curveAlongi_SumupDelay40-149_TOF"<<0<<"-"<<3999<<".dat";
ofs2.open(oss2.str().c_str(),std::ios::out);
ofs3.open(oss3.str().c_str(),std::ios::out);

std::vector<double> TOFR1, TOFR2;
unsigned int startingDelay1 = 31, endingDelay1 = 39,
             startingDelay2 = 40, endingDelay2 = 149,
             TOF_low = 300,
             TOF_up = 1500;

for(unsigned int i=0; i<4000; i++) {
  double sum1 = 0.,
         sum2 = 0.;
```

```
  for(unsigned int j=startingDelay1; j<=endingDelay1; j++)
    sum1 += sweepAveR[i][j]/9.;
  for(unsigned int j=startingDelay2; j<=endingDelay2; j++)
    sum2 += sweepAveR[i][j]/110.;

  if(i>=TOF_low && i<TOF_up) {
    TOFR1.push_back(sum1);
    TOFR2.push_back(sum2);
  }

  double ebe=11.09-(9.2329e-13/((i*1.e-9-9.8341e-9)*(i*1.e-9-9.8341e-9))
                    -0.39569);

  ofs2<<std::setw(15)<<std::left<<i
      <<std::setw(20)<<std::left<<std::setprecision(6)<<sqrt(i)
      <<std::setw(20)<<std::left<<std::setprecision(6)<<ebe
      <<std::setw(20)<<std::left<<std::setprecision(6)<<std::fixed<<sum1
      <<std::endl;
  ofs3<<std::setw(15)<<std::left<<i
      <<std::setw(20)<<std::left<<std::setprecision(6)<<sqrt(i)
      <<std::setw(20)<<std::left<<std::setprecision(6)<<ebe
      <<std::setw(20)<<std::left<<std::setprecision(6)<<std::fixed<<sum2
      <<std::endl;
}

// Mean and SD analysis
double sqrtTOF_low = sqrt(TOF_low),
       sqrtTOF_up  = sqrt(TOF_low+TOFR1.size()),
       EBE_low = 11.09-(9.2329e-13/((TOF_low*1.e-9-9.8341e-9)
                      *(TOF_low*1.e-9-9.8341e-9))-0.39569),
       EBE_up=11.09-(9.2329e-13/(((TOF_low+TOFR1.size())*1.e-9-9.8341e-9)
                     *((TOF_low+TOFR1.size())*1.e-9-9.8341e-9))-0.39569);

std::ostringstream oss_MeanSD[6];
std::ofstream ofs_MeanSD[6];
oss_MeanSD[0]<<"sumupDelay31-39_TOF"<<TOF_low<<"-"<<TOF_up
             <<"-Step25_MeanSD.dat";
oss_MeanSD[1]<<"sumupDelay31-39_sqrtTOF_"<<sqrtTOF_low<<"-"<<sqrtTOF_up
             <<"-Step0Pt5_MeanSD.dat";
oss_MeanSD[2]<<"sumupDelay31-39_EBE_"<<EBE_low<<"-"<<EBE_up
             <<"-Step0Pt1_MeanSD.dat";
oss_MeanSD[3]<<"sumupDelay40-149_TOF"<<TOF_low<<"-"<<TOF_up
             <<"-Step25_MeanSD.dat";
oss_MeanSD[4]<<"sumupDelay40-149_sqrtTOF_"<<sqrtTOF_low<<"-"<<sqrtTOF_up
             <<"-Step0Pt5_MeanSD.dat";
```

```
oss_MeanSD[5]<<"sumupDelay40-149_EBE_"<<EBE_low<<"-"<<EBE_up
             <<"-Step0Pt1_MeanSD.dat";
for(unsigned int i=0; i<6; i++) {
  ofs_MeanSD[i].open(oss_MeanSD[i].str().c_str(),std::ios::out);
  if(i==0||i==3) {
    ofs_MeanSD[i]<<std::setw(20)<<std::left<<"Group"<<std::setw(20)
                 <<std::left<<"N"
                 <<std::setw(20)<<std::left<<"TOF0"<<std::setw(20)
                 <<std::left<<"Mean"<<std::setw(20)<<std::left<<"SD"
                 <<std::endl;
  }
  else if(i==1||i==4) {
    ofs_MeanSD[i]<<std::setw(20)<<std::left<<"Group"<<std::setw(20)
                 <<std::left<<"N"
                 <<std::setw(20)<<std::left<<"sqrtTOF0"<<std::setw(20)
                 <<std::left<<"Mean"<<std::setw(20)<<std::left<<"SD"
                 <<std::endl;
  }
  else {
    ofs_MeanSD[i]<<std::setw(20)<<std::left<<"Group"<<std::setw(20)
                 <<std::left<<"N"
                 <<std::setw(20)<<std::left<<"EBE0"<<std::setw(20)
                 <<std::left<<"Mean"<<std::setw(20)<<std::left<<"SD"
                 <<std::endl;
  }
}

for(unsigned int i=0; i<TOFR1.size(); i+=25) {
  double mean1 = 0., sd1 = 0.,
         mean2 = 0., sd2 = 0.;
  for(unsigned int j=i; j<i+25; j++) {
    mean1 += TOFR1[j]/25.;
    mean2 += TOFR2[j]/25.;
  }
  for(unsigned int j=i; j<i+25; j++) {
    sd1 += (TOFR1[j]-mean1)*(TOFR1[j]-mean1);
    sd2 += (TOFR2[j]-mean2)*(TOFR2[j]-mean2);
  }
  sd1 = sqrt(sd1/24.);
  sd2 = sqrt(sd2/24.);

  ofs_MeanSD[0]<<std::setw(20)<<std::left<<i/25<<std::setw(20)
               <<std::left<<25<<std::setw(20)<<std::left<<i+TOF_low
               <<std::setw(20)<<std::left<<std::setprecision(6)<<mean1
               <<std::setw(20)<<std::left<<std::setprecision(6)<<sd1
```

```
                <<std::endl;
  ofs_MeanSD[3]<<std::setw(20)<<std::left<<i/25<<std::setw(20)
                <<std::left<<25<<std::setw(20)<<std::left<<i+TOF_low
                <<std::setw(20)<<std::left<<std::setprecision(6)<<mean2
                <<std::setw(20)<<std::left<<std::setprecision(6)<<sd2
                <<std::endl;
}

std::map<double,double> sqrtTOFR1, sqrtTOFR2;
for(unsigned int i=0; i<TOFR1.size(); i++) {
  sqrtTOFR1[sqrt(i+TOF_low)] = TOFR1[i];
  sqrtTOFR2[sqrt(i+TOF_low)] = TOFR2[i];
}
std::map<double,double>::const_iterator itr1 = sqrtTOFR1.begin(),
                                        itr2 = sqrtTOFR2.begin();
unsigned int gc_sqrtTOF = 0;

for(double sqrtTOF0=sqrtTOF_low; sqrtTOF0<sqrtTOF_up; sqrtTOF0+=0.5) {
  double mean1 = 0., sd1 = 0.,
         mean2 = 0., sd2 = 0.;
  std::vector<double> temp1, temp2;
  while(itr1->first<sqrtTOF0+0.5 && itr1!=sqrtTOFR1.end()) {
    mean1 += itr1->second;
    mean2 += itr2->second;
    temp1.push_back(itr1->second);
    temp2.push_back(itr2->second);
    itr1++;
    itr2++;
  }
  mean1 = mean1/temp1.size();
  mean2 = mean2/temp2.size();
  for(unsigned int j=0; j<temp1.size(); j++) {
    sd1 += (temp1[j]-mean1)*(temp1[j]-mean1);
    sd2 += (temp2[j]-mean2)*(temp2[j]-mean2);
  }
  sd1 = sqrt(sd1/temp1.size());
  sd2 = sqrt(sd2/temp2.size());

  ofs_MeanSD[1]<<std::setw(20)<<std::left<<gc_sqrtTOF<<std::setw(20)
                <<std::left<<temp1.size()<<std::setw(20)<<std::left
                <<std::setprecision(6)<<sqrtTOF0
                <<std::setw(20)<<std::left<<std::setprecision(6)<<mean1
                <<std::setw(20)<<std::left<<std::setprecision(6)<<sd1
                <<std::endl;
  ofs_MeanSD[4]<<std::setw(20)<<std::left<<gc_sqrtTOF<<std::setw(20)
```

```
                <<std::left<<temp2.size()<<std::setw(20)<<std::left
                <<std::setprecision(6)<<sqrtTOF0
                <<std::setw(20)<<std::left<<std::setprecision(6)<<mean2
                <<std::setw(20)<<std::left<<std::setprecision(6)<<sd2
                <<std::endl;
  gc_sqrtTOF++;
}

std::map<double,double> EBER1, EBER2;
for(unsigned int i=0; i<TOFR1.size(); i++) {
  double ebe = 11.09-(9.2329e-13/(((i+TOF_low)*1.e-9-9.8341e-9)
                    *((i+TOF_low)*1.e-9-9.8341e-9))-0.39569);
  EBER1[ebe] = TOFR1[i];
  EBER2[ebe] = TOFR2[i];
}
std::map<double,double>::const_iterator itr11 = EBER1.begin(),
                                        itr22 = EBER2.begin();
unsigned int gc_EBE = 0;

for(double EBE0=EBE_low; EBE0<EBE_up; EBE0+=0.15) {
  double mean1 = 0., sd1 = 0.,
         mean2 = 0., sd2 = 0.;
  std::vector<double> temp1, temp2;
  while(itr11->first<EBE0+0.15 && itr11!=EBER1.end()) {
    mean1 += itr11->second;
    mean2 += itr22->second;
    temp1.push_back(itr11->second);
    temp2.push_back(itr22->second);
    itr11++;
    itr22++;
  }
  mean1 = mean1/temp1.size();
  mean2 = mean2/temp2.size();
  for(unsigned int j=0; j<temp1.size(); j++) {
    sd1 += (temp1[j]-mean1)*(temp1[j]-mean1);
    sd2 += (temp2[j]-mean2)*(temp2[j]-mean2);
  }
  sd1 = sqrt(sd1/temp1.size());
  sd2 = sqrt(sd2/temp2.size());

  ofs_MeanSD[2]<<std::setw(20)<<std::left<<gc_EBE<<std::setw(20)
               <<std::left<<temp1.size()<<std::setw(20)<<std::left
               <<std::setprecision(6)<<EBE0
               <<std::setw(20)<<std::left<<std::setprecision(6)<<mean1
               <<std::setw(20)<<std::left<<std::setprecision(6)<<sd1
```

```
                <<std::endl;
  ofs_MeanSD[5]<<std::setw(20)<<std::left<<gc_EBE<<std::setw(20)
                <<std::left<<temp2.size()<<std::setw(20)<<std::left
                <<std::setprecision(6)<<EBE0
                <<std::setw(20)<<std::left<<std::setprecision(6)<<mean2
                <<std::setw(20)<<std::left<<std::setprecision(6)<<sd2
                <<std::endl;
  gc_EBE++;
}

// Clearup
std::cout<<"\nFile Closing......\n"<<std::endl;

for(int k=0; k<10; k++) {
  ifs_P[k].close();
  ifs_E[k].close();
  ifs_D[k].close();
}
ofs1.close();
ofs2.close();
ofs3.close();
for(unsigned int i=0; i<6; i++) ofs_MeanSD[i].close();

std::cout<<"Clearup......\n"<<std::endl;

for(int i=3999; i>=0; i--) {
  delete [] sweepAveBG[i];
  delete [] sweepAveD[i];
}
delete [] sweepAveBG;
delete [] sweepAveD;

for(int k=9; k>=0; k--) {
  for(int i=3999; i>=0; i--) {
    delete [] BG[k][i];
    delete [] D[k][i];
  }
}
for(int k=9; k>=0; k--) {
  delete [] BG[k];
  delete [] D[k];
}
delete [] BG;
delete [] D;
```

```
 std::cout<<"Program Finished Successfully!\n"<<std::end;
}
```

B.2 Data File Creation for SAS Input Procedures in Appendices A.3 and A.4

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <iomanip>
#include <string>
#include <vector>
#include <map>
#include <cmath>

double probAtTOF(int,double,double);

int main()
{
 std::ifstream ifs_P[10], ifs_E[10], ifs_D[10];
 std::string   inFileName;

 // Input P E D
 std::cout<<"\nReading 10 P, E and D files......"<<std::endl;
 for(unsigned int k=0; k<10; k++) {
   std::ostringstream oss_P, oss_E, oss_D;
   oss_P<<"P"<<k+1<<".dat";
   oss_E<<"E"<<k+1<<".dat";
   oss_D<<"D"<<k+1<<".dat";
   ifs_P[k].open(oss_P.str().c_str(),std::ios::in);
   ifs_E[k].open(oss_E.str().c_str(),std::ios::in);
   ifs_D[k].open(oss_D.str().c_str(),std::ios::in);
 }

 std::cout<<"\nProgram Running......\n"<<std::endl;

 // Initialization
 double** sweepAveBG = new double*[4000];
 double** sweepAveD  = new double*[4000];
 for(unsigned int i=0; i<4000; i++) {
   sweepAveBG[i] = new double[151];
   sweepAveD[i]  = new double[151];
 }
 for(unsigned int i=0; i<4000; i++) {
```

```
  for(int j=0; j<151; j++) {
    sweepAveBG[i][j] = 0.;
    sweepAveD[i][j]  = 0.;
  }
}
double*** BG = new double**[10];
double*** D  = new double**[10];
for(unsigned int k=0; k<10; k++) {
  BG[k] = new double*[4000];
  D[k]  = new double*[4000];
}
for(unsigned int k=0; k<10; k++) {
  for(unsigned int i=0; i<4000; i++) {
    BG[k][i] = new double[151];
    D[k][i]  = new double[151];
  }
}
for(unsigned int k=0; k<10; k++) {
  for(unsigned int i=0; i<4000; i++) {
    for(int j=0; j<151; j++) {
      BG[k][i][j] = 0.;
      D[k][i][j]  = 0.;
    }
  }
}

// Calculate sweepAveBG, sweepAveD, sweepAveR, BG and D
int p, e, d;
for(unsigned int k=0; k<10; k++) {
  for(unsigned int i=0; i<4000; i++) {
    for(unsigned int j=0; j<151; j++) {
      ifs_P[k]>>p;
      ifs_E[k]>>e;
      ifs_D[k]>>d;
      sweepAveBG[i][j] += (p+e)*0.1;
      sweepAveD[i][j]  += d*0.1;

      BG[k][i][j] = (p+e);
      D[k][i][j]  = d;
    }
  }
}

const double K = 5.02;
```

```
double sweepAveR[4000][151];
for(unsigned int i=0; i<4000; i++) {
for(unsigned int j=0; j<151; j++) {
    sweepAveR[i][j] = sweepAveD[i][j]-K*sweepAveBG[i][j];
  }
}

// Curve along Delay_WTSumupTOF
std::ostringstream oss;
std::ofstream ofs;
oss<<"curveAlongj_Delay0-149_WTSumTOF.dat";
ofs.open(oss.str().c_str(),std::ios::out);
for(unsigned int j=0; j<=39; j++) {
  double sum_j = 0.;
  for(unsigned int i=0; i<=4000; i++) {
    sum_j += sweepAveR[i][j]*probAtTOF(i,28.9134,2.2450);
  }
  ofs<<std::setw(12)<<std::left<<j<<std::setw(15)<<std::left
     <<std::setprecision(4)<<std::fixed<<sum_j<<std::endl;
}
for(unsigned int j=40; j<=149; j++) {
  double sum_j = 0.;
  for(unsigned int i=0; i<=4000; i++) {
    sum_j += sweepAveR[i][j]*probAtTOF(i,28.9164,1.8966);
  }
  ofs<<std::setw(12)<<std::left<<j<<std::setw(15)<<std::left
     <<std::setprecision(4)<<std::fixed<<sum_j<<std::endl;
}

// Clearup
std::cout<<"\nFile Closing......\n"<<std::endl;

for(int k=0; k<10; k++) {
  ifs_P[k].close();
  ifs_E[k].close();
  ifs_D[k].close();
}
ofs.close();

std::cout<<"Clearup......\n"<<std::endl;

for(int i=3999; i>=0; i--) {
  delete [] sweepAveBG[i];
  delete [] sweepAveD[i];
}
```

```
 delete [] sweepAveBG;
 delete [] sweepAveD;


 for(int k=9; k>=0; k--) {
   for(int i=3999; i>=0; i--) {
     delete [] BG[k][i];
     delete [] D[k][i];
   }
 }
 for(int k=9; k>=0; k--) {
   delete [] BG[k];
   delete [] D[k];
 }
 delete [] BG;
 delete [] D;

 std::cout<<"Program Finished Successfully!\n"<<std::endl;
}

double probAtTOF(int TOF, double u, double s)
{
 const double PI = 3.1415926;
 double prob = 0.,
        lower = 0.5*(sqrt(TOF)+sqrt(TOF-1)),
        upper = 0.5*(sqrt(TOF+1)+sqrt(TOF)),
        dx = (upper-lower)/10000.;

 for(double x=lower; x<=upper; x+=dx) {
     prob += (1./sqrt(2.*PI*s*s))*exp(-0.5*(x-u)*(x-u)/(s*s))*dx;
 }

 return prob;
}
```

Appendix C

The Mathematica codes

C.1 Data Visualization in Section 4.1.1.4

```
Directory[]

SetDirectory["/Users/zhuofeihou/STAT/MSP_Meeting/Meeting_100115/"]

FileNames[]

RData = ReadList["R_iPrime-jPrime_k=5.02.txt", Real, RecordLists -> True]

g1 = ListPlot3D[RData, PlotRange -> All, Mesh -> {50, 10},
     MeshStyle -> Gray, AxesLabel -> {"TOF (i')", "Delay (j')", "R"},
     PlotLabel -> "3D Plot of R with k=5.02, z=[-0.15, 10]",
     Ticks -> {{0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50},
               {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
               {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}}]

g2 = ListPlot3D[RData, PlotRange -> {-0.15, 0.4}, Mesh -> {50, 10},
     MeshStyle -> Gray, AxesLabel -> {"TOF (i')", "Delay (j')", "R"},
     PlotLabel -> "3D Plot of R with k=5.02, z=[-0.15, 0.4]",
     Ticks -> {{0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50},
               {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10},
               {0, 0.1, 0.2, 0.3, 0.4, 0.5}}]

Show[GraphicsArray[{{g1, g1}, {g2, g2}}]]

g3 = ListContourPlot[RData, PlotRange -> All, Frame -> True,
     PlotLabel -> "2D Contour Plot of R with k=5.02, z=[-0.15, 10]"]

g4 = ListContourPlot[RData, PlotRange -> {-0.15, 0.4}, Frame -> True,
     PlotLabel -> "2D Contour Plot of R with k=5.02, z=[-0.15, 0.4]"]

g5 = Show[GraphicsArray[{g3, g4}]]
```

```
Export["2D_Contour_R_iPrime-jPrime_k=5.02.png",g5,"PNG",ImageSize->1000]

g6 = ListDensityPlot[RData, PlotRange -> All,
     PlotLabel -> "2D Density Plot of R with k=5.02, z=[-0.15, 10]"]

g7 = ListDensityPlot[RData, PlotRange -> {-0.15, 0.4},
     PlotLabel -> "2D Density Plot of R with k=5.02, z=[-0.15, 0.4]"]

g8 = Show[GraphicsArray[{g6, g7}]]

Export["2D_Density_R_iPrime-jPrime_k=5.02.png",g8,"PNG",ImageSize->1000]
```