Summer 6-20-2019

TRIPLET POPULATION DYNAMICS AND EXCITED STATE RELAXATION IN CHALCOGENOPHENE POLYMERS

Benjamin D. Datko
University of New Mexico - Main Campus

Benjamin D. Datko

Candidate

Chemistry and Chemical Biology

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. John K. Grey , Chairperson

Dr. Jeff Rack

Dr. Terefe G. Habteyes

Dr. David Dunlap

TRIPLET POPULATION DYNAMICS AND EXCITED STATE RELAXATION IN CHALCOGENOPHENE POLYMERS

by

BENJAMIN D. DATKO

B.S., Chemistry, Temple University 2013

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Chemistry

The University of New Mexico
Albuquerque, New Mexico

July, 2019

DEDICATION

To my best friend, and the love of my life.

ACKNOWLEDGMENTS

I first want to thank and give my appreciation to my mentor and advisor Dr. John K. Grey. His support, patience, and insight allowed for completion of the work within this dissertation. The research and my Ph.D. experience benefited from Dr. Grey's willingness for supporting both theoretical and experimental endeavors.

I thank both present and past group members of the Grey research group, including Alan Thomas, who first mentored me on the operations of the lab, Dave Walwark and Griffin Canning, who I thank for their discussion and insight into the research challenges I encountered. I thank all the undergraduate research students who I had the pleasure to mentor and who helped with the work, including Dana Portlock, Maggie Berrens, Ryan Grimm, and Heather McClurg.

I want to thank my friend and co-author, Dr. Maksim Livshits, especially for his effort and discussion. I want to thank my family for their continuous encouragement and support. Lastly, I thank my best friend and fiancée; I could not imagine reaching this accomplishment without her.

TRIPLET POPULATION DYNAMICS AND EXCITED STATE RELAXATION IN CHALCOGENOPHENE POLYMERS

by

BENJAMIN D. DATKO

B.S., Chemistry, Temple University 2013

Ph.D., Chemistry, University of New Mexico

ABSTRACT

The conventional understanding of intersystem crossing in multichromophoric conjugated polymers is usually depicted via a pure electronic model, neglecting contributions of vibrations or conformational order. Obtaining accurate structure-function correlations on spin-conversion processes involving photogenerated singlet excitons to triplet excitons and the excited state dynamics requires sensitivity to the subtle conformational ordering within conjugated polymers. This dissertation seeks to understand the kinetics of multi-exciton singlet-triplet interactions and the excited state relaxation of chalcogen containing (S, Se) conjugated polymers. Utilizing single molecule modulation spectroscopy allows determination of triplet formation of individual conjugated polymer chains and aggregates. This technique resolves triplet-induced fluorescence quenching to ascertain the dynamics of the triplet population. In parallel, we have utilized the solutions to the probabilistic master equation describing the time-dependent kinetics of triplet formation. Finally, investigating the excited state relaxation of strongly aggregating, non-emissive poly(3-decylthieneylenevinylene) (P3DTV) and its heavy atom analog poly(3-decyl-seleneylenevinylene) (P3DSV) we demonstrate an alternative hypothesis for the observed ultrafast excited state dynamics.

DECLARATION

This dissertation is written in the form of seven chapters, five of which are papers (Chapters 3 through 7) which are published. Chapter 1 provides an introduction to the body of the work. Chapter 2 gives a detailed description of the instrumentation and experimental methods. The citations of the published work are listed below:

Chapter 3: Phys. Chem. Chem. Phys. 2017, 19 (41), 28239–28248.

Chapter 4: J. Phys. Chem. C 2018, 122 (17), 9718–9725.

Chapter 5: Sci. Rep. 2019, 9 (1), 817.

Chapter 6: Phys. Chem. Chem. Phys. 2018, 20 (34), 22159–22167

Chapter 7: J. Phys. Chem. Lett. 2019, 10 (6), 1259–1263.

All publications include my mentor and supervisor, Professor Dr. John K. Grey as the corresponding author. Chapters 3 through 4 includes professor Martin Heeney as a co-author, whose lab synthesized and provided the chemical material. Chapter 6 and Chapter 7 includes authorship from professor Dr. Yang Qin whose lab synthesized all chemical materials and Dr. Maksim Livshits and professor Dr. Jeff Rack assisting in experimentation.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

I_E the intensity of the excitation.

h Planck's constant.

k_{ISC}^{R} reverse intersystem crossing.

k_r the rate constant for radiative deactivation.

k_{IC} the rate constant for internal conversion.

k_{ISC} constants for intersystem crossing.

k_{exc} the rate constant for excitation.

AOM acoustic optical modulator.

APD avalanche photodiode.

CCD charge coupled device.

CCDF complementary cumulative distribution functions.

DCB ortho-dichlorobenzene.

DiI 1,1-didocyl-3,3,3,3-tetramethyl-indocarbocyanine perchlorate.

dTV alkyl substituted trans-thienylene-vinylene dimer.

EMCCD electron-multiplying charge coupled device.

F8BT (poly(9,9′-dioctylfluorene cobenzothiadiazole)).

I(t) time-dependent emission intensity.

KDa kilodalton.

MCA multichannel analyzer.

MEH-PPV poly[2-methoxy, 5-(29-ethyl-hexyloxy)-p-phenylenevinylene].

MLE maximum likelihood estimate.

P3DTV poly(3-decyl-seleneylenevinylene).

P3DTV poly(3-decylthieneyl-enevinylene).

P3HS poly(3-hexylselenophene).

P3HT poly(3-hexylthiophene).

P3OT poly 3-octyl-thiophene.

PPV-PPyV poly(p-phenylene vinylene)-poly(p-pyridylene vinylene).

RHS right hand side.

rR resonance raman.

S sulfur.

$\mathbf{S}_0$ ground singlet state.

$\mathbf{S}_1$ first excited singlet state.

Se selenium.

$\mathbf{T}_1$ lowest excited triplet state.

TCSPC time-correlated single photon counting.

1.0 Introduction

1.1 Motivation

Spin forbidden triplet excitons (S=1) in semiconducting, multichromophore conjugated polymers have longer lifetimes then spin allowed singlet excitons (S=0). The longer lifetime results from the transition of the triplet excited state to the singlet ground state being strictly forbidden by spin selection rules. The large disparity of lifetimes between excited states of different spins in conjugated organic semiconductors can be a benefit[1–6] as well as a detriment[7–9] for possible applications. For example, longer lifetimes can hurt the efficiency of organic light-emitting diodes[10–12] due to an increased likelihood to quench emissive singlet excitons. For light energy harvesting, conjugated polymers used in bulk heterojunction systems (i.e. blends of polymers and fullerene acceptors) are dependent on photocarriers reaching a donor-acceptor interface for charge separation. These devices can significantly benefit from the longer triplet lifetime since there would be an increased chance for the triplet to diffuse to a donor-acceptor interface within the heterostructure layer.[4,13,14] Understanding how to manage the complex kinetics of singlet and triplet formation, energy migration, and electronic communication in conjugated polymers has proven difficult, despite of active multidisciplinary research.

Investigating the formation of triplet excitons and interactions of triplets with other excitonic species is made difficult by the strong correlation between morphology and photophysics in conjugated polymers. The intrinsic nature of the conjugated polymer, being an assembly of covalently linked quasi-chromophores and having varying degrees of polydispersity contributes to the complex correlation between morphology and photophysics. The

chromophores responsible for the electronic absorption are small subunits existing from 2 to 12 monomer units.[15,16] These subunits will then have overlapping electronic transitions manifesting as an inhomogeneously broadened absorption spectrum. Depending on the proximity and relative orientation of the absorbing subunits, the excitation (exciton) can efficiently energy transfer between segments both along and across polymer chains.[16–23] Not only can the chromophores undergo efficient electronic energy transfer, but each excitation on a chromophore has a chance of intersystem crossing from a singlet to a triplet exciton. There can be a significant triplet population even for conjugated polymers with a low quantum yield of intersystem crossing under continuous illumination.[10,24] These interactions are difficult to determine using bulk spectroscopy because the measured response is an average over all the possible conformation the polymer may adopt.

Single molecule spectroscopy has made unraveling the structure-function relation of multichromophoric interactions in conjugated polymers possible.[25–28] Single molecule spectroscopy measures the emission of individual, isolated conjugated polymers in an ultradilute inert polymer matrix. Sampling each polymer chain or nanoaggregate allows for the unraveling of the heterogenous spectral and kinetic response. Despite single molecule spectroscopy advantages for investigating conjugated polymers, heterogeneous structure and photophysics are still observed even at the single polymer chain level.

A foundational understanding of the photophysics and morphology of conjugated polymers was built on investigating both single polymer chains and nanoaggregates of poly[2-methoxy, 5-(29-ethyl-hexyloxy)-p-phenylenevinylene] (MEH-PPV).[29–31] Even though investigators have studied MEH-PPV in varying experimental conditions,[32–35] there is still a lack of understanding of singlet and triplet interactions at the single polymer chain level with more significant rates of triplet formation. The work within this dissertation seeks to answer “How do multple exciton singlet-triplet interact on spatially confined polymer chains and aggregates?” and “How are excited state relaxation and triplet formation limited in strongly aggregating conjugated polymers?”. Obtaining accurate structure-function

correlations on the spin-conversion of singlet excitons to triplet excitons and the excited state dynamics is key and requires sensitivity to the subtle conformational ordering within the conjugated polymer. A diverse set of experimental approaches and new kinetic modeling of multi-exciton interactions are used to investigate chalogen containing conjugated polymers.

1.2 Determining the role of conformation and intersystem crossing in single molecule photophysics of multichromophoric conjugated polymers

Photophysics of both single and multichromophoric systems have been investigated by measuring the time-dependent emission intensity, I(t), at the single molecule level. The time-dependent emission intensity trace of the emitted photons are typically measured from isolated, diffraction limited spots using continuous wave excitation. The first measured conjugated polymers was a poly(p-phenylene vinylene)-poly(p-pyridylene vinylene) (PPV-PPyV) copolymer measured at ultra-dilute concentrations in an inert polymer host of PMMA.[29] The I(t) of all molecules of PPV-PPyV showed discrete intensity jumps between few "on" intensity levels and "off" intensity near the background level on time scales from milliseconds to seconds. The discrete jumps are widely known as "blinking." Discrete jumps between "on" and "off" in multichromophoric materials are unexpected since conceptually conjugated polymers could be thought of as non-interacting chromophores resulting in a continuous exponentially decaying intensity trace.[36] The observation of transition between "on" and "off" were accompanied by irreversible photobleaching. Photobleaching is the irreversible photooxidation of the polymer chain resulting in "permanent darkness" of the intensity transient most likely due to the lack of absorption after the photooxidation. Two conclusions were made from these first observations, (i) the singlet exciton is effectively "funneled" to lower energy subunits on the polymer chain from which radiative emission can occur and (ii) at the funnel an efficient quencher of the singlet exciton can be "installed" sequentially quenching further emission. Therefore, discrete jumps between

emitting and non-emitting "dark" state is a consequence of interactions between neighboring chromophores. The reversible quenching of emission observed in I(t) is a direct consequence of the polymer chain morphology.

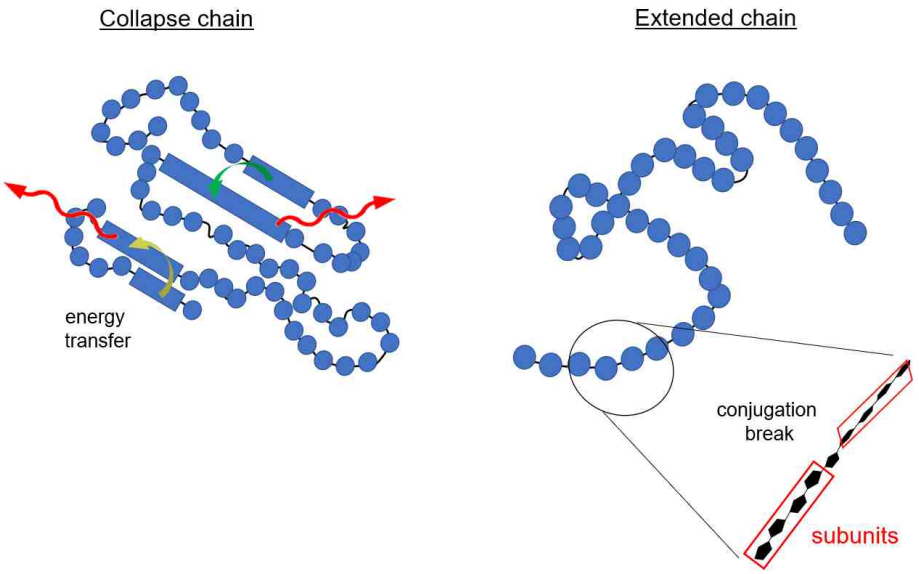

Figure 1.1: Cartoon depictions of two conjugated polymer conformation. Left is of a collapsed chain showing energy transfer to extend conjugated subunits, drawing as rectangles. Right is of an extended chain emphasizing the separation of the subunits and highlighting conjugation breaking via torsional rotation.

In the experiment mentioned above, the conformation of the polymer adopts a collapsed conformation aligning chain segments parallel to each other. This conformation is known as a defect cylinder, a depiction of the conformation is shown in Figure 1.1a. Combine simulation and experimental measurements of the anisotropy of the transition dipole moment of isolated conjugated polymer chains gives evidence of the morphology adopting an elongated ellipsoid like conformation.[37] The proximity of the aligned polymer chains allow for efficient energy transfer of singlet excitons most likely through a Forster type mechanism. Energy transfer is highly dependent on the conformation of the polymer, which can be somewhat controlled by processing conditions of the single molecule sample. Huser et al.[38] demonstrated this dependence by spin casting poly[2-methoxy, 5-(29-ethyl-hexyloxy)-p-phenylenevinylene] (MEH-PPV) in toluene verses chloroform. Casting in toluene results

in the stepwise switching between emissive and "dark" levels. Casting the polymer in chloroform results in a continuous exponentially decaying I(t) resulting in an extended conformation known as a defect coil, depiction shown in Figure 1.1b. Conformation of the conjugated polymer can limit the observed photophysics at the single molecule level but does not give further evidence to the identity of the quencher or the nature of the "dark" state.

Vanden Bout et al.[29] showed the rate of transitioning between the emissive "on" states to the "off" state is dependent on the rate of excitation in single polymer chains of PPV-PPyV; thus the quencher is photochemically formed.[16,36] With careful control over the presence of oxygen, Yu et al.[16] demonstrated that oxygen dramatically increases singlet quenching in MEH-PPV. The dependence of oxygen and excitation intensity gives significant evidence that the quenching species is an oxidative chemically defect formed through interactions between triplet excitons and oxygen. Molecular oxygen has a triplet ground state configuration allowing energy transfer between the triplet exciton on the polymer chain producing singlet oxygen. Singlet oxygen can then readily reacts with C=C present in many conjugated polymers. Oxygen interaction can be both reversible and irreversible due to the observation of transitioning back from the "off" state to the "on" state in I(t) traces.

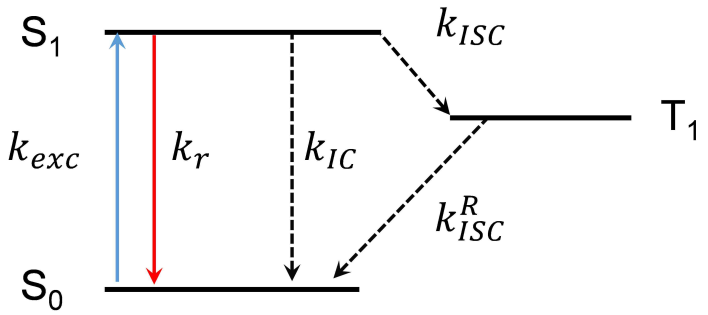

Figure 1.2: Jablonski diagram depicting the rate constants of various electronic process following absorption of a photon. **Key** : S_0 is the singlet ground state, S_1 is the first singlet excited state, T_1 the first excited triplet state, k_{exc} is the rate constant for excitation, k_r the rate constant for radiative deactivation of $S_1 \rightarrow S_0$, k_{IC} is the rate constant for internal conversion, k_{ISC} and k_{ISC}^R are the rate constants for intersystem crossing and reverse intersystem crossing, respectfully.

The kinetics between the chromophore and molecular oxygen is illustrated by a three-state Jablonski diagram of the ground state singlet, the first excited state singlet, and the first excited state triplet. If molecular oxygen is near the chromophore while the excitation is in the triplet state, T_1, energy transfer can then take place. The rate-limiting step for the interaction of the chromophore and oxygen is either the lifetime of the triplet state, the inverse of the rate constant for reverse intersystem crossing k_{ISC}^{R}, or the rate of diffusion of oxygen through the solid host. Analogous interactions between the rate of intersystem crossing and oxygen are seen in a single chromophore system, such as DiI (1,1-didocyl-3,3,3,3-tetramethyl-indocarbocyanine perchlorate).[39,40] With the removal of oxygen, DiI has a large increase in the photostability and allows for the direct measurement of the intersystem crossing.[40] The polymer host can also limit the diffusion of oxygen thus affecting the measured rate of photobleaching and intersystem crossing.[41]

Beyond the observed effect of oxygen and intersystem crossing there are significant differences between the kinetics of single and multichromophoric systems. In single chromophores without the presence of oxygen, I(t) traces show "bursts" of counts before entering long "off" times.[40] The burst is due to cycling from S_0 and S_1 right before intersystem crossing. While in the triplet state the chromophore can no longer absorb incoming photons and appears dark with a sudden drop in counts for long periods, as long as seconds. Comparatively to conjugated polymers, the observed one-step bleaching in I(t) limited by oxygen interactions but only when the conformation allows for efficient energy transfer to a few emitting sites. Once oxygen is significantly removed the I(t) traces show a steady, continuous stream of photons with few photobleaching events where single spots under continuous irradiation can last up to minutes without photobleaching even if the polymer is in a collapsed conformation.[16] Yet singlet and triplet fluctuations can still be present but not resolvable due to limiting time resolution when measuring I(t) traces. Investigating singlet and triplet interactions for conjugated polymers has proven challenging due variation in the sample preparation of single molecule experiments.[24] Further developments in experimen-

tation and insights of the kinetics are necessary beyond interrogating I(t) traces of single polymer chains.

1.3 Measuring the kinetics of exciton-exciton interactions of single conjugated polymers chains

Deducing exciton-exciton interactions such as singlet-triplet and triplet-triplet from I(t) alone in conjugated polymers is limited by the time resolution necessary to observe the interactions and averaging the fluctuations induced varying populations of excitonic species away. Yip et al.[36] demonstrate how "on" / "off" transitions in I(t) traces can be masked due to averaging over many chromophores using both Monte Carlo simulations and observing I(t) from dye molecules (180 molecules) in a polymer bead. Beyond just washing away fluctuations from exciton-exciton interactions, the dynamics proceed on much faster time scales typically used to measure I(t). The dwell time in the measurement is limited to 0.1 ms while triplet formation can be on the order of nanoseconds to microseconds. Understanding of the complex kinetics of varying populations of exciton species in conjugated polymers requires both the time resolution and sensitivity towards the polymer chain conformation.

Developments from the Barbara group toward measuring and understanding of multi-exciton interactions was first done using fluorescence correlation spectroscopy,[42] then further development to fluorescence modulation spectroscopy of single polymer chains and aggregates.,[25,26,30] the later will be the focus. Detection of both singlet-triplet and triplet-triplet interactions is accomplished through synchronous averaging many repeat cycles with a square wave pulse inducing quenching of singlet emission from a build-up of triplet populations. Three aspects of the experimental data are necessary to emphasize: (i) at time zero the polymer chain contains zero triplets i.e., triplet free polymer (ii) as the polymer is continually irradiated triplet populations build up inducing quenching of emission observed (iii) dynamics lead to an eventual steady state of the emission and triplet populations. The experimental data produces exponential-like decay, which a kinetic model can then be im-

posed to extract out photophysical rate constants such as the rate of singlet-triplet quenching and triplet-triplet annihilation.

The governing kinetics were assumed to follow a two-state reversible scheme where the conjugated polymer spends the majority of the time occupying either the ground singlet state or the first triplet excited state. The two-state scheme assumes a separation of fast and slow processes occurring on the single polymer chain. Separation of such processes into fast are singlet emission, singlet-singlet annihilation, internal conversion, intersystem crossing, and triplet-triplet annihilation. Slow processes include reverse intersystem crossing and the formation of triplets through the singlet state under low excitation ($k_{exc} <<< k_{ISC}$). Typically, bulk parameters for conjugated polymers are consistent with the distinction of fast and slow separation. Assuming "fast" or efficient triplet-triplet annihilation limits the number of triplet excitons present. Therefore, the resulting kinetics observed are from the reversible transition between the ground and first triplet excited state, shown below.

$$\mathrm{S}_0 \underset{k_r}{\overset{k_f}{\rightleftharpoons}} \mathrm{T}_1 \tag{1.1}$$

where the forward and backward first-order rate constants are given by

$$k_f = k_{exc} k_{ISC} \tau_{fl} \tag{1.2a}$$

$$k_b = k_{ISC}^R + k_{exc} k_{ISC} \tau'_{fl} \tag{1.2b}$$

where k_{ISC}^R is the reverse intersystem crossing rate constant, k_{ISC} is the intersystem rate constant, τ_{fl} is the fluorescence lifetime without a triplet, τ'_{fl} is the fluorescence lifetime with a triplet present, and k_{exc} is the rate of excitation given by equation 1.3.

$$k_{exc} = \frac{I_E \sigma}{h\nu} \tag{1.3}$$

Where I_E is the intensity of the excitation, σ is the absorption cross-section, h is Planck's

constant, and ν is the frequency of the excitation light.

The success of the technique to both identifying singlet-triplet and triplet-triplet interactions and formulating a kinetic scheme of conjugated polymers cannot be understated. But the assumption of either efficient or infinite triplet-triplet annihilation limits the application of the model to conjugated polymers with larger rates of triplet formation. Palacios et al.[43] showed qualitatively similar behavior of quenching in poly 3-octyl-thiophene (P3OT) with the assumption of efficient triplet-triplet annihilation leading to one triplet responsible for the observed kinetics. The quantum yield of intersystem crossing is larger in thiophene containing conjugated polymers compared to MEH-PPV, the nominal quantum yield of the nominal quantuam yield of ~30%[44–46] while the quantum yield of intersystem crossing for MEH-PPV is ~1.25%.[47] Therefore, an assumption of infinite triplet-triplet annihilation may not be valid for chalcogen containing conjugated polymers. Thomas et al.[48] showed the inability of determining reliable rates of both singlet-triplet and triplet-triplet interactions from the two-state model. To determine and expand our understanding of triplet interactions on single polymer chains and aggregates requires the use of the full multistate photodynamical model without an assumption of limiting triplet population.

1.4 Existence of multiple triplets on single polymer chains: leaving the infinite triplet-triplet annihilation regim

Moving beyond an assumption, efficient triplet-triplet annihilation i.e., infinite triplet-triplet annihilation requires incorporating higher order states of more than one triplet. Yu et al.[42] established a kinetic population state model to systematically describing the interactions for conjugated polymers accounting localized multichromophoric behavior. The state model describes the number of singlets and triplets current present on the conjugated polymer. Each chromophore of the polymer can occupy either the singlet ground state, the first excited singlet state, and the first excited triplet state. The kinetics on the polymer chain is assumed to be homogenous such that only the total number of singlets and triplets are

considered. This implies the excitons are not limited by diffusion. Sequentially counting the number of singlet excitons and triplet excitons present on a conjugated polymer chain creates a state space of two dimensions where one dimension is the number of singlets and the other being the number of triplets. The "coordinate system" for the state space is given by [#S_1 , #T_1] where #S_1 is the number of excited singlet excitons present and #T_1 is the number of excited triplet state excitons. Transitions between the states are governed by photophysical kinetics, some of which are shown in the Jablonski diagram Figure 1.2, as well as bimolecular interactions when the state has a sufficient number of excitonic species. Varying interactions between two chromophores are shown in Figure 1.3.

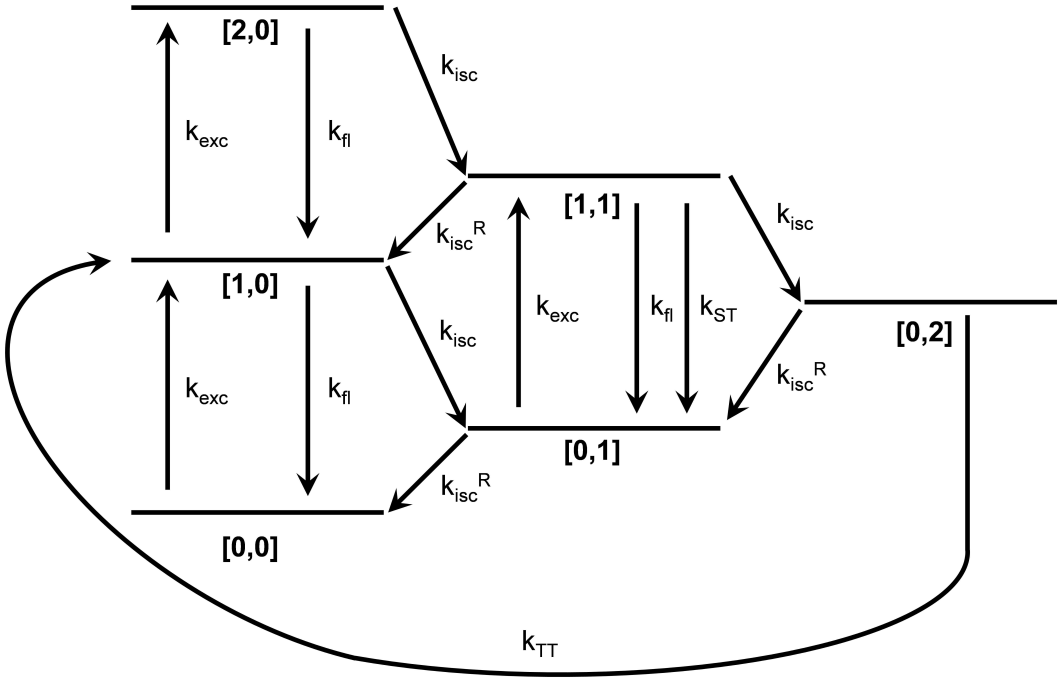

Figure 1.3: Kinetic scheme considering the interactions between two chromophores. **Key** : The rate constant for singlet-triplet quenching is k_{ST} and the rate constant for triplet-triplet annihilation is k_{TT}. All other rate constants are identified in Figure 1.2

Even with just two chromophores, the number of states and interactions are complex. A challenge for describing interactions within the conjugated polymer is a large number of states needed. Examining the two chromophores case one can count the number of ways two chromophores can occupy the ground singlet state, the first excited singlet state, and

the first excited triplet state. The possible arrangement of the number of singlet and triplet excitons for two chromophores are shown in Figure 1.4 along with representation using [#S_1, #T_1]. When increasing the number of chromophores, the number of states needed to arrange the occupancy (ground, singlet, and triplet) is an ever growing and diverging series of numbers. One can write the set of six differential equations for the probability of occupying the state shown in Figure 1.3. Implementing a solution to the set of first order differential equation shown in Figure 1.3 can be solved by any numerical integration library to obtain the time-dependent solutions. However, the assumption of just two chromophores may not be widely applicable to most conjugated polymers.

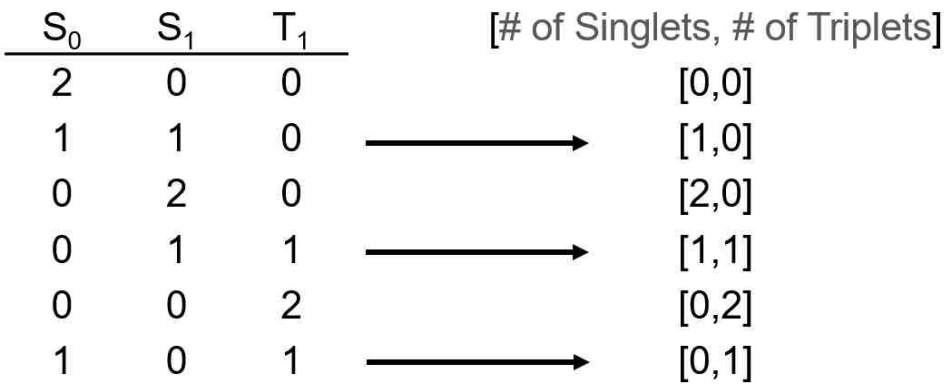

Figure 1.4: Shown above is the total combination of states needed to describe the interactions between two chromophores and the representation of each state in terms of counting singlet and triplet excitons.

A more systematic approach is needed to expand beyond two chromophores. It's not apparent from the presentation above, but the kinetics described is similar to a stochastic kinetics formalism for a governing chemical master equation. A master equation is a discrete difference equation describing the time-dependence of the conditional probability of having N number of chemical species at a given time t.[49] One can derive a master equation for any given set of elementary reactions, but the difficulty is in implementing a numerical solution.[50] Simple first order elementary reactions and a few cases for bimolecular reactions have analytical solutions.[51–53] While more complex reactions may have analytical solutions for the steady state while the time-dependent solutions require numerical integration.[54]

Writing a chemical master equation for the above sets of states shown in Figure 1.3, requires two stochastic difference equations for both the singlet and triplet populations. Gruber et al.[55] implemented a numerical solution of the two chemical master equations for singlet and triplet interactions for modeling singlet-triplet annihilation in light harvesting complex II complexes but showed the time-dependence of the singlet excitons can be replaced with the steady-state values. Considering both the timescale at which the dynamics are measured in the fluorescence intensity modulation and when using of low rates of excitation, we can assume singlet excitons are at steady-state with a low occupation, ignoring higher order terms of more than one singlet exciton. The separation of fast and slow is similar to what was proposed originally but we exclude triplet-triplet annihilation from the separation. The kinetics left to consider is the formation and decay of the triplet excitons. Barzykin and Tachiya[56] applied a stochastic chemical master equation, similar to the kinetic population state model of Yu et al.,[42] detailing the triplet dynamics of MEH-PPV and F8BT (poly(9,9′- dioctylfluorene cobenzothiadiazole)) considering just three chemical reactions: the gain of one triplet, the loss of one triplet, and the loss of two triplets through assumed pairwise annihilation. The numerical results demonstrate at low excitation for both MEH-PPV and F8BT having more than one triplet was rare in cases with a slow triplet-triplet annihilation. In comparison, at high excitation the likelihood of multiple triplets increases, assuming finite triplet-triplet annhihilation. Assuming the presence of one triplet exciton when considering conjugated polymers with efficient rates of intersystem crossing is invalid and detailing the kinetics of populations of triplet excitons is then necessary.

1.5 Looking beyond the pure electronic view of spin-orbit coupling

An increase in the conversion of singlet excitons into triplet excitons is expected when substituting from sulfur to selenium in the backbone of the conjugated polymer. This expectation is derived from a perturbative treatment of intersystem crossing where the magnitude of the spin-orbit mixing and the energetic splitting between the first excited singlet and first

excited triplet determines the intersystem crossing rate constant k_{ISC}. This conventional picture is a pure electronic depiction which does not explicitly consider vibrations or conformational order, which can alter the intersystem rate. Anomalous disappearance of triplet formation in thin films of regio-regular poly(3-hexylthiophene) (P3HT) compared to regio-random P3HT gives evidence of the complex structure-function of intersystem crossing in P3HT.[57–60] Further evidence of the complex structure-function relation of triplet formation is shown in single molecule investigation of well-controlled self-assembly nanofibers of P3HT.[48] Where a non-perturbative mechanism is proposed for the observed efficient triplet formation in P3HT nanoaggregates showing J-aggregate spectral signatures (strong 0-0 transition compared to 0-1 sideband).[48] These results have opened up discussions on the role of structural and electronic contributions in intersystem crossing.

The perturbative treatment of intersystem crossing is shown as an example to formulate the conventional understanding of transitions between singlet and triplet states.[61,62] Consider a wave function resulting from spin-orbit coupling between a singlet and triplet state.

$$\Psi_{SO} = \Psi_T^0 + \lambda \Psi_S^0 \tag{1.4}$$

Where Ψ_{SO} is the perturbed wave function which results from the mixing from Ψ_T^0 pure spin, zero-ordered triplet excited state wave function and Ψ_S^0 is the pure spin zero-order singlet state wave function. We assume the coupling of just one singlet state for simplicity. In general, any appropriate singlet which can mix could be considered, not necessarily the first singlet state. The extent of mixing is given by λ from perturbation theory. Where λ

$$\lambda = \frac{< \Psi_S^0 | H_{SO} | \Psi_T^0 >}{E_T^0 - E_S^0} \tag{1.5}$$

H_{SO} is the spin-orbit coupling Hamiltonian. If more singlet states were included then λ would be equal to the sum over k singlet state, having the form of eq. 1.5. Considering the

transition dipole moment integral, given by

$$\mu_{TS} =< \Psi_T|\mu|\Psi_S > \tag{1.6}$$

Substituting the perturbed spin-orbit coupling wave function for the triplet wave function, eq. 1.4 into eq. 1.6 for Ψ_T, results in

$$\mu_{TS} =< \Psi_{SO}|\mu|\Psi_S > \tag{1.7a}$$

$$\mu_{TS} =< \Psi_T^0|\mu|\Psi_S^0 > +\lambda < \Psi_S^0|\mu|\Psi_S^0 > \tag{1.7b}$$

The first term on the right-hand side (RHS) of eq. 1.7b goes to zero due to spin selection rules; both wave functions are pure spin, zero-order. For the second term of eq. 1.7b on the RHS μ can be brought out since the transition dipole moment only operates on the electronic wavefunctions and Ψ_S^0 is a pure spin wave function, resulting in $\mu_{TS} = \lambda\mu$. The resulting probability of the transition is then proportional to the square of the transition moment, $|\mu_{TS}|^2 \propto |\lambda^2|$. The degree of the coupling between singlet and triplet state manifest in λ shown in the matrix element $< \Psi_S^0|H_{SO}|\Psi_T^0 >$ and the inverse of the energy difference of the pure spin, zero-ordered states. Above we used the transition dipole moment integral as an example, but we can notice the general way for two states of different multiplicities to have a non zero transition probability of is through the mixing of singlet character into the triplet state.

For the radiationless transition of intersystem crossing, we can use Fermi's golden rule to express the rate of intersystem crossing between S_1 and T_1. From a similar procedure shown above, we can then estimate the rate as

$$k_{ISC} \propto \Big[\frac{< \Psi_S^0|H_{SO}|\Psi_T^0 >}{E_T^0 - E_S^0}\Big]^2 \tag{1.8}$$

Equation 1.8 is the resulting approximation of a pure electronic perturbative treatment of

intersystem crossing. From this approximation, the rate of intersystem crossing can be inferred by simply inspecting both the contributions of the spin-orbit coupling Hamiltonian in the numerator and the exchange energy in the denominator. The exchange energy has been shown to be finite in most conjugated polymers;[10,63,64] therefore, the variance of the rate of intersystem crossing can be dominated by the spin-orbit coupling Hamiltonian. Using atomistic spin-orbit coupling parameters for the heavy atom (S, Se) as a proportionality for the matrix element one can infer the expected enhancement of triplet formation when moving down the period. However, this approximation follows from an assumption the heavy atom is incorporated in the electronic transition of the singlet state and cannot account for observations of structural contributions towards the intersystem crossing. Caution is needed when rationalizing efficient intersystem crossing from the approximation, equation 1.8. There is growing evidence of efficient triplet formation proceeding through a vibrationally assisted intersystem crossing mechanism in chalogen containing chromophores, and oligomers.[44,65–68]

1.6 Organization of the content

In chapter 2, details are given on the instrumentation and experimental methods used throughout the dissertation. In chapter 3, we examine the effect of triplet formation and interaction with emissive singlet excitons in poly(3-hexylselenophene) (P3HS) using single molecule spectroscopy. Due to the propensity of aggregation observed in various solvents, P3HS is diluted to ultra-low levels in order to resolve intrinsic singlet-triplet interactions at the single chain level. By measuring single molecule fluorescence intensity modulation, we determined the reverse intersystem crossing rate constant. Utilizing the determined value enabled modeling of the likelihood of triplet occupancy at the single polymer chain level, at steady-state. The unexpected observation of the disappearance of singlet-triplet quenching at the onset of P3HS aggregation gives further insight into the structure-function of triplet formation. This work initiates the investigation into both the role structure in

P3HS (chapter 4) and the development of the time-dependence solution of stochastic triplet occupancy (chapter 5).

Chapter 4, investigates the role of triplet formation in the photoinduced oxidation through a detailed statistical analysis of the blinking transitions observed in I(t) at the single molecule level. Using a high boiling point solvent to cast P3HS and P3HT in an inert host exposed to oxygen allows both the control of the conformation (collapsed defect cylinder) and observations of triplet-oxygen interactions. Comparing the rate of transition between "on" and "off" in similar chain lengths of P3HS and P3HT demonstrated the contribution of the heavy atom towards triplet-oxygen energy transfer.

Chapter 5, further develops the stochastic chemical master equation seen in chapter 1 by implementing the time-dependent solution. Numerical simulations of the complex interactions of multichromophoric stochastic kinetics are used to model the birth and death of triplet excitons at the single molecule level. Using previously determined reverse intersystem crossing constant for P3HS and P3HT in chapter 1 allows determination of other rate constants such as singlet-triplet quenching, triplet-triplet annihilation, and the ability to count the most likely number of triplets on a single polymer chain.

Chapter 6, presents a detailed study into the ultrafast dynamics of poly(3-decylthieneyl-enevinylene) (P3DTV) and it is heavy atom analog poly(3-decyl-seleneylenevinylene) (P3DTV) to elucidate if singlet-fission, spin allowed conversion of a photoexcited singlet exciton into two "free" triplet excitons, is operable. By contrasting the ultrafast dynamics of P3DTV to P3DSV using a combination of resonance Raman and transient absorption, we show no resolvable contribution of the heavy atom toward triplet formation in P3DSV. The result demonstrates an alternative hypothesis for the observed ultrafast excited state dynamics in P3DTV and P3DSV as Franck-Condon state relaxation involving many displaced vibrational modes.

Chapter 7, studying the small conjugated oligomer alkyl-substituted thienylene–vinylene dimer, dTV, an analog chromophore for P3DTV, we show unambiguous triplet formation

in dilute solution. The triplet formation is proposed through a large conformation displacement in the excited state enhancing intersystem crossing. Giving further evidence triplet formation in this class of polymer does not originate from singlet-fission.

2.0 Instrumentation and methods

2.1 Sample Preparation for single molecule spectroscopy

Single molecule samples were spun cast in a polystyrene matrix on rigorously cleaned glass substrates. Glass substrates were cleaned in sequential sonication (20 minutes) of trichloroethylene, acetone, and then methanol (TAM). Substrates were blown off with high purified nitrogen and further cleaned by UV-ozone exposure. Single molecule dilutions typically started from a maximum optical density of ~0.05 - 0.01 then serial diluted down until ~10^{-9} - 10^{-10} M. It's necessary to check the spot density of samples as a function of dilution to verify if samples have reached single molecule limit. This was done by collecting fluorescence images of freshly spun, unsealed concentrations of the analyte until a linear spot density as a function of sample dilution was observed. Samples were spun cast dynamically inside a nitrogen glove box. The weight percent of the polymer matrix used ranged from 2-5%. Final thickness of spun cast polystyrene matrix was ~100 nm, verified thickness using a KLA-Tencor Alpha Step 500 profilometer. Readers are referred to the seminal paper for polystyrene spin casting where Hall et al.[69] describes the square root dependence of spin speed on film thickness. Empirically 100 nm has worked best but it's important to keep thickness large enough to avoid interactions with the glass substrate.[24] Samples were further verified to be at the single molecule limit by observing intermittency of their intensity trace and diffraction limited spot size in the fluorescence image.[29]

To remove unwanted photochemistry with oxygen, samples were first pumped down in a deposition chamber to 10-7 torr for half an hour and then sealed with 100 to 200 nm of aluminum.[16] Once a sample was sealed measurements were made the day of or at the

most the next day. After measurements or not in use, samples were stored in a nitrogen environment, a glove box. English et al.[40] has shown oxygen may still diffuse through microscopic pinholes of the metal overcoat if left out in ambient conditions.

2.2 Single molecule scanning confocal microscopy

The schematic for the scanning confocal microscope used for single molecule fluorescence is shown in Figure 2.1. The microscope body is a Zeiss Axio Observer D1 equipped with a nanopositioning stage. Fluorescence images are constructed by raster scanning a diffraction limited spot from a high numerical aperture objective (oil immersion, 1.4 NA 100x magnification, Zeiss Plan-Apochromate) across the sample using a nanopositioning stage. A home-built LabView program controls the operation of the raster scanning, collection and saving of the signal. Excitation is either from an argon or krypton ion laser (Melles Griot / Spectra-Physics). Detection of the emission is either through an avalanche photodiode (Perkin Elmer or ID Quantique) or dispersed through a spectrograph (Shamrock SR-303i) utilizing an EMCCD (Andor Newton). Appropriate excitation filters are used before the detector and interference filters for the excitation to diminish contribution of plasma lines when collecting spectra.

Progression of a single molecule experiment proceeds with acquiring a fluorescence image locating single molecule spots. The fluorescence image is acquired by raster scanning the sample in the X-Y plane over the objective for a given resolution and dwell time. With the image, spots are translated over the microscope objective for collection. Signal acquired depends on the experiment which includes spectra, monitoring the time-dependent intensity of the emitted photons, fluorescence lifetime, and intensity-modulation photoluminescence (see next section). An APD (Perkin Elmer) is used to measure the intensity of the emitted photons of individual spots with a limiting time resolution of ~0.1 ms, using continuous wave excitation. Fluorescence lifetimes are measured using time-correlated single photon counting (TCSPC) with a picosecond pulsed diode laser (Edinburgh Instruments). The

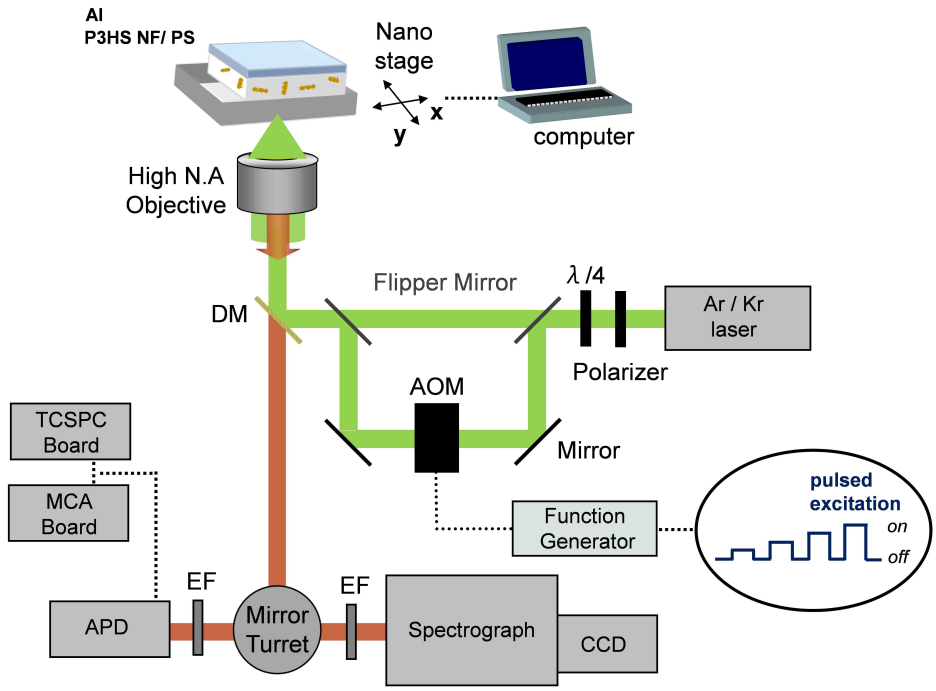

Figure 2.1: Block diagram of the scanning confocal microscope used for both single molecule fluorescence detection and Raman spectroscopy. The flipper mirror path to the AOM is utilized during the measurement of single molecule fluorescence intensity modulation. **Key** : N.A. numerical aperture; DM dichroic mirror; Ar and Kr argon and krypton AOM acoustic optical modulator; TCSPC time-correlated single photon counting; MCA multichannel analyzer; APD avalanche photodiode; EF edge filter; CCD charge coupled device.

emitted photons are measured with an APD (ID Quantique) using a SPC-130-EMN board (Becker & Hickl).

2.3 Fluorescence intensity-modulated single-nanoparticle photoluminescence spectroscopy

Measuring single-triplet interactions on single polymer chains requires the time-resolution and sensitivity to observe the interactions beyond was is feasible when measuring the I(t) trace. Thus, intensity-modulated single molecule spectroscopy is used to resolve multiple triplet populations at the single chain level.[25,26,70] This technique ascertains the pulse-induced kinetics of triplet formation through synchronous averaging of the fluorescence emission with a high repetition rate of square pulse excitations. The sequence of square

pulses are shaped from either a krypton or argon ion laser using an acoustic optical modulator (IntraAction Corp) and a function generator (Fluke 10 MHz). Repeat cycles ranging from 1 Hz to a few kHz can induce singlet-triplet quenching through build up of a triplet population. Emitted photons are measured using an APD with a multichannel analyzer (MCA-3, FAST ComTec). The multichannel analyzer has a limiting time-resolution of 100 ns per bin. Several pulses are used in a single waveform to measure intensity dependence on one single spot. The time between the pulses are sufficiently long to ensure complete recovery of ground state before the next pulse arrives.[70]

2.4 Epifluorescence microscopy and particle tracking of single molecule fluorescence

A schematic of the epifluorescence microscope is shown in Figure 2. The sample is illuminated using a 488 nm PhoxX laser diode through a high numerical aperture objective (oil immersion, 1.4 NA, Zeiss Plan-Apochromate) with a 63x magnification attached to a piezo focus driver (Physik Instrumente). The body of the microscope is an Axiovert 200 M with a manual 3-axis stage using differential micrometers. The excitation is focused to the back aperture of the objective to ensure the whole sample is illuminated. The emission is collected through the same objective passing through the dichroic mirror toward the detector. A telescope lens pair is used to magnify the image onto the electron-multiplying charge coupled device (EMCCD) after passing through an appropriate excitation filter to further block any excitation light transmitted through the dichroic mirror. The detector is an iXon Ultra 888 EMCCD manufactured by Andor Technologies. Epifluorescence has its advantage over confocal microscopy by allowing a high throughput collection of single molecule spots during an acquisition. The challenge with epifluorescence is the processing and analyzing the data collected.

The nominal experiment collected with the epifluorescence setup shown are the time-dependent trace of emitted photons, I(t), of any single molecule in the field of view. The I(t) traces are constructed from the raw digital video data. The traces are built from the raw

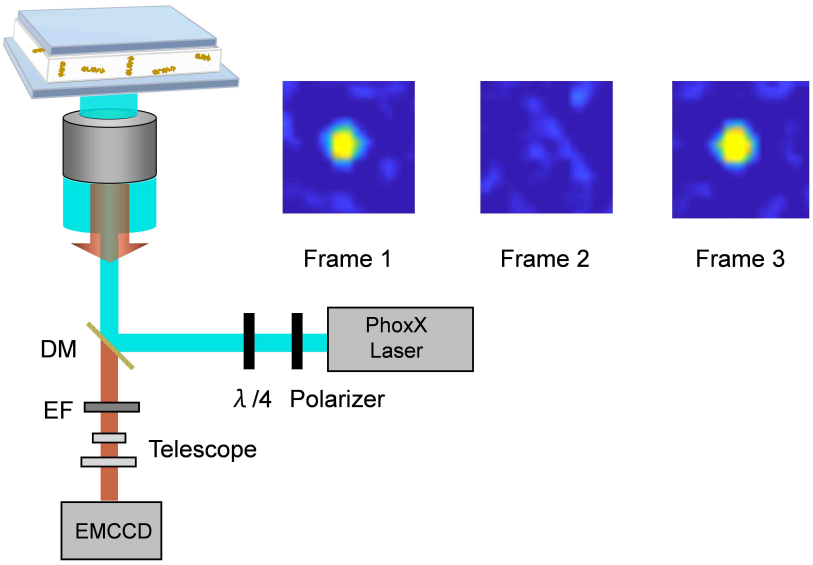

Figure 2.2: Block diagram of the epifluorescence microscope used single molecule fluorescence detection. The consecutive frames show a representative of the data acquired and depicting a blinking event of single polymer chain. **Key** : DM dichroic mirror; EF edge filter; EMCCD electron multiplying charge coupled device.

video using an in-house algorithm and user interface, implemented in MATLAB. The algorithm to acquire the I(t) traces from the video is based off the work of Crocker and Grier[71] and utilizes the available subroutines found on the repository.[72] The algorithm and the user interface implement five principles of particle tracking: (i) correcting for imperfections in the individual frames (ii) locating candidate particle positions (iii) refine candidate positions (iv) discriminating "false" particles (v) build the intensity trace. Unlike Crocker and Grier[71] where the experiment monitors colloidal particles leaving and entering the focal volume in a solution, our experiments have particles which are stationary in a solid host. Thus, once a particle's coordinate is found emitting in a frame, the particle position is known for the entirety of the acquisition regardless of blinking.

In brief, the algorithm corrects for both a non-uniform background intensity and random noise during collection, afterwards, candidate particles coordinates can be located. Particle locations are found on the constructed maximum pixel projection from the entire acquisition. The maximum pixel projection reconstructs a single frame using each maximum pixel intensity throughout the stack of frames. Using the maximum image projection allows ef-

ficient collection of single frame blinking events. The maximum pixel projection is used to locate candidate particle coordinates by iteratively looping through each pixel and listing the local maximum based on a user defined threshold found in a given radius, also set by the user. With the list of coordinates the algorithm integrates a given radius centered on the particle and tabulates the intensity per frame to construct the I(t) trace. False particles can now be removed from this first round of selection by statistically comparing attributes of the I(t) trace, such as average brightness, to background I(t) traces constructed from integrating in the void between the chosen particles. Background I(t) traces can either be chosen by hand or routinely chosen at random. Background coordinates within the image can be randomly selected by sampling random integers between the dimensions of the image. Then a degree of overlap can be calculated between the randomly selected coordinates verses particle coordinates. If the background spots are found to be overlapping with particle candidates, they are removed from further analysis. After removal, background spots are integrated in the same manner as candidate particles. This allow for an efficient random sampling of the sample with a low risk of integrating a real particle. Having both signal and noise is necessary for further analysis of I(t) such as choice of thresholding between "on"/"off" to determine the time characteristic of transitioning. Another iteration can take place towards tuning the particle detection threshold and integration diameter to find the best balance between correctly identifying particles and sampling the background.

2.5 Absorption, excitation and emission spectroscopy

Steady-state electronic absorption and emission has been a vital tool towards understanding the excited state properties and characterizing the emitting state of conjugated polymers. Attention must be given to the solvent of choice, temperature of which the sample is measured, concentration of sample to control for aggregation, and storage of the analyte in solution.[73–75] Measurements of the electronic absorption utilized a Shimadzu UV-2550. Measurements of the excitation and emission spectroscopy utilized an Edinburgh Instru-

ments FLS 980 with appropriate excitation and long pass filters (Newport). Monitoring the absorption and emission as a function of concentration was simply done by diluting from a known optical density until surpassing the limit of detection of both instruments. When measuring emission, maximum optical density of the samples was no more than ~0.1 to limit contributions from inner filter effects. Samples were stored on the bench top in the dark when monitoring the effect of storage on the electronic absorption and or emission. For characterizing the absorption and emission for comparison of single molecule spectra the analyte was dissolved in the same solvent used to spin cast the sample.

2.6 Resonance and non-resonant Raman spectroscopy

Resonance Raman (rR) is an indispensable tool for detailing the excited state dynamics of conjugated polymers. Understanding the electronic transition of the fundamental chromophore of the conjugated polymers is difficult because an intrinsic inhomogeneous absorption. Resonance Raman allows for the selectivity of the chromophore within the conjugated polymer and through the interpretation of the spectra, the excited state dynamics can be detailed. Raman is a two photon spectroscopy where an incident photon with energy $\hbar\omega_I$ an initial wavefunction ψ_i from a lower potential to an upper potential (when off resonant the upper potential known as a "virtual state") and then the emission of a photon with energy $\hbar\omega_S$ returns the system to state ψ_f, where f is a vibrational quantum number of the ground electronic state. When the frequency of the incident photon is equal to the frequency of the scattered photon $\omega_I = \omega_S$ the scattering is elastic or known as Rayleigh scattering. When $\omega_I \neq \omega_s$ the scattering is known as Raman scattering and the intensity of Raman scattering can be measured as a function of either the incident photon frequency or the scattered photon frequency. In the case of rR the incident photon, ω_I, is on resonance with an electronic transition and the nature of the scattering is no longer from the "virtual state" but the excited state potential. A challenge when interpreting the spectra is understanding the contribution to the intensity when the incident photon is either on resonance

or off-resonant.

From a classical perspective of Raman scattering derived from Placzek,[76–78] Raman transitions arise from the static change in the polarizability along a vibrational normal mode. The intensity of the Raman scattering can be approximated by

$$I_{Raman}(\omega_I) \propto \frac{\partial \alpha}{\partial Q_k}^2 \tag{2.1}$$

In the equation above, α is the polarizability tensor and Q_k is the vibrational coordinate for the k^{th} normal mode. As such only vibrations that change the polarizability of the molecule will have any Raman intensity. But if we want to use Raman to understand a chromophore's ground and excited electronic configuration a quantum approach expressing the Raman scattering in terms of molecular electronic and vibrational states is necessary. Describing the quantum aspects of Raman scattering with the Kramers-Heisenberg-Dirac (KHD)[79,80] expression is not easy but the complexity can be avoided if we consider a time-dependent perspective.[81,82] From the time-dependent approach of Raman scattering developed by Heller, Zink, and coworkers[82–85] demonstrate the significance of the vibrational wavepacket dynamics along the excited state. The intensity of the Raman scattering is

$$I_{i-f} \propto \omega_I \omega_S^3 [\alpha_{fi}]^* [\alpha_{fi}] \tag{2.2}$$

with the time-dependent Raman scattering cross-section given by

$$\alpha_{fi} = \frac{i}{\hbar} \int_0^\infty < \phi_f | \phi(t) > \exp\{i(\omega_k + \omega_I - E_{0-0})t - \Gamma t)\} dt \tag{2.3}$$

Where $< \phi_f | \phi(t) >$ is the autocorrelation function of the final vibrational wavepacket, ϕ_f, and the time-dependent vibrational wavepacket, $\phi(t)$; Γ is the phenomenological damping factor; ω_k is the zero-point vibrational frequency of the k^{th} normal mode; $E_{0\text{-}0}$ is the energy of the electronic origin. The focus is on the overlap term $< \phi_f | \phi(t) >$ in both computing

and understanding the significance. An analytical expression of the overlap term exist if we assume[84,85] (i) the potentials of both ground and electronic are harmonic, (ii) no Duschinsky effects meaning no alteration of the vibrational frequency and rotation of the normal coordinate in the excited state (iii) the electronic transition dipole moment has no dependencies on the normal coordinate (iv) no change in the force constant between ground and excited. The total overlap for k total symmetric normal modes has the form

$$\begin{aligned} < \phi_f|\phi(t) >= \prod_k \Big\{ \exp\Big[-\frac{\Delta_k^2}{2}(1-\exp(-i\omega_k t)) - \frac{i\omega_k t}{2}\Big] \\ \times(1-\exp(-1\omega_k t))^{n_k} \times \frac{(-1)^{n_k}\Delta_k^{n_k}}{(2^{n_k}n_k!)^{1/2}} \exp(-i\omega_0 t) \end{aligned} \tag{2.4}$$

Where ω_k is vibrational frequency (cm^{-1}), Δ_k is the displacement (dimensionless) of the k^{th} normal mode and n_k is the vibrational quantum number of the k^{th} normal mode in the ground state.

The intensity of the Raman arises from the dynamics of the propagating wavefunction on the excited surface. A key parameter towards the governing dynamics of the vibrational wavepacket is then the displacement, Δ between the equilibrium geometry of the ground and excited state. From this understanding displacement can be estimated from the relative intensities observed in the rR spectra. Thus, leading to historically named Savin's formula,[79,84,85] also known as the short-time approximation[81,86]

$$\frac{I_k}{I_{k'}} = \frac{[Intensity, mode1]}{[Intensity, mode2]} = \frac{\omega_k^2\Delta_k^2}{\omega_{k'}^2\Delta_{k'}^2} \tag{2.5}$$

The above set of equations allows for a robust modeling for quantitative features of the wavepacket dynamics and a fitting procedure for vibrationally resolved electronic absorption. Determining the relevant frequencies which show a resonance enhancement and displacements from the pairwise comparison of their intensities one can simulate the electronic

absorption[85,86] using

$$I(\omega) = C\omega \int_{-\infty}^{\infty} e^{i\omega t} < \phi|\phi(t) > dt \tag{2.6}$$

Where $I(\omega)$ is the intensity in photons per unit volume per unit time at frequency ω and C is a constant. The overlap for on specific normal mode is

$$< \phi|\phi(t) >= \exp\{-\frac{\Delta_k^2}{2}(1 - \exp(-i\omega_k t) - \frac{i\omega_k t}{2})\} \tag{2.7}$$

The procedure for using the above equations to both fit the electronic absorption and determine quantities of the wavepacket dynamics is to measure both resonance Raman and off-resonance Raman. Comparison between the spectra are made to observe for resonance enhancements of the Raman intensities. Displacements can be first approximated using equation 2.5 then the relevant frequencies shown to have resonance enhancement are used to simulate the absorption using equations 2.6 and 2.7. Iterations are completed by calculating the absorption and comparing to the experimental measured absorption. One can iteratively decide to fix parameters such as the electronic origin, add or remove vibrational frequencies, and or adjust displacement until convergence. This procedure is implemented in both MATLAB and Igor (WaveMetrics) is performed by trial and error. A similar procedure is implemented in Orca Advance Spectral Analysis[87,88] (ASA) which allows for automation of the fitting procedure, incorporation of many modes, efficient implementation of combination bands and overtones, as well as predictions assisted with electronic structure theory. Resonance Raman is measured using the confocal microscope coupled to a spectrograph (Shamrock SR-303i) utilizing an EMCCD (Andor Newton) using the ion gas laser for excitation, schematic shown in Figure 2.1. Off-resonance Raman measurements use a DXR Smart Raman (Thermo Scientific) equipped with a 780 nm laser. Samples are prepared in solid via drop casting onto clean glass substrates, cleaned in the same way described in sample preparation of this same chapter.

3.0 Effect of a heavy heteroatom on triplet formation and interactions in single conjugated polymer molecules and aggregates

3.1 Introduction

The tendency of many conjugated polymers to form non-covalent π-stacked aggregates has significant implications for the relaxation pathways of photogenerated singlet (S=0) excitons.[89,90] Moreover, the conformational and packing order also influence these dynamics although direct correlations between structure and measured branching ratios is often lacking.[59,91–93] In particular, the formation of spin-forbidden triplet (S=1) excitons represents a serious loss channel in thin film optoelectronic device settings[1] yet, it has proven exceedingly difficult to accurately resolve the contributions of these fine structure factors using conventional absorptive spectroscopic probes. Fortunately, molecular perspectives of subtle conformational ordering and packing interactions can be obtained by single molecule spectroscopy approaches, which also reveal intrinsic heterogeneity and population dynamics that usually are averaged away in ensemble techniques.[25]

In the conventional second-order perturbative picture of intersystem crossing, both the magnitude of the spin-orbit mixing and energetic splitting of the singlet and triplet states (ΔE_{ST}) determine the intersystem crossing rate constant (k_{ISC}). The amount of singlet-triplet mixing depends on the strength of the spin-orbit coupling matrix element whereas energy splitting (i.e., S_1-T_n) is determined by the exchange interaction. In the framework of conjugated organics, the spin-orbit interaction is usually small (<5 meV)[94,95] and ΔE_{ST} large (~0.5—1.0 eV)[63] which translate into small kISC values and quantum yields, Φ_{ISC} < 5%. Increasing spin-orbit coupling as well as reducing ΔE_{ST} can enhance triplet

yields by introducing heavy atoms into the conjugated backbone or lowering exchange splitting by spatially delocalizing electron-hole pairs, respectively. However, this pure electronic picture does not explicitly take into account the roles of vibrations as well as conformational and packing order, which can significantly modify intersystem crossing rates and yields due to its localized nature in conjugated organics.[96,97] The existence of multiple, structurally distinct conformers can further obscure mechanistic details of triplet formation in conjugated polymers. Interestingly, Monkman and co-workers and Koehler and co-workers extensively investigated triplets in a range of conjugated polymers in solution and found comparable singlet-triplet splitting energies regardless of structure and conformational qualities.[1,63,98]

In other cases, anomalously large triplet populations and rapid triplet formation dynamics have been reported in crystalline organic systems, such as acenes and some conjugated polymers, that are proposed to arise from singlet fission.[99,100] Recent work has suggested this process is mediated by intermolecular, or, interchain charge transfer states.[101] In an earlier study, we prepared highly purified, single chain aggregates of poly(3-hexylthiophene) (P3HT) exhibiting highly delocalized singlet exciton character with J-aggregate spectral characteristics and efficient triplet formation.[75,102] Time-resolved, single molecule electric field dependent spectroscopy studies of P3HT J-aggregates demonstrated the involvement of interchain charge transfer states as triplet precursors on time scales <5 ns.[48] King et al. also reported enhanced triplet formation in polymers with excited state charge transfer character that could be tuned by changing the dielectric environment (i.e., solvent polarity).[103] Steiner et al. further investigated singlet-triplet interactions in P3HT single chains and observed efficient quenching in addition to photon anti-bunching indicative of a single photon emitter.[104] These studies underscore the fact that fine structural qualities (i.e., polymer chain ordering and aggregation) can indeed have a significant influence on triplet formation mechanisms in polymer chains but correlations between structure and time-dependent triplet populations remain poorly understood.

Here, triplet formation and interactions with emissive singlets are investigated in poly(3-hexylselenophene) (P3HS). Like its sulfur analog, P3HS readily aggregates in solution resulting in limited solubility even at very low concentrations (<10^{-7} M).[105] In addition to strong aggregation, the introduction of the heavier selenium atom significantly shortens the singlet exciton lifetime from ~500 ps (P3HT) to ~8 ps.[44] This effect was assigned to facile triplet formation on time scales of ~26 ps due to greater spin-orbit coupling and densities of higher energy triplet states in the vicinity of the first excited singlet state (S_1).[44]

We use single molecule fluorescence excitation intensity modulation spectroscopy to probe triplet population dynamics and singlet-triplet interactions on nano- to millisecond time scales.[26] Single P3HS molecules are excited with rectangular shaped laser pulses of varying intensity to examine the fluorescence dynamics of individual molecules or aggregates. When the laser turns on, no triplets are present but, at $t > 0$, substantial triplet population buildup occurs (i.e., multiple triplets on a single P3HS chain) leading to rapid quenching of the singlet fluorescence intensity on time scales <10 μs. Upon increasing concentrations of dilute solid dispersions near the onset of aggregation, evidence of triplet-induced fluorescence quenching vanishes. Although similar effects occur in lesser-ordered P3HT aggregates,[106–108] this result is particularly surprising because of the ability of single P3HS chains to support more than one triplet.[44] We propose that efficient non-radiative processes involving highly displaced torsional vibrations or formation of short-lived interchain polaron pairs are probably responsible for suppression of triplet formation or enhanced relaxation in P3HS aggregates. Overall, the sensitivity of P3HS triplet population dynamics underscores the importance of structure and packing on photophysical pathways and material functionality.

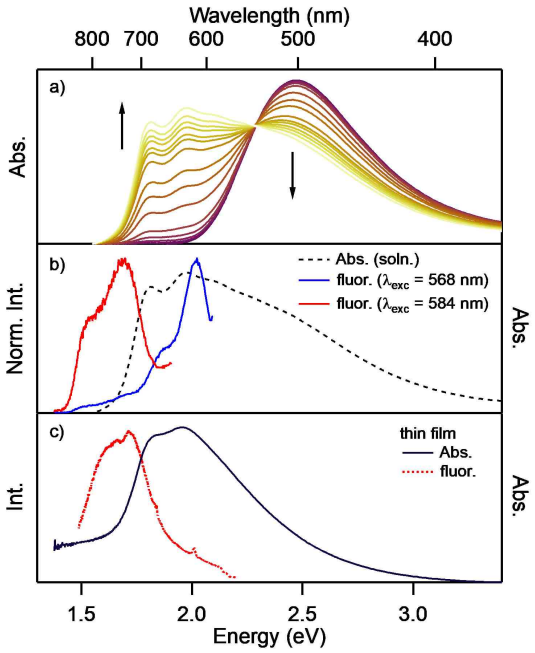

Figure 3.1: a) Absorption spectra of P3HS solutions (toluene) measured at 15 min intervals over a 4 hr time span. Arrows indicate the evolution of spectral features (note the characteristic isosbestic point at ca. 620 nm, 2.2 eV). b) Fluorescence spectra (solid traces) with corresponding absorption spectra (dashed traces) of fresh (blue) and aged (red) solutions excited at 584 nm and 568 nm, respectively. c) Absorption and fluorescence spectra of P3HS thin films.

3.2 Result and Discussion

3.2 Spectroscopic signatures of P3HS aggregates and isolated chains

A key goal of this work is to understand how the heavier selenium heteroatom, conformational ordering and aggregation of P3HS chains influence relaxation of singlet excitons and conversion into spin-forbidden triplets. In order to reliably distinguish between single chains and aggregates, it is first necessary to characterize the spectroscopic signatures of both species. P3HS is known to have limited solubility in addition to associating into large, branched fibrillar networks in a variety solvents.[109] Electronic absorption spectra offer a useful view into the relative amounts and quality of aggregates and their excitonic properties. Figure 3.1a shows electronic absorption spectra (optical densities ~0.1) over a period of ~4 hours following gradual cooling from heating toluene solutions to ~400 K to promote

complete solvation. New transitions emerge upon formation of aggregates that are red-shifted by ~0.5 eV compared to absorption maxima of solvated P3HS chains. A partially resolved vibronic progression of ~1400 cm^{-1} is observed corresponding to the excited state frequency of the totally symmetric CC ring breathing mode. These features are consistent with aggregates and the isosbestic point at ~580 nm (~2.2 eV) confirms the presence of two structurally distinct chromophores. We do not attempt to estimate exciton coupling strength in P3HS aggregates but, the overall similarity of absorption spectra to P3HT thin film aggregates is consistent with H-aggregate or, interchain, character.[90,110] Additionally, P3HS aggregate absorption spectra are red-shifted from those of P3HT because of the heavier selenium atom. It is also noteworthy that a fraction of P3HS chains do not aggregate as seen from the persistent solution-like, amorphous, absorption feature at ~2.5 eV (500 nm).

Figure 3.1b displays P3HS fluorescence spectra of an aggregated P3HS solution excited on resonance with amorphous and aggregated chains. At higher energy excitation (λ_{exc} = 568 nm), solvated P3HS chains are selectively excited that exhibit a partially resolved fluorescence line shape is observed with a progression frequency interval of ~1400 cm^{-1} (blue solid trace) and noticeably stronger $I_{0\text{-}0}$ peak. This lineshape is characteristic of intra-chain excitons typically found on single polymer chains with high monomer co-planarity.[90,110,111] As P3HS chains begin to aggregate, fluorescence spectra gradually red-shift with lower signal-to-noise ratios (red trace), consistent with known spectroscopic trends of π-stacked aggregates. These structures can be selectively excited with lower energy photons (λ_{exc} = 584 nm) and line shapes become non-Poissonian suggesting distortions from interchain exciton coupling effects.[112] We found no evidence of line shape distortions from reabsorption of emitted light (inner filter effect) as expected from the low optical densities of these solutions (<0.1). For comparison, thin film optical spectra were measured (Fig. 3.1c) that bear remarkable similarity to aged solutions. Both the amounts and quality of aggregates may be tuned by choice of solvent[113] and we examined this effect by dissolving P3HS in chlorobenzene. Figure 3.2 shows absorption spectra measured over extended intervals fol-

lowing initial cooling of heated solutions and, unlike toluene solutions, absorption spectra, demonstrate much lower aggregation content despite that fluorescence spectra exhibit partially resolved vibronic structure similar to toluene solutions in Fig. 3.1b.

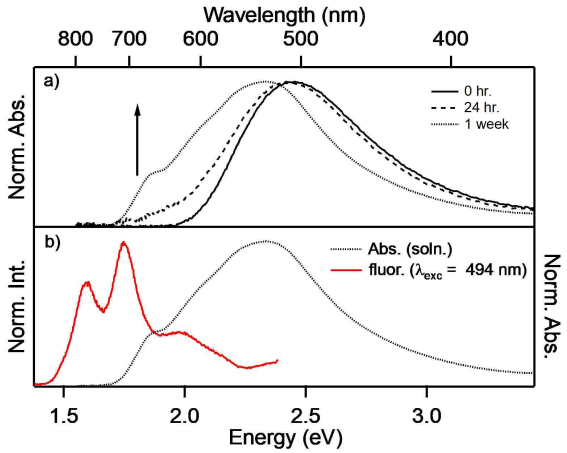

Figure 3.2: a) Absorption spectra of P3HS solutions (chlorobenzene) over a 1-week time span. b) Fluorescence spectra (solid traces) with corresponding absorption spectra (dashed traces).

To determine if variations in nano-morphology exist between samples, TEM imaging was performed on toluene and chlorobenzene dispersions shown in Figure 3.3a and b, respectively. Both samples show dendritic and bundle-like structures as reported earlier,[109] with characteristic widths approaching ~100 nm. However, samples processed from toluene have 'blurry' regions surrounding the web-like nanofibers corresponding to either non-aggregating fractions or smaller aggregates.[109] The lack of significant aggregate absorption in chlorobenzene samples suggests that the large web-like structures are comprised of smaller aggregates interspersed with non-aggregating chains. The relatively low molecular weight of the P3HS sample (~30 KDa) as well as solvent interactions could also play a role in the cooperativity between π-stacking and alkyl side group interactions that influence the formation and morphology of hierarchical structures.[109]

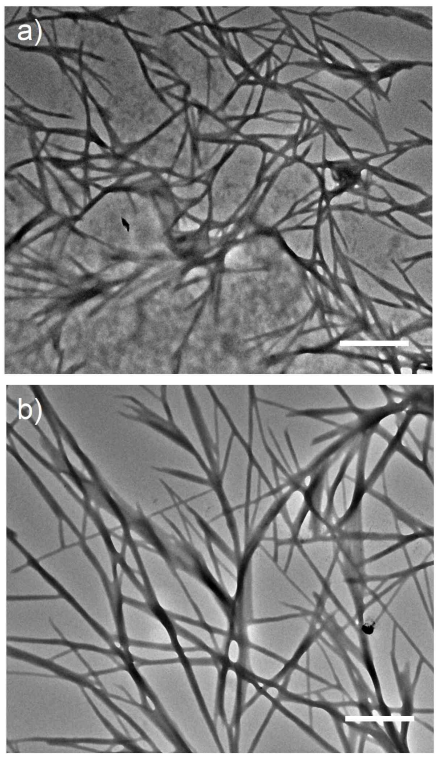

Figure 3.3: TEM images of P3HS cast from toluene (a) and chlorobenzene (b). Scale bar = 1 μm.

3.2 Effect of conformational order and aggregation on triplet formation yields and singlet-triplet interactions

In order to track the evolution of triplet formation and singlet-triplet interactions in single chains and aggregates, it is essential that P3HS chains be well solvated and dispersed within a solid host support matrix (e.g., polystyrene). Because chlorobenzene and related solvents tend to promote large agglomerates, we instead focus on toluene dispersions that were found to yield better quality thin films. Following spin-casting into dilute thin films and deposition of a metal overcoating to prevent unwanted photochemistry, single molecule spectroscopy and imaging techniques are employed to resolve signatures of single isolated chains and aggregates.

Figure 3.4 shows fluorescence images of P3HS dispersed in polystyrene thin films of two different concentrations with corresponding representative spectra excited at 2.18 eV

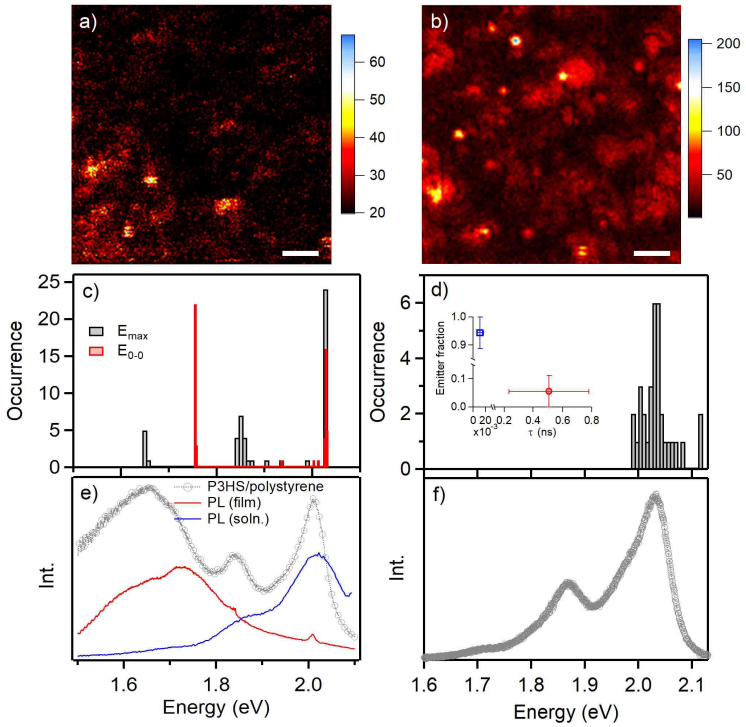

Figure 3.4: Fluorescence images (a, b), histograms (c, d) and representative fluorescence spectra (e, f) of concentrated (ca. 10-8 M) and dilute (ca. 10-9 M) P3HS dispersed in polystyrene (toluene), respectively. Scale bar = 1 μm. Fluorescence energy histograms are generated by using both the maximum (Emax) and energy of the electronic origin (E0-0). The inset in (d) shows average and standard deviations of single P3HS chain fast (blue square) and slow (red open circle) fluorescence lifetime decay amplitudes and times. For comparison, representative thin film (red) and dilute solution (blue) are displayed with averaged fluorescence spectra of concentrated P3HS dispersions in e).

(568 nm). As noted earlier, this excitation energy preferentially excites isolated P3HS chains and comparisons of concentrated and dilute sample images, reveal a large disparity in the areal density of emitters. Figs. 3.4a,b correspond to ~10^{-8} M (concentrated) and ~10^{-9} M (dilute) of P3HS in polystyrene dispersions cast from toluene, respectively. It is peculiar that the lesser-concentrated sample exhibits larger areal densities and intensities suggesting either strong self-quenching or low excitation efficiencies at the excitation energy used in the higher concentrated sample. These possibilities are more closely examined in the following.

We next classify emitters in both samples using either the energy of electronic origin

($E_{0\text{-}0}$) or energy of the maximum intensity (E_{max}) values to assess if new transitions emerge upon dispersing P3HS into polystyrene. Comparison of fluorescence energy distributions indicate that a substantial fraction of P3HS in the concentrated sample remains in an aggregated state as seen from the presence of at least two emitting species closely resembling the solvated and aggregated P3HS forms shown in Fig. 3.1b, blue and red traces, respectively. At lower concentrations, mostly diffraction-limited spots are observed and spectra show similar features as the solvated form of P3HS although ensemble averaging in solutions leads to linewidth broadening in the latter. The prominent 0-0 transition in dilute spectra are consistent with isolated P3HS chains that also coincide with their E_{max} values. These features are also representative of intrachain exciton character, which is expected because of the greater quinoidal character of P3HS.[75,114] It is also noteworthy that energies of the dilute P3HS samples are in good agreement with earlier fluorescence spectra of dilute P3HS solutions in CS_2.[44]

Once P3HS chains begin to aggregate, red-shifted and broadened fluorescence features emerge. While excimer-like species may be responsible for the low energy emission feature, we did not observe longer lifetimes as expected from this type of excited state interaction (data not shown). Fluorescence images were also excited with 1.92 eV (647 nm) light from samples of comparable concentrations only showed larger spots and fluorescence emission from the lower energy form. Similar imaging experiments performed on the dilute samples at this lower excitation energy were devoid of any spots (data not shown). The cluster of peaks at ~1.87 eV in Fig. 3.4c correspond to the 0-1 vibronic sidebands of the solvated form which is confirmed by comparison with dilute P3HS dispersions in Figs. 3.4d, f.

Fluorescence images and distributions in Fig. 3.4 demonstrate that aggregates are dissociated in highly dilute P3HS dispersions (i.e., $<10^{-9}$ M). Moreover, comparison of average intensities suggests enhanced non-radiative decay contributions in P3HS aggregates, which has been observed in related systems and proposed to result from a static quenching mechanism.[115,116] Related single molecule spectroscopy studies of nominally non-emissive

polymers have also demonstrated restoration of fluorescence yields from non-aggregated chains by dilution into solid hosts. For example, Hu et al. found that dispersion of non-emissive poly(thienelyenevinylene) (PTV) chains into PMMA could restore emission from the solvated form, which is identical to the effect of diluting P3HS to very low levels as seen in Fig. 3.4.[116] Average fluorescence lifetimes of isolated P3HS chains were measured (Fig. 3.4d, inset) that were dominated by an instrument-limited decay component (<50 ps, >90%, blue open square). This is in contrast to typical values reported from dilute P3HT samples of ~500 ps.[44] A simple explanation for the faster decay dynamics in isolated P3HS chains could originate from its lower band gap energy. According to the energy gap law (weak coupling limit), this would result in a larger value of the non-radiative decay component by about a factor of 3. However, this estimate is still too small to rationalize experimental trends in lifetimes and quantum yields. It is also important to stress that application of the energy gap law requires no significant geometric distortions of the nuclear framework between ground and emitting state potentials, which is not the case in either solvated or aggregated P3HS chains.[117]

Additional insights into the origins of the faster fluorescence decay dynamics in P3HS can be inferred from transient absorption spectroscopy studies of P3HT and P3HS dilute solutions. Pensack et al. reported significantly faster singlet exciton lifetimes for the latter (~500 ps vs. ~8 ps) by monitoring the stimulated emission signal contribution to transient decays. In addition, these authors observed very low fluorescence quantum yields for P3HS ($\sim 4 \times 10^{-3}$). Similar trends were observed for poly(3-hexyltellurophene) (P3HTe) that have much shorter singlet exciton lifetimes (~0.3 ps) compared to P3HS. With the aid of time-dependent density functional theoretical calculations, these behaviors were assigned to efficient triplet generation due to the presence of the heavier heteroatom (larger spin-orbit coupling) and higher-lying triplet states nearly degenerate with the photogenerated singlet exciton.[44] It is important to mention that spin-orbit coupling in this class of molecules is expected to be small in the limit of idealized C_{2h} symmetry but should be relaxed in lower

symmetry groups.[65,118]

In a related single molecule study of triplet formation in single chain P3HT aggregates with high planarity, we demonstrated that intrachain exciton delocalization increases the likelihood of encountering a charge transfer state in their lifetime.[48] Carriers occupying charge transfer states can then preferentially recombine at later times (i.e., >5 ns) to triplets according to spin recombination statistics. Similar mechanisms involving charge transfer states have been proposed in singlet fission active systems where greater singlet exciton delocalization promotes faster triplet formation.[119] In order to determine if charge transfer states contribute to P3HS triplet formation, we measured fluorescence decays of single chains in the presence of a time-dependent electric field (i.e., square-wave) at 1 KHz. Electric field-dependent fluorescence decays showed no modulation which casts doubt on significant charge transfer state character or other triplet precursors present during the singlet exciton lifetime. However, if these states are sufficiently short-lived (i.e., beyond our available time resolution, ~50 ps) then their presence may go undetected in this experiment.

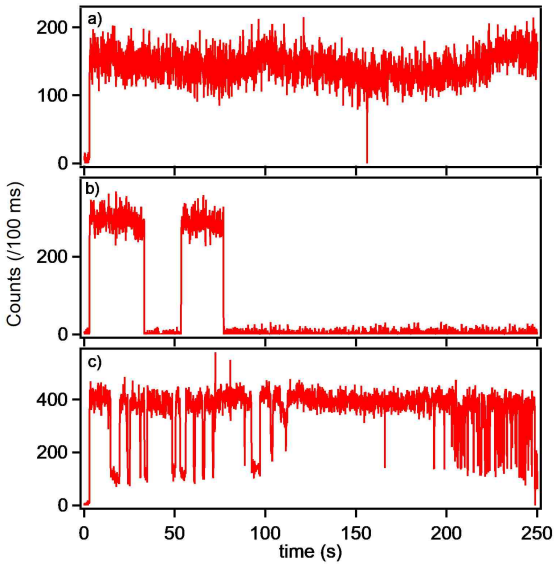

Figure 3.5: Representative fluorescence intensity transients of dilute P3HS solid dispersions classified according to typical time scales of intermittency (on/off behaviour). a) stable, b) long (blinking), and c) flickering. Proportions of each class are provided in the text.

Although triplets form on fast time scales in P3HS, their populations may often per-

sist over much longer time scales appearing as oxygen-related photochemistry from sensitization of reactive species (e.g., singlet oxygen). Time integrated intensity transients of single molecules offer a convenient means to investigate such interactions on time scales >1 ms. We measured fluorescence intensity transients of dilute P3HS dispersions sealed with an aluminum overcoating to avoid complications common in oxygen saturated samples. Figure 3.5 shows representative intensity transients collected from over 90 isolated P3HS chains displaying a wide range of responses. Most molecules (~65%) were photostable over the total measurement duration (>3—5 min), which are represented in Fig. 3.5a. The rest exhibit some intermittency (on/off behaviour) with time scales of ~1 ms up to tens of seconds. For example, approximately 25% of all molecules displayed long on/off time periods (i.e., seconds to tens of seconds, Fig. 3.5b) that has been assigned previously to reversible interactions with quenching defects.[25]

Fast flickering behaviour occurred in <10% of single P3HS chains (e.g., Fig. 3.5c) with characteristic on/off times ranging between ~1 ms up to 100 ms. This is representative of triplet quenching of emissive singlet excitons and has been reported in fluorescence intensity transients for many conjugated organic molecules.[120] Photostable P3HS molecules also showed lower average counts than observed in molecules with distinct intermittency. It is possible that triplets may still be present in these molecules and that fast intermittency occurs below the available time resolution of the multi-channel scaler (~1 ms). Another possibility is that photostable spots are aggregates but this was ruled out from corresponding fluorescence spectra that only exhibit line shapes resembling those in Fig. 3.4f.

While informative, fluorescence intensity transients alone are not sufficient to fully resolve triplet-related interactions and processes occurring over the triplet lifetime. For this reason we turn to an alternative approach to interrogate singlet-triplet and triplet-triplet interactions on faster time scales in addition to resolving ground state recovery kinetics. Single molecule excitation intensity modulation spectroscopy was performed on dilute P3HS solid dispersions by exciting the fluorescence of single molecules with a train of rectangular

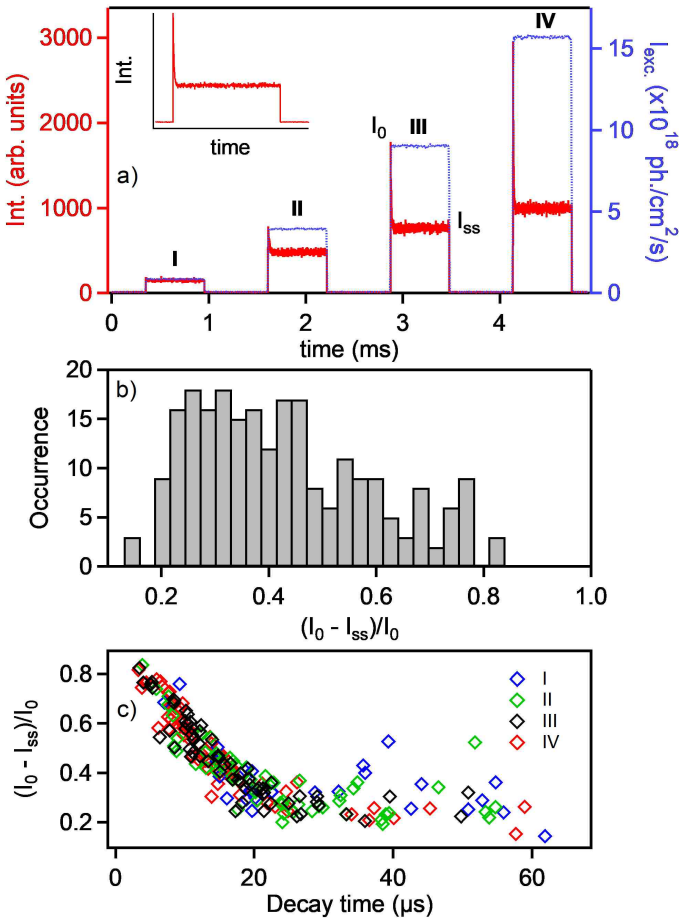

Figure 3.6: a) Representative MCA transient from a single P3HS chain (red trace). Inset: MCA transient from a single rectangular pulse. Roman numerals correspond to the excitation pulse sequence (blue trace) for λexc=568 nm (2.18 eV). The prompt and steady state intensities are denoted (I0 and ISS, respectively). b) Histogram of observed fluorescence intensity modulation quenching depth fraction. c) Scatter plots of quenching depths versus decay time constants for each excitation pulse intensity in the pulse cycle.

laser pulses of linearly increasing intensities (i.e., stair-step shaped pulse series). Details of the experimental setup and procedure have been described previously.[102] In brief, responses of single molecules excited with stair-step pulse shapes are synchronously averaged for many thousands of cycles (>104) using a multi-channel analyzer (MCA) and single photon counting detection. The presence of dynamic singlet quenchers can be resolved on ~100 ns up to ~10 ms time scales from MCA transients. Because various charged and neutral species with net spin can quench emissive singlets, care in sample preparation and measure-

ment conditions are required to properly distinguish these different quenchers. For example, both triplets and polarons are capable of quenching fluorescence but their respective contributions can be resolved according to known lifetimes and from careful control of sample environment. The longer lifetimes of triplets and their ability to sensitize reactive oxygen species greatly simplifies their identification as quenchers in intensity modulated signals. This is accomplished by selective choice of modulation frequency range (i.e., similar to known triplet lifetimes) and controlled exposure to ambient oxygen.[102] Figure 3.6a shows a representative excitation intensity modulated fluorescence response from a single P3HS chain (red trace) and typical peak excitation pulse intensities were approximately 4, 11, 25, and 58 W/cm^2 (200 Hz repetition rate) per pulse (blue trace). We classify MCA transients according to quenching depth and decay time, both of which are dependent on the excitation intensity as well as the triplet yield. For molecules displaying fluorescence intensity modulation, the prompt intensity (I_0) decays to a steady state value (I_ss) on time scales typically ranging from ~1—10 μ s, consistent with a triplet-induced quenching mechanism. The nature of the singlet-triplet annihilation has been proposed to follow a long-range, dipole-dipole (Förster-type) energy transfer mechanism although contributions from a Dexter-type mechanism are also possible.[25] Quenching dynamics from MCA transients typically follow single exponential decay behavior (see Fig. 3.6a inset) and the effective decay rate constant is related to both the forward (k_f) and reverse (k_b) rate constants involving formation and relaxation of the triplet (vide infra). Figure 3.6b shows the histogram compiled from quenching depth fraction ($I_0 - I_{ss}/I_0$) and, similar to fluorescence spectra and intensity transients in Figs. 3.4 and 3.5, MCA intensity transients display a wide range of responses reflecting the intrinsic heterogeneity of the P3HS samples. Over 80% of all molecules investigated (>90) experienced fluorescence intensity modulation, even at very low intensities (<1 W/cm^2) and the modulation depth, or, contrast ratio, serves as a useful metric for understanding the population and depopulation kinetics involving the triplet manifold.

Figure 3.6c displays the quenching modulation depth and the measured decay time for

each excitation pulse in the sequence (i.e., I-IV). The near linear correlation between these values over the range of excitation intensities indicates that non-linear processes do not contribute to the observed dynamics. Assuming that singlet-triplet quenching is solely responsible for the observed fluorescence quenching, the triplet lifetime may be estimated if excitation intensities are low enough and time intervals between pulses are sufficiently long to allow all triplets to relax before the next pulse in the sequence. Palacios et al. extracted estimates of the reverse intersystem crossing rate constants (k'_{ISC}) from T_1 to S_0 of a polythiophene derivative using this approach.[43] They found a value of ~6×10^5 s^{-1} (τ~17 μs) at an excitation intensity of ~0.4 W/cm^2.[43] Similar values at higher excitation intensities were observed that also fall within the typical range used here. By fitting fluorescence quenching decays at low excitation intensities (e.g., ~10 W/cm^2 or less) we estimate an average quenching time constant of ~8 μs. The smaller values could reflect larger triplet populations, enhanced singlet-triplet interaction or shorter triplet lifetimes. It is important to also point out that this approach of estimating triplet lifetimes directly from quenching dynamics tacitly assumed only one triplet present per polymer chain.

We next use a modified version of the stair-step pulse sequence to ascertain the decay of triplet quenchers in the dark, which should provide a better estimate of the triplet lifetime. For example, upon reaching a steady state intensity level, populations return back to the ground state once the pulse turns off which should be exactly $(k'_{ISC})^{-1}$. By comparing the prompt intensities (I_0) of two pulses delayed temporally with respect to each other, it is possible to estimate these values if the spacing between pulses is similar to the reverse intersystem crossing relaxation time. If another pulse excites the molecule before the triplet population falls to zero, the I_0 values are diminished with respect to the first pulse. Figure 3.7a shows an example of the two pulse experiment for a single P3HS molecule and averaged prompt intensity ratios ($I_{[1]}/I_{[2]}$) recorded for various pulse delay times (Δt) from ~60 P3HS molecules are shown in Fig. 3.7b. The same measurements were performed on isolated P3HT chains of similar molecular weight that are also displayed in Fig. 3.7b.

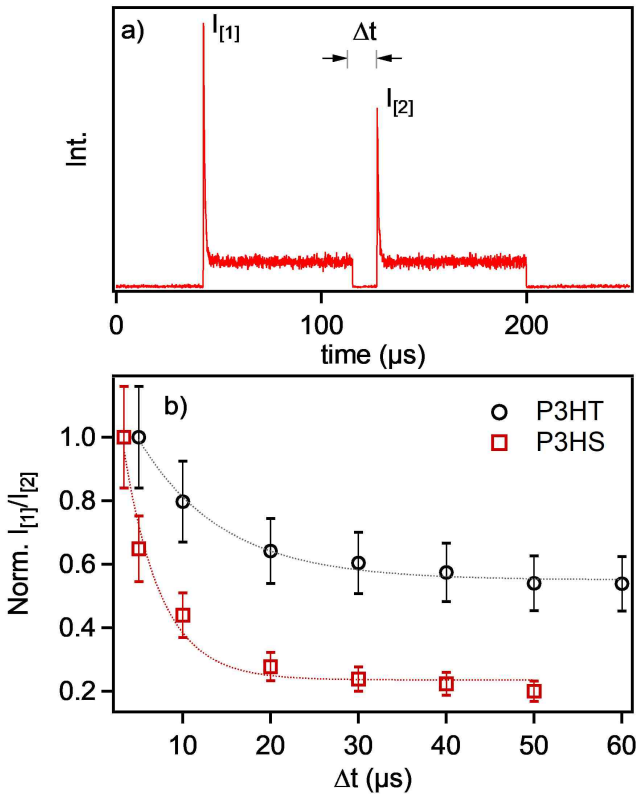

Figure 3.7: a) Variable delay two-pulse experiment to reveal triplet relaxation time scales in the dark (i.e., $T_1 \rightarrow S_0$). b) Comparisons of triplet relaxation kinetics for single P3HS and P3HT chains.

Characteristic times for the $T_1 \rightarrow S_0$ conversion were estimated to be ~4 μs $\pm$ 0.8 μs and ~10 μs $\pm$ 1 μs for P3HS and P3HT samples, respectively, by fitting to a single exponential decay function. This estimate of $(k'_{ISC})^{-1}$ is about a factor of 2 smaller than those obtained from fitting low excitation intensity MCA transient quenching decays. We expect estimates of $(k'_{ISC})^{-1}$ determined from the two-pulse MCA experiment to be more accurate because only triplets should be present after the first laser pulse turns off. Furthermore, this experiment ensures that no pre-pulse triplet population exists in the regular stair-step MCA measurements described above meaning that all processes originate from the photo-prepared state, S_1.

In previous applications of the excitation intensity modulated single molecule spectroscopic technique used here, fluorescence quenching dynamics were simulated using a

coupled, multi-state photodynamic model that assumed small intersystem crossing quantum yields (<5%) and large (infinite) triplet-triplet annihilation rate constants (i.e., $k_{TT} >> k_{ISC}$). This scenario permits application of the steady-state approximation to the governing rate equations involving short-lived, multi-triplet configurations (i.e., more than one triplet per chain). The multi-level system can then be reduced to an effective two-state model where the molecule spends the most time in either S_0 or T_1 with never more than one triplet occupying the molecule at any given time.[26] While these assumptions greatly simplify simulations of the time-dependent S_0 and T_1 populations and quenching dynamics behavior, it is likely that they are not valid for single P3HS chains. For example, based on the larger k_{ISC} values of P3HS measured previously, multiple triplets are probably present immediately after the laser turns on. If k_{TT} is infinitely fast, then these triplets will annihilate until there remains only one per P3HS chain. However, if singlet-triplet annihilation is inefficient (i.e., based on poor spectral overlap between the singlet emission and triplet absorption spectra), then fluorescence intensity modulation depths should be small and quenching dynamics should be slow, which is not the case even for low excitation intensities. If k_{TT} is also finite (i.e., comparable or not much larger than k_{ISC}), then multiple triplets may persist for longer times. Even if a Dexter-type exchange component of singlet-triplet quenching is operative, the localized nature of the triplet and short singlet lifetimes imply its overall contribution is also small.

Experimental evidence of more than one triplet present on P3HS chains is perhaps most apparent from the fast quenching dynamics and significant discrepancy between estimates of the average $(k'_{ISC})^{-1}$ values determined by fitting MCA fluorescence quenching decays at low excitation intensities (~8 μs) compared to those obtained from the delayed two-pulse experiment (~4 μs). The central approximation in using the former approach is that only one triplet is present, hence, fluorescence quenching is dictated entirely by the singlet-triplet interaction (k_Q). However, when multiple triplets exist on a single chain due to the finite nature of k_{TT} or inefficient singlet-triplet quenching, this assumption breaks down. This

effect is also apparent when comparing triplet relaxation dynamics and quenching depths in P3HT and P3HS chains of similar molecular weight (Fig. 3.7b). Namely, triplet populations are much lower in the sulfur analog.

Using the assumptions that triplet-triplet annihilation is finite and singlet-triplet quenching is inefficient, Barzykin and Tachiya derived a stochastic model for the time-dependent probability of finding n triplets on a single polymer chain, $P_n(t)$.[56]

$$\begin{aligned}\frac{d}{dt}P_n(t) = k_{f,n-1}P_{n-1}(t) - \left[k_{f,n} + nk_b + \frac{1}{2}n(n-1)k_{TT}\right]P_n(t) \\ +k_b(n+1)P_{n+1}(t) + \left[\frac{1}{2}(n+1)(n+2)k_{TT}\right]P_{n+2}(t)\end{aligned} \tag{3.1}$$

Here, $k_{f,n}$ is the effective forward rate constant for triplet formation,

$$k_{f,n} \simeq \frac{k_{exc}k_{ISC}\tau_{fl}}{1 + nk_Q\tau_{fl}} \tag{3.2}$$

where k_{exc} is the excitation rate constant ($= I_{exc}\sigma$), I_{exc} is the excitation intensity, and σ is the absorption cross-section, k_{TT} is the first order triplet-triplet annihilation rate constant in a molecule with two triplets, k_{ISC} is the intersystem crossing rate constant, k_Q is the singlet-triplet quenching rate constant and τ_{fl} is the fluorescence lifetime.[56] The rate constant of the reverse (intersystem crossing) is given by,

$$k_b = k'_{ISC} \tag{3.3}$$

and the time-dependent fluorescence intensity is expressed as,

$$I(t) = I(0)\sum_{n=0}^{\infty}\frac{P_n(t)}{1 + nk_{QST}\tau_{fl}} \tag{3.4}$$

The rate equation describing $P_n(t)$ does not have an analytical solution, however, the steady-

state solution is,

$$P_n(\infty) = \frac{2^{(\beta-1-3n)/2}\alpha^n}{n!}\frac{I_{\beta-1+n}(\alpha/\sqrt{2})}{I_{\beta-1}(\alpha)} \tag{3.5}$$

where $\alpha = 4(k_f/k_{TT})^{1/2}$, $\beta = 2k_b/k_{TT}$, and $I_p(x)$ denotes the modified Bessel function of the first kind.[56]

We now use this model to estimate the probability of more than one triplet being present on single P3HS chains based on experimentally measured parameters. Using a typical value of the absorption cross-section of conjugated polymers of ~10^{-15} cm^2, k_{exc} ranges between ~10^5-10^7 s^{-1} in our experiment. We then use the measured values of k_{ISC} and the singlet exciton lifetime (~τ_{fl}) from Pensack et al. (3×10^{10} s^{-1} and 18×10^{-12} s, respectively)[44] in addition to estimates of $(k'_{ISC})^{-1}$ (~2×10^5 s-1) from the delayed two-pulse experiment (Fig. 3.7). It is also assumed that k_{TT} and is comparable to k_{ISC} and we vary this parameter to determine conditions for generating non-zero probabilities of more than one triplet per P3HS chain. Lastly, similar to the findings of Tachiya and co-workers,[56] we found that $k_{f,n}$ was insensitive to k_Q using the experimental values outlined above, which allows the following approximation, $k_{f,n} \sim k_{f,0} = k_{exc}k_{ISC}\tau_{fl}$.

Figure 3.8 shows simulated $P_n(\infty)$ values for two limiting cases encountered in the modulated fluorescence experiment. The first assumes that k_{TT} is identical to k'_{ISC} then varies k_{exc} to alter the effective triplet formation rate constant, $k_{f,0}$. These results are collected in Fig. 3.8a that demonstrate the probability of a single polymer chain supporting multiple triplets increases as expected when kexc increases. At the upper limit of k_{exc} used in our experiments, there is significant likelihood of having more than two triplets per chain, which explains efficient fluorescence quenching by triplets despite inefficient singlet-triplet annihilation as expected via the Förster energy transfer mechanism. We next fixed kexc at a moderate level (5×10^6 s^{-1}) and vary k_{TT} to assess how multiple triplet configurations respond to faster triplet annihilation. Increasing k_{TT} leads to rapid depopulation of multi-triplet configurations such that all probability shifts to only one triplet is present that occurs even when k_{TT} is only slightly larger than k'_{ISC}. This situation corresponds exactly to earlier

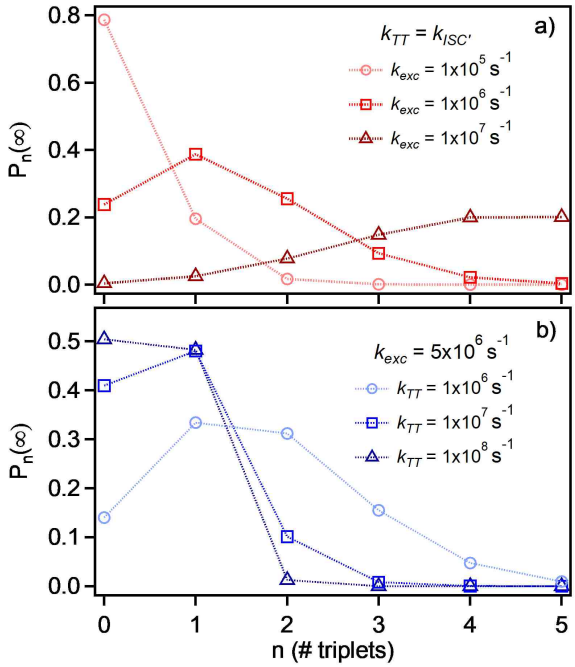

Figure 3.8: Simulated triplet occupancies (P_n) for single P3HS chains while varying kexc assuming that $k_{TT} = k_{ISC}$ (a) and varying k_{TT} while holding kexc constant (5×10^6 s^{-1}) (b).

assumptions that multiple triplets undergo rapid annihilation until only one triplet remains. These simulations illustrate that this limit is only a good approximation when intersystem crossing quantum yields Φ_{ISC} remain low (e.g., < 5%).

3.2 Loss of triplet-induced fluorescence quenching in P3HS aggregates

As shown earlier in Fig. 3.4, the concentration threshold for aggregate formation in P3HS is relatively low. In order to probe how aggregation affects triplet formation and interactions found in single chains, we measured MCA transients from P3HS dispersions with appreciable aggregation (i.e., ~10^{-8} M or higher). Figure 3.9a shows a representative MCA response from an aggregated P3HS dispersion obtained by recording the signal while raster scanning over the sample. This approach was necessary because of the lack of well-resolved fluorescence spots in images (see Fig. 3.4a). Remarkably, there is an abrupt disappearance of fluorescence intensity modulation, even for larger average excitation intensities (>100

W/cm^2). Qualitatively, this result suggests that interchain π-stacking interactions between nearby P3HS segments introduce new and efficient deactivation pathways that either bypass triplet formation, or, promote rapid annihilation or relaxation of triplets immediately after formation.[59] Because of the relatively large apparent kISC values in isolated chains, it was expected that even low triple population levels would be present regardless of the polymer aggregation state. If triplets were relaxing or annihilating more efficiently in aggregates then this scenario would correspond to the special case mentioned earlier where k_{TT} becomes infinite leading to only one triplet at any time. In addition, fluorescence intensity modulation depths would be weaker and quenching dynamics would be slower as seen in aggregates of other conjugated polymers.[30] Neither was observed in P3HS samples with substantial aggregation and we now evaluate changes in predicted $P_n(t)$ for each of these cases in an ad-hoc fashion (i.e., $k_{ISC} \sim 0$ or $k_{TT} \sim \infty$).

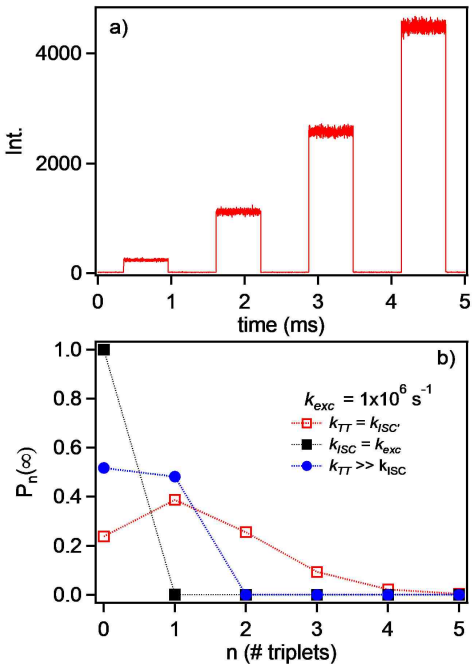

Figure 3.9: a) Representative MCA transient of an aggregated dilute P3HS solid dispersion obtained by collecting modulated fluorescence while raster scanning the modulated laser excitation beam over the film. b) Simulated triplet occupancies (P_n) using the ad-hoc approach of the stochastic triplet model (see text). Rate constants of triplet formation and annihilation were varied to assess regimes where triplet formation is bypassed because aggregation.

Figure 3.9b shows examples of predicted triplet occupancies in the regimes of hypothetical inefficient triplet formation and fast triplet annihilation in P3HS aggregates. Using the previous condition of $k'_{ISC} = k_{TT}$ (Fig. 3.8a) as a benchmark (red open squares), we find that decreasing kISC results in no triplets formed (black squares) whereas increasing k_{TT} collapses the system to the earlier approximation of only one triplet present (blue circles). The latter case seems plausible as P3HS chains aggregate since triplets would be in much closer proximity. However, as mentioned above, no evidence of triplet-induced fluorescence quenching was observed in aggregating P3HS samples, which are expected to have non-zero populations of at least one triplet per chain. To our knowledge, there is no known mechanism causing reductions of k_{ISC} over several orders of magnitude (assuming no other processes are affected by aggregation) thus bypassing triplet formation. We propose that loss of triplet-induced fluorescence modulation in P3HS aggregates probably originates from other non-radiative processes that outcompete k_{ISC}.

Previous near-infrared transient absorption and ultrafast fluorescence spectroscopy of aggregating polymer thin films (e.g., P3HT) reported no signatures of triplets.[59,106] Instead, polaronic features appeared on time scales of <1 ns that were assigned to an auto-ionization of the singlet exciton.[59,106] Paquin et al. examined this mechanism in detail using time-resolved PL spectroscopy of P3HT thin films where excitons acquire greater charge transfer character making them more susceptible for dissociation into polaron pairs.[107] If a similar mechanism becomes operative in P3HS aggregates, excitons could acquire interchain charge transfer character on time scales faster than those reported for triplet formation.[65] As mentioned earlier, no evidence of appreciable charge transfer character was observed in P3HS chains. However, it is conceivable that relaxation from these states occurs on time scales faster than our available instrument resolution (~50 ps).

Recent theoretical work has also shed light on the roles of large vibrational displacements on polaron formation occurring on ultrafast time scales[121] that could be significant for bypassing triplet formation in P3HS aggregates. Additional investigations are required to

accurately sort out how aggregation regulates excited relaxation pathways. We are presently examining the effect of solvent on P3HS aggregation, which could shed new light on the manner in which these molecules interact and the impact on excited state deactivation channels.

3.3 Conclusions

The strong tendency of P3HS chains to aggregate has important implications for the relaxation of photoexcited singlet excitons. Only by diluting P3HS to ultralow concentrations ($<10^{-9}$ M), it becomes possible to discern intrinsic photophysical properties. The large triplet formation yields in isolated chains lead to rapid fluorescence quenching on time scales <10 μs. Using a stochastic model with previously measured rate constants, we demonstrate that triplet-triplet and singlet-triplet interactions are comparable or smaller than the intersystem crossing rate constant which allows configurations of multiple triplets to exist on a single P3HS chain. However, at the onset of aggregation, triplets are either completely bypassed or quenched due to a more efficient non-radiative process likely involving short-lived polaronic species and large interchain torsional displacements. Our results demonstrate that, even when efficient intersystem crossing and large average triplet populations are present, faster decay mechanisms dominate once P3HS chains aggregate.

4.0 Resolving Anomalous Heavy Atom Effects from Discrete Triplet Mediated Photochemistry Events on Single Conjugated Polymer Chainss

4.1 Introduction

The formation of spin-forbidden, triplet excited states in conjugated organic polymeric semiconductors removes chromophores from photoexcitation cycling and eventually leads to degradation over time by sensitization of reactive oxygen species, e.g., singlet oxygen.[1] Intersystem crossing between singlet and triplet manifolds is governed by the strength of spin-orbit coupling as well as the singlet-triplet exchange energy[94] and there is considerable interest in harnessing triplet attributes, such as longer exciton lifetimes (diffusion lengths), using heavy atom substitution schemes.[122–125] However, basic design rules have relied mainly on pure electronic descriptions2 that do not explicitly account for structural factors,[126] including chain conformation and packing.[65,94] This has often led to significant discrepancies in the understanding of triplet processes between molecular- and material levels, namely, the lack of appreciable triplet populations in bulk forms despite large triplet yields at the molecular level.[44,127,128]

Single molecule fluorescence spectroscopic approaches can bridge this gap in understanding by resolving discrete triplet interactions with emissive singlet excitons in addition to triplet mediated photochemistry affecting material stability.[42,102,104] Importantly, polymer photophysical outcomes, i.e., triplet yields and triplet-related processes, are strongly influenced by polymer chain conformations and aggregation that may be further complicated by intrinsic heterogeneity.[129] For example, polymer chains in extended conformations behave as independent, multi-chromophoric systems that may support more than one

triplet at a time.[127] On the other hand, efficient excitation energy funneling to as little as one chromophore typically dominates in collapsed conformations. This behavior manifests as discrete (i.e., "on/off") intermittency in fluorescence intensity transients due to reversible quenching interactions with either reactive oxygen species or triplets.[16,37,38,130] Here, we use conformational selection schemes to confine triplet processes to as few chromophores as possible in order to resolve the impact of heteroatom substitution on triplet formation and interactions with oxygen in structurally similar polymer chains. We are particularly interested in gaining molecular perspectives of the factors governing triplet formation and efficacies of oxygen sensitization as well as subsequent interactions with polymer segments. This information is often difficult to obtain from bulk thin films yet essential for understanding and mitigating processes responsible for degradation causing performance losses in devices.

Isolated chains of poly(3-hexylthiophene) (P3HT) and poly(3-hexylselenophene) (P3HS) of similar molecular weight (~30 KDa) and regioregularity were investigated using high throughput single molecule spectroscopic imaging. Fluorescence intensity transients are constructed from each resolved spot in widefield images and time-dependent triplet occupancies are inferred from discrete blinking events. Triplets can transfer their energy to oxygen leading to reactive species that act as reversible and irreversible fluorescence quenchers (e.g., charge transfer interactions and chemical defects, respectively). The efficacies and characteristics of fluorescence quenching depend on the oxygen diffusion rate, triplet formation yields and residence times for individual chromophores, which are expected to vary significantly between P3HS and P3HT molecules due to the substantially larger spin-orbit coupling constant of the former. Because current understanding of heavy atom effects and triplet mediated oxygen photochemistry has been limited primarily to ensemble level experiments, relatively few details surrounding heterogeneity effects on triplet formation yields, relaxation dynamics and sensitization efficiencies at the molecular level are available.

At the single molecule level, fluorescence intermittency is observed in both polymers in addition to irreversible photobleaching at longer times. Interestingly, we find that P3HS

has significantly longer "on" times than P3HT chains, or, alternatively, shorter-lived "off" times. This result suggests lower average (steady-state) triplet occupancies in P3HS despite the substantially larger triplet formation yields reported earlier. Consequently, isolated P3HS molecules tend to be about 2× brighter that probably represent the higher molecular weight fraction or a distinct conformer of the sample. Although aggregation effects are known to severely reduce triplet populations[127] - which may explain longer average "on" times in P3HS - recent studies have also demonstrated that larger aggregates are much less emissive than isolated chains.[115] Together with the fact that all P3HS molecules experience similar frequencies of intermittency and photobleaching as P3HT, we rule out aggregation as the source of longer "on" times. We then take advantage of detailed statistical modeling to shed light on these unexpected behaviors using a hidden two-state Markov chain model to simulate fluorescence intensity transients and transition probabilities between "on" and "off" states. By including a transient "bleach" state that can only be accessed when the system occupies the "off" state, we are able to reproduce experimental trends very well. Comparing parameters from this effective hidden Markov chain model to expected average "on/off" times (i.e., $\langle \tau_{(on/off)} \rangle$) from a simple two-state photophysical model, we find that transition frequencies (probabilities) to, or, remaining on, the "on" are much larger in P3HS. In other words, P3HS chains do not spend as much time in the triplet precursor before returning to the ground state singlet thereby diminishing the likelihood of oxygen sensitization. As possible origins for this effect, we conjecture that conformational qualities and possibly strong interactions between the selenium heteroatom with oxygen (e.g., charge transfer complexes) could lead to significant deviations from expected large triplet populations and fluorescence quenching behavior. The former possibility was verified by preparing P3HS samples in a lower boiling point solvent that favors elongated conformations and more than one triplet per chain. Lastly, we conjecture that synthetic strategies to tune spin-orbit mixing and triplet character should also consider how fine structural qualities, such as conformation and packing, modulate triplet yields and processes critical for

determining material functionality and stability.

4.2 Results and Discussion

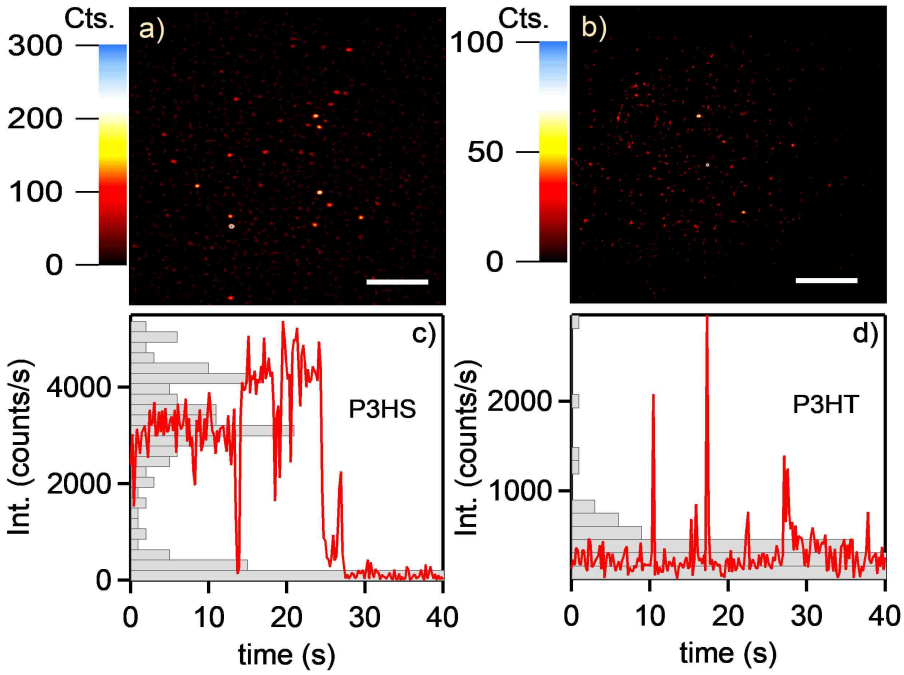

Figure 4.1: Representative fluorescence images of P3HS (a) and P3HT (b) chains dispersed in polystyrene cast from ultra-dilute DCB solutions (scale bar = 4 μm). Representative integrated fluorescence intensity transients and intensity histograms of single P3HS (c) and P3HT (d) chains.

Figure 4.1 a,b shows representative fluorescence images of P3HS and P3HT chains, respectively. Larger estimated triplet formation quantum yields and lower fluorescence quantum yields (Φ_{isc} and Φ_{fl}, respectively) for P3HS[44] initially led us to expect greater intermittency and poor photostability (i.e., long periods in dark states) compared to P3HT. However, upon inspection of images, isolated P3HS chains appeared significantly brighter on average suggesting lower triplet occupancies. Instrumental factors were ruled out on the basis of the constant CCD spectral response over the fluorescence energy range of both polymers, which are similar for dilute solutions. No significant aggregation features were observed in fluorescence spectra although P3HS displayed a small red-shifted feature pos-

sibly arising from a minority fraction of larger folded chains with substantial interchain contacts. P3HS is also much less soluble in most organic solvents than P3HT of similar molecular weight.[105] For comparison, P3HS dispersions were prepared from toluene solutions that exhibit drastically lower signal-to-noise ratios and photostability. Based on an earlier study, we found that low fluorescence counts from these samples results from selection of elongated conformations and nonzero probabilities of multiple triplets at any given time.[127] Furthermore, P3HS aggregates tend to be much less emissive due to more efficient non-radiative relaxation mechanisms that also suppress triplet formation,[127] allowing us to exclude contributions from minority aggregates as the origin of larger intensities.

Fluorescence intensity transients offer much more informative views of the structural factors responsible for differences in average intensity from images. Figs. 4.1 c,d show representative transients for single P3HS and P3HT chains, respectively, and over 90% of isolated spots displayed discrete intermittency behavior, further ruling out significant aggregation in either sample. The appearance of stochastic blinking events is consistent with emission often from as little as one chromophore per chain although ~30% of transients showed evidence of multiple emitting levels. This behavior was more prevalent in P3HS and with larger molecules (i.e., the higher molecular weight fraction) suggesting either incomplete energy funneling or, a distribution of fluorescence intensity levels for a particular chromophore. Interestingly, this feature was most apparent after emission switched back on after visiting a quenched state indicating a distribution of intensity levels for single chromophores rather than transitions between multiple emitting sites on the chain.

The relatively low fluorescence yields of both molecules also necessitated implementing a rigorous method to sort transients to exclude noise effects by comparing the standard deviation of the particle intensities to that of the background. Using established methods for resolving intensity fluctuations from noise in transients, we generate an intensity threshold for each polymer system according to, $I_{threshold} = \mu_{noise} + 6\sigma$, where μ_{noise} is the mean noise level and σ is its standard deviation. Individual idealized transient levels are com-

pared to $I_{threshold}$ and only particles above this threshold were accepted, which produces more reliable statistics by effectively removing particles that resemble background noise. We validated the sorting procedure by comparing the intensity distributions of all rejected particles below Ithreshold to backgrounds (i.e., no distinct fluorescence spots in corresponding images) and found similar behaviors.

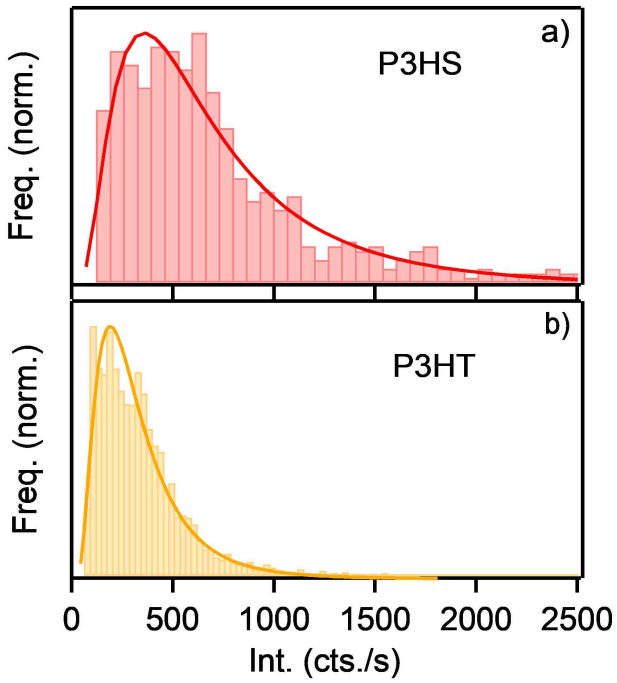

Figure 4.2: "On" intensity distributions from sorted single chain fluorescence intensity transients for over 200 P3HS (a) and P3HT (b) molecules. Distributions were fitted using a log-normal function (solid line).

From qualitative comparisons in Fig. 4.1, we now plot the intensity distributions of sorted fluorescence intensity transients that are shown in Figure 4.2. Samples are comprised of over 200 individual molecules for each polymer system confirming, on average, that P3HS exhibits significantly larger (~2×) fluorescence intensities. Although contributions from aggregates were ruled out earlier, it is possible that the larger molecular weight fraction adopts different conformations from smaller chains leading to larger average fluorescence counts. This can also be explained from the fact that larger chains are more likely to undergo chain collapse resulting in more efficient energy funneling. Similar trends were observed

in P3HT molecules of varying molecular weight but average intensities tended to saturate with very large chains.[128]

At this stage, it is more perhaps informative to examine the ensemble statistical distributions of "on" and "off" times for P3HS and P3HT molecules by plotting complementary cumulative distribution functions (CCDF, eq. 4.1) that are displayed as log-log plots in Figure 4.3.

$$P(t_{on,off}) = \left(\frac{t_{on,off}}{t_{min}}\right)^{-\alpha+1} \tag{4.1}$$

We assume that "on" intensity levels follow the log-normal distributions in Fig. 4.2 resulting in an effective two-level description. This classification was justified on the basis that multiple intensity levels usually were apparent after the particle visited the "off" state as opposed to the more characteristic signature of multi-chromophore emission where the system randomly transitions between different "on" levels. This choice of model is further justified below by comparing experimental and simulated transients.

From Figure 4.3 a, P3HS chains exhibit much longer "on" times than P3HT. By analyzing the "on/off" time distributions, it is possible to gain insight on the conformational qualities and the nature of the emitter(s). Since DCB is a high boiling point solvent, individual chains usually have ample time to adopt preferred conformations during film casting. We begin by comparing experimental "on" and "off" CCDF responses to synthesized data generated by a uniform sampling of random numbers then transforming into a normalized power law distribution[131] that showed good agreement. This behavior is consistent with single chromophore emitters, however, a general consensus surrounding accepted exponent ranges for single polymer chains is lacking. Asymptotic deviations were always apparent at longer times (see Fig. 4.2), originating from the finite transient acquisition time because of irreversible photobleaching typically occurring at >30s. Binning simulated power law "on/off" distributions with the experimental time resolution (~0.2s) produced better agreement and confirmed this feature is a systematic deviation from idealized power law behavior.[45]

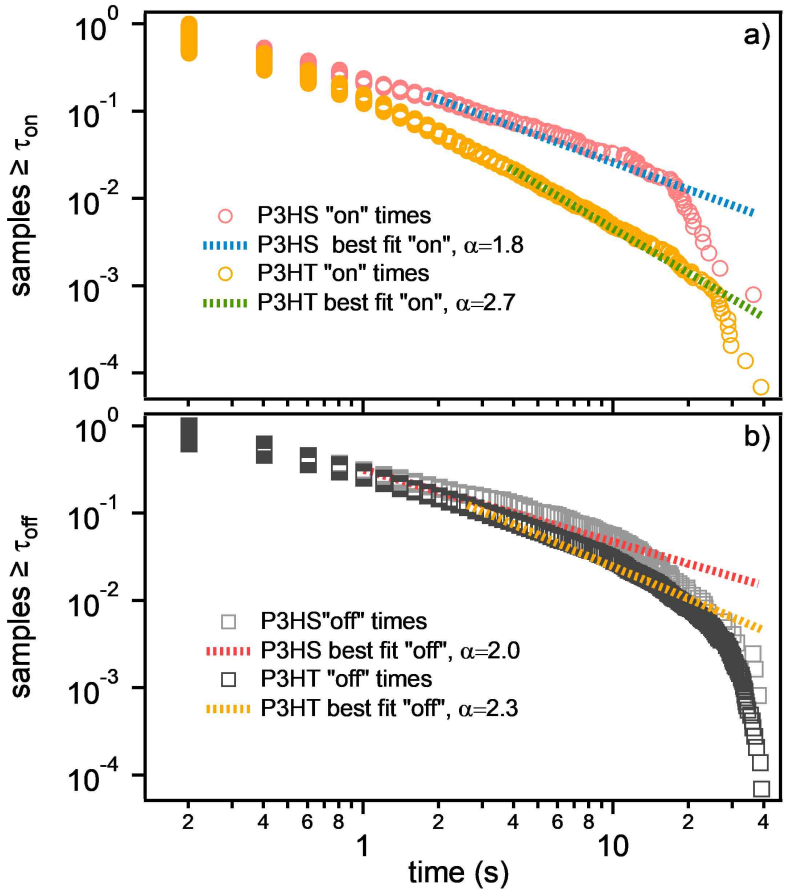

Figure 4.3: Complementary cumulative distribution functions (CCDF) for P3HS and P3HT "on" (a) and "off" (b) ensemble times from single chains . Best-fit power law curves are included for each molecule and time distribution using a procedure described in detail in the Supporting Information.

We next undertook a more rigorous analysis procedure to generate a best-fit power law model and verify the nature of emitters. First, t_{min} (the shortest time that experimental data follow ideal power law behavior[132]) was determined using Kolmogorov-Smirnoff statistics followed by a maximum likelihood estimate (MLE) approach to accurately determine α.[133,134] The goodness of fit was assessed from a null hypothesis test using a random sampling of a theoretical power law distribution. The confidence interval, p-value, was chosen to be 0.1 and, overall, experimental "on/off" time distributions agreed with the power law model (i.e., p-values were in the range of 0.4 — 0.8). Best-fit power law curves generated from this procedure are shown with experimental data in Fig. 4.3 for P3HS and P3HT

chains. Interestingly, "off" time distributions for both polymers have nearly identical exponents ($\alpha \approx$ 2.0-2.2), consistent with a similar mechanism for fluorescence quenching or, occupying the "off" state. Significantly longer average P3HS "on" times are reflected in the smaller α value of 1.8 (compared to a α value of 2.7 for P3HT) indicating a low probability for transitioning to, and, remaining in the "off" state (viz. triplet precursors). Although virtually all molecules display telltale evidence of emission from as little as one chromophore, larger experimental "on" time distributions may suggest more than one chromophore could be involved in P3HS. Because of the lower solubility of P3HS, it is possible that a fraction of chains adopt conformations different from those of smaller chains (i.e., lower average intensities). Together with the finding of larger average "on" intensities from Fig. 4.2, the most plausible explanation is that the larger P3HS weight fraction more readily exists in the collapsed form. This can be further rationalized in terms of the greater quinoidal character of P3HS that should hinder collapse of smaller chains due to greater backbone rigidity.[105] Lastly, based on the fact that all P3HS molecules undergo discrete blinking in addition to permanent photobleaching at longer times - similar to that of P3HT chains in this study – confirms the lack of macroscopic aggregates.

Unfortunately, unlike small organic dye molecules (i.e., single dipole emitters), further comparisons of statistical emission behaviors from single P3HS and P3HT chains to other conjugated polymers are much less straightforward.[135] This originates primarily from the extreme heterogeneity (i.e., molecular weight polydispersity and conformation selection depending on solvent processing) of polymer systems. For example, it is possible that a single chain may support multiple excitons and charged species (polarons) leading to a wide range of interactions that may influence statistical behaviors. These aspects complicate comparisons and interpretation of "on" and "off" behaviors, especially from the fact that most processes occur on time scales much faster than typical fluorescence transients. However, the overwhelming majority of fluorescence transient responses are indicative of single chromophore emission and we apply an effective two-state model to understand the

statistical behaviors of both P3HS and P3HT chains.

As a first step, we estimate expected average "on" and "off" times, or, transition frequencies based on known photophysical constants reported earlier for P3HS and P3HT systems. This entails a two-state, single chromophore model that assumes the molecule spends most time in the ground electronic state (S_0) and lowest energy triplet excited state (T_1). The average "on" and "off" transition frequencies are then described by:

$$\langle \tau_{on} \rangle^{-1} = \Phi_{isc} k_{exc} \tag{4.2}$$

$$\langle \tau_{off} \rangle^{-1} = k'_{isc} \tag{4.3}$$

where Φ_{ISC}, k_{exc}, and k'_{ISC} are the intersystem crossing quantum yield, excitation rate constant and reverse intersystem crossing rate constant, respectively. Φ_{ISC} values are estimated from earlier photophysical studies[44,59] and we assume similar absorption cross-sections for each polymer (viz., k_{exc}). This yields predicted $\langle \tau_{on} \rangle^{-1}$ values of ~3×10^4 s^{-1} and ~1×10^4 s^{-1} for P3HS and P3HT, respectively, that correspond to transition frequencies much faster than the fluorescence image frame acquisition rate. However, these values can serve as a useful guide to predict experimental transient behaviors, specifically, transition probability distributions (vide infra). Estimates of k'_{ISC} are taken from previous single molecule spectroscopic investigations of ground state recovery following buildup of steady-state triplet populations, which are ~3×10^5 s^{-1} and ~5×10^4 s^{-1} for P3HS and P3HT, respectively.[43,127] Similar to "on" times, average experimental "off" times for both polymers are longer than typical triplet shelving events predicted from eq. 4.3 which instead are determined from triplet-mediated oxygen interactions that quenches fluorescence via charge transfer interactions.[16,136,137] Qualitatively, P3HS transitions to the triplet manifold faster than P3HT, however, these triplets tend to relax much faster in the former thus returning the chromophore to S_0 where it can again undergo photoexcitation cycling. These trends are consistent with experimental observations of significantly shorter-lived "off" states (i.e., faster transition

from “off” states) in P3HS although these estimates do not provide a complete picture of kinetic factors responsible for experimental behaviors.

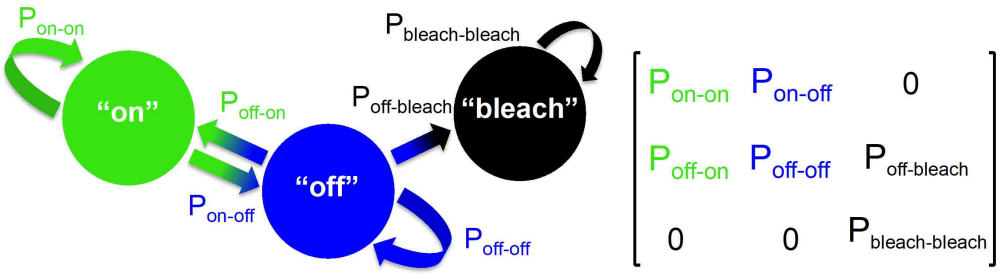

Figure 4.4: Hidden two-state Markov chain model and transition matrix describing events remaining on, and leaving “on” and “off” states. A transient “bleach” state is included that can only be accessed by transitioning from the “off” state.

Because our experiment probes triplet photochemistry under continuous illumination, it is more practical to evaluate the probabilities of transitioning to and from “on/off” states. For comparison, the simple two-state model in eqs. 4.2 and 4.3 only considers transitions to and from “on” and “off” states, e.g., $S_0 \rightleftharpoons T_1$, and does not account for oxygen-related events. Expanding the simple two-state picture is also necessary since the power law fitting analysis of experimental “on/off” time distributions in Fig. 4.3 do not explicitly assume any particular distribution of transition probabilities. Stochastic kinetic modeling was performed using a hidden two-state Markov chain model in Figure 4.4 that includes a transient “bleach” state to replicate experimental observations of irreversible photobleaching at long times. In this description, the “bleach” state may only be accessed if the system currently occupies the “off” state and transitions to the “off” state can only be distinguished from the “bleach” state if the system switches back to the “on” level. If this does not occur during the remaining course of the acquisition, it is not possible to discern if the system has transitioned to the “bleach” state or has become trapped in a longer “off” period. For these reasons, we only count events up to the last transition to a quenched level, effectively removing contributions from irreversible photobleaching to the “on/off” statistics.

To assess the validity and predictive capability of this model, we first generate theoret-

ical distributions assuming exponentially distributed probabilities for transitioning to and from "on", "off" and "bleach" states,

$$P(\tau) = \frac{1}{|\tau|} exp(-t/\tau) \tag{4.4}$$

which are transformed into a CCDF ($1 - P(\tau)$) for P3HS and P3HT "on" and "off" times. We assume the system can remain on any of the three states, transition to another state, and, lastly, become trapped on the "bleach" state. No a priori knowledge of absolute probabilities of transitioning between states is necessary as the model only assesses relative probability determined by the CCDF for each state. Probability sampling involves transitioning randomly between these distributions, which is described in the following.

We first consider transitions to either "on" or "off" states (i.e., P_{off-on} and P_{on-off}, respectively) which are inferred from the simple photophysical model in eqs. 4.2 and 4.3 and from experimental trends. Importantly, transition probabilities for remaining on either of these states are exact complements (i.e., P_{on-on} and $P_{off-off}$, respectively) and, for this reason, it is only necessary to reference transitions to the terminal state. Previous single molecule spectroscopic studies estimated oxygen diffusion encounter frequencies of ~5 s^{-1},[16,25,29] which is similar to charge tunneling time constants between the molecule and surrounding medium.[138] Because of the similarity in experimental P3HS and P3HT "off" time distributions, we fix transition probability time constants for transitioning to the "off" state (τ_{on-off}) at 0.2s suggesting that $\langle \tau_{off} \rangle^{-1}$ (eq. 4.3) is actually determined by oxygen diffusion encounter frequencies, $k_{diff}^{O_2}$. In other words, the "off" state is independent of the heteroatom (i.e., k'_{isc}) unlike observed "on" time distributions.

With transition probabilities to the "off" state fixed, transition probability time constants for transitioning to "on" state (τ_{off-on}) were adjusted to best replicate experimental trends. These values were found to be 0.005s and 5s for P3HS and P3HT, respectively. We speculate that the large discrepancy in transition probabilities to the "on" state probably originate

from the molecular structure and environment, which (similar to "off" times) were not explicitly considered in the simple photophysical model (eqs. 4.2 and 4.3). It is instructive to point out that the time constants determine the probability for transitioning to a particular state and are not directly comparable to transition frequencies between S_0 and T_1 states determined from eqs. 4.2 and 4.3. For example, the smaller value of τ_{off-on} for P3HS indicates that the likelihood of remaining in the "off" state is much lower than in P3HT consistent with experimental observations of P3HS remaining in the "on" state for much longer times. Interestingly, the mean probability (i.e., $1/\tau$) for transitioning to the "on" state is within an order of magnitude with estimated $\langle\tau_{on}\rangle^{-1}$ values in P3HS. Conversely, the discrepancy is much larger in P3HT chains that may result from stronger than expected interactions with oxygen (vide infra).

Monte Carlo simulations of fluorescence intensity transients were performed next using the Metropolis algorithm that randomly samples the theoretical "on/off" CCDFs described above (including the transient "bleach" state) to determine the Markov time step.[139] Figure 4.5 shows representative simulated transients with Gaussian noise added to replicate experimental signal-to-noise ratios. Simulated "on" intensity levels also follow the log-normal distribution shown in Fig. 4.2 that recovers the apparent multi-level behavior similar to experiment (i.e., "on" levels only change in intensity following a transition to the "off" state). Time-dependent state occupancies are superimposed alongside simulated transients that, overall, reproduce experimental intermittency behaviors reasonably well within the hidden two-state Markov model.

Although counts from molecules experiencing permanent photobleaching were not included if the system did not switch back to the "on" state during the sampling window, it was necessary to use different transition probability time constants to the transient "bleach" state for each polymer. Specifically, P3HS simulated transients required a τ_{bleach} of 0.05s whereas that of P3HT was determined to be 50s. The range in these values can be explained in terms of experimental trends and predicted transition probability distributions

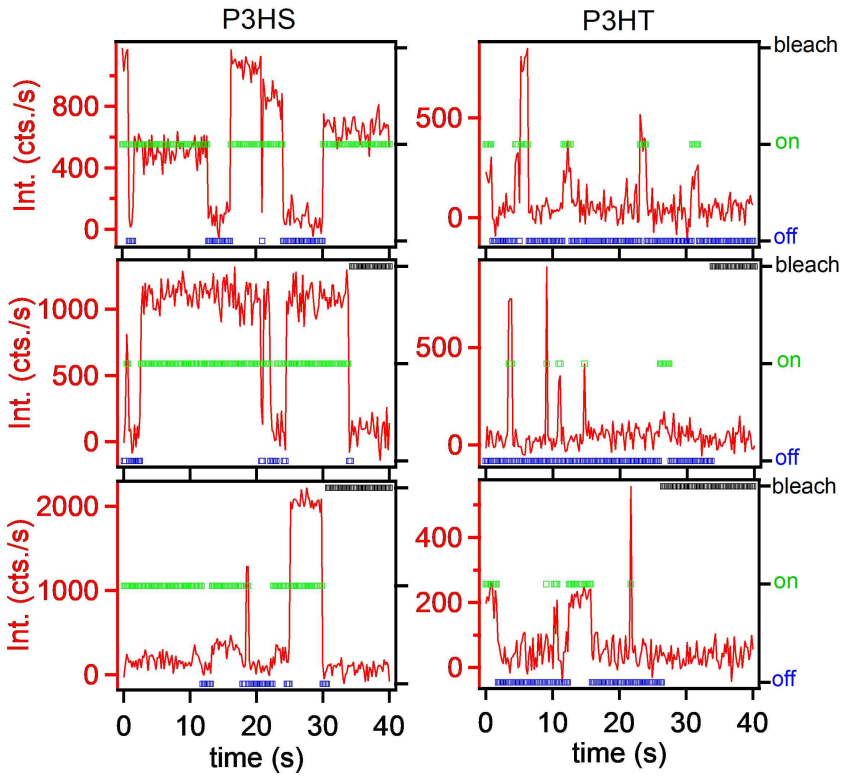

Figure 4.5: Simulated fluorescence intensity transients based on CCDF from exponentially distributed transition probabilities described in eq. 4.4 for P3HS and P3HT.Transients were simulated using a conventional Monte Carlo approach assuming that "on" intensity levels follow the empirical long-normal behavior in Fig. 4.2. State occupancies are shown on the right axis of each graph represented by different levels accessed in the course of the transient (not to scale).

for all states in the Markov chain. Because P3HS have a much smaller probability for remaining in the "off" state, a smaller relative τ_{bleach} compared to τ_{off} implies a greater likelihood of transitioning to the photobleached state upon entering the "off" state. On the other hand, shorter "on" times observed in P3HT result in a greater likelihood that the chromophore should enter the "bleach" state (assuming τ_{bleach} are the same) since it spends a greater amount of time in the "off" state. However, this scenario implies that all P3HT molecules undergo irreversible photobleaching immediately upon transitioning to the "off" state, which is rarely observed experimentally. Instead, a larger value of τ_{bleach} increases the probability that the system will make return visits to the "on" state albeit short-lived. It is also important to point out that the ratios of τ_{bleach}/τ_{on} are the same for both polymers

meaning an equal likelihood of transitioning to either state from their respective "off" state. This behavior is more easily envisaged by evaluating transition probabilities directly from transients.

Using the same Markov chain model in Figure 4.4, we now recast experimental and simulated transients into probability distributions by generating the transition matrix for each molecule by counting events above ("on") and below ("off") the characteristic intensity threshold defined earlier. This approach also ensures the two-state description is maintained by assuming "on" intensities are distributed via the log-normal function (Fig. 2). We sample over 200 simulated and experimental P3HS and P3HT transients each and Figure 4.6 shows histograms of probabilities for "on" (a, c, respectively) and "off" (b, d, respectively) transitions for each polymer. Similar to power law distributions in Fig. 4.3, experimental "off" P3HS and P3HT distributions are nearly identical, although slight deviations appear in simulated distributions (i.e., probabilities are shifted to lower average values). Nonetheless, the excellent agreement between experimental and simulated "off" times effectively confirm our hypothesis that transitions to this state follow the same mechanism and thereby only depend on oxygen encounter frequencies. The model correctly predicts that simulated and experimental "on" distributions vary greatly between P3HS and P3HT as identified previously from the power law distributions in Fig. 4.3. These results confirm that longer P3HS "on" times at low excitation intensities originate from unexpectedly low triplet occupancies thereby minimizing the likelihood of interactions with oxygen.

Unexpectedly low P3HS triplet yields may stem from formation of charge transfer complexes with oxygen that should alter photophysical pathways. This mechanism was invoked earlier to explain excited state deactivation and degradation mechanisms in solid P3HT films in the presence of oxygen.[140,141] It is important to stress, however, that bulk P3HT thin films contain a large fraction of semi-crystalline aggregates (i.e., π-stacked chains) with negligible triplet populations.[48,59,128] We had considered earlier that the lower solubility of P3HS[105] could result in a small amount of aggregates but, this was ruled out on

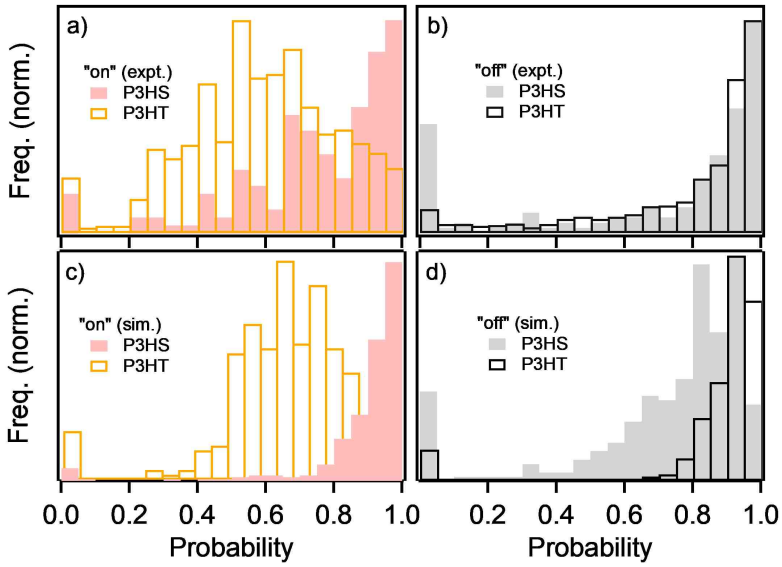

Figure 4.6: "On" and "off" probability distributions from over 200 P3HS and P3HT simulated (a, b, respectively) and experimental (c,d, respectively) fluorescence intensity transients assuming a hidden two-state Markov chain model (see Figure 4.4) including a transient "bleach" state.

the basis of discrete fluorescence intermittency behavior and irreversible photobleaching occurred in both P3HS and P3HT chains on similar time scales. Another intriguing possibility involves possible differences in the polymer chain conformational state, in particular, the greater quinoidal character of P3HS favors greater backbone rigidity and hinders chain collapse. This scenario could result in incomplete excitation energy funneling and a greater likelihood of multi-chromophore interactions in addition to exciton (i.e., triplet-triplet) annihilation becoming operable.[56] At the present, it is not possible to reliably discern these multi-particle interactions, however, good agreement is obtained between experimental and simulated transition probabilities based on the hidden two-state Markov chain model which confirms that emission is probably dominated by single chromophores with a log-normal intensity distribution. It is possible that P3HS and P3HT interact differently with oxygen which may explain the large disparities between transition probabilities of "on" states. Earlier studies have demonstrated substantial, but, non-bonding, interactions between oxygen and sulfur and selenium based heterocycles.[142] However, we are not able to speculate how

such interactions might be affected by polymer conformations. We are currently investigating fluorescence blinking of these molecules in different environments (i.e., oxygen depleted environments) but, signal-to-noise ratios are usually lower because triplets are no longer quenched by energy transfer to oxygen.

4.3 Conclusions

By confining triplet processes on structurally similar polymer chains, we have demonstrated anomalous heavy atom effects that are not easily resolved from conventional ensemble experiments. These factors are not usually considered in materials design schemes for tuning spin-orbit coupling despite their importance for determining material functionality and stability. This was most apparent from comparisons of predictions from the effective two-state Markov chain model that, despite showing good agreement with experiment, are not consistent with expectations from photophysical constants measured at the ensemble level. Polymer chain conformation characteristics may in fact play a more important role than expected as revealed from the statistical behavior of fluorescence intensity transients. For this reason, a comprehensive model accounting for multi-particle interactions could perhaps reconcile experimental trends with those predicted from simpler single chromophore, two-state model used here. Lastly, our results offer some potentially encouraging perspectives into the basic understanding of material degradation processes via triplet-induced oxygen sensitization, namely, faster ground state recycling diminishes the likelihood of forming defects that permanently alter material performance in optoelectronic devices.

5.0 Population dynamics of multiple triplet excitons revealed from time-dependent fluorescence quenching of single conjugated polymer chains

5.1 Introduction

Conjugated organic polymers have demonstrated promise in solar cell applications but extreme heterogeneity and efficient intrinsic loss mechanisms,[143,144] such as rapid non-radiative excitation energy dissipation, are responsible for large disparities between measured and predicted efficiencies. There is now widespread interest for mitigating performance losses by generating multiple excitons per photon absorbed or extending exciton lifetimes.[3,125,145] Singlet fission – the generation of two triplet excitons from one singlet exciton – and heavy atom substitution to increase triplet character, respectively, have attracted the most attention,[3,99,101,146–149] but applications to conjugated polymers are limited.[150,151] In fact, mechanistic studies of singlet fission have concentrated on small molecule arrays with well-defined chromophore orientations. Furthermore, efforts to increase triplet exciton character and tune triplet interactions in polymers may be complicated by contributions from large vibrational displacements along high frequency modes that modulate spin-orbit coupling strength.[152]

Perhaps the most significant obstacle for effectively utilizing multi-exciton generation and harvesting strategies can be traced to the multi-chromophoric nature of polymers (i.e., many conjugated segments of varying length) and variable inter-chromophore coupling due to conformational heterogeneity.[153] For example, the longer lifetimes of triplets creates complex photophysical scenarios due to the presence of multiple excitonic states of different spin on many chromophore segments that interact over a broad range of time scales

(e.g., ~ 10^{-12} - 10^{-3} s). Interestingly, previous pulse radiolysis studies found that isolated conjugated polymer chains can support a large number of triplets (~30) simultaneously.[63] While much of the current focus has emphasized elucidating triplet formation mechanisms on ultrafast time scales, relatively little is known about how populations of multiple triplet excitons evolve on longer time scales (i.e., comparable to triplet lifetimes).

Unfortunately, resolving multi-excitonic interactions at the materials level in polymers is complicated from a myriad of competing decay channels arising from intermolecular interactions and aggregation.[150,154] However, by dispersing polymers into inert glassy hosts, intermolecular interactions and packing heterogeneity are negated in addition to spatially confining excitonic kinetics and interactions to a single polymer chain (SPC). Single molecule spectroscopy can then be used to interrogate triplet population dynamics and interactions with emissive singlets by monitoring fluorescence quenching on nanosecond to millisecond time scales.[136,155–157] However, much of the earlier investigations of excitonic processes at the SPC level focused on energy transfer within the singlet manifold where funneling between chromophore sites typically dominates responses.[16,19,38,158,159] This regime is most prevalent when yields of spin-forbidden triplet excitons are small (<10%) although the presence of even one triplet can have significant consequences at the SPC level.[1,42,160] Triplet interactions with emissive singlets are commonly inferred from intermittency behavior of SPC fluorescence intensity transients in the form of flickering (i.e., fast cycling between "on" and "off" intensity levels) on millisecond time scales[161] or blinking behavior due to sensitization of reactive oxygen species occurring on time scales of seconds.[141] More specialized single molecule spectroscopic tools have proven effective for exposing triplet interactions on faster time scales, such as, fluorescence correlation spectroscopy[104] and excitation intensity modulation spectroscopy.[26] While these approaches can access population dynamics of excitonic configurations, kinetic models describing discrete excitonic interactions are much more complicated when multiple triplets are involved. For this reason, it is generally assumed that rates describing multi-excitonic triplet configurations and processes

(i.e., triplet-triplet annihilation) reach a steady state condition immediately following photoexcitation.[42] In other words, triplet diffusion or the triplet-triplet annihilation rate constant are assumed to be infinitely fast leading to only one triplet at a time. This limit effectively reduces singlet-triplet interactions to a simple two-state description where the system spends most time in either the lowest energy triplet (T_1) or ground electronic state (S_0)[42] although crucial details of multi-exciton interactions are lost.

We use single molecule excitation intensity modulation spectroscopy to probe triplet induced fluorescence quenching dynamics in SPCs of single poly(3-hexylthiophene) (P3HT) and poly(3-hexylselenophene) (P3HS), which have reported triplet yields exceeding 30%.[44,162] Although it is assumed that triplet formation follows a conventional perturbative mechanism, these systems are excellent models for understanding the full implications of heavy atom substitution at the SPC level in addition to resolving interactions between multiple triplets and triplet population dynamics. Quenching behavior is modeled by calculating the time-dependent probabilities of $n = 0, 1, 2, \ldots nth$ triplet population dynamics using a stochastic photodynamic model based on the Smith-Ewart differential difference equation originally developed to describe polymerization/emulsion kinetics.[54,163–166] We use the basic formalism of Barzykin and Tachiya[56] and employ the approach of Birtwistle and coworkers[165] to discretize and solve the Smith-Ewart model using the Gauss-Seidel iterative approach. Unlike effective two-state models mentioned earlier, this model incorporates finite triplet diffusion and triplet-triplet annihilation rate constants resulting in nonzero probabilities of multiple triplets at the SPC level. Simulations of triplet population and fluorescence quenching dynamics of P3HS and P3HT SPCs show good agreement with experiment when triplet-triplet annihilation rate constants are comparable to the natural first order triplet decay rate constant. Interestingly, despite larger reported triplet formation and decay rate constants in P3HS,[44] we observed larger steady-state probabilities of multiple triplets in P3HT chains. This unexpected result highlights the importance of accounting for all kinetic factors regulating population dynamics of multiple triplets that are difficult to

obtain from ensemble level studies. Furthermore, singlet-triplet interactions were found to be much stronger in P3HS probably from substantial red-shifts in singlet exciton electronic transitions due to the heavier selenium heteroatom![105] leading to improved spectral overlap between singlet donors and triplet acceptors. Overall, we demonstrate a robust and informative method for resolving the evolution of multiple triplet excitons to help bridge the gap in understanding of population dynamics on longer time scales.

5.2 Results and Discussion

Single molecule excitation intensity modulation spectroscopy was used to interrogate the presence of multiple interacting triplet excitons. Figure 5.1a shows a representative fluorescence image of well-dispersed P3HS SPCs in polystyrene. Importantly, intrinsically low fluorescence quantum yields (~10^{-3}) due to efficient triplet formation lead to low signal-to-noise ratios and individual spots often show 'streaking' due to singlet quenching by triplets. SPC fluorescence emission and quenching dynamics are resolved by exciting individual molecules using sequences of rectangular shaped laser pulses displayed in Fig. 5.1b where pulse characteristics are tailored according to the expected triplet lifetimes.[26,127] Like other fluorescence-based single molecule probes, this technique relies on a loss of signal as an indicator of singlet-triplet interactions which, in the case of the target materials, can be very large leading to rapid quenching and low steady-state intensities. This effect can be seen in Fig. 5.1c which shows representative triplet-induced fluorescence quenching behavior of a single P3HS molecule in addition to average decay times and modulation depths from over 40 SPCs (inset). When the laser first turns on (t_0) no triplets are present but, over time, triplet populations increase causing fluorescence quenching via singlet-triplet energy transfer (annihilation). Average triplet populations are also affected by intrinsic decay (i.e., reverse intersystem crossing) and triplet-triplet annihilation that depend on SPC structural and electronic factors.[1,98] Taken together, these processes cause the initial intensity, $I(0)$, to decay to a steady-state level, I_{ss}, and quenching depths can serve as a useful metric

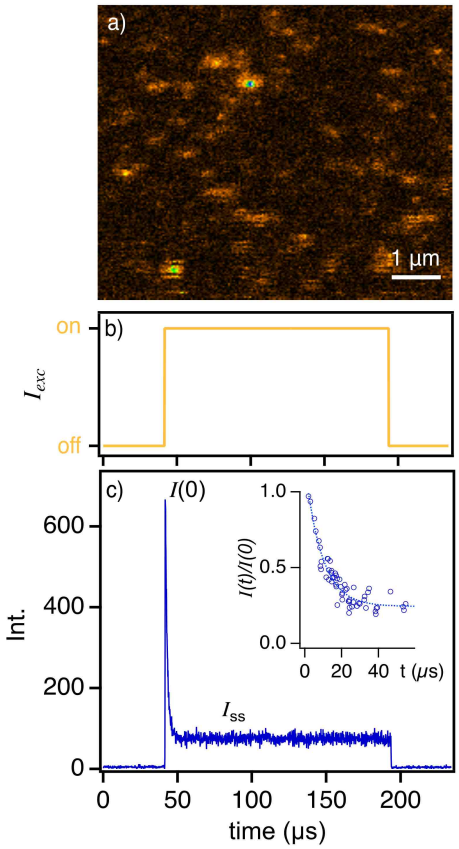

Figure 5.1: a) Representative fluorescence image of SPCs dispersed in polystyrene matrices. b) Rectangular laser excitation pulse waveform used to excite fluorescence in single polymer chains (top). c) Example of fluorescence quenching dynamics in a single polymer chain with multiple triplets. Immediately after the laser turns on, intensities begin at an initial value, $I(0)$), then, as triplet occupancies increase, decay to a non-zero steady state value, I_{ss}, usually within or faster than the triplet lifetime. Inset: Histogram of quenching depths and decay times from over 40 P3HS molecules with an exponential decay fit as a guide for the eye.

of both time-dependent triplet occupancies and the strength of singlet-triplet interactions. For ease of comparison with simulated quenching curves (vide infra), we report fluorescence quenching depths as the fraction of quenched fluorescence normalized to $I(0)$, i.e., $I(t)/I(0)$.

Because of the large variation in responses from SPCs of both polymers,[102,127] we used a modified version of excitation intensity modulation technique described earlier[102] where two rectangular pulses of the same duration and intensity are temporally delayed and $I(0)$

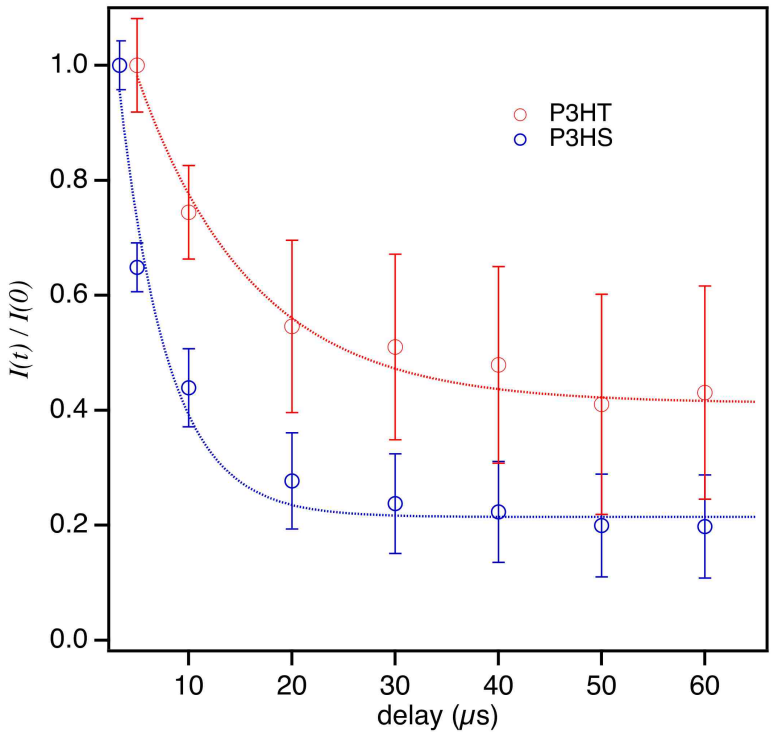

Figure 5.2: Experimental fluorescence quenching behavior of P3HS and P3HT comprised of ensemble averaged SPC data from a variable delay two-pulse approach recorded for various delay time intervals. Quenching dynamics are represented as $I(t)/I(0)$ curves.

values of each are recorded with pulse delay time and converted into $I(t)/I(0)$ curves. Figure 5.2 shows experimental single molecule fluorescence intensity quenching data obtained from P3HS (blue) and P3HT (red) single molecules of similar molecular weight prepared under the same conditions. The first pulse achieves a steady-state triplet population condition and the second pulse excites the SPC before triplet populations fully relax. Varying the time delay between the two pulses reveals triplet relaxation dynamics of SPCs that are comparable to quenching decay times directly from transients from low excitation intensities (e.g., Fig. 5.1b). Responses in Fig. 5.2 are averaged over many (>40) SPCs of each polymer providing a better comparison to ensemble level measurements in addition to further exposing the role of the heteroatom on triplet formation efficiencies and singlet-triplet and triplet-triplet interactions. Average quenching depths of P3HS and P3HT SPCs range from ~60% to ~80% of $I(0)$, respectively, (e.g., corresponding to Iss values of ~0.4 and

~0.2, respectively). The larger average quenching depths and faster quenching dynamics of P3HS SPCs are consistent with larger triplet occupancies and singlet-triplet quenching rates but, it is not possible to ascertain the relative magnitudes of associated rate constants by inspection alone. The former depend on triplet formation yields as well as the triplet and fluorescence lifetimes, however, it is not straightforward to directly infer relative contributions of triplet-triplet annihilation from $I(t)/I(0)$ curves.[56,167]

Barzykin and Tachiya[56] had previously overcame the limitations imposed by assumptions of infinitely fast triplet-triplet annihilation by introducing an expanded stochastic photodynamic model to simulate the time-dependent evolution of multiple triplet populations from fluorescence intensity modulation data. Importantly, these authors only considered quenching behaviors from conjugated polymers with relatively low triplet yields (<10%) and, consequently, smaller populations of multiple triplet exciton configurations. We now simulate population dynamics of multiple triplets and their interactions with emissive singlets by adopting a similar approach as Barzykin and Tachiya, which benefits from the fact that triplet decay kinetics are orders of magnitude slower than singlet decay time scales.[56] The model accounts for the existence of multiple chromophores where each can occupy either S_0, S_1 or T_1 states at any given time where only the numbers of each state determine the overall configuration and not the specific location on the SPC.[56] The kinetic scheme for triplet formation and decay is given as follows for n triplets on an SPC that follows the same format as the original Smith-Ewart model,

$$\text{SPC}_n \xrightarrow{k_{f,n}} \text{SPC}_{n+1} \tag{5.1}$$

$$\text{SPC}_n \xrightarrow{nk_b} \text{SPC}_{n-1} \tag{5.2}$$

$$\text{SPC}_n \xrightarrow{\frac{1}{2}n(n-1)k_{TT}} \text{SPC}_{n-2} \tag{5.3}$$

where $k_{f,n}$ is the forward rate constant for generating a triplet on an SPC with n triplets. It is assumed that triplet formation and decay mechanisms are governed by intersystem crossing via spin-orbit singlet-triplet mixing although the specific triplet generation mechanism is inconsequential. Now, the full form of $k_{f,n}$ is,

$$k_{f,n} = \frac{k_{exc} k_{ISC} \tau_{fl}}{1 + k_{ISC}\tau_{fl} + n k_{QST} \tau_{fl}} \tag{5.4}$$

where k_{exc} is the excitation rate constant, k_{ISC} is the intersystem crossing rate constant, τ_{fl} is the fluorescence lifetime, and k_{QST} is the singlet-triplet quenching rate constant. Triplets decay to the S_0 state via first order reverse intersystem crossing which is given by $k_b (= k'_{ISC})$, or, from triplet-triplet annihilation described by k_{TT}, the pseudo first order rate constant in the Smith-Ewart description. We use reported values for k_{ISC}, τ_{fl} and k'_{ISC} measured in dilute solutions or in solid dispersions which are held invariant to extract k_{TT} and k_{QST} estimates from fluorescence quenching curves. P3HS has much larger reported k_{ISC} values of 3×10^{10} s^{-1} compared to P3HT of ~1×10^{9} s^{-1} due to larger singlet-triplet spin-orbit mixing.[44,59,168] Likewise, fluorescence lifetimes in P3HS are ~26 ps that were estimated from singlet exciton lifetimes from stimulated emission decays as well as from lifetimes measured at the SPC level.[44,127] Reported τ_{fl} values in solvated P3HT chains are ~500 ps but care must be taken to ensure no appreciable aggregation exists resulting in a larger contribution from a fast decay component associated with torsional relaxation within aggregate π-stacks.[91] Triplet lifetime ($1/k'_{isc}$) estimates were generated from previous single molecule intensity modulation investigations yielding values of ~2×10^{5} s^{-1} and ~1×10^{4} s^{-1}, for P3HS and P3HT, respectively.[102,127]

The time-dependent probability of n triplets, $P_n(t)$, can now be obtained by solving the

Smith-Ewart differential difference equation,[163,164,166]

$$\frac{d}{dt}P_n(t) = k_{f,n-1}P_{n-1}(t) - \left[k_{f,n} + nk_b + \frac{1}{2}n(n-1)k_{TT}\right]P_n(t) \\ +k_b(n+1)P_{n+1}(t) + \left[\frac{1}{2}(n+1)(n+2)k_{TT}\right]P_{n+2}(t) \tag{5.5}$$

And the time-dependent fluorescence intensity, I(t), is expressed as,[56]

$$I(t) = I(0)\sum_{n=0}^{\infty}\frac{P_n(t)}{1+nk_{QST}\tau_{fl}} \tag{5.6}$$

where $I(0)$ is the initial (normalized) intensity ($t = 0$). Details for solving eq.5.5 are provided in and Figure 5.3 shows a diagram describing the probabilities and transition rates involving zero (a), one (b), and two (c) triplets on an SPC. Three principle rates drive the formation of any state, namely, i) gain of one triplet from intersystem crossing, ii) loss of one triplet from reverse intersystem crossing, and iii) loss of two triplets through annihilation. For any number of triplets, each probability is governed by the same principle rates. At $t = 0$ (i.e., when the laser turns on), the probability of having zero triplets decreases due to a constant forward (zero order) rate constant, $k_{f,0}$. As triplet occupancies increase, new rates of loss processes become important. In addition to population losses through reverse intersystem crossing, which varies linearly with the current number of triplets, triplet-triplet annihilation becomes operative for more than one triplet proportional to the square of the current occupancy. Importantly, this model implies that, if probabilities of any two states are equal and non-zero, the current state ($P_n(t)$) will always gain infinitely more triplets from annihilation of a higher state (i.e., $P_{n+2}(t)$) than losing triplets from annihilation to a lower state, (i.e., $P_{n-2}(t)$).

It is first instructive to consider a useful approximation where $k_{f,n}$ is constant ($k_{f,n}$ ~$k_{f,0} = k_{exc}k_{ISC}\tau_{fl}$). This regime enables estimates of $P_n(t)$ at long times, or, $P_n(\infty)$ without the need to solve the Smith-Ewart equation exactly,[56] which is helpful for estimating steady-state multi-triplet populations using only known photophysical constants

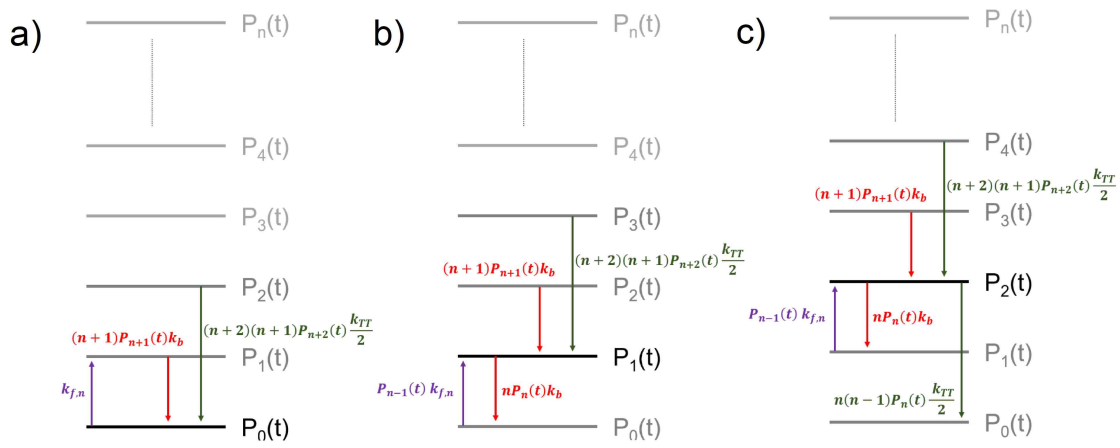

Figure 5.3: Schematic representation of the model showing the influence of the three principle rates on the probability of zero, one, and two triplets in the conjugated polymer, panel a, b, and c respectively.

(vide supra). In an earlier study of P3HS SPCs, we used this assumption and found non-zero probabilities of more than one triplet on an SPC when k_{TT} was similar to k'_{ISC}.[127] These preliminary estimates of steady-state triplet populations revealed that the assumption of infinitely large k_{TT} is not valid in SPCs with large triplet yields (large k_{ISC}). This result can be checked easily by using the condition $k_{TT} >> k'_{ISC}$. For example, rapid triplet-triplet annihilation to just one triplet on an SPC requires unrealistically large values of k_{QST} to achieve $I(t)/I(0)$ comparable to experiment.[56] In order to accurately reproduce experimental fluorescence quenching behavior, the Smith-Ewart differential difference equation must be solved numerically to calculate $Pn(t)$. We applied the general method of Birtwistle et al.[165] to solve the Smith-Ewart model and estimate k_{TT} and k_{QST} with significant nonzero multi-triplet populations.

Unfortunately, the difficulty in extracting reliable values of k_{TT} in either P3HS or P3HT at the ensemble level usually due to the disappearance of triplet signatures or ambiguous dynamics[59] necessitates first exploring realistic ranges of possible values. This was accomplished by varying k_{TT} and comparing calculated quenching behavior to experiment for fixed values of k_{QST}. Figure 5.4 shows $I(t)/I(0)$ behaviors for varying k_{TT} at fixed k_{QST} values referenced to the reported fluorescence lifetimes, τ_{fl}. Estimates of k_{QST} from

other polymers at the SPC level were typically in the range of ~10^8-10^{10} s^{-1} which provides an additional benchmark for establishing realistic limits of this parameter.[42] We imposed a lower limit on k_{TT} corresponding to k'_{ISC} for each polymer and horizontal arrows in Fig. 5.4 depict the expected range of k_{TT} based on averaged experimental steady-state intensities (I_{ss}) values of ensemble quenching curves in Fig. 5.2. Comparison of quenching behaviors reveals a larger sensitivity to k_{TT} in P3HS based on experimental quenching depths indicating stronger singlet-triplet interactions. It is also interesting to note that triplet lifetimes decrease drastically with increasing triplet densities due to faster triplet-triplet annihilation.[63] Although the presence of multiple triplets in P3HT and P3HS is apparent, it is doubtful that the relatively low excitation intensities used here (ca. 10-100 W/cm^2) ever enter regimes where the bimolecular annihilation process vastly exceeds linear triplet decay mechanisms.

The results also reveal that an increase in k_{QST} accompanies increases in k_{TT} values due to the fact that fewer triplets are present. Furthermore, we show in the following that experimentally measured quenching depths and dynamics place fundamental limits on k_{QST} that support our assumption of k_{TT} is comparable to k'_{ISC}. It is also interesting to note that when k_{QST} is small in both polymers ($k_{QST} * \tau_{fl} < 10^{-2}$), $I(t)/I(0)$ behaviors are practically invariant of k_{TT} indicating no interactions between singlets and triplets regardless of their occupancies. This regime could represent the case when SPC conformations are extended and low excitation intensities (i.e., small k_{exc}), thus requiring excitons to diffuse over longer distances, and triplet lifetimes are short compared to τ_{fl}. Additionally, if τ_{fl} is relatively small (e.g., P3HS), larger triplet occupancies or singlet-triplet interactions (viz. k_{QST}) are needed to produce appreciable quenching.

Using the assumption, $k_{TT} = k'_{ISC}$, k_{QST} is next varied over several decades to assess the sensitivity of singlet-triplet interactions and their effect on $I(t)/I(0)$. Figure 5.5 shows simulated fluorescence quenching depths and, from inspection, larger k_{QST} are required for P3HS consistent with stronger singlet-triplet interactions. This effect can be rationalized by the fact that the heavier selenium atom causes significant red-shifting of singlet electronic

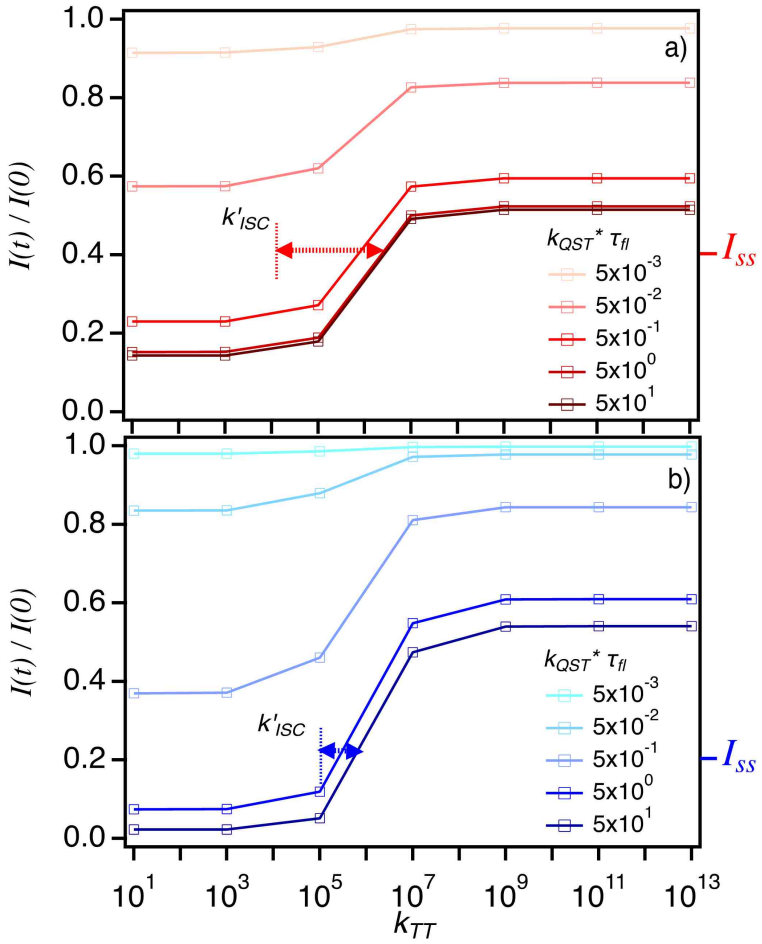

Figure 5.4: a) Fluorescence intensity quenching of P3HT (a) and P3HS (b) calculated from assuming a constant product of the singlet-triplet quenching rate constant (kQST) while varying the triplet-triplet annihilation rate constant (k_{TT}). We assume a lower limit of the latter by referencing to the reverse intersystem crossing rate constant (k'_{ISC}). Arrows represent the likely range of k_{TT} values at the average steady state intensity (I_{ss}).

transitions[105] which should result in better spectral overlap between singlet donor emission and triplet acceptor absorption and, consequently, larger k_{QST}. Furthermore, based on experimental $I(t)/I(0)$ curves, P3HS displays substantially larger quenching depths which implies that k_QST must be sufficiently large compared to the excited state lifetime. Additional insights for validating the choice of k_{TT} and k_{QST} values can be obtained from triplet formation quantum yields (Φ_{ISC}). According to experimental estimates of k_ISC and τ_{fl} from previous ensemble measurements provided earlier and in Table 5.1, Φ_{ISC} values for P3HT are ~0.5 compared to P3HS of ~0.8 leading to the expectation of larger triplet

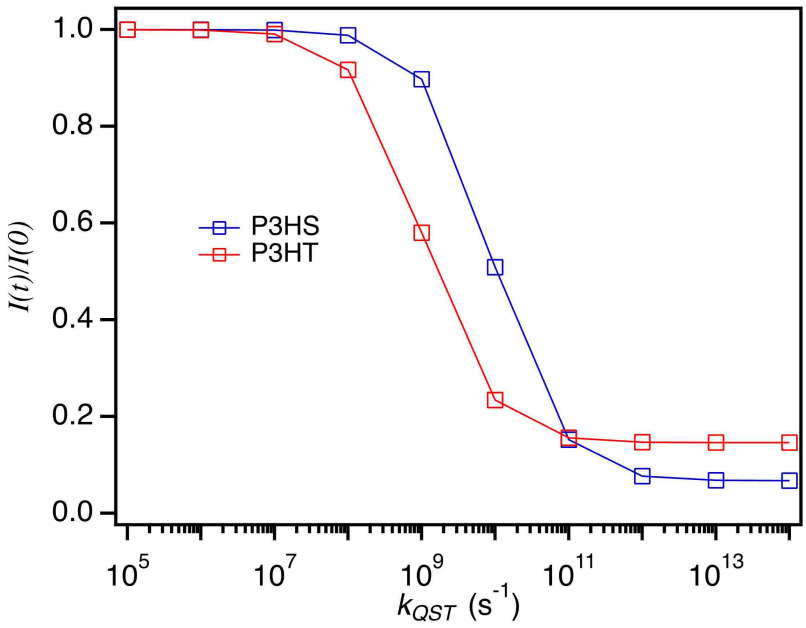

Figure 5.5: Fluorescence intensity quenching of P3HS and P3HT calculated from the assumption of $k'_{ISC} = k$ while varying k_{QST}.

occupancies in the former. However, triplet decay by reverse intersystem crossing back to S0 and triplet-triplet annihilation is faster in P3HS, reflected in estimates of k'_{ISC} of 2×10^5 s^{-1} compared to 1×10^4 s^{-1} of P3HT.[127] Rapid triplet population decay from larger triplet-triplet annihilation and reverse intersystem crossing rate constants in P3HS drastically reduce triplet occupancies despite their larger formation yields, requiring larger values of k_{QST} to achieve fluorescence quenching depths comparable to experiment. We now sim-

Table 5.1: Fluorescence quenching simulation parameters for P3HT and P3HS.

Parameter	P3HT	P3HS
k_{exc} (s^{-1})	1×10^6	1×10^6
k_{ISC} (s^{-1})	1×10^9	3×10^{10}
k'_{QST} (s^{-1})	1×10^4	2×10^5
k_{QST} (s^{-1})	5×10^8	8×10^{10}
k_{TT} (s^{-1})	1×10^4	2×10^5
τ_{fl} (s)	500×10^{-12}	26×10^{-12}

ulate time-dependent fluorescence quenching ($I(t)/I(0)$) behavior and triplet populations for P3HS and P3HT SPCs using eqs. 5.5 and 5.6 based on estimates of k_{TT} and k_{QST} from Figs. 5.4 and 5.5. Table 5.1 summarizes all parameter values used in the simulations and results are displayed in Figure 5.6 and compared to experimental data in Fig 5.2. Values of k_{exc} were generated assuming an absorption cross-section of ~10^{-15} cm^2 for each polymer with an excitation power density of ~10 W/cm^2 at 568 nm. To facilitate comparison between polymers we plot I(t)/I(0) curves by multiplying the simulation time step by k'_{ISC} for each polymer in addition to displaying fluorescence quenching with time. Similar to experiment, P3HS shows faster quenching dynamics and larger quenching depths as expected for larger k_{QST} and multiple triplets. As predicted in Figs. 5.4 and 5.5, shallow and slower quenching in P3HT can result from either inefficient singlet quenching by triplets (lower k_{QST}) or faster triplet-triplet annihilation as well as shorter-lived singlet exciton (S_1) states. Because measured τ_{fl} values are over an order of magnitude larger in P3HT compared to P3HS, the latter possibility can be easily ruled out. Earlier work on SPCs containing only light atoms reported k_{QST} values within the same range as generated here although these studies also assumed infinitely fast k_{TT}. Since this latter regime allows only one triplet quencher at a time, larger k_{QST} are needed to achieve substantial quenching depths and faster quenching dynamics. For example, comparison of quenching behavior between earlier studies and ours indicate that exceptionally large k_{QST} values (e.g., >10^{12} s^{-1}) would be necessary to produce observed quenching depths found in P3HS and P3HT chains. This observation confirms that triplet-triplet annihilation must be held finite in systems with larger triplet yields and occupancies.

Time-dependent triplet occupancies for $I(t)/I(0)$ curves (Fig. 5.6) are next shown in Figure 5.7 depicting the evolution of multiple triplet configurations over the excitation pulse duration (~100 μs). Comparing triplet population dynamics to quenching behaviors reveals some interesting trends that are not immediately obvious from reported photophysical constants. First, steady-state triplet populations are much larger in P3HT despite having lower

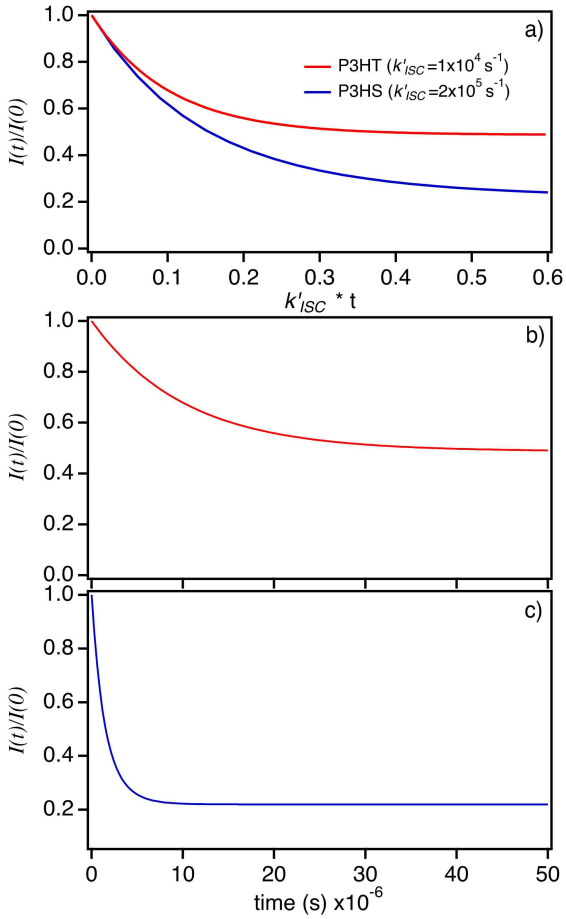

Figure 5.6: Fluorescence quenching of P3HS and P3HT SPCs. a) $I(t)/I(0)$ plots scaled to k'_{ISC} for each polymer. Time-dependent quenching depth curves for P3HT (b) and P3HS (c).

Φ_{ISC} values. This effect arises from slower triplet-triplet annihilation and reverse intersystem crossing as well as the longer excited state lifetimes. The shallow quenching depths and slower dynamics also indicates that singlet-triplet interactions are significantly weaker in P3HT which, assuming a resonant energy transfer mechanism, is probably from lower spectral overlap compared to P3HS. Secondly, Fig. 5.7 reveals that P3HS triplet occupancies reach a steady-state condition much faster than P3HT. Calculating $P_n(t)$ – allowing for up to 15 triplets – reveals that P3HS triplet dynamics are complete by ~$3 \pm 1\mu s$ compared to ~$25 \pm 6\mu s$ in P3HT resulting in steady-state triplet populations ($P_n(\infty)$) of ~2 ± 2 triplets for P3HS compared to ~4 ± 3 triplets on P3HS chains. The lower values of $P_n(\infty)$ from

P3HS chains are a consequence of both faster triplet relaxation and triplet-triplet annihilation for multiple triplet configurations coupled with a very short effective excited state lifetime. In contrast, weaker singlet-triplet interactions, slower triplet-triplet annihilation and longer-lived excited states in P3HT result in larger triplet occupancies despite having lower triplet formation yields.

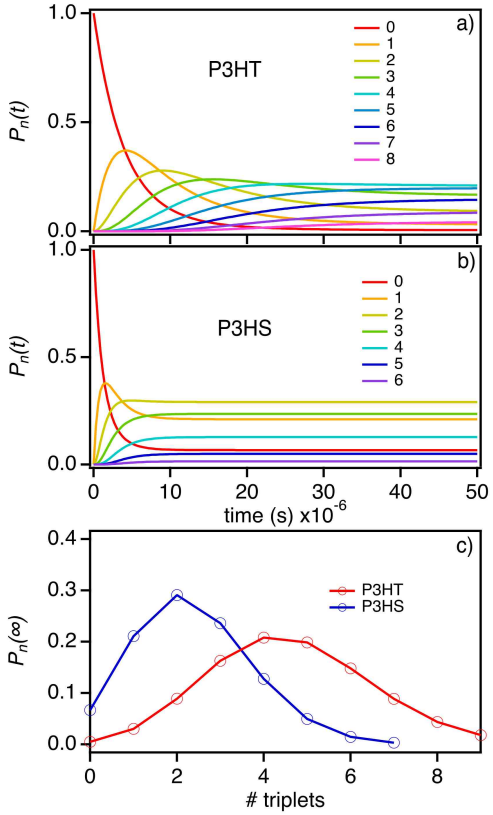

Figure 5.7: Time-dependent populations of triplets in P3HT (a) and P3HS (b) for multiple triplet configurations. c) Steady-state triplet populations ($P_n(\infty)$) for each polymer.

Plots of Pn(t) dynamics in Fig. 5.7 reveal useful insights into the roles of triplet lifetimes and interactions affecting triplet population decay that are essential for properly interpreting $I(t)/I(0)$ behavior of both systems. This was particularly apparent in the case of large quenching depths and faster quenching dynamics in P3HS despite having smaller triplet occupancies at longer times. This result appears counterintuitive when only considering triplet formation yields which have been shown earlier to depend on the heteroatom. It is also informative to point out how the choice of k_{TT} and k_{QST} used in the simulations

of $I(t)/I(0)$ curves and $P_n(t)$ dynamics in Figs. 5.6 and 5.7, respectively, can be validated from comparison with experiment. For example, we considered scenarios where $k_{TT} > k'_{ISC}$ and $k_{TT} < k'_{ISC}$ and varied k_{QST} to tune quenching responses that can be compared directly to experiment. We set an upper limit of ~10^{11} s^{-1} which is comparable to the fastest observed singlet-triplet quenching rate constant reported in conjugated polymers.[42] When $k_{TT} > k'_{ISC}$ triplet occupancies shift to lower values (i.e., $P_n(\infty)$ shifts to lower values) requiring larger k_{QST} to achieve the same quenching depth seen experimentally. However, upon comparison with experimental $I(t)/I(0)$ curves, poor agreement arises from quenching dynamics behaviors, i.e., simulated quenching dynamics are much faster. On the other hand, by allowing $k_{TT} < k'_{ISC}$, k_{QST} decreases drastically as shown in Fig. 5.5 due to much larger triplet occupancies (i.e., $P_n(\infty)$ shifts to larger values). Although previous pulse radiolysis work on larger polymer chains found evidence that an SPC can support up to 30 triplets,[98] it is doubtful that such large steady-state occupancies are possible here since both polymers are relatively small (ca. 30 KDa).

Lastly, photophysics of polymers are highly dependent on the SPC conformational qualities and we expect that values of k_{TT} and k_{QST} to fluctuate with sample preparation conditions (viz. solvent) and from molecule-to-molecule. In fact, previous single molecule spectroscopic investigations of singlet-triplet interactions found drastic variations in fluorescence responses with molecular size[104] and order.[102] We expect that SPC conformational qualities should have a large impact on k_{TT}. Here, we do not expect either P3HS or P3HT to be able to self-aggregate which should attenuate triplet diffusion and triplet-triplet annihilation. Further examination of this effect is beyond the scope of the present paper due to large variations in solubility of both polymers although large variations in fluorescence quenching often appear with very subtle changes in chain conformation and order.[102,128] In addition to triplet-triplet annihilation, self-aggregation in larger polymer chains has been proposed to increase singlet-triplet interactions.[104,122,169] However, reliably sorting out the dominant energy transfer mechanism has proven difficult. Nonetheless, long-range, Forster-type en-

ergy transfer mechanisms should not be affected by conformation as much as exchange mediated, Dexter-type energy transfer.

5.3 Conclusions

We have demonstrated new perspectives of kinetic factors governing the occupancies of triplets in conjugated polymers with large triplet yields on time scales relevant to optoelectronic devices. This feature, along with the multi-chromophoric nature of polymers, increases the likelihood of multiple excitonic species existing on an SPC at any given time. Our simulations have revealed several critical points that should be taken into consideration when designing excitonic materials for optoelectronic applications. Specifically, multi-exciton generation and heavy atom substitution approaches to generate and harvest many triplets. Perhaps the most noteworthy of these involves the kinetic competition between triplet formation and decay mechanisms that determine triplet occupancies on longer time scales. This effect was most pronounced in P3HS where, despite larger triplet yields, faster relaxation through first and second order processes lowered triplet occupancies compared to P3HT. We expect these processes in addition to singlet-triplet interactions to be strongly dependent on the SPC conformation, which can be simulated by adjusting the associated rate constants. Overall, our approach can now provide a clearer link between triplet formation (e.g., singlet fission or increased triplet admixture) on sub-picosecond time scales and triplet interactions on much longer time scales.

6.0 Unravelling the enigma of ultrafast excited state relaxation in non-emissive aggregating conjugated polymers

6.1 Introduction

Non-covalent, π-stacked aggregates of conjugated polymers significantly modify electronic properties and relaxation pathways of photogenerated singlet excitons.[89,112] Because many photophysical outcomes are decided on time scales <1 ps, aggregates can have a profound impact on the functionality and performance of polymers in optoelectronic device settings. One of the most important, but often overlooked, dynamical components of polymer photophysics is Franck-Condon state relaxation involving several displaced vibrational coordinates.[170–172] These modes are typically the high frequency, symmetric stretching motions of the conjugated backbone and it is generally assumed that only one mode, e.g., the totally symmetric C=C stretching mode (~1400-1500 cm^{-1}), has a significant displacement. This simplified picture of electron-phonon coupling often suffices although rapid excitation energy dissipation into multiple vibrational coordinates may complicate dynamics assignments and place fundamental limitations on energy harvesting yields.

We investigate the roles of Franck-Condon active vibrations in excited state relaxation dynamics of aggregating conjugated polymers similar to archetype systems used in solar cell applications (e.g., poly(3-alkylthiophenes), P3AT). Alkyl substituted poly(thieneylenevinylenes) (PTV) display intriguing photophysics and absorb a larger fraction of NIR photons. For this reason, PTVs were expected to produce larger solar cell power conversion efficiencies,[173–175] but devices rarely surpass 2% in power conversion efficiencies despite extensive optimization efforts.[174,176,177] The origins of poor performance were proposed to originate

from unsuitable morphologies and short-lived excited states.[178,179] The former characteristic can be traced to facile aggregation, even in dilute solutions, thus preventing intimate mixing with electron acceptors (i.e., fullerenes). Reported excited state lifetimes in PTV derivatives are <1 ps also resulting in low (<10^{-4}) photoluminescence (PL) quantum yields.[180] Electric dipole forbidden states residing in the mid-gap region were invoked to explain rapid relaxation dynamics and non-emissive nature of PTVs.[181,182] Assuming idealized C_{2h} point group symmetry, photogenerated $1B_u$ singlet exciton states relax non-radiatively to a lower energy $2A_g$ excited state then back to the $1A_g$ ground electronic state.[181–183] Experimental evidence for the $2A_g$ excited state was inferred from photoinduced absorption spectra[184] and large dispersion in resonance Raman spectra.[185] Musser et al. next amended this model by claiming the $1B_u$ singlet excited state first undergoes activated intra-chain singlet fission on time scales of ~45 fs followed by triplet pair relaxation to the $2A_g$ excited state on ps time scales.[186]

The prospect of efficient singlet fission in polymers raises interesting possibilities for generating and harvesting multiple electron-hole pairs.[147,148,187] Solution processability also provides an attractive and cost-effective testbed for verifying photovoltaic efficacies involving triplets. However, many aspects of singlet fission in polymers, especially PTV-type systems, remain poorly understood. Most notably, several stable structural forms may coexist[188] (e.g., aggregates and amorphous chains) that change with processing conditions. Moreover, large variations in photophysical branching ratios for polymer chains of different regioregularity and conformations have been documented[59,189,190] that can complicate optimization strategies.

We investigate vibrational Franck-Condon activity and excited state dynamics of an alkyl PTV derivative, poly(3-decylthieneylenevinylene) (P3DTV) and its heavy atom analog, poly(3-decylseleneylenevinylene) (P3DSV) (see Figure 6.1) using resonance Raman and transient absorption spectroscopy. These systems are advantageous for connecting early vibrational Franck-Condon activity to excited state relaxation dynamics on longer

time scales and their dependence on structural (aggregation) characteristics. Interestingly, Vanden Bout and co-workers recently demonstrated that dispersion of a related PTV derivative into solid inert hosts restores PL emission.[116] This result suggests the proposed role of mid-gap $2A_g$ excited states may not be as pervasive as originally believed.

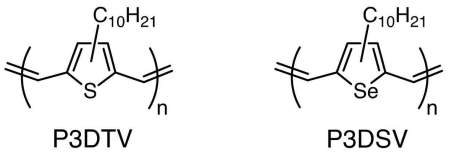

Figure 6.1: Structures of P3DTV and P3DSV.

Using detailed resonance Raman spectroscopy approaches, we selectively peer into Franck-Condon excited state vibrational relaxation activity for different polymer structural forms. Interestingly, both P3DSV and P3DTV systems display remarkably similar patterns of extended and rich harmonic progressions (i.e., overtones and combination transitions) involving multiple skeletal vibrational modes. Transient absorption spectroscopy measurements on and off resonance with aggregate absorption transitions revealed two principal transient decay components with time constants similar to earlier studies.[181] No evidence of ultrafast triplet formation was observed in either system, despite lower aggregate content and the presence of the heavier selenium atom in P3DSV. Barford and co-workers further noted very small spin-orbit coupling for π-stacked aggregates,[95] which should further negate heavy atom effects regardless of the triplet formation mechanism.

The prevalence of aggregates in addition to substantial displacements of many vibrational coordinates determined from a Raman intensity analysis leads us to conclude that efficient non-radiative vibrational energy dissipation dominates excited state relaxation dynamics. Importantly, these photophysics can be explained straightforwardly using simple multi-dimensional harmonic oscillator models without the need for complex multi-step processes involving optically inaccessible electronic states. Further evidence of anharmonic couplings responsible for efficient excitation energy dissipation into low frequency inter-

molecular modes promoted by large Franck-Condon activity was reported earlier by Vardeny and co-workers in the form of strain waves in solid films.[191,192]

6.2 Results and discussion

P3DTV and P3DSV were synthesized and characterized according to Guoshun et al.[193] and we begin by considering the characteristics of linear optical spectra of both polymers in different media. Figure 6.2 shows absorption spectra of P3DTV and P3DSV in dilute chlorobenzene solutions and thin films. Both polymers are sparingly soluble and remain aggregated even at low concentrations (e.g., $<10^{-7}$ M) similar to other aggregating polymers.[194] Weakly resolved vibronic structure is apparent with an average interval of ~1400 cm^{-1}, nearly identical to many polythiophene derivatives.[110] However, in PTVs, this interval actually consists of multiple displaced skeletal vibrations in the Franck-Condon state.[195] P3DSV absorption maxima are red-shifted by ~0.11 eV due to the heavier selenium heteroatom but lineshape features possess similar vibronic patterns as P3DTV.

It is also useful to point out that the electronic origin (0-0) transition of both polymers changes significantly between solution and solid forms. Specifically, the lowest energy resolved absorption features in both samples are separated by one vibronic interval although the high-energy tails overlap. Previous linear spectroscopy studies of PTV systems have proposed that two electronic origin transitions are present corresponding to polymorphs. Gavrilenko et al. performed theoretical simulations on a model PTV oligomer and concluded that changes in the absorption onset region arose from two structurally distinct aggregate forms with different side group packing motifs.[74] These authors investigated model oligomers with different packing arrangements of the alkyl side chains and found evidence of two separate transitions separated by nearly one vibronic interval although vibronic progressions were not included.[74] A simpler, alternative explanation is that the 0-0 peak strength varies with the ordering characteristics of π-stacked aggregates, which follows the current consensus of a single electronic origin (0-0 peak) with a vibronic progression built

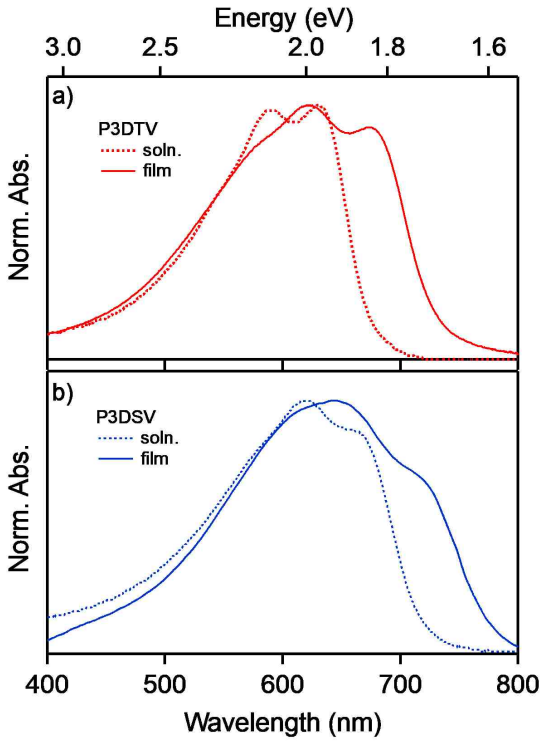

Figure 6.2: a), b) Absorption spectra of P3DTV and P3DSV, respectively, solutions (dashed) and thin films (solid).

on this transition.[90] According to the weakly coupled H-aggregate exciton model, the 0-0 transition is forbidden by symmetry although vibronic sidebands (0-n, n>1) are allowed.[110] The 0-0 peak strength is especially sensitive to intra- and interchain aggregate order, i.e., this transition becomes weakly allowed when aggregates possess significant disorder,[196,197] such as large torsional distortions between π-stacked monomers.[118] An important caveat is that only one vibrational mode is displaced between ground and excited state potential energy surfaces,[198] which is not the case for PTV derivatives.

To further home in on the nature of the lowest energy excitonic transitions in both P3DSV and P3DTV compounds, we examined the effect of polymer concentration on the absorption lineshapes over a broad range of concentrations. As samples become more dilute, the main absorption lineshape only decreases in its integrated absorbance with no significant changes in the vibronic pattern or energies indicating that the packing integrity is not affected, rather, only the aggregate concentration. If the weakly resolved vibronic line-

shape consisted of multiple electronic origins, it is unlikely that these would exhibit similar behavior over this concentration range.

PTV systems are nominally non-emissive[179] (e.g., quantum yields $<10^{-4}$), making it difficult to establish a good correspondence between absorption and PL spectra of distinct structures. Recent single molecule spectroscopy studies by Hu et al. unequivocally demonstrated that dilute dispersion within an inert solid host restores emission.[116] Interestingly, the observed PL maximum energy overlaps with the main absorption lineshape suggesting that only aggregates are non-emissive and these structures reabsorb emitted light from solvated, non-aggregated chains.

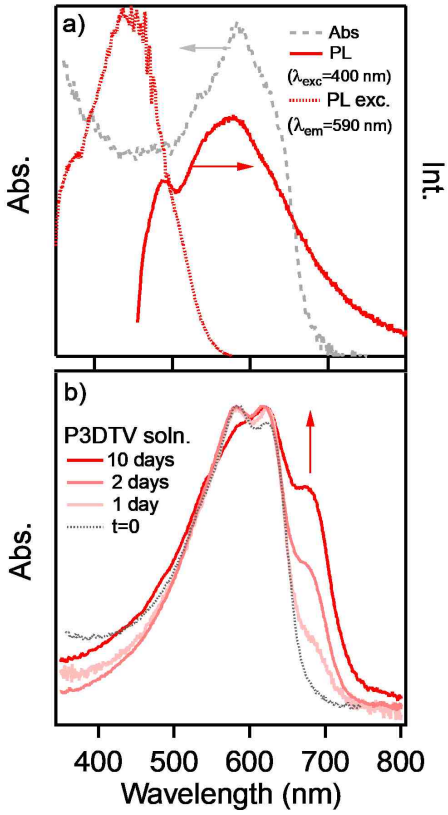

Figure 6.3: a) PL spectrum of the most dilute P3DTV solution (0.15 mg/L) with its corresponding excitation spectrum. The absorption spectrum is shown for comparison (gray dotted trace). b) Time-dependent normalized absorption spectra of a P3DTV solution over 10 days (0.6 mg/L) showing the growth of the characteristic low energy 0-0 feature near the onset.

Figure 6.3a shows PL and excitation spectra of the most dilute P3DTV sample (~0.15 mg/L) revealing a broad lineshape in the energetic vicinity of the main aggregate absorption spectrum for P3DTV, similar to that of Hu et al.[116] No PL emission was observed for the P3DSV sample regardless of concentration, which is a consequence of efficient non-radiative deactivation. Similar reductions in PL intensities have also been observed in heavy atom analogs of related polythiophenes (namely, P3HT) where PL quantum yields fall precipitously with larger heteroatoms in the 5-member ring.[44] PL excitation spectra of the blue-shifted P3DTV emitter reveal a broad and unresolved lineshape on the high energy tail of the absorption spectrum (gray trace) representing solvated P3DTV chains.

The fact that PL emission can be restored upon dilution suggests that mid-gap forbidden excited states are likely at much higher energy and not accessible using one-photon spectroscopies. However, this result does not explain why dominant aggregate structures are non-emissive. Recent studies have investigated aggregates and concluded they are intrinsically non-emissive probably from efficient charge generation via exciton dissociation or polaronic nature of photoexcitations.[115] It is first informative to consider the nature of aggregates in PTV and related polymers as a general consensus does not yet exist. In particular, we seek to confirm whether absorption spectra are comprised of multiple overlapping transitions from polymorphs accidentally separated by one vibronic interval, or, from a single origin with a vibronic progression built on this transition.

Following initial dilution, small color changes were observed akin to self-assembly induced formation of aggregates in polythiophenes.[199] Absorption spectra of P3DTV solutions (~0.6 mg/L) were monitored over several days while stored under nitrogen (Figure 6.3b) and a characteristic growth of a peak exactly matching the 0-0 energy from thin film spectra were observed. The remarkable similarities of absorption spectra of these aged solutions to the thin film lineshape in Fig. 6.2a confirms that aggregates are the dominant absorbers. It is also interesting to note that vibronic sidebands show very little change, consistent with expectations from an H-aggregate type interaction. Similar behavior was

observed in P3DSV samples although lineshapes of aged solutions do not exactly resemble corresponding thin films and spectral changes were not as pronounced as seen in P3DTV. The results outlined here suggest that both P3DTV and P3DSV chains exist as small aggregates in solution and eventually ripen to larger ones as encountered in thin films. The persistence of the characteristic aggregate lineshape to very low concentrations (e.g., Figs 6.2a,b) supports this view indicating single P3DTV and P3DSV chains probably adopt collapsed conformations. As multiple chains associate or coalesce at higher concentrations, aggregate ordering characteristics are disrupted causing increasing strength of the 0-0 transition.[90,112]

In the case of partially resolved vibronic lineshapes of polymer aggregates, it is common to apply the weakly coupled aggregate exciton model.[90] However, caution is necessary for extending this description to explaining trends in vibronic absorption lineshapes of P3DTV and P3DSV aggregates. For example, in P3HT aggregates, it is a good assumption that only the dominant C=C symmetric stretching vibration (~1400 cm^{-1}) of the thiophene ring contributes to the vibronic structure. On the other hand, the vibronic pattern and interval of P3DTV and P3DSV derivatives – while bearing likeness to that of P3HT – actually contain contributions of many displaced skeletal vibrational modes spanning a broad range of frequencies (e.g., ~400-1600 cm^{-1}).[195] The coalescence of multiple progressions gives rise to weakly resolved progression intervals in one apparent frequency that often does not match any mode in the Raman spectrum. This is known as the 'missing mode effect' and is most prevalent in molecules with many Franck-Condon active vibrations with substantial excited state displacements.[85] Additionally, coupling between vibrational modes (e.g., via Duschinsky rotation) are possible which further complicate views of excitonic coupling.

It is now useful to further examine the multi-dimensional excited state potential landscape of P3DSV and P3DTV and vibrational mode specific Franck-Condon dynamics. Figure 6.4 shows resonance Raman spectra of P3DTV and P3DSV thin films excited at 488 nm (a) and 780 nm (b) corresponding to post- and pre-resonance regimes, respectively (com-

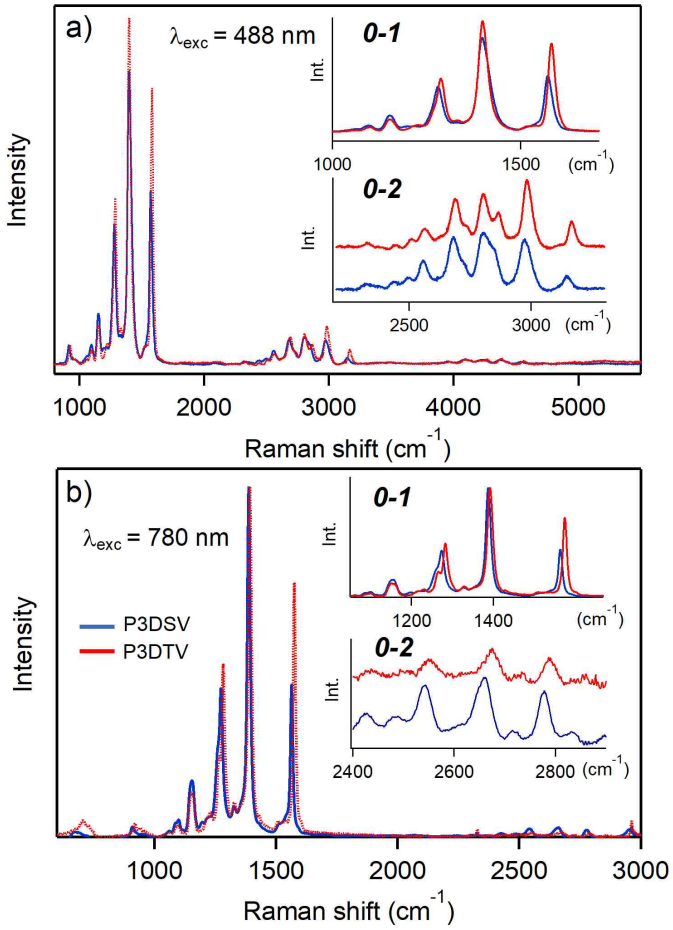

Figure 6.4: Resonance Raman spectra of P3DTV (red) and P3DSV (blue) excited at 488 nm (a) and 780 nm (b). Insets: Expanded fundamental (0-1) and first overtone (0-2 regions).

parison of Raman spectra excited with other wavelengths are included in the Supporting Information). In an earlier resonance Raman investigation of two alkyl substituted PTV derivatives, we found that aggregates and isolated chains co-exist simultaneously in solution and solid-state.[195] Clearly, the existence of multiple structures can severely complicate the interpretation of spectroscopic signals, which is especially important to sort out for studies involving ultrafast pulsed laser excitation (vide infra).

We focus on the dominant displaced vibrational modes of the conjugated backbone of each polymer mainly involving the CC symmetric stretching motions.[200,201] In particular, the vinylene CH stretch (~1270—1280 cm^{-1}), CC ring stretch (~1390—1400 cm^{-1}), and vinyl CC stretch (~1570—1580 cm^{-1}) modes display the largest sensitivity to subtle changes

in chain packing order and the largest resonance enhancements consistent with a larger excited state displacement. Moreover, these modes dominate overtone and combination transitions further underlying their importance in excited state geometrical rearrangements. Comparison between P3DTV and P3DSV Raman spectra also reveal discernible red-shifts (~10 cm^{-1}) of vibrations involving the vinylene group on the latter in addition to subtle decreases of the C-S-C (C-Se-C) bending vibration at ~720 cm^{-1} due to the heteroatom.

Perhaps the most interesting and revealing feature in both polymers is the appearance of rich progressions of overtone and combination transitions (0-2 clusters highlighted as insets) that persist for up to three harmonics. Importantly, these transitions encode valuable insights into Franck-Condon vibrational dynamics following photon absorption. First, similar intensity distributions are observed with pre-resonant excitation (i.e., 780 nm excitation, Fig. 6.4b) but only the first harmonic region is resolved. This excitation regime samples fast dynamics often resulting in self-cancellation of the Raman wavepacket by rapid oscillations of the dominant imaginary term at times longer than one vibrational period.[82] The fact that the first overtone-combination cluster of peaks is observed under pre-resonance conditions demonstrates significant wavepacket motion on the multi-dimensional excited state surface. Vertical projection of the Raman wavepacket with photon absorption samples the slope of the excited state potential energy surface, which imparts momentum along the path(s) of steepest descent.[82] This corresponds to the high frequency, largest displaced backbone modes of both P3DTV and P3DSV systems indicating the excited state evolves mainly along these modes. Furthermore, the bound nature of these excited state potentials and appearance of multiple harmonics demonstrates that the wavepacket makes several return visits to the Franck-Condon region. Because overtone-combination intensities develop at later times (e.g., several vibrational periods), these transitions are usually subject to broadening effects. The fact that linewidths in overtone-combination clusters remain relatively narrow over the entire spectral window demonstrates that the wavepacket is well described within the coherent state approximation over the excited state lifetime. Additionally, no ev-

idence of interference effects from state crossings,[202] such as anomalous intensities or dips in excitation profiles, are observed and the system can be modelled using simple two-state harmonic oscillator potentials.

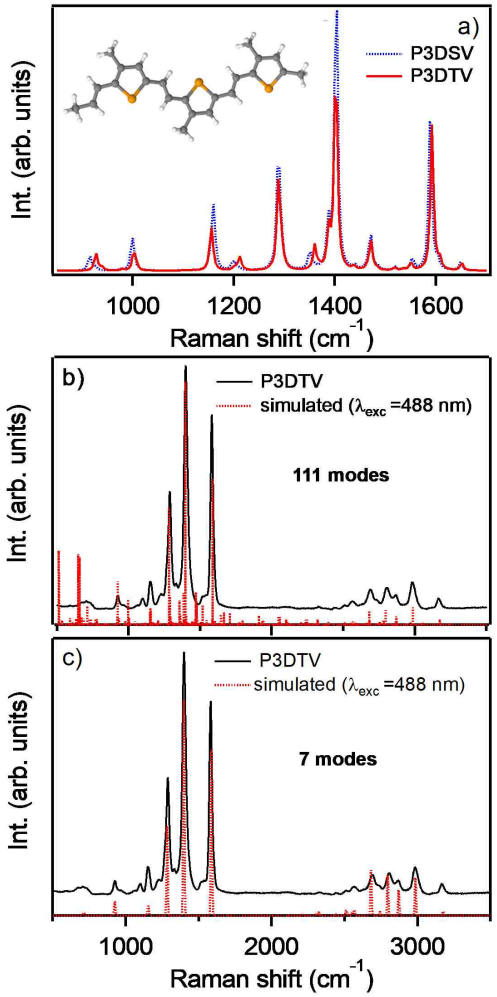

Figure 6.5: Comparison of Raman frequencies of high frequency CC stretching modes of P3DTV and P3DSV.

We now undertake density functional theoretical (DFT) simulations to calculate Raman intensities under resonant and non-resonant conditions. Figure 6.5a shows simulated ground state Raman spectra of P3DTV and P3DSV small molecule surrogates (inset) under non-resonant conditions (i.e., no wavepacket evolution on the excited state potential energy surface). Only slight red-shifts of the dominant skeletal CC vibrations are apparent with substitution for the heavier selenium heteroatom, similar to experiment. Frequencies of

Table 6.1: Comparison of Raman frequencies of high frequency CC stretching modes of P3DTV and P3DSV.

Assignment[a]	Vinyl C-H strech	Ring C=C strech	Vinyl C=C strech
P3DSV			
Raman shift (cm^{-1}) (λ_{exc} = 488 nm)	1280	1401	1573
Raman shift (cm^{-1}) (λ_{exc} = 780 nm)	1271	1388	1564
B3LYP/6-31G(d) (cm^{-1})	1286	1400	1586
P3DTV			
Raman shift (cm^{-1}) (λ_{exc} = 488 nm)	1287	1400	1573
Raman shift (cm^{-1}) (λ_{exc} = 780 nm)	1283	1392	1575
B3LYP/6-31G(d) (cm^{-1})	1285	1399	1590

[a] Comparison to theoretical frequencies were made only for ground state conditions

Raman-active skeletal vibrations between theory and experiment are included in Table 6.1.

Time-dependent density functional theory (TD-DFT) was used next to calculate excited state electronic structures and Raman intensities were simulated using Independent Mode Displacement Harmonic Oscillators (IMDHO). Figures 6.5b,c show theoretical Raman spectra including excited state contributions for all Raman-active vibrational modes and only the largest displaced modes of the model oligomer, respectively. Mode-specific displacements are determined from differences in the ground and excited state geometries and simulated spectra show excellent agreement with experiment, especially in the overtone-combination region. Interestingly, Raman spectra generated using only the largest displaced modes (i.e., over 80% of the total vibrational reorganization energy) reproduces experimental patterns relatively well although this approximation does not capture the full complexity of excited state geometrical rearrangements. Resonance Raman spectra were not simulated for the P3DSV derivative using TD-DFT methods because of the need to optimize selection of appropriate pseudo-potentials for core electrons. Nonetheless, the re-

markable similarity between experimental Raman patterns of P3DTV and P3DSV derivatives (Fig. 6.4) demonstrates similar displacements of skeletal vibrations.

Simulated resonance Raman intensities also provide useful perspectives of the time scales of Franck-Condon vibrational relaxation which usually require ultrashort pulsed laser excitation. For example, the phenomenological wavepacket damping constant used in the simulations was ~100 fs (350 cm^{-1}) that produces several recurrences (i.e., return visits to the Franck-Condon region) in the high frequency displaced skeletal vibrations listed in Table 6.1. Franck-Condon wavepacket dynamics can persist for up to ~500 fs, similar to reported singlet exciton lifetimes[181,186] demonstrating that non-radiative vibrational activity represents a large contribution to excited state relaxation. These characteristics now offer clearer views into the non-emissive behavior of PTV aggregates.

6.2 Evidence of vibrational dynamics on longer time scales: transient absorption spectroscopy

Because resonance Raman spectroscopy is sensitive to the early vibrational dynamics of the Franck-Condon excited state, it is informative to connect this regime to ascertain the long term the fate of excitons. Transient absorption spectroscopy measurements were performed on P3DTV and P3DSV derivatives in dilute chlorobenzene solutions. To our knowledge, heavy atom effects on the photophysics of PTV systems have not been reported previously, which presents an interesting scenario to examine both the roles of structure and anticipated spin-orbit effects on previously reported singlet fission processes. It is useful to note that triplet signatures often vanish when appreciable aggregation exists,[59] although a fraction of non-aggregated P3DSV and P3DTV chains should also be present. Transient absorption dynamics were investigated at several pump excitation wavelengths to resolve how the large and relatively long-lived structural deformations in the excited state are affected by polymer aggregation and their influence on photophysical branching ratios.

Transient absorption pump-probe spectra and dynamics of P3DSV and P3DTV dilute

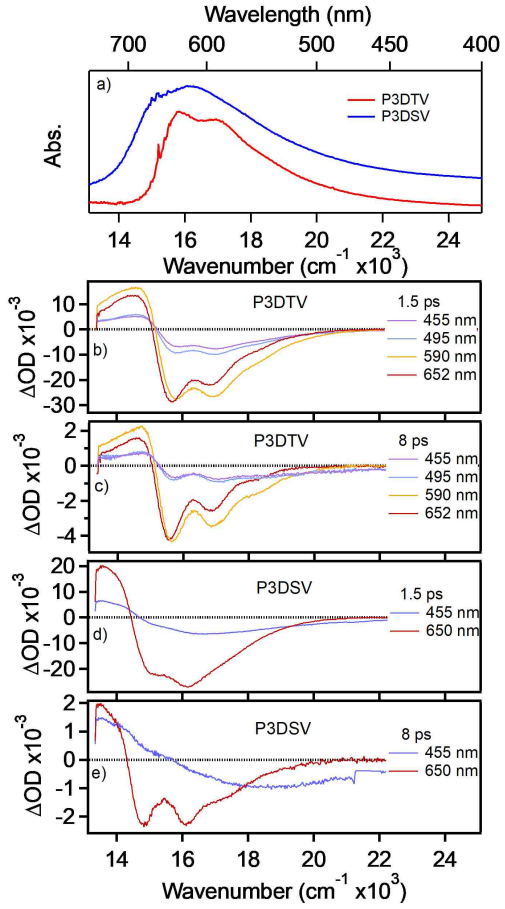

Figure 6.6: a) Electronic absorption spectra of P3DTV and P3DSV. Transient absorption spectra at two probe delay times, 1.5 ps and 8 ps, of P3DTV and P3DSV (b), c) and d), e), respectively. Pump excitation wavelengths are shown for each system and fluences were typically <1 mJ/cm^2.

solutions are shown in Figure 6.6 along with steady-state absorption spectra for comparison. Pump excitation wavelengths (energies) were tuned over the entire ground state absorption range and evidence of multiple absorbers was present albeit in different proportions for both systems. Transient absorption spectra are displayed at two probe delay times, namely, 1.5 ps and 8 ps (the entire series of each polymer is shown in the Supporting Information) and both systems exhibit prominent ground state bleach and excited state absorption components. It is first interesting to note the time and pump wavelength dependence on the ratios of the first two resolved vibronic peaks (0-0 and 0-1) in ground state bleach components. At longer delay times and pump excitation wavelengths, the 0-0 transition dominates indicating re-

laxation to longer conjugated segments and selective excitation, respectively. Furthermore, the bleach spectral lineshapes do not exactly match those of ground state absorption consistent with the presence of more than one chromophore (i.e., aggregate structure). These trends demonstrate that aggregates dominate transient spectra in addition to the presence of aggregates with different packing qualities. Like studies of related materials, relaxation to longer chains segments by the usual energy transfer cascade type mechanisms leads to the appearance of linewidth sharpening and larger 0-0 strengths at longer times.

We analyzed the dynamics of spectral components using a singular value decomposition approach revealing one principle kinetic component for the longest pump wavelength and two components for the other pump wavelengths used. Details of the model and fitting procedure are included in the Supporting Information and individual fitting of the principle kinetic components was performed assuming a bi-exponential model. Two main spectral components were obtained, and the larger component had time constants of ~1.5 ps and 60 ps compared to 0.6 ps and ~2 ps for the smaller component. The relative contribution was dependent on the pump excitation wavelength with the larger component completely dominating at longer wavelengths. The larger component is assigned as relaxation and thermalization dynamics within aggregates and subsequent recombination.

The smaller component showed faster dynamics that became more prominent at shorter pump wavelengths that probably involves minority, non-aggregated chains. We assign the 0.6 ps and ~2 ps time constants of this component to excited state formation and recombination, respectively, based on the fact that the former exhibits rise dynamics on subpicosecond time scales. The smaller component also experiences small blue-shifts at longer delay times that decays to background with no longer-lived kinetic features. Despite that the relative contributions of fast and slow decay components changes with pump wavelength, no change in time constant was observed for these spectral components regardless of pump wavelength.

The dominance of aggregate contributions in both steady-state and time-resolved spec-

tra of P3DSV and P3DTV and their large vibrational displacements along multiple vibrational modes are consistent with observed facile non-radiative relaxation dynamics and non-emissive behaviour. It is remarkable that similar dynamics and time constants were observed in the P3DSV derivative. Although P3DTV did not exhibit a clear ground state bleach signal corresponding to the proposed contribution of shorter, non-aggregated chains, a higher energy bleach component became apparent in the 8 ps delay transient spectra of P3DSV (Fig. 6.6e). These trends reflect differences in the amounts of non-aggregated chains between samples and their relative contributions to the ground and excited state dynamics.

Perhaps the most striking feature from these results is the relatively minor effect of the heavier selenium atom, or, expected spin-orbit coupling strength.[95] This is in contrast to earlier reports of larger intersystem crossing rate constants in P3AT derivatives where bigger heteroatoms yielded shorter excited state lifetimes due to increased triplet formation rates.[44] We posit that rapid excitation energy dissipation via many Franck-Condon vibrational modes circumvents triplet formation, similar to electroluminescence studies showing a much lower triplet formation constant.[203] Because this pathway dominates in aggregates, it is possible that triplets may be populated in minority non-aggregated chains that was assessed by measuring transient absorption spectra in the near-infrared (NIR) spectral region.

Triplet and polaronic states typically absorb in the NIR region and Figure 6.7 compares transient absorption spectra of P3DTV and P3DSV excited with 652 nm photons and probed up to 1300 nm. A broad excited state absorption feature appears in both systems that red-shifts by ~1000 cm^{-1} in ~400 fs then decays to background on time scales <10 ps. Comparison with excited state absorption features in Fig. 6.6 demonstrates these NIR transients are consistent with excited state absorption features involving the photogenerated singlet exciton. The significant red-shifts appearing in both systems likely originate from vibrational relaxation and solvent reorganization occurring on time scales up to a few ps.[204]

Overall, results from steady-state and time-resolved spectroscopy measurements are

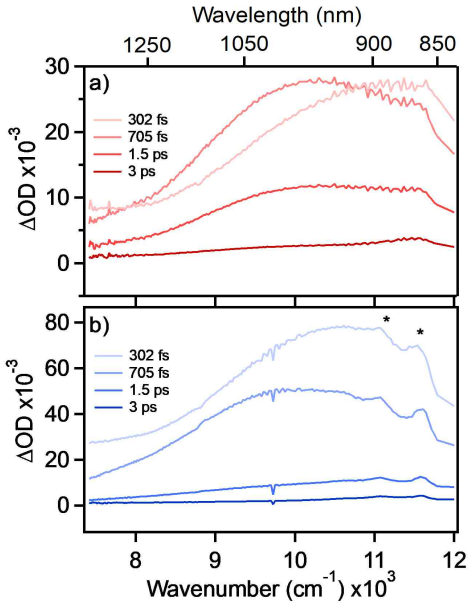

Figure 6.7: Transient absorption spectra of P3DTV (a) and P3DSV (b) probed in the NIR region. Pump excitation wavelength for each series was 652 nm. Asterisks denote an artifact due to the probe continuum.

similar to earlier reports noting rapid excited state relaxation dynamics that are typically complete in <10 ps.[181,184,186] Most notably, we observed the same low amplitude tails on longer time scales in transient absorption spectra inferred as direct evidence of the $2A_g$ excited state. It is also interesting to note the changes in vibronic maxima from ground state bleach signals in both polymers where different 0-0/0-1 ratios were observed at different pump excitation wavelengths. The lack of clear isosbestic points in the transient spectra confirms the presence of multiple isolated aggregate structures which also manifests as large frequency dispersion effects in resonance Raman spectra[195] taken earlier as direct evidence of the forbidden $2A_g$ excited state from interference effects.[185] We now propose a revised photophysical model for PTV type systems in Figure 6.8 that only requires a two-state, undistorted harmonic oscillator description with significant wavepacket motion on the $1B_u$ excited state. Importantly, this model can fully account for photophysical responses of both systems especially efficient vibrational deactivation of aggregates.

Lastly, measured time scales and absorption features of transient spectra were not con-

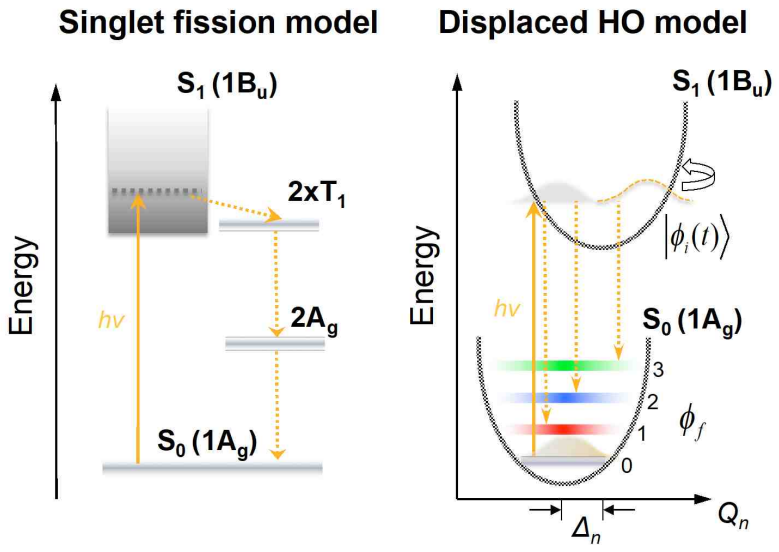

Figure 6.8: Singlet fission model from Ref.[186] and displaced harmonic oscillator (HO) model probed by resonance Raman spectroscopy from this work.

sistent with appreciable triplet formation, presumably from singlet fission. We conjecture that previous measurements probably interrogated a larger fraction of non-aggregated chains that often exhibit larger triplet populations.[59] However, the larger fraction of non-aggregated chains and larger expected spin-orbit coupling in P3DSV should produce unambiguous triplet features, which was not the case. Interestingly, we performed preliminary transient absorption studies on a small molecule PTV-type oligomer in dilute solution that showed unambiguous triplet signatures which cannot originate from singlet fission.

6.3 Conclusion

We have shown that the primary relaxation pathway for alkyl substituted PTV systems involves vibrational energy dissipation from high frequency Franck-Condon active modes into lower frequency intermolecular vibrations. The prevalence of aggregates in these and related polymers explains their non-emissive nature which can be alleviated by dilution. This effect was most apparent from the restoration of PL emission demonstrating the absence of mid-gap excited states with low oscillator strengths. The lack of appreciable heteroatom influence may be attributed to strong coupling between π-stacked polymer chains

although relatively little is known about the dependence of spin-orbit coupling on excitonic interactions. Transient absorption spectra of both polymers showed similar dynamics confirming the dominance of vibrational relaxation in aggregates that appears to bypass previously proposed relaxation pathways. Our results demonstrate that useful insights of Franck-Condon vibrational activity can be readily obtained from resonance Raman intensities that provide a more cost-effective approach for screening materials.

7.0 Large Excited State Conformational Displacements Expedite Triplet Formation in a Small Conjugated Oligomer

7.1 Introduction

Spin-forbidden triplet excited states in conjugated organic molecules have a large bearing on functionality and performance at the materials level.[130,205130] In most cases, intersystem crossing yields are small and relaxation within the singlet manifold typically dominates excited state dynamics.[206] On the other hand, conjugated heterocycles (e.g., thiophenes) often exhibit substantial triplet yields that vary significantly with size, conformation and packing order.[102,207,208] Because there is now widespread interest in harvesting triplets to improve performance metrics of optoelectronic devices, such as solar cells,[147,148] comprehensive pictures of molecular electronic and structural factors governing transitions to and from the triplet manifold are needed.

Intersystem crossing events are most conveniently described from a pure electronic perspective that only consider vertical energies as inputs for estimating couplings and rate constants.[209] However, the role of vibrational motions in mediating triplet formation and relaxation has received greater consideration.[152,207] There is also increased attention for potentially harnessing specific vibrational motions to regulate transitions between different spin state manifolds[210] that may eventually be possible via synthetic or processing approaches although this topic has remained relatively unexplored.

Here, we investigate the roles of mode-specific vibrational displacements on triplet formation dynamics in an alkyl substituted trans-thienylene-vinylene dimer (dTV, see Figure 7.1), a small molecule analog of poly(thienylene-vinylenes) (PTV). PTV photophysics,

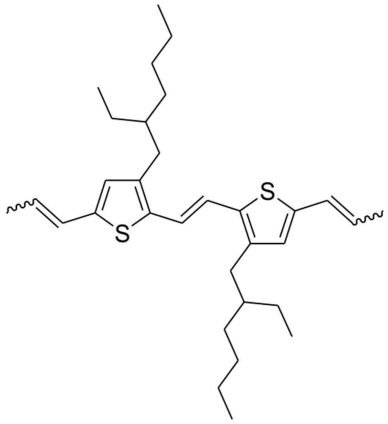

Figure 7.1: Structure of the trans-thienylene-vinylene dimer (dTV)

though well studied, have remained controversial largely due to the short-lived nature of their excited states.[181,185,186] For example, dipole-forbidden mid-gap excited states (i.e., $2A_g$) have been invoked to explain rapid excited state relaxation and non-emissive behavior commonly observed in these systems.[181,182,192] This model was next amended by Musser et al. to include a singlet fission process within 45 fs followed by relaxation of the triplet pair to the $2A_g$ state and subsequent non-radiative decay to the ground electronic state.[186] However, Hu et al. demonstrated that fluorescence emission could be restored by dispersing a PTV derivative in a solid inert host, causing aggregates to dissociate.[116] Our group also recently demonstrated rich Franck-Condon vibrational dynamics in aggregating PTVs which promotes efficient non-radiative relaxation.[211] While a comprehensive photophysical model describing the interdependence on molecular structure is still lacking, minimizing complications from aggregation and heterogeneity effects is essential to obtain unambiguous views of photophysical pathways.[212] To this end, we demonstrate that dTV can serve as model system for resolving vibrational contributions to intersystem crossing events in related molecules[213] and non-aggregating polymeric counterparts.

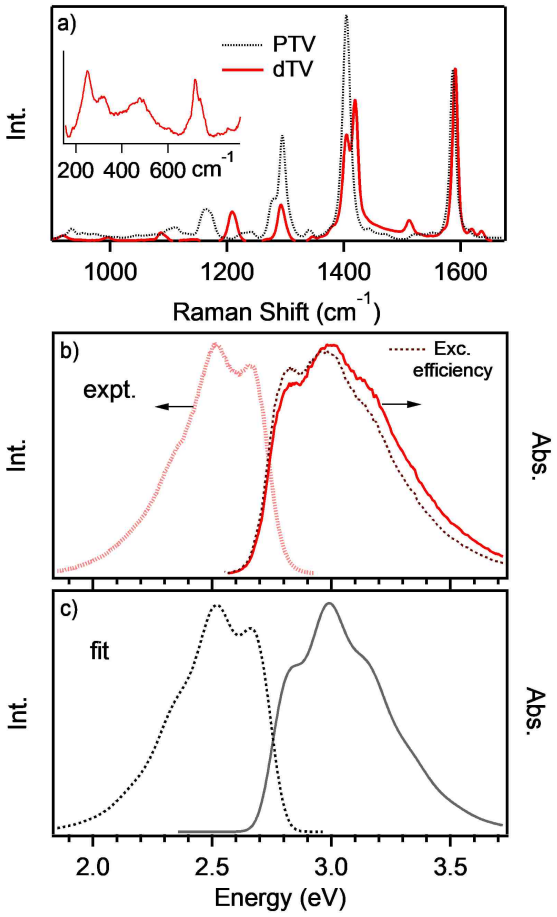

Figure 7.2: a) Pre-resonant Raman spectra of dTV and a poly(thienylene-vinylene (PTV) analog for comparison in the CC stretching region. Inset: enlarged low frequency region of the dTV Raman spectrum. b) Electronic absorption and fluorescence emission spectra of the dimer in dilute solution (O.D. < 0.1). c) Simulated electronic spectra using Raman-active vibrations from a).

7.2 Results and Discussion

Figure 7.2 shows Raman spectra (a) and electronic absorption and fluorescence emission spectra (b) of dilute dTV solutions. Raman spectra were measured under off resonance conditions (785 nm) showing activity in multiple skeletal vibrations. Comparison between Raman patterns of a related PTV derivative reveals similar features confirming intra- and inter-ring CC symmetric stretching vibrations that were assigned previously.[200,201] Both absorption and fluorescence emission spectra display a partially resolved vibronic progression

with an average interval of ~1300 cm^{-1} representing the coalescence of multiple displaced skeletal modes. Stokes shifts of ca. 4000 cm^{-1} are apparent indicating large excited state geometric distortions.

Table 7.1: Fit parameters used for dTV absorption and emission spectra simulations.

Mode (cm^{-1})	Δ (dimensionless)
1590	0.85
1420	0.6
1400	0.6
1290	0.55
1210	0.6
920	0.5
700	0.7
580	0.5
340	1.8
$E_{0\text{-}0}$ (cm^{-1})	22200
Γ (cm^{-1})	200

We next use a simple undistorted harmonic oscillator model for up to nine modes (i.e., dominant displaced modes from Raman spectra) and a single electronic origin ($E_{0\text{-}0}$) to simulate spectral line shapes in Fig. 7.2 b and obtain excellent agreement with experiment. Relative vibrational displacements were estimated using the short-time approximation ($I_1/I_2 = \omega_1^2\Delta_1^2/\omega_2^2\Delta_2^2$) from Raman intensities in Fig. 7.2 a which are then scaled to the overall displacement determined from the overall absorption spectra widths (Table 7.1).[81] Importantly, it was necessary to include highly displaced low frequency modes in order to obtain good agreement with the large observed Stokes shifts and vibronic linewidth broadening characteristics. Inspection of Raman spectra in the low frequency regime reveals activity involving the vinylene group, especially, out-of-plane torsional motions (ca. 300-340 cm^{-1}). To this end, we used an effective frequency of 340 cm^{-1} and vary the displacement

to replicate experimental features that represents the dominant linewidth broadening contribution resulting in shifts spectral origins to higher (lower) energy for absorption (emission) transitions. Earlier spectroscopic investigations of phenylene-vinylene oligomers noted similar behavior at room temperature where spectral broadening was attributed to large torsional displacements involving the vinylene group.[214] Fluorescence lifetime measurements in dilute solutions were also measured revealing a single decay component of ~1.7 ns and excitation spectra overlap well with absorption line shapes consistent with electronic transitions involving a single chromophore.

The fact that highly displaced low frequency torsional motions must be explicitly included in the vibronic analysis demonstrates excited state geometries are markedly different than the ground electronic state. We next measured transient absorption spectra of dTV dilute solutions to gain additional views into how these vibrational displacements impact photophysical branching ratios following relaxation of the Franck-Condon state. Figures 7.3 a,b show transient spectra of dTV in degassed chlorobenzene (CB) and dispersed in nujol, respectively. Spectral features are similar between the two samples, but dynamics change significantly depending on the surrounding medium. Two isosbestic points are labeled in Figure 7.3 a,b by arrows that indicate the growth and decay of two species (ground state and excited state). Spectral components were obtained via multi-exponential global kinetic analysis of the singular value decomposition pump-probe data set, as well as single wavelength multi-component fits yielding identical time constants.

In both media, spectra obtained at short time delays (several picoseconds) show a structured bleach feature strongly resembling the ground state absorption lineshape, and two partially resolved excited state absorption features with maxima at ~2.2 eV and ~1.7 eV. Not shown are the thermalization dynamics of the S_1 state most likely involving the high frequency skeletal vibrations are associated with a time constant of ~250 fs. The ~2.2 eV and ~1.7 eV absorption transitions merge into a single excited state absorption at ~2.5 eV and a bleach feature that persists on microsecond timescales. Global kinetic analysis reveals

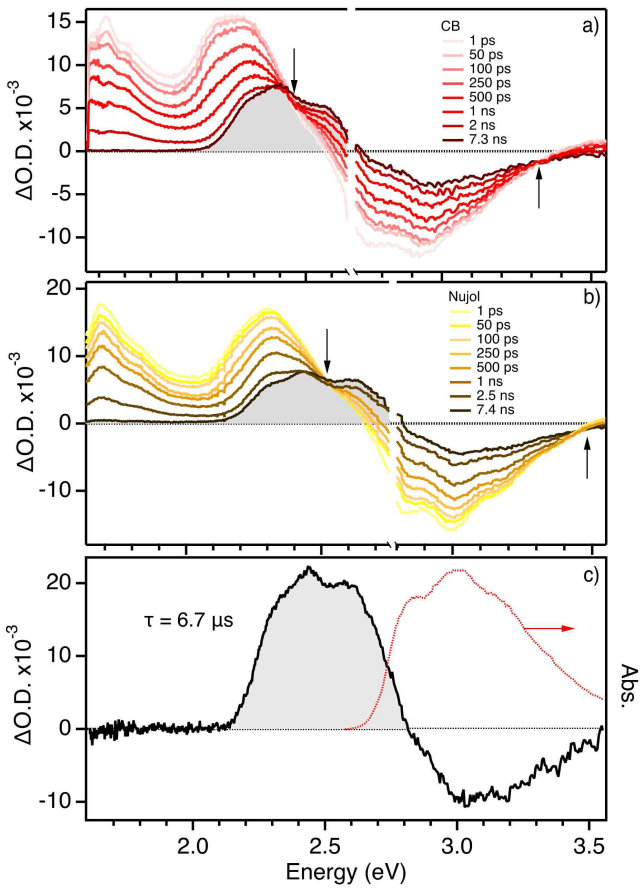

Figure 7.3: Broadband transient absorption spectra of dTV in dilute CB (a) solutions and nujol (b) dispersions over various probe delay times. Pump excitation energies were typically 2.6-2.8 eV and arrows depict resolved isosbestic points in the spectral dynamics. c) Transient absorption spectrum of dTV from flash photolysis measurements in degassed dilute CB solution. A steady-state absorption spectrum (red dotted trace) is included for reference.

time constants of 260 $\pm$ 100 ps, 1430 $\pm$ 190 ps, and 6.7 $\pm$ 0.2 μs in CB, and 210 $\pm$ 40 ps, and 1640 $\pm$ 70 ps in nujol. Nanosecond absorption spectra were not obtained for nujol dispersions although the lifetime of longer-lived component is expected follow similar trends. In the presence of air, the long-time constant in CB decreases to ~200 ns. Based on the lifetime range and sensitivity to environment, the long-lived component can be assigned as relaxation of the lowest energy triplet (T_1) to the ground electronic state (S_0) whereas assignments of faster dynamics are less straightforward

To further home-in on factors involved in transitioning to the triplet manifold, we turn to

density functional theory (DFT) simulations. Figure 7.4 displays HOMO and LUMO isosurfaces (0.03 e/Å^3) of the dTV molecule calculated at the B3LYP/6-31G(d) level of theory in vacuum with the branched ethyl-hexyl groups replaced by methyl groups. The ground state (S_0) geometry was optimized in C_{2h} symmetry and time-dependent DFT (TD-DFT) simulations at the same level of theory were employed to obtain five lowest-energy excited states utilizing the fully optimized S_0 as a reference state. Importantly, the transition with the largest oscillator strength (1.3) corresponds to 100% HOMO→LUMO character and vertical excitation energies were 2.75 eV, in excellent agreement with the HOMO-LUMO energy gap and experiment.

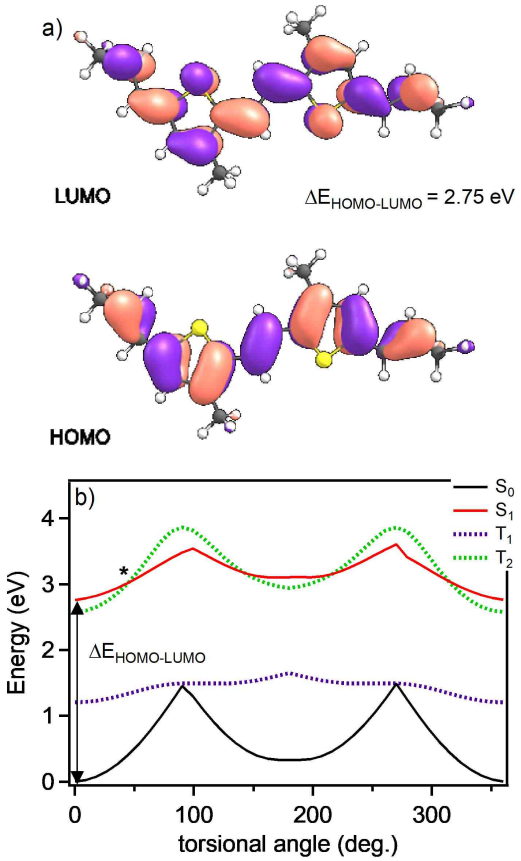

Figure 7.4: a) HOMO and LUMO isosurfaces (0.03 e/Å3) for the dTV model compound. b) Potential energy surfaces for S0, S1, T1, and T2 states calculated using the torsional displacement coordinate. The HOMO-LUMO energy gap is indicated and the asterisk denotes the crossing between the distorted S1 and T2 excited states (see text).

We next calculated potential energy surfaces for the lowest energy singlet (S_0, S_1, S_2) and triplet (T_1, T_2, T_3) states along the most prominent displaced torsional mode, i.e., the effective 340 cm^{-1} mode. Comparison with simulated Raman spectra confirms that modes in this low frequency region involve out-of-plane dTV torsional motions that are known to exert a large influence on the photophysics of related molecules.[215–217] Potential energy surfaces are shown in Fig. 7.4 b revealing large energetic barriers for converting to the cis form of dTV in S_0. These barriers are substantially relaxed in the S1 state although the trans form remains the lowest in energy by ~1 eV. We also performed TD-DFT simulations by optimizing the dTV structure in the S1 state and found a small decrease (~0.17 eV) in the locations of potential minima compared to vertically projecting the ground state geometry.

It is useful to point out the predicted intersection between S1 and a higher energy triplet state, T_2, corresponding to a distortion of 50° along the torsional coordinate. These profiles indicate that the T_2 state is the likely conduit for entering the triplet manifold due to minimization of the singlet-triplet energy gap.[209,218] Interestingly, earlier work on related compounds proposed that intersystem crossing requires large conformational deformations leading to isostructural cisoid singlet and triplet excited states.[217] Moreover, recent non-adiabatic excited state simulations of model thiophene oligomers reported intersystem crossing via the S_1-T_2 intersection as the principal relaxation channel.[219] This process was also found to be driven by inter-ring torsional motions and ring opening deformations that becomes impeded in larger molecules. These authors predicted the initial intersystem crossing time constant to be ~140 ps followed by relaxation to the lowest energy T_1 state occurred on much longer time scales (~1.3 ns) and eventual recovery to S_0 (~7 μs).[219]

Based on predictions from DFT simulations and prior work on related molecules,[217–219] we assign the ca. 200 ps decay constant in both media to transitions involving a distorted (cisoid) S_1 state, consistent with the broader spectral lineshape at ~2.2 eV. This geometry enables facile intersystem crossing to the nearly isoenergetic T_2 state with the same geometry due to minimization of the singlet-triplet energy gap. A caveat of this assignment is that

it requires excess excitation energy to surmount the potential energy barrier of torsional motion to access the cisoid S_1 state. Slower transitions to T_1 are also possible probably involving the relaxed S_1 state that we assign as the ca. 1.4 ns dominated by the ~1.7 eV absorption feature. The appearance of the prominent isosbestic point in transient absorption spectra (Fig. 7.3) demonstrates that both pathways to T_1 are operative. However, conformational distortions expedite the intersystem crossing transition which is consistent with high sensitivity of associated dynamics to the surrounding medium. Figure 7.5 illustrates the proposed relaxation mechanisms in dTV.

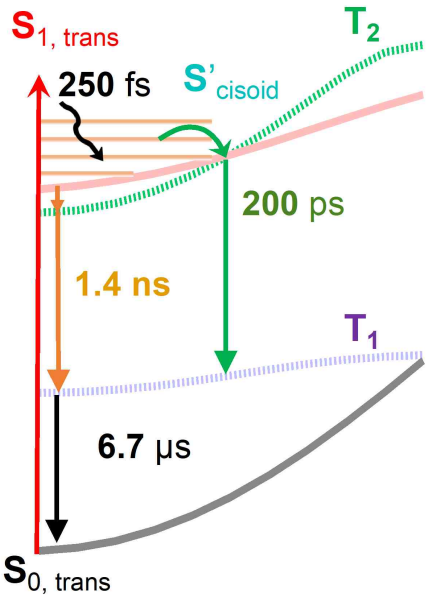

Figure 7.5: Proposed excited state relaxation pathways of dTV.

Although earlier work demonstrated the prevalence of ring opening and ring deformations as possible channels for accessing the triplet manifold, it is likely that such mechanisms would result in the appearance of distinct photoproducts (e.g., cis conformers).[219] These forms should possess markedly different absorption energies and lineshapes over time. No significant changes in absorption lineshapes were observed indicating that intersystem crossing and relaxation within the triplet manifold dominate relaxation dynamics.

7.3 Conclusions

We have demonstrated the importance of large conformational displacements in transitioning into the triplet manifold in a small thienylene-vinylene oligomer. Our results offer additional perspectives of triplet signatures appearing on fast time scales in related polymer systems that are usually interpreted on the basis of multi-step processes involving pure electronic states. The fact that triplet signatures vanish in aggregating polymer analogs of dTV and are instead dominated by non-radiative relaxation involving many vibrational modes demonstrates that specific geometric distortions govern intersystem crossing in these systems. Furthermore, controlling aggregation characteristics as well as vibrational displacements may be accomplished straightforwardly by simply varying processing approaches.

8.0 Future Work

8.1 Counting triplets on single polymer chains for solar cells

There is a growing interest to utilize triplet excitons within organic photovoltaics containing conjugated polymers. Current organic photovoltaics utilizing conjugated polymers and non-fullerene acceptors are now achieving >15%[220] power conversion efficiencies, yet this is still well below the Shockley-Queisser (S-Q) limit of 33%.[221] Multiple exciton generation (MEG) is proposed to theoretically beat the S-Q limit.[222–224] Designing conjugated polymers for multiple exciton generation can not only further increase the power conversion efficiency but could surpass the S-Q limit. Current design strategies for developing conjugated polymers for increased yields of triplet excitons is limited by measuring the complex interactions of multiple exciton species. Improving our understanding of singlet-triplet quenching and triplet-triplet annihilation in conjugated polymers is a key goal, and can be achieved through further development of pairing stochastic modeling and single molecule kinetics.

Single molecule spectroscopy of conjugated polymers allows measuring the distributions of rates for singlet and triplet formation. Unlike bulk spectroscopy where the measurement is reduced to an ensemble average over all conformations of the polymer chains. Improvement towards efficiently fitting time-dependent stochastic kinetics with each single polymer chain measured will allow for greater insight of the complex interactions between chromophores. Current effort toward this goal is underway using efficient optimization algorithms developed in the machine learning community and applying the stochastic kinetics developed in this dissertation toward understanding triplet formation in J-aggregate

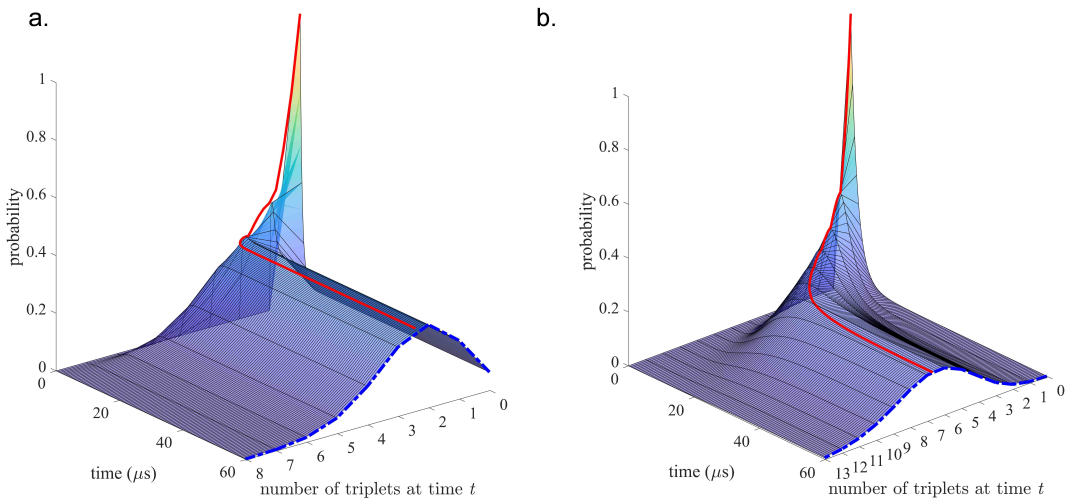

Figure 8.1: Conditional probability surfaces of the triplet occupancy showing the time-dependence of the probility of having N triplets at time t for P3HS (a) and P3HT (b). The red solid line shows the time-dependence of the average number of triplets. The blue dashed line highlights the resulting steady-state distribution of the number of triplets

P3HT nanofibers. This combination allows for direct determination of rate constants for conjugated polymers not easily ascertained from bulk spectroscopy (singlet-triplet quenching and triplet-triplet annihilation) and gives the ability to follow the dynamics of discrete triplet populations. Figure 9.1 shows the dynamics for both P3HS and P3HT using the values and procedures shown in chapter 5. From inspection of Figure 9.1, one can notice the distribution of the triplet occupancy on a single chain evolves in time, highlighting the complexity in the kinetics. Overall the work presented can be used for further investigated triplet exciton kinetics in conjugated polymers to provide new design strategies for optimizing yields of excitonic species by balancing singlet-triplet and triplet-triplet interactions.

9.0 Appendices

9.1 Appendix A: Steady-state solution to the time-dependent triplet probability master equation

The rate equation of the conditional probability of having n triplets at time t, $P_n(t)$ does not have an analytical solution, however the steady-state solution does have an analyical solution when assuming a forward constant rate. The steady-state solution is,

$$P_n(\infty) = \frac{2^{(\beta-1-3n)/2}\alpha^n}{n!}\frac{I_{\beta-1+n}(\alpha/\sqrt{2})}{I_{\beta-1}(\alpha)} \tag{9.1}$$

where $\alpha = 4(k_f/k_{TT})^{1/2}$, $\beta = 2k_b/k_{TT}$, and $I_p(x)$ denotes the modified Bessel function of the first kind.[56] Shown below is the output of the MATLAB implentation of the above equation.

To call the function use:

```
% P3HT Values
kisc = 1e9;
kiscR = 1e4;
tauFL = 500e-12;
ktt = 1e4;
kexc = 1e6;
kst = 5e8;
Nt = 14;
```

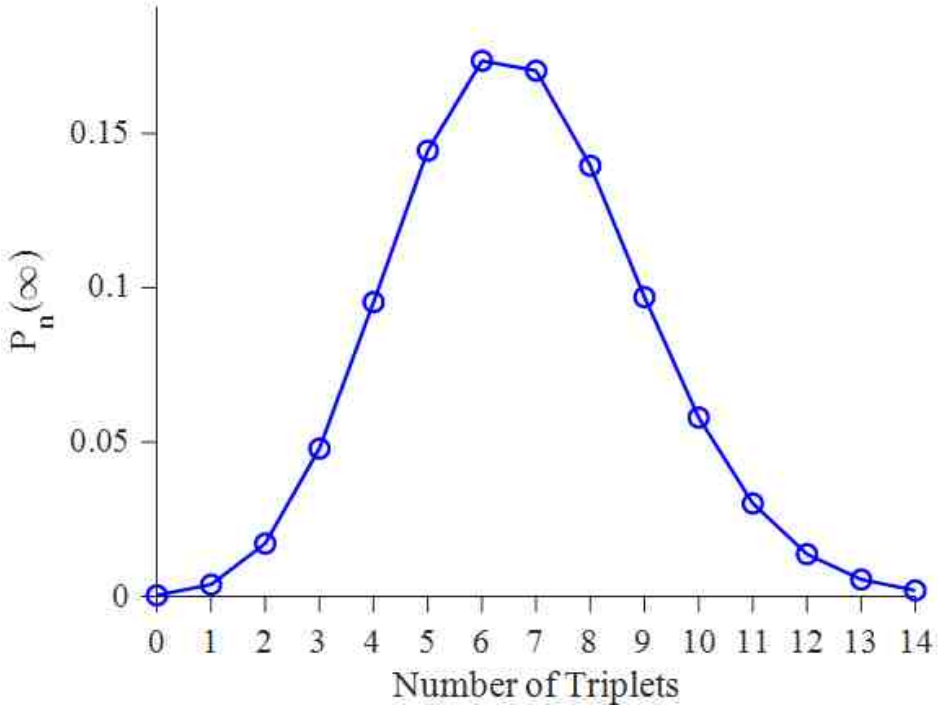

Figure 9.1: Example output from the function `SimTripletTachiya`. See text for input parameters.

```
Results = SimTripletTachiya(...
kisc,...
kiscR,...
tauFL,...
ktt,...
kexc,...
Nt,...
0);
```

MATLAB implentation:

```
function Results = SimTripletTachiya(kisc,kiscR,tauFL,ktt,
   kexc,Nt,YorN,kq,plotYorN)
% To call function:
```

```
%   >> Results = SimTripletTachiya(3e10,2e5,18e-12,2e3,8e4
   ,5,0);
% Input:
%   kisc: rate constant for intersytem crossing
%   kiscR: rate constant for reverse intersytem crossing
%   tauFL: fluoresence lifetime
%   ktt: rate constant for triplet-triplet annihliation -->
   Vary ktt
%   kexc: rate constant of excitation --> Vary kexc
%   Nt: number of triplets
%   YorN: Either 1 or 0 where
%       1 = run simulation calculating kf0 dependent on the
   number of
%       triplets
%       0 = run simulation calculating kf0 independent on
   the number of
%       triplets
% Output:
%   Results = (Nt+1,2);
%       where Results(:,1) is k0f for every iteration of 0:
   Nt
%       and Results(:,2) is Pn for every iteration of 0:Nt
%
% References: 1. Barzykin AV; Tachiya M.
%               J. Phys. Chem. B, 2006, 110 (13), pp 7068
    7072
%             2. Pensack RD; Song Y; McCormick TM; Jahnke AA
```

```
   ; Hollinger J;
%                Seferos DS; Scholes GD.
%                J. Phys. Chem. B 2014, 118 (9), 2589 2597
%
% P3HS experimental inputs
% kisc = 3e10 Taking from Pensack
% kiscR = 2e5 rev ISC from expt
% tauFL = 26e-12 Taking from Pensack
% ktt = 2e10 --> Vary ktt
% kexc = 8e6 ~1e20 ph/cm2/s, sigma=1e-15 --> Vary kexc
% Nt = 5
%
% Parameters used to reporduce Figure 3b in Reference 1
% kisc = 7e6
% kiscR = 2e3
% tauFL = 3e-9
% ktt = 5e5
% kexc = 5e4
% Nt = 3


% check to make sure user enter 1 or 0 for YorN
msg = ['Please enter 1 or 0. 1 for calculating kf dependent
   on Nt',...
    '0 for calculating kf0 with forward constant rate'];

if ~(YorN == 1) && ~(YorN == 0)
```

```
    warning(msg);
    Results = [];
    return
end

% calculate beta
beta = (2 * kiscR)/ktt;

% set up storage Pn and kf0
Pn = zeros(Nt,1);
kf0 = zeros(Nt,1);

% check to see if user wants to change kf0 on every n
if YorN == 1
    % kq is singlet triplet quenching
%     kq = 1e5;

    % loop through the number of triplets desited starting
       with zero
    for n=0:Nt

        % calculate kf0
        kf0(n+1)=kexc*kisc*tauFL/(1+(kq*tauFL));

        % calculate alpha
        alpha = 4*sqrt((kf0(n+1)/ktt));
```

```
        a = 2^((beta-1-3*n)/2);
        b = alpha^n;

        % first bessel function
        nu1 = beta - 1 + n;
        Z1 = alpha/sqrt(2);

        % 2nd bessel function
        nu2 = beta - 1;
        Z2 = alpha;

        % put it all together
        Pn(n+1) = ((a * b)/factorial(n)) * (besseli(nu1,Z1)/
           besseli(nu2,Z2));
    end

    Results = [kf0, Pn];
else

    % set up kf0
    kf0 = (kexc*kisc*tauFL);

    % set up alpha
    alpha = 4*sqrt((kf0/ktt));

    for n=0:Nt
```

```
        a = 2^((beta-1-3*n)/2);
        b = alpha^n;

        % first bessel function
        nu1 = beta - 1 + n;
        Z1 = alpha/sqrt(2);

        % 2nd bessel function
        nu2 = beta - 1;
        Z2 = alpha;

        % put it all together
        Pn(n+1) = ((a * b)/factorial(n)) * (besseli(nu1,Z1)/
           besseli(nu2,Z2));
    end

    % remap kf0
    kf0 = repmat(kf0,Nt+1,1);

    Results = [kf0, Pn];
end

if nargin < 9
    plotYorN = 1;
end

if plotYorN
```

```
% plot results
figure;

strXLabel = 'Number of Triplets';
strYLabel = 'P_n(\infty)';
astrPlotLineWidth = 1.5;
astrPlotMarkerSize = 8;
iFontSize = 14;
strFontName = 'Times';

plot(0:Nt,Results(:,2),'b-o',...
    'MarkerSize',astrPlotMarkerSize,...
    'LineWidth',astrPlotLineWidth);

set(gca,...
    'FontSize',iFontSize,...
    'FontName',strFontName,...
    'XMinorTick', 'off',...
    'YMinorTick', 'off',...
    'TickDir', 'out',...
    'TickLength', [0.02 0.02],...
    'XTick', 0:1:Nt,...
    'Xlim', [0 Nt],...
    'Ylim', [0 max(Results(:,2))*1.1]);

box off
```

```
    xlabel(strXLabel);
    ylabel(strYLabel);

    % if YorN
    %     str = {'ksic = ', num2str(kisc);
    %     'kiscR = ', num2str(kiscR);
    %     'tauFL = ', num2str(tauFL);
    %     'ktt = ', num2str(ktt);
    %     'kexc = ', num2str(kexc);
    %      'kq = ', num2str(kq)};
    % else
    %     str = {'ksic = ', num2str(kisc);
    %     'kiscR = ', num2str(kiscR);
    %     'tauFL = ', num2str(tauFL);
    %     'ktt = ', num2str(ktt);
    %     'kexc = ', num2str(kexc);};
    % end

    % annotation('textbox',[0.55,0.3,.3,.5],'String',str)

end

end
```

9.2 Appendix B: Numerical solution for the time-depdence of $P_n(t)$

We employ the approach of Birtwistle and Blackley[165,225] to solve the Smith-Ewart equation, implented in MATLAB with named function `nTripletTimeDepend` (see below). The full form is solved using the iterative approach of Gauss-Seidel, and discrete time intervals. The discrete form then becomes,

$$\begin{aligned}\frac{P_{n,q+1}-P_{n,q}}{h} = (1-\theta)\Big[&(P_{n-1,q}-P_{n,q})k_{f,n} + \{(n+1)P_{n+1,q} - nP_{n,q}\}k_b \\ &+\{(n+2)(n+1)P_{n+2,q} - n(n-1)P_{n,q}\}\frac{k_{TT}}{2}\Big] \\ +\theta\Big[&(P_{n-1,q+1}-P_{n,q+1})k_{f,n} + \{(n+1)P_{n+1,q+1} - nP_{n,q+1}k_b\} \\ &+\{(n+2)(n+1)P_{n+2,q+1} - n(n-1)P_{n,q+1}\}\frac{k_{TT}}{2}\Big]\end{aligned} \tag{9.2}$$

Changing from continuous to discrete time intervals was made through substitution of $t = qh$, where q is the ever incrementing timestep and h is the time interval for the step. Here, $P_n(t)$ was replaced with $P_(n,q)$ and $\frac{d}{dt}P_n(t)$ was replaced with the forward finte difference, $(P_{n,q+1} - P_{n,q})/h$. The value θ a weighting parameter between zero and one. The value is set fixed to 0.5 as the authors prescribed in their discussion of the stability of the iterative procedure.[165] Eq. 9.2 can be solved with the following initial conditions,

$$t = 0 \begin{cases} P_{0,0} = 1, & \\ P_{n>0,0} = 0, & \text{for } n = 1, 2, ..., n_{max} \\ P_{0,1} = & \\ \Big(\theta\{k_b P_1, 1 + k_{TT}P_{2,1}\} + \frac{1}{h} - (1-\theta)k_{f,n}\Big)/\Big(\frac{1}{h} + \theta k_{f,n}\Big) & \end{cases} \tag{9.3}$$

It is important to note the terms $P_{n-1,q}$ and $P_{n-1,q+1}$ are set to zero when $n-1 < 0$. The full form of the Smith-Ewart equation in our model includes infinite number of triplet states,

and therefore must be truncated at a given n_{max}.

The function `nTripletTimeDepend` may be called with no inputs, when done so a dialog will appear asking the user for all nessceary parameters. The dialog is shown in figure 9.2. Otherwise the function can be called from the command line.

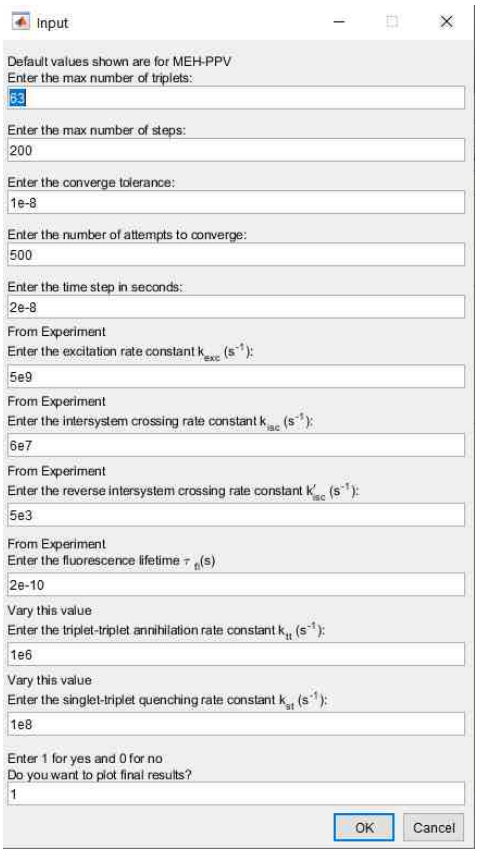

Figure 9.2: Input dialog if the function `nTripletTimeDepend` is called with no inputs.

The example below shows how to call the function from the commandline:

```
% P3HT Values
kisc = 1e9;
kiscR = 1e4;
tauFL = 500e-12;
ktt = 1e4;
kexc = 1e6;
kst = 5e8;
```

```
nMax = 63;

qMax = 300;


Nt = 13;


timestep = 1/(500*kiscR);


[Pfinal,t,I,P_avg,P_var]= nTripletTimeDepend(nMax,qMax,1e
   -8,500,timestep,kexc,kisc,kiscR,tauFL,ktt,kst,1,'constant
   ');
```

If the user chooses to plot the results, Figures 9.3 through 9.5 will be displayed. Figure 9.3 show the total probability and convergence of the error, which is used to verify the simulation hasn't leaked probability and has converged within the given tolerance. The average and variance of the triplet population is shown in figure 9.4. Where the values shown are calculated from the final converged $P_n(t)$ using the defination of the expectation and variance,[226–228]

$$< n >_t = \sum_{n=0}^{n_0} nP_n(t) \tag{9.4a}$$

$$\sigma_t^2 = \sum_{n=0}^{n_0} n^2 P_n(t) - < n >_t^2 \tag{9.4b}$$

Figure 9.5 shows the normalized time-dependent fluorescence intensity where $I(0)$ is assumed to be one and $I(t)$ is

$$I(t) = I(0) \sum_{n=0}^{\infty} \frac{P_n(t)}{1 + nk_{QST}\tau_{fl}} \tag{9.5}$$

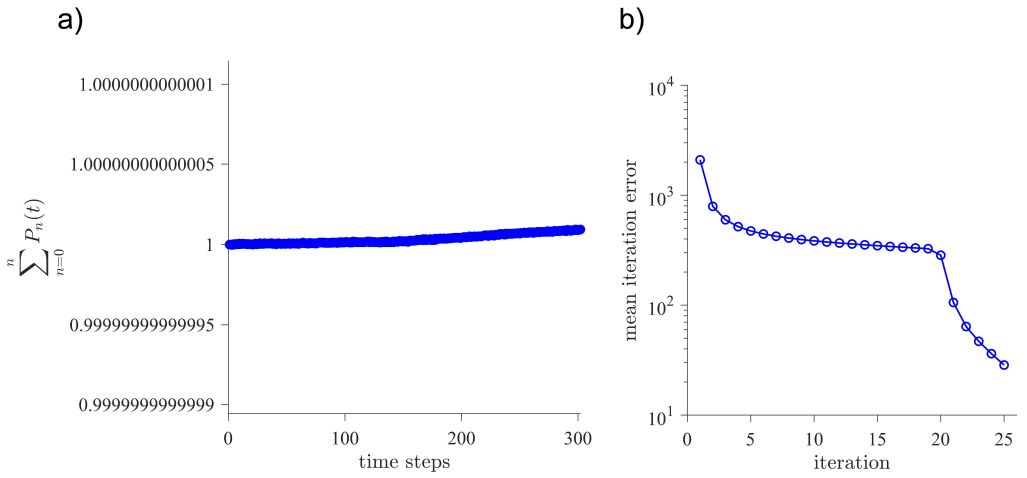

Figure 9.3: a) The sum of all probability over the length of the simulation where no significant probability leakage is observed. b) The trace of the mean iteration error. Using the above parameters and tolerance of $1E-8$ the simulation convergences within within 26 steps having a max error of $3.9612e-09$

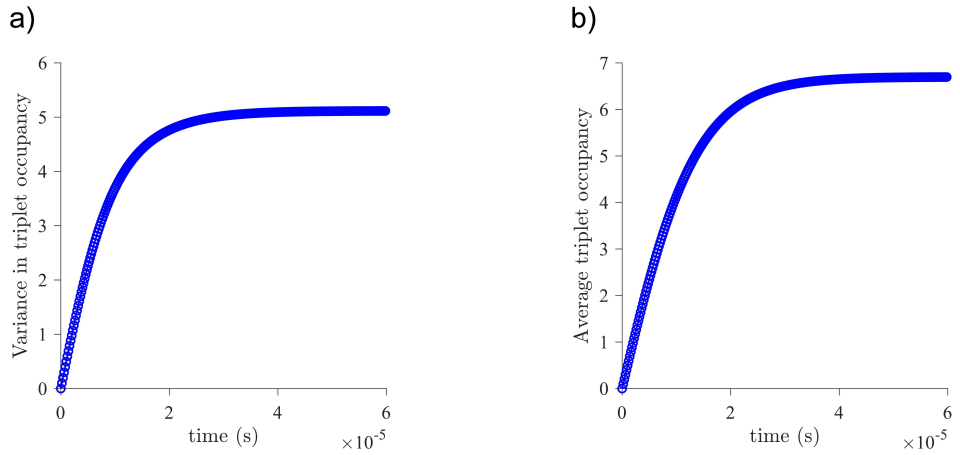

Figure 9.4: The variance (a) and average (b) time-dependence of having n triplet occupancy.

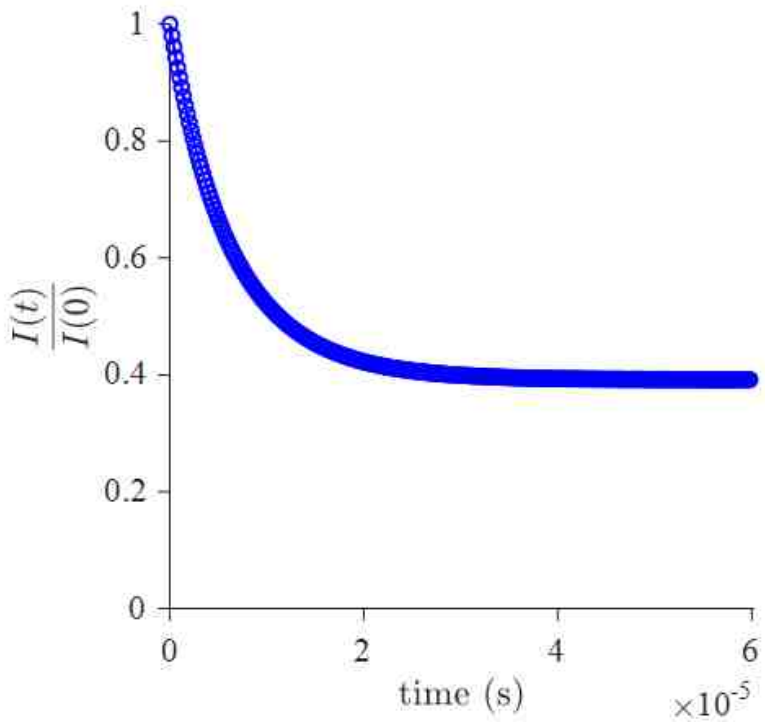

Figure 9.5: The normalized time-dependent fluorescence intensity, $I(t)/I(0)$.

```
function [Pfinal,t,I,P_avg,P_var] = nTripletTimeDepend(
    varargin)
%%
% Author: Benjamin D. Datko
%
% 12 Dec, 2017
%% Purpose:
% Solves the time-dependent Smith-Ewart differential
    difference
% equation using finite differences and Gauss-Seidel
    iterative method.
% Will calculate the time evolution of n triplets for a
    given time
```

```
% with discrete timesteps.
%
%% Input:
% If function is called with zero inputs the function will
   ask for all
% relvant data before starting
%
%   nMax = the maximum number of triplets to calculate
   including zero
%   qMax = the maximum number of time steps
%   tol = the tolerance for the convergence of error.
   recommend pick
%       something small, 1e-8
%   iterNum = the number of attempts to converge with the
   given 'tol'
%   timestep = time step in seconds (s^-1)
%   kexc = rate constant of excitation (s^-1)
%   kisc = rate constant of intersystem crossing (s^-1)
%   kiscR = rate constant of reverse intersystem crossing (s
   ^-1)
%   tauFL = fluorescence lifetime (s)
%   ktt = rate constant of triplet-triplet annihilation
%   kst = rate constant of singlet quenching by triplets
%   PlotYorN = yes = 1 no = 0, question to plot results
   after done
%       calculating
%% Optional Input:
```

```
% The last input allows the user to choose how the effective
    rate
% constant of triplet formation to be calculated. Either
   three words
% shown below and you must have quotes around the word.
%
% If no value is given the default value is 'constant'.
%
%  'constant' = assumes the constant forward rate
%
% $$ k_{f,n} \approx k_{f,0} = k^0_f  = k_{exc} k_{isc} \tau
    _{fl}$$
%
%  'full' = uses the full form for the rate
%
% $$ k_{f,n} = \frac{k_{exc} k_{isc} \tau _{fl}}{1 + k_{isc}
    \tau _{fl} + n k_{st} \tau _{fl}}$$
%
%  'approx' = uses the approximate form of the rate
%
% $$ k_{f,n} = \frac{k_{exc} k_{isc} \tau _{fl}}{1+n k_{st}
   \tau _{fl}} $$
%
%% Output:
%  Pfinal = resulting matrix of the time-dependent
   probabilities for n
%   triplets
```

```
%       The rows of Pfinal are the time steps starting with
   row 1 at t = 0
%       The columns of Pfinal are the time-dependent
   probability of the
%           nth triplet starting with column 1 at n = 0
%       m x n where:
%           m = the maximum number of time steps, 'qMax'
%           n = the maximum number of triplets, 'nMax'
%  t = the linear spaced time vector starting at 0 ending at
%   (qMax-1)*timestep incremented by timestep
%  I = the time-dependent intensity vector for all triplets.
    same size as t
%
% $$ I(t) = I(0) \sum^{\infty}_{n=0} \frac{P_n(t)}{1+nk_{st
   }\tau _{fl}} $$
%
%% References:
% # Barzykin AV; Tachiya M. J. Phys. Chem. B 2006, 110 (13),
    7068 7072. <http://pubs.acs.org/doi/abs/10.1021/
   jp056088q Link>
% # Birtwistle, D. T.; Blackley, D. C. J. Chem. Soc. Faraday
    Trans. 1981, 77 (6), 1351 1351 DOI: 10.1039/f19817701351
   . <http://pubs.rsc.org/en/content/articlelanding/1981/f1/
   f19817701351#!divAbstract Link>
% # Datko B.D; Thomas A.K; Grey J.K; Fei Z; Heeney M. Phys.
   Chem. Chem. Phys. 2017, 19 (41), 28239 28248 DOI:
   10.1039/c7cp05476a. <http://pubs.rsc.org/-/content/
```

```
   articlehtml/2017/cp/c7cp05476a Link>
%
%% Notes
% The original Smith-Ewart differential difference was
   developed for
% free-radical polymerisation. Regardless, the equation is
   easily adaptable
% to triplet formation on a conjugated polymer. [Ref 1] All
   equations and
% code has been rewritten to reflect this change. For the
   reader's sake
% when reading Ref 2, a key is given for relating terms from
    free-radical
% polymerisation to triplet formation:
%
% $\sigma = k_{f,n}$ the effective rate constant of triplet
   formation
%
% $k = k^{\prime}_{isc}$ the reverse intersystem crossing
   rate constant
%
% $\chi = \frac{k_{tt}}{2}$ first order rate constant of
   triplet-triplet formation
%
%
%% Example Output
% The figures in the bottom of this document were generated
```

```
   using default
% values for all inputs. To reproduce these results call the
    function as
% such:
%
%  [Pfinal,t,I] = nTripletTimeDepend;
%
% Hit "Ok" when prompted to enter values and then select "
   Select" to
% calculate with the constant forward rate assumption. Then
   the number of
% convergence steps it took to converge will display on the
   command line
% and the three figures will then be displayed.

% if the function is called with no inputs will ask the user
if nargin == 0
    prompt = {sprintf('Default values shown are for MEH-PPV\
       nEnter the max number of triplets:'),...
        'Enter the max number of steps:',...
        'Enter the converge tolerance:',...
        'Enter the number of attempts to converge:',...
        'Enter the time step in seconds:',...
        sprintf('From Experiment\nEnter the excitation rate
           constant k_{exc} (s^{-1}):'),...
        sprintf('From Experiment\nEnter the intersystem
           crossing rate constant k_{isc} (s^{-1}):'),...
```

```
    sprintf('From Experiment\nEnter the reverse
       intersystem crossing rate constant k^{\\prime}_{
       isc} (s^{-1}):'),...
    sprintf('From Experiment\nEnter the fluorescence
       lifetime \\tau _{fl}(s)'),...
    sprintf('Vary this value\nEnter the triplet-triplet
       annihilation rate constant k_{tt} (s^{-1}):'),...
    sprintf('Vary this value\nEnter the singlet-triplet
       quenching rate constant k_{st} (s^{-1}):'),...
    sprintf('Enter 1 for yes and 0 for no\nDo you want
       to plot final results?')};
dlg_title = 'Input';
num_lines = 1;
defaultans = {'63','200','1e-8','500','2e-8','5e9','6e7'
   ,'5e3','2e-10','1e6','1e8','1'};
options.Interpreter = 'tex';
answer = inputdlg(prompt,dlg_title,num_lines,defaultans,
   options);

% check to see if user canceld
if isempty(answer)
    % user canceld
    Pfinal = [];
    t = [];
    I = [];
    return
end
```

```
    inputs = zeros(length(answer),1);
    for i=1:length(answer)
        inputs(i) = str2double(answer{i});
    end

    % get the inputs
    nMax = inputs(1);
    qMax = inputs(2);
    tol = inputs(3);
    iterNum = inputs(4);
    timestep = inputs(5);
    kexc = inputs(6);
    kisc = inputs(7);
    kiscR = inputs(8);
    tauFL = inputs(9);
    ktt = inputs(10);
    kst = inputs(11);
    PlotYorN = inputs(12);

    which_kf = choose_kf;

else
    % get the inputs
    nMax = varargin{1};
    qMax = varargin{2};
    tol = varargin{3};
```

```
    iterNum = varargin{4};
    timestep = varargin{5};
    kexc = varargin{6};
    kisc = varargin{7};
    kiscR = varargin{8};
    tauFL = varargin{9};
    ktt = varargin{10};
    kst = varargin{11};
    PlotYorN = varargin{12};

    if nargin == 13
        which_kf = varargin{13};
    else
        which_kf = 'constant';
    end
end

switch which_kf

    case 'constant'
        % default is the forward constant rate assumption
        % constant forward rate assumption
        kf = kexc * kisc * tauFL;
        kf = repmat(kf, [nMax + 1, 1]);

    case 'full'
        % full effective rate constant
```

```
        kf = zeros(nMax + 1, 1);
        for n=0:nMax
            kf(n+1) = (kexc * kisc * tauFL)./(1+(kisc*tauFL)
               +(n*kst*tauFL));
        end

    case 'approx'
        % approximat effective rate constant
        kf = zeros(nMax + 1, 1);
        for n=0:nMax
            kf(n+1) = (kexc * kisc * tauFL)./(1+(n*kst*tauFL
               ));
        end
end

% create the initial guess matrix P. P must be size(qMax +
   2, nMax +3)
% first row of P is the initial conditions such that
% P(1,:) = [1, zeros(1,qMax+1] -> (1, 0, 0, 0, 0, ....)
P = zeros(1, nMax + 1);
P(1,1) = 1;
tmp = repmat([0.99, linspace(1e-6, 1e-7, nMax )], [qMax + 1
   1]);
P = [P; tmp];
P = [P, zeros(qMax + 2, 2)];

Pold = P(1:qMax+2,1:nMax+1);
```

```
b = zeros(qMax+1,nMax+1);
theta=0.5;

err_Store = zeros(iterNum,qMax);

% only start time when the user askes for plotting results
if PlotYorN
    tic
end

% loop over 0 to q
for q=0:qMax
    iteration = 1;
    err = 100000;
    while max(err(:))>tol %iteration <= iterNum &&
        % loop over 0 to n
        for n=0:nMax

            if n - 1 < 0
                tmp_b = 0;
                tmp_P = 0;
            else
                tmp_b = kf(n+1)*P(q+1,n+1-1);
                tmp_P = kf(n+1)*P(q+1+1,n+1-1);
            end

            b_part1 = P(q+1,n+1)*((1/timestep)-(1-theta)*(kf
```

```
            (n+1)+n*kiscR+n*(n-1)*(ktt*0.5)));
        b_part2 = (1-theta)*(tmp_b+kiscR*(n+1)*P(q+1,n
            +1+1)+(ktt*0.5)*(n+2)*(n+1)*P(q+1,n+1+1+1));
        b(q+1,n+1) = b_part1 + b_part2;

        %disp(['q = ', num2str(q), ' n = ', num2str(n),
            ' iteration = ', num2str(iteration)]);
        P_numerator = theta*(tmp_P+kiscR*(n+1)*P(q+1+1,n
            +1+1)+(ktt*0.5)*(n+2)*(n+1)*P(q+1+1,n+1+2)) +
             b(q+1,n+1);
        P_denominator = (1/timestep) + theta*(kf(n+1)+n*
            kiscR+n*(n-1)*(ktt*0.5));


        P(q+1+1,n+1) = P_numerator/P_denominator;

    end

    err = abs((P(q+2,1:nMax+1) - Pold(q+2,:)) ./ P(q
        +2,1:nMax+1));
    Pold = P(1:qMax+2,1:nMax+1);

    %disp(['iteration = ', num2str(iteration), ' err =
        ', num2str(max(err(:)))]);
    err_Store(iteration,q+1) = max(err(:));
    iteration = iteration + 1;
end
```

```
end

% only stop time when the user askes for plotting results
if PlotYorN
    toc
end

% Display output
if iteration >= iterNum
    str = ['Hit the maximum number of iterations: ', num2str
       (iteration-1),...
        ' Max err = ', num2str(max(err(:)))];
    disp(str)
elseif max(err(:)) <= tol && PlotYorN
    str = ['Converged to tol ', num2str(tol), ' within ',
       num2str(iteration),' Max err = ', num2str(max(err(:))
       )];
    disp(str)
end

% clean up any unwatned zeros
rowsWithZeros = any(err_Store == 0, 2);
err_Store = err_Store(~rowsWithZeros, :);

% calculate intenisty assuming I(0) is 1
Pfinal = P(1:qMax,1:nMax);
```

```
P_avg = zeros(qMax,1);

P_var = zeros(qMax,1);

% the definition of the expectation value or average value
   at time t
for n=0:nMax-1

    P_avg = P_avg + n*Pfinal(:,n+1);

end

% the definition of the variance
for n=0:nMax-1

    P_var = P_var + (n - P_avg).^2.*(Pfinal(:,n+1));

end

t = (0:qMax-1)*timestep;
t = t';

I = zeros(qMax,1);

for n=0:nMax-1

    I = I + Pfinal(:,n+1)/(1+(n*kst*tauFL));
```

```
end



% plot results if user wants
if PlotYorN
    % plot opitions
    astrPlotLineWidth = 1.25;
    astrPlotMarkerSize = 6;
    iFontSize = 14;
    strFontName = 'Times';

    % plot err vs iteration
    figure;
    %     subplot(1,3,1);
    semilogy(mean(err_Store,2),'b-o',...
        'MarkerSize',astrPlotMarkerSize,...
        'LineWidth',astrPlotLineWidth);

    MaxIterHitPlus1 = (size(mean(err_Store,2),1) + 1);

    set(gca,...
        'FontSize',iFontSize,...
        'FontName',strFontName,...
        'XMinorTick', 'off',...
        'YMinorTick', 'on',...
```

```
    'TickDir', 'out',...
    'TickLength', [0.02 0.02],...
    'Xlim', [0 MaxIterHitPlus1],...
    'Box','off');

axis(gca,'square');

strXLabel = 'iteration';
strYLabel = 'mean iteration error';

xlabel(strXLabel,'Interpreter','latex')
ylabel(strYLabel,'Interpreter','latex')

set(gcf,'Color','w');

% plot P_avg(t) vs time
figure;
plot(t,P_avg,'b-o',...
    'MarkerSize',astrPlotMarkerSize,...
    'LineWidth',astrPlotLineWidth);

set(gca,...
    'FontSize',iFontSize,...
    'FontName',strFontName,...
    'XMinorTick', 'off',...
    'YMinorTick', 'off',...
    'TickDir', 'out',...
```

```
    'TickLength', [0.02 0.02],...
    'Box','off');

set(gcf,'Color','w');

axis(gca,'square');

strXLabel = 'time (s)';
strYLabel = 'Average triplet occupancy';

xlabel(strXLabel,'Interpreter','latex')
ylabel(strYLabel,'Interpreter','latex')

% plot P_var(t) vs time
figure;
plot(t,P_var,'b-o',...
    'MarkerSize',astrPlotMarkerSize,...
    'LineWidth',astrPlotLineWidth);

set(gca,...
    'FontSize',iFontSize,...
    'FontName',strFontName,...
    'XMinorTick', 'off',...
    'YMinorTick', 'off',...
    'TickDir', 'out',...
    'TickLength', [0.02 0.02],...
    'Box','off');
```

```
set(gcf,'Color','w');

axis(gca,'square');

strXLabel = 'time (s)';
strYLabel = 'Variance in triplet occupancy';

xlabel(strXLabel,'Interpreter','latex')
ylabel(strYLabel,'Interpreter','latex')

% plot I(t)/I(0) vs time
%     subplot(1,3,2);
figure;
plot(t,I,'b-o',...
    'MarkerSize',astrPlotMarkerSize,...
    'LineWidth',astrPlotLineWidth);

set(gca,...
    'FontSize',iFontSize,...
    'FontName',strFontName,...
    'XMinorTick', 'off',...
    'YMinorTick', 'off',...
    'TickDir', 'out',...
    'TickLength', [0.02 0.02],...
    'Xlim', [0 timestep*(qMax+1)],...
    'Ylim', [0 1],...
```

```
    'Box','off');

axis(gca,'square');

strXLabel = 'time (s)';
strYLabel = '$$ \frac{I(t)}{I(0)} $$';

xlabel(strXLabel,'Interpreter','latex')
ylabel(strYLabel,'Interpreter','latex')

set(gcf,'Color','w');

% check if probability is greater than 1
%     subplot(1,3,3);
figure;
plot(sum(P,2),'b-o',...
    'MarkerSize',astrPlotMarkerSize,...
    'LineWidth',astrPlotLineWidth);

set(gca,...
    'FontSize',iFontSize,...
    'FontName',strFontName,...
    'XMinorTick', 'off',...
    'YMinorTick', 'off',...
    'TickDir', 'out',...
    'TickLength', [0.02 0.02],...
    'Xlim', [0 size(err_Store,2) + 1],...
```

```
        'Box','off');

    axis(gca,'square');

    strXLabel = 'time steps';
    strYLabel = '$$ \sum^n_{n=0} P_n(t) $$';

    xlabel(strXLabel,'Interpreter','latex')
    ylabel(strYLabel,'Interpreter','latex')

    set(gcf,'Color','w');

end

% this function asks the user how they want to calculate the
    forward rate
% when no agruments are given
    function choice = choose_kf

        f = figure('Position',[300 300 250 150],'Name','
           Select One');
        set(f, 'MenuBar', 'none', 'ToolBar', 'none');
        movegui('center');
        uicontrol('Parent',f,...
            'Style','text',...
            'Position',[20 80 210 40],...
            'String','How should I calculate the effective
```

```
            rate constant of triplet formation?');

    popup = uicontrol('Parent',f,...
        'Style','popup',...
        'Position',[75 65 100 25],...
        'String',{'constant';'full';'approx'},...
        'Callback',@popup_callback);

    uicontrol('Parent',f,...
        'Position',[89 20 70 25],...
        'String','Select',...
        'Callback','delete(gcf)');

    value = get(popup,'Value');
    items = get(popup, 'String');
    choice = char(items(value,:));


    % Wait for d to close before running to completion
    uiwait(f);

    function popup_callback(popup,~)
        idx = popup.Value;
        popup_items = popup.String;
        choice = char(popup_items(idx,:));
    end
end
```

```
end
```

9.3 Appendix C: Single Molecule Experiment Image Software

Below are screen shots of the software in use, description of the code is in chapter 2.

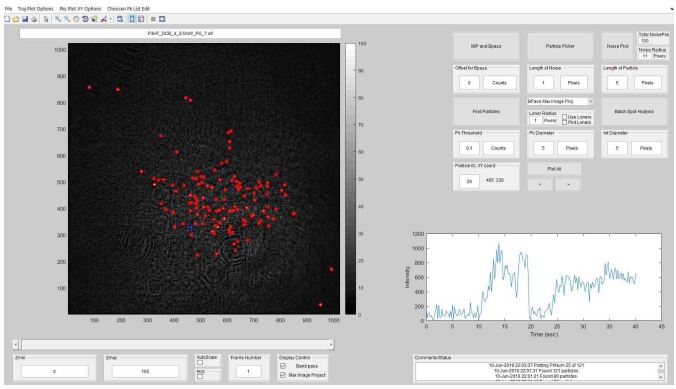

Figure 9.6: Image of user interface indicating the found particles shown in red and transients generated in the bottom right.

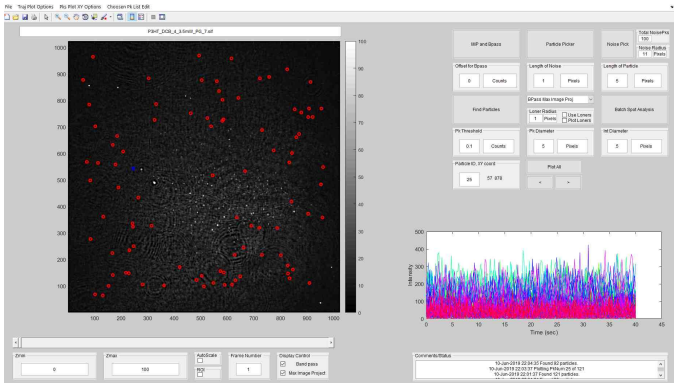

Figure 9.7: Image indicating noise particles found in red and transients generated in the bottom right.

9.4 Appendix D: Code for `SME_Image` software

The code below is shown ”AS IS”. There will be some issues and idiosyncrasies. Requires both `SME_ImageUI.m` and `acquisitionImage.m` are in same directory.

Dependcies:

- multiWaitbar by Ben Tordoff[229]
- Andor's[1] MATLAB sif reader[230]

[1] Contact Andor for the latest version

```
classdef SME_ImageUI < handle

    properties (Access = private)
        % Figure for the UI
        Figure

        % basic information
        appFilename
        appDatadirectory
        appNumFrames
        appSizeOfFrames
        appKineticCycleTime
        appFilepath

        % Menus for the figure and Items for the Menus
        % file menu options
        FileMenu
        LoadImageMenu
        SaveWorkMenu
        SaveAsMAT
        SaveAsCSV

        % particle trajectory plot menu
        PkTrajPlotMenu
        TimePlotChoice
        FramePlotChoice
```

```
% particle XY plot menu
PkXYPlotMenu

Pk_XY_Results
PkPlotXY_MIP
PkPlotXY_PP
PkPlotXY_CSV
NoisePksXY

ChoicePksXY_PP



% Edit choosen Particle List
ChoosenPksListMenu
DeletePksFromList

% Comment box to display error
CommentTitle
CommentEdit

% Name, axis, and slider for image
FilenameEdit
Axis
ImageSlider

% PushButton for load
```

```
LoaderPush

% PlotChoice
PlotChoiceTitle
PlotChoiceDropDown

% Push button for 'M'ax 'I'mage 'P'roject and 'B'and
    Pass
MIPandBpassPush

% Push button, image selection, edit for lnoise,
   edit for lobject,
% and the edit for threshold all for bpass
lnoiseTitle
lnoiseEdit
lnoiseUnitEdit
lobjectTitle
lobjectEdit
lobjectUnitEdit
thresholdEditTitle
thresholdEdit
thresholdUnitEdit

% Push button for particle picker
ppPush

% Push button for noise particle
```

```
NoisePksPush

% Noise pick radius title, edit, and unit edit
NoiseRadiusTitle
NoiseRadiusEdit
NoiseRadiusUnitEdit

% Noise pick number title and edit
NoiseNumberTitle
NoiseNumberEdit

% Push button, image selection, edit for particle
   threshold, and
% the edit for the particle size
pkFindPush
pkFindDropDown
pkThresholdTitle
pkThresholdEdit
pkThresholdUnitEdit
pkSizeTitle
pkSizeEdit
pkSizeUnitEdit

% Intergration Diameter title, edit, and unit edit
intDiameterTitle
intDiameterEdit
intDiameterUnitEdit
```

```
% Edit Zmin Zmax
ZminTitle
ZminEdit
ZmaxTitle
ZmaxEdit
AutoScaleChkBx
AutoScaleTitle

% Edit Frame number
CurrentFrameNumberTitle
CurrentFrameNumberEdit

% ROI ChkBx
ROIChkBx
ROITitle

% ZoomOrigInfo and ZoomCoord
ZoomOrigInfo
ZoomNewInfo

% Analyze axis for hist and traj
AnalyzeAxis

% Current ParticleTitle, ParticleNum, and ParticleXY
ParticleTitle
ParticleNum
```

```
ParticleTextXY

% Particle Up and down pushbutton and and particle
   Plot all
% pushbutton
ParticleUp
ParticleDown
ParticlePlotAll

% display control
DisplayControlTitle
DisplayControlBpassChkBx
DisplayControlBpassTxt
DisplayControlMaxProjImageChkBx
DisplayControlMaxProjImageTxt

% loner parameters
lonerDiameterTitle
lonerDiameterEdit
lonerDiameterUnitEdit
lonerDiameterChkBox
lonerDiameterTxt
lonerDiameterPlotChkBx
lonerDiameterPlotTxt

% Pushbutton for Batch particle analysis
BatchSpotAnalysisPush
```

```
end % end for properties (private)

properties
    ImageData
end % end for properties (public)

methods
    % Constructor
    function app = SME_ImageUI

        %% Figure UI
        app.Figure = figure('Visible','on','NumberTitle'
           ,'off',...
            'Color',[0.8 0.8 0.8],'Tag','SME_Window','
               ToolBar',...
            'figure','MenuBar','none',...
            'Position', 0.6 * get(0,'ScreenSize'));

        % Move the window to the center of the screen
        movegui(app.Figure,'center');

        % sets the figure close function. This lets the
           class know
        % that the figure wants to close and thus the
           class should
        % cleanup in memory as well
```

```
% adopted from https://www.mathworks.com/
   matlabcentral/fileexchange/33816-example-
   using-a-matlab-class-to-control-a-gui
% Accessed 5 Sept 2017
set(app.Figure, ...
    'closerequestfcn', @(src,event) Close_fcn(
       app, ...
    src, event));

% Create the zoomHandle
zoomHandle = zoom;
zoomHandle.ActionPostCallback = @app.
   UpdateZoomInfo;

%% Uimenu file menu, and items
app.FileMenu = uimenu(app.Figure,'Label','File')
   ;
app.LoadImageMenu = uimenu(app.FileMenu,...
    'Label','Load Image','Callback',@app.
       loadImageCallback);
app.SaveWorkMenu = uimenu(app.FileMenu,'Label','
   Save Work');
app.SaveAsMAT = uimenu(app.SaveWorkMenu,...
    'Label','Save As *.mat','Callback',@app.
       SaveWorkCallback);
app.SaveAsCSV = uimenu(app.SaveWorkMenu,...
    'Label','Save As *.csv','Callback',@app.
```

```
        SaveWorkCallback);

% Particle Trajectory plot menu and items
app.PkTrajPlotMenu = uimenu(app.Figure,...
    'Label','Traj Plot Options');
app.TimePlotChoice = uimenu(app.PkTrajPlotMenu,'
   Label',...
    'Plot x-axis with Time','Checked','on',...
    'Callback',@app.PkTrajPlotMenuCallback);
app.FramePlotChoice = uimenu(app.PkTrajPlotMenu,
   'Label',...
    'Plot x-axis with Frame number',...
    'Callback',@app.PkTrajPlotMenuCallback);

% Particle Plot XY menu and items
app.PkXYPlotMenu = uimenu(app.Figure,...
    'Label','Pks Plot XY Options');

% plot opitions for Bpass Max image proj
   particles
app.Pk_XY_Results = uimenu(app.PkXYPlotMenu,'
   Label',...
    'Pk Result XY');

app.PkPlotXY_MIP = uimenu(app.Pk_XY_Results,'
   Label',...
    'BPass Max Image Proj','Checked','on','
```

```
        Callback',@app.PkXYPlotMenuCallback);

% plot opitions for particle picker
app.PkPlotXY_PP = uimenu(app.Pk_XY_Results,'
   Label',...
    'Particle picker','Callback',@app.
       PkXYPlotMenuCallback);

% plot opitions for CSV particles
app.PkPlotXY_CSV = uimenu(app.Pk_XY_Results,'
   Label',...
    'CSV','Callback',@app.PkXYPlotMenuCallback);

app.NoisePksXY = uimenu(app.Pk_XY_Results,'Label
   ',...
    'Noise Particles','Callback',@app.
       PkXYPlotMenuCallback);

app.ChoicePksXY_PP = uimenu(app.PkXYPlotMenu,'
   Label',...
    'Choosen Particles','Callback',@app.
       PkXYPlotMenuCallback);



% Pk List edit choosen
app.ChoosenPksListMenu = uimenu(app.Figure,...
```

```
    'Label','Choosen Pk List Edit');
app.DeletePksFromList = uimenu(app.
   ChoosenPksListMenu,'Label',...
    'Delete Particles','Callback',@app.
       DeletePksFromListCallback);

%% Edit comment box
% Position of the title box for Comments
CommentTitlePos = [0.61 0.01 0.38 0.08];

app.CommentTitle = uipanel('Parent',app.Figure,'
   Title',...
    'Comments/Status',...
    'Units','Normalized','Position',
       CommentTitlePos);

% Position of the edit box is relative to the
   title
CommentEditPos = CommentTitlePos + [0.005 0.005
   -0.01 -0.0275];

app.CommentEdit = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
    'min',0,'max',2,...
    'Style','edit','String',[datestr(datetime),
       ' Welcome!'],...
```

```
    'Position',CommentEditPos,'Enable','Inactive
       ');

%% Pushbutton MIP and Bpass
MIPandBpassPushPosition = [0.67 0.9 0.1 0.08];

% Push button and image selection for the
   average image
app.MIPandBpassPush = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','pushbutton','String','MIP and Bpass
       ',...
    'Callback',@app.MaxImageProjectCallback,...
    'Position',MIPandBpassPushPosition);

%% Pushbutton particle picker
app.ppPush = uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','pushbutton','String','Particle
       Picker',...
    'Callback',@app.ParticlePickerCallback,...
    'Position', [0.78 0.9 0.1 0.08]);

%% Pushbutton to pick noise particles
app.NoisePksPush = uicontrol('Parent',app.Figure
   ,...
```

```
    'Units','Normalized',...
    'Style','pushbutton','String','Noise Pick',
       ...
    'Callback',@app.NoisePksPushCallback,...
    'Position', [0.89 0.9 0.05 0.08]);

%% Edit Noise Radius Title
% Position for the Noise Radius Title
NoiseRadiusTitlePosition = [0.9425 0.9 0.05
   0.04];

app.NoiseRadiusTitle = uipanel('Parent',app.
   Figure,...
    'Title','Noise Radius',...
    'Units','Normalized',...
    'Position',NoiseRadiusTitlePosition);

% Position for the NoiseRadiusEditPosition
   relative to the
% NoiseRadiusTitlePosition
NoiseRadiusEditPosition =
   NoiseRadiusTitlePosition + ...
    [0.003 0.003 -0.03 -0.02];%[0.9455 0.9065
       0.02 0.0300];

app.NoiseRadiusEdit = uicontrol('Parent',app.
   Figure,...
```

```
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Position',NoiseRadiusEditPosition);

% Position for NoiseRadiusUnitEditPosition
   relative to
% NoiseRadiusTitlePosition
NoiseRadiusUnitEditPosition =
   NoiseRadiusTitlePosition + ...
    [0.025 0.003 -0.03 -0.02];

app.NoiseRadiusUnitEdit = uicontrol('Parent',app
   .Figure,...
    'Units','Normalized',...
    'Style','edit','String','Pixels',...
    'Position',NoiseRadiusUnitEditPosition,...
    'Enable','Inactive');

%% Edit Noise Number Title
% Position for the Noise Radius Title
NoiseNumberTitlePosition = [0.9425 0.9425 0.05
   0.04];

app.NoiseNumberTitle = uipanel('Parent',app.
   Figure,...
    'Title','Total NoisePks',...
    'Units','Normalized',...
```

```
    'Position',NoiseNumberTitlePosition);

% Position for the NoiseRadiusEditPosition
   relative to the
% NoiseRadiusTitlePosition
NoiseNumberEditPosition =
   NoiseNumberTitlePosition + ...
    [0.003 0.003 -0.03 -0.02];%[0.9455 0.9065
       0.02 0.0300];

app.NoiseNumberEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Position',NoiseNumberEditPosition);

%% Edit offset for bpass
thresholdEditTitlePosition = [0.67 0.81 0.1
   0.08];

app.thresholdEditTitle = uipanel('Parent',app.
   Figure,...
    'Title','Offset for Bpass',...
    'Units','Normalized',...
    'Position',thresholdEditTitlePosition);

% Position for the threshold Edit relative to
```

```
    threshold title
thresholdEditPosition =
    thresholdEditTitlePosition - ...
     [-0.01 -0.009 0.07 0.04];

app.thresholdEdit = uicontrol('Parent',app.
    Figure,...
     'Units','Normalized',...
     'Style','edit','String','0',...
     'Position',thresholdEditPosition);

% Position for threshold Unit relative to
    threshold title
thresholdUnitEditPosition =
    thresholdEditTitlePosition - ...
     [-0.045 -0.009 0.05 0.04];

app.lobjectUnitEdit = uicontrol('Parent',app.
    Figure,...
     'Units','Normalized',...
     'Style','edit','String','Counts',...
     'Position',thresholdUnitEditPosition,'Enable
        ','Inactive');

%% Edit length of noise
lnoiseTitlePosition = [0.78 0.81 0.1 0.08];
```

```
app.lnoiseTitle = uipanel('Parent',app.Figure,
   ...
    'Title','Length of Noise',...
    'Units','Normalized',...
    'Position',lnoiseTitlePosition);

app.lnoiseEdit = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Position',lnoiseTitlePosition + [0.01 0.009
        -0.07 -0.04]);

app.lnoiseUnitEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','Pixels',...
    'Position',lnoiseTitlePosition + ...
    [0.05 0.009 -0.06 -0.04],'Enable','Inactive'
       );

%% Edit length of particle
% Position for the lobject Title
lobjectTitlePosition = [0.89 0.81 0.1 0.08];

app.lobjectTitle = uipanel('Parent',app.Figure,
   ...
```

```
    'Title','Length of Particle',...
    'Units','Normalized',...
    'Position',lobjectTitlePosition);

app.lobjectEdit = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Position',...
    lobjectTitlePosition + [0.01 0.009 -0.07
       -0.04]);

% Position for lobject Unit relative to lobject
   title
lobjectUnitEditPosition = lobjectTitlePosition -
    ...
[-0.05 -0.009 0.06 0.04];

app.lobjectUnitEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','Pixels',...
    'Position',lobjectUnitEditPosition,'Enable',
       'Inactive');

%% Pushbutton-Popupmenu find particles
% Push button, image selection, edit for
```

```
    particle threshold,
% and the edit for the particle size
app.pkFindPush = uicontrol('Parent',app.Figure,
    ...
     'Units','Normalized',...
     'Style','pushbutton','String','Find
        Particles',...
     'Callback',@app.pkFindPushCallback,...
     'Position',[0.67 0.72 0.1 0.08]);

app.pkFindDropDown = uicontrol('Parent',app.
    Figure,...
     'Units','Normalized',...
     'Style','popupmenu',...
     'String',{'BPass Max Image Proj' 'Particle
        Pick' 'CSV' 'Noise'},...
     'Position',[0.78 0.77 0.1 0.03]);

%% Edit Loner Diameter
% Position for the loner diameter Title
lonerDiameterTitlePosition = [0.78 0.72 0.1
    0.045];

app.lonerDiameterTitle = uipanel('Parent',app.
    Figure,...
     'Title','Loner Radius',...
     'Units','Normalized',...
```

```
    'Position',lonerDiameterTitlePosition);

% Position for the lonerDiameterEditPosition
   relative to the
% lonerDiameterTitlePosition
lonerDiameterEditPosition =
   lonerDiameterTitlePosition + ...
    [0.003 0.003 -0.08 -0.02];%[0.783 0.723 0.02
        0.025];

app.lonerDiameterEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Position',lonerDiameterEditPosition);

% Position for lonerDiameterUnitEditPosition
   relative to
% lonerDiameterTitlePosition
lonerDiameterUnitEditPosition =
   lonerDiameterTitlePosition + ...
    [0.025 0.003 -0.08 -0.02];

app.lonerDiameterUnitEdit = uicontrol('Parent',
   app.Figure,...
    'Units','Normalized',...
    'Style','edit','String','Pixels',...
```

```
    'Position',lonerDiameterUnitEditPosition,...
    'Enable','Inactive');

% Position for lonerDiameterPlotChkBx position
   relative to
% lonerDiameterTitlePosition
lonerDiameterPlotChkBxPosition =
   lonerDiameterTitlePosition + ...
    [0.052 0.003 -0.085 -0.03]; %[0.8320 0.7280
       0.0150 0.0150];

app.lonerDiameterPlotChkBx = uicontrol('Parent',
   app.Figure,...
    'Units','Normalized',...
    'Style','checkbox','Value',0,...
    'Position',lonerDiameterPlotChkBxPosition);

% Position for lonerDiameterPlotTxt relative to
% lonerDiameterTitlePosition
lonerDiameterPlotTxtPosition =
   lonerDiameterTitlePosition + ...
    [0.06 0.003 -0.062 -0.03]; %[0.8400 0.7230
       0.0380 0.0150];

app.lonerDiameterPlotTxt = ...
    uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
```

```
    'Style','text','String','Plot Loners',...
    'Position',lonerDiameterPlotTxtPosition);

% Position for lonerDiameterChkBox position
   relative to
% lonerDiameterTitlePosition
lonerDiameterChkBoxPosition =
   lonerDiameterTitlePosition + ...
    [0.052 0.018 -0.085 -0.03]; %[0.8320 0.7230
       0.0150 0.0150];

app.lonerDiameterChkBox = uicontrol('Parent',app
   .Figure,...
    'Units','Normalized',...
    'Style','checkbox','Value',0,...
    'Position',lonerDiameterChkBoxPosition);

% Position for lonerDiameterPlotTxt relative to
% lonerDiameterTitlePosition
lonerDiameterTxtPosition =
   lonerDiameterTitlePosition + ...
    [0.06 0.018 -0.062 -0.03]; %[0.8400 0.7230
       0.0380 0.0150];

app.lonerDiameterTxt = ...
    uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
```

```
    'Style','text','String','Use Loners',...
    'Position',lonerDiameterTxtPosition);

%% Pushbutton Batch particle densities
BatchSpotAnalysisPushPosition = [0.89 0.72 0.1
   0.08];

            % Push button and image selection
               for the average image
app.BatchSpotAnalysisPush = uicontrol('Parent',
   app.Figure,...
    'Units','Normalized',...
    'Style','pushbutton','String','Batch Spot
       Analysis',...
    'Callback',@app.batchSpotsCallback,...
    'Position',BatchSpotAnalysisPushPosition);

%% Edit Pk Threshold
% Position for the particle threshold Title
pkThresholdTitlePosition = [0.67 0.63 0.1 0.08];

app.pkThresholdTitle = uipanel('Parent',app.
   Figure,...
    'Title','Pk Threshold',...
    'Units','Normalized',...
    'Position',pkThresholdTitlePosition);
```

```
% Position for the particle Threshold edit
   relative to the
% particle Threshold title
pkThresholdEditPosition =
   pkThresholdTitlePosition - ...
    [-0.01 -0.009 0.07 0.04];

app.pkThresholdEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','0.6',...
    'Position',pkThresholdEditPosition);

% Position for particle Threshold Unit relative
   to particle
% Threshold title
pkThresholdUnitEditPosition =
   pkThresholdTitlePosition - ...
    [-0.045 -0.009 0.054 0.04];

app.pkThresholdUnitEdit = uicontrol('Parent',app
   .Figure,...
    'Units','Normalized',...
    'Style','edit','String','Counts',...
    'Position',pkThresholdUnitEditPosition,...
    'Enable','Inactive');
```

```
%% Edit Pk Diameter
% Position for the particle size Title
pkSizeTitlePosition = [0.78 0.63 0.1 0.08];

app.pkSizeTitle = uipanel('Parent',app.Figure,
   ...
    'Title','Pk Diameter',...
    'Units','Normalized',...
    'Position',pkSizeTitlePosition);

% Position for the particle size edit relative
   to the
% particle size title
pkSizeEditPosition = pkSizeTitlePosition - ...
    [-0.01 -0.009 0.07 0.04];

app.pkSizeEdit = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
    'Style','edit','String','3',...
    'Position',pkSizeEditPosition);

% Position for particle size Unit relative to
   particle
% size title
pkSizeUnitEditPosition = pkSizeTitlePosition -
   ...
```

```
[-0.045 -0.009 0.054 0.04];

app.pkSizeUnitEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','Pixels',...
    'Position',pkSizeUnitEditPosition,...
    'Enable','Inactive');

%% Edit integration diameter
% Position for the particle size Title
intDiameterTitlePosition = [0.89 0.63 0.1 0.08];

app.intDiameterTitle = uipanel('Parent',app.
   Figure,...
    'Title','Int Diameter',...
    'Units','Normalized',...
    'Position',intDiameterTitlePosition);

% Position for the integration Diameter edit
   relative to the
% integration Diameter title
intDiameterEditPosition =
   intDiameterTitlePosition - ...
    [-0.01 -0.009 0.07 0.04];

% edit box for integration Diameter
```

```
app.intDiameterEdit = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','edit','String','3',...
    'Position', intDiameterEditPosition);

% Position for integration Diameter Unit
   relative to
% integration Diameter title
intDiameterUnitEditPosition =
   intDiameterTitlePosition - ...
    [-0.045 -0.009 0.054 0.04];

% edit box for unit integration diameter
app.intDiameterUnitEdit = uicontrol('Parent',app
   .Figure,...
    'Units','Normalized',...
    'Style','edit','String','Pixels',...
    'Position',intDiameterUnitEditPosition,...
    'Enable','Inactive');

%% Edit particle XY
ParticelTitlePos = [0.67 0.54 0.1 0.08];

app.ParticleTitle = uipanel('Parent',app.Figure,
   ...
    'Title','Particle ID, XY coord',...
```

```
    'Units','Normalized',...
    'Position',ParticelTitlePos);

ParticleNumPos = ParticelTitlePos - ...
    [-0.0100 -0.0060 0.070 0.0400];

app.ParticleNum = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Position',ParticleNumPos);

app.ParticleTextXY = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','text','String','1024 1024',...
    'Position',ParticelTitlePos + [0.045 0.0135
       -0.065 -0.055]);

%% Pushbutton particle up, down, and plot all
% ParticleDown pushbutton to move down through
   the list of
% particles found

% positoin for the plot all push button
ParticlePlotAllposition = [0.78 0.5824 0.082
   0.0376];
```

```
% Plot All push button to plot all trajectories
   on same plot
app.ParticlePlotAll = uicontrol('Parent',app.
   Figure,...
    'Units','Normalized',...
    'Style','pushbutton','String','Plot All',...
    'Callback',@app.PlotAllPksTrajPushCallback,
       ...
    'Position', ParticlePlotAllposition);

app.ParticleDown = uicontrol('Parent',app.Figure
   ,...
    'Units','Normalized',...
    'Style','pushbutton','String','<',...
    'Callback',@app.ParticleUpDownCallback,...
    'Position',ParticlePlotAllposition + [0
       -0.0424 -0.043 0]);

% ParticleUp pushbutton to move up through the
   list of
% particles found
app.ParticleUp = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
    'Style','pushbutton','String','>',...
    'Callback',@app.ParticleUpDownCallback,...
```

```
    'Position',ParticlePlotAllposition + [0.042
       -0.0424 -0.043 0]);

%% Axis-Text-Slider image axis, filename, image
   slider
app.FilenameEdit = uicontrol('Parent',app.Figure
   ,...
    'Units','Normalized',...
    'Style','edit','String','filename',...
    'Position',[0.0261 0.96 0.4779 0.03],'Enable
       ','Inactive');

app.Axis = axes('Parent',app.Figure,'Units','
   Normalized',...
    'Box','on','YDir','normal','DataAspectRatio'
       ,[1 1 1],...
    'DataAspectRatioMode','manual',...
    'Position',[0.045 0.2 0.55 0.75],'Tag','
       ImageAxis');

app.AnalyzeAxis = axes('Parent',app.Figure,...
    'Units','Normalized','Box','on',...
    'Position',[0.63 0.183 0.35 0.24]);

app.ImageSlider = uicontrol('Parent',app.Figure,
   ...
    'Units','Normalized',...
```

```
    'Style','slider',...
    'Enable','Inactive',...
    'Callback',@app.ImageFigSliderCallback,...
    'Position',[0.0161 0.1 0.4779 0.03]);

addlistener(app.ImageSlider,'Value','PostSet',
   ...
    @app.ImageFigSliderCallback);

%% Edit Zmin Zmax
% Position for the Zmin size Title
ZminTitlePosition = [0.0161 0.015 0.125 0.075];

app.ZminTitle = uipanel('Parent',app.Figure,...
    'Title','Zmin','Units','Normalized',...
    'Position',ZminTitlePosition);

% Position for Zmin edit position relative to
   Zmin
% title
ZminEditPosition = ZminTitlePosition + ...
    [0.012 0.0097 -0.025 -0.04];

app.ZminEdit = uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','edit','String','0',...
    'Callback',@app.imageRescaleCallback,...
```

```
    'Position',ZminEditPosition);

% Position for the Zmax size Title
ZmaxTitlePosition = [0.1511 0.015 0.125 0.075];

app.ZmaxTitle = uipanel('Parent',app.Figure,...
    'Title','Zmax','Units','Normalized',...
    'Position',ZmaxTitlePosition);

% Position for Zmax edit position relative to
   Zmax
% title
ZmaxEditPosition = ZmaxTitlePosition + ...
    [0.012 0.0097 -0.025 -0.04];

app.ZmaxEdit = uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','edit','String','0',...
    'Callback',@app.imageRescaleCallback,...
    'Position',ZmaxEditPosition);

%% Checkbox AutoScale
% Position for the AutoScaleChkBx Title
AutoScaleChkBxTitlePosition = [0.2861 0.05475
   0.04 0.03525];

app.AutoScaleTitle = uipanel('Parent',app.Figure
```

```
    ,...
     'Title','AutoScale','Units','Normalized',...
     'Position',AutoScaleChkBxTitlePosition);

% Position for AutoScaleChkBx position relative
    to
% AutoScaleChkBx Title
 AutoScaleChkBxPosition =
     AutoScaleChkBxTitlePosition + ...
     [0.0039 0.005475 -0.025 -0.02025]; %[0.29
        0.060225 0.0150 0.0150];

app.AutoScaleChkBx = uicontrol('Parent',app.
    Figure,...
     'Units','Normalized',...
     'Style','checkbox','Value',1,...
     'Callback',@app.imageRescaleCallback,...
     'Position',AutoScaleChkBxPosition);

%% Check box ROI
% Position for the ROITitle
ROITitlePosition = [0.2861 0.015 0.04 0.03525];

app.ROITitle = uipanel('Parent',app.Figure,...
     'Title','ROI','Units','Normalized',...
     'Position',ROITitlePosition);
```

```
% Position for ROIChkBx position relative to
% ROITitle
ROIChkBxPosition = ROITitlePosition + ...
    [0.0039 0.005475 -0.025 -0.02025];

app.ROIChkBx = uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','checkbox','Value',0,...
    'Callback',@app.ROICallback,...
    'Position',ROIChkBxPosition);

%% Edit current frame number
% Position for the CurrentFrameNumberTitle
CurrentFrameNumberTitlePosition = [0.3361 0.015
   0.0625 0.075];

app.CurrentFrameNumberTitle = uipanel('Parent',
   app.Figure,...
    'Title','Frame Number','Units','Normalized',
       ...
    'Position',CurrentFrameNumberTitlePosition);

% Position for CurrentFrameNumberEdit position
   relative to
% CurrentFrameNumberTitle
CurrentFrameNumberEditPosition = ...
    CurrentFrameNumberTitlePosition + ...
```

```
    [0.012 0.0097 -0.025 -0.04];

 app.CurrentFrameNumberEdit = uicontrol('Parent'
    ,app.Figure,...
    'Units','Normalized',...
    'Style','edit','String','1',...
    'Callback',@app.imageFnumCallback,...
    'Position',CurrentFrameNumberEditPosition);

%% Edit and checkbox display
% Position for the DisplayControlTitle
DisplayControlTitlePositon = [0.4086 0.015
   0.0775 0.075];

app.DisplayControlTitle = uipanel('Parent',app.
   Figure,...
    'Title','Display Control','Units','
       Normalized',...
    'Position',DisplayControlTitlePositon);

% Position for DisplayControlBpassChkBx position
    relative to
% DisplayControlTitlePositon
DisplayControlBpassChkBxPosition = ...
    DisplayControlTitlePositon + ...
    [0.005 0.035 -0.0475 -0.06]; %[0.29 0.060225
        0.0150 0.0150];
```

```
app.DisplayControlBpassChkBx = uicontrol('Parent
   ',app.Figure,...
    'Units','Normalized',...
    'Style','checkbox','Value',0,...
    'Callback',@app.DisplayControlCallback,...
    'Position',DisplayControlBpassChkBxPosition)
       ;

% Position for DisplayControlBpassTxtPosition
   relative to
% DisplayControlBpassChkBxPosition
DisplayControlBpassTxtPosition = ...
    DisplayControlBpassChkBxPosition + ...
    [0.01 0 0.03 0];

app.DisplayControlBpassTxt = ...
    uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','text','String','Band pass',...
    'Position',DisplayControlBpassTxtPosition);

% Position for DisplayControlMaxProjImageChkBx
   position
% relative to DisplayControlTitlePositon
DisplayControlMaxProjImageChkBxPosition = ...
    DisplayControlTitlePositon + ...
```

```
    [0.005  0.01 -0.0475 -0.06];

app.DisplayControlMaxProjImageChkBx = ...
    uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','checkbox','Value',0,...
    'Callback',@app.DisplayControlCallback,...
    'Position',
       DisplayControlMaxProjImageChkBxPosition);

% Position for
   DisplayControlMaxProjImageTxtPosition
   relative
% to DisplayControlMaxProjImageChkBxPosition
DisplayControlMaxProjImageTxtPosition = ...
    DisplayControlMaxProjImageChkBxPosition +
       ...
    [0.01 0 0.03 0];

app.DisplayControlMaxProjImageTxt = ...
    uicontrol('Parent',app.Figure,...
    'Units','Normalized',...
    'Style','text','String','Max Image Project',
       ...
    'Position',
       DisplayControlMaxProjImageTxtPosition);
```

```
    end % end for SME_ImageUI (Constructor)

end % end for methods (public)


%% Callbacks
methods (Access = private)
    %% class deconstructor
    % handles the cleaning up of the class &
    % figure. Either the class or the figure can
       initiate the
    % closing condition, this function makes sure both
       are
    % cleaned up
    % adopted from https://www.mathworks.com/
       matlabcentral/fileexchange/33816-example-using-a-
       matlab-class-to-control-a-gui
    % Accessed 5 Sept 2017
        function delete(app)
            % remove the closerequestfcn from the figure
               , this
            % prevents an infitie loop with the
               following delete
            % command
            set(app.Figure,  'closerequestfcn', '');
            % delete the figure
```

```
            delete(app.Figure);
            % clear out the pointer to the figure -
               prevents memory
            % leaks
            app.Figure = [];
        end

        %function - Close_fcn
        %
        %this is the closerequestfcn of the figure. All
           it does here is
        %call the class delete function (presented above
           )
        function app = Close_fcn(app, ~, ~)
            delete(app);
        end
%
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %% Callback Comment message board
        function commentBoxUpdate(app,mgs)

            % get the current messages
            oldmsgs = cellstr(get(app.CommentEdit,'
               String'));

            % update oldmsgs
```

```
        oldmsgs = [cellstr([datestr(datetime),mgs]);
            oldmsgs];

        % update CommentEdit
        set(app.CommentEdit,'String',oldmsgs);

    end

    %% Callback Load Image
    function loadImageCallback(app,~,~,filename,
       datadirectory,...
            BatchDatYorN)

        if nargin <= 3

            % check to see if the user has already
               picked a
            % directory
            if isempty(app.appDatadirectory)
                theFilePathStart = pwd;
            else
                theFilePathStart = app.
                   appDatadirectory;
            end

            % The user picks a file
            % Browse for the image file
```

```
    [filename,datadirectory]=uigetfile(...
        {'*.dat;*.sif',...
        'Accpted Files (*.dat, *.sif)';...
        '*.*', 'All Files (*.*)'},...
        'Choose Image file',theFilePathStart
           );

    if filename == 0
        % User canceled
        return;
    end
end

% create an instance of the dataRaw obj
dataRaw = acquisitionImage(filename,
   datadirectory);

% store the information into the app obj
app.appFilename = filename;
app.appDatadirectory = datadirectory;

% Check the file either for dat or sif
fileEnd = strsplit(filename,'.');
fileEnd = fileEnd{end};
switch fileEnd
    case 'sif'
        %% case sif load
```

```
% First need to check if user
   installed andor
% sifread

% multiWaitbar 0 of 8
multiWaitbar( 'Loading Image', 0 );

if ispc
    AndorFP = fullfile(matlabroot,'\
       toolbox\Andor');
    if exist(AndorFP,'dir') ~=7
        msg = ' Andor sif reader not
            found. Load data as *.
           dat';
        app.commentBoxUpdate(msg);
        % Close multiWaitbar
        multiWaitbar( 'Loading Image
           ', 'Close' );
        return
    end
elseif ~ispc
    AndorFP = fullfile(matlabroot,'/
       toolbox/Andor');
    if exist(AndorFP,'dir') ~= 7
        msg = ' Andor sif reader not
            found. Load data as *.
           dat';
```

```
            app.commentBoxUpdate(msg);
            % Close multiWaitbar
            multiWaitbar( 'Loading Image
                ','Close' );
            return
        end
    end



    % update multiWaitbar 1 of 8
    multiWaitbar( 'Loading Image', 0.125
        );

    % using sif reader to load in data
    rc=atsif_setfileaccessmode(0);
    absfilepath = fullfile(datadirectory
        ,filename);

    % make sure sif reader can read the
       file
    try
        rc=atsif_readfromfile(
            absfilepath);
    catch
        msg = ' Sif reader could not
            read file. Make sure you
```

```
        selected a *.sif file.';
    app.commentBoxUpdate(msg);
    % Close multiWaitbar
    multiWaitbar( 'Loading Image', '
        Close' );
    return
end

% update multiWaitbar 2 of 8
multiWaitbar( 'Loading Image', 0.25
    );

% check again to sif reader
    successfully opened
% file
if rc == 22002

    % 0 = read signal from sif file
        and not bkg
    % (e.g. = 2)
    signal=0;
    % check there is a signal
        present
    % present - 1 if present, 0 if
        not
    [rc,present]=
        atsif_isdatasourcepresent(
```

```
    signal);
if present
    % query the number of frames
        contained in
    % the file (e.g. in the
       instance of a
    % kinetic series there may
       be more than 1
    [rc,no_frames] =...
        atsif_getnumberframes(
           signal);

    % update multiWaitbar 3 of 8
    multiWaitbar( 'Loading Image
       ', 0.375 );

    % retrieve the number of
       pixels in each
    % frame
    [rc,imSize] =
       atsif_getframesize(signal
       );

    % update multiWaitbar 4 of 8
    multiWaitbar( 'Loading Image
       ', 0.5 );
```

```
% get the dimensions of the
   frame to open
[rc,left,bottom,right,top,
   hBin,vBin] =...
    atsif_getsubimageinfo(
       signal,0);

% update multiWaitbar 5 of 8
multiWaitbar( 'Loading Image
   ', 0.625 );

% retrieve the frame data
[rc,data] =
   atsif_getallframes(...
    signal,imSize*no_frames)
       ;

% update multiWaitbar 6 of 8
multiWaitbar( 'Loading Image
   ', 0.75 );

% check for type of acq (e.g
    FVB or image)
[rc,pattern] = ...
    atsif_getpropertyvalue(
       signal,...
    'ReadPattern');
```

```
% make sure sif is an image
   pattern
if pattern ~= '4'
    msg = ' FVB pattern in
       sif file. Select an
       image file.';
    app.commentBoxUpdate(msg
       );
    % close multiWaitbar
    multiWaitbar( 'Loading
       Image', ...
        'Close' );
    return
end

% get KineticCycleTime
[rc,KineticCycleTime] = ...
    atsif_getpropertyvalue(
       signal,...
    'KineticCycleTime');

% update multiWaitbar 7 of 8
multiWaitbar( 'Loading Image
   ', 0.875 );

% reshape the 2D array to a
```

```
    3D array for
% display
width = ((right - left)+1)/
    hBin;
height = ((top-bottom)+1)/
    vBin;
data = reshape(data',width,
    height,[]);

% update multiWaitbar 8 of 8
multiWaitbar( 'Loading Image
    ', 1 );

% display the first frame
imagesc(data(:,:,1),'Parent'
    ,app.Axis);
colormap(app.Axis,gray);
colorbar(app.Axis);
set(app.Axis,'YDir','normal'
    ,...
     'DataAspectRatio',[1 1
        1],...
     'DataAspectRatioMode','
        manual');

  app.ZoomOrigInfo = get(app
     .Axis,...
```

```
        {'Xlim','Ylim'});

    % Update Zmin Zmax
    set(app.ZminEdit, 'String',
       ...
        num2str(min(min(data
           (:,:,1)))));

    set(app.ZmaxEdit, 'String',
       ...
        num2str(max(max(data
           (:,:,1)))));

    % close multiWaitbar
    multiWaitbar( 'Loading Image
       ', ...
        'Close' );

    % Update text and slider
    set(app.FilenameEdit,'String
       ',filename);
    if no_frames == 1
        set(app.ImageSlider, '
           Min',0,...
            'Max',no_frames,'
               Value',1,...
            'SliderStep', [1 1]/
```

```
            no_frames);
        set(app.ImageSlider,...
            'Enable','on');
    else
        set(app.ImageSlider, '
           Min',1,...
            'Max',no_frames,'
               Value',1,...
            'SliderStep', [1 1]/
               no_frames);
        set(app.ImageSlider,...
            'Enable','on');
    end

    % change PlotChoiceDropDown
       to raw
    set(app.PlotChoiceDropDown,'
       Value',1);

    % Update data information
    % Check to see if width and
       height are
    % equal
    if width == height
        % doesn't mater, image
           is square
        dataRaw.sizeOfFrames =
```

```
        width;
    app.appSizeOfFrames =
        width;
else
    dataRaw.sizeOfFrames = [
        width, height];
    app.appSizeOfFrames = [
        width, height];
end

dataRaw.numFrames =
    no_frames;
dataRaw.kineticCycleTime =
    ...
     KineticCycleTime;
dataRaw.filepath =
    absfilepath;
dataRaw.images = data;
% how get images "dataRaw.
    images(:,:,1);

app.ImageData = dataRaw;
% how get images
% "app.ImageData.images
    (:,:,1);"

% update basic info for app
```

```
            obj
        app.appNumFrames = no_frames
            ;
        app.appKineticCycleTime =
            ...
             KineticCycleTime;
        app.appFilepath =
            absfilepath;

        % update CommentEdit
        msg = [' Acquisition ',
            filename, ' Loaded'];
        app.commentBoxUpdate(msg);


    else
        msg = [' Could not load file
            . sif reader rc ERROR ',
            num2str(rc)];
        app.commentBoxUpdate(msg);
        % Close multiWaitbar
        multiWaitbar( 'Loading Image
            ', 'Close' );
        return
    end

else
```

```
            msg = [' Could not load file.
               sif reader rc ERROR ',
               num2str(rc)];
            app.commentBoxUpdate(msg);
            % Close multiWaitbar
            multiWaitbar( 'Loading Image', '
               Close' );
            return
        end


    case 'dat'
        %% case dat load

        % check to see if you are running a
           batch job
        % if you are  should have more than
           3 arg passed to
        % call back
        % otherwise turn BatchDataYorN to
           false
        if nargin <= 3
            BatchDatYorN = false;
        end

        if ~BatchDatYorN
            % Prompt to get the size of CCD
```

```
    chip and the number
% of trames
prompt = {['Enter the size of
    the frame:' ...
     ' Example: 1024, 512'], ...
     ['Enter Number of frames:'
        ...
     ' Note: One is the minimum'
        ],...
     ['Enter the time step in
        secs '...
     'between frames: Example:
        0.2, 4']};
dlg_title = 'Size of the chip';
numlines = 1;

if ~isempty(app.appSizeOfFrames)
     defaultValues = {num2str(app
        .appSizeOfFrames),...
         num2str(app.appNumFrames
            ),...
         num2str(app.
            appKineticCycleTime)
            };

     answer = inputdlg(prompt,
        dlg_title,numlines,
```

```
        defaultValues);
else
    answer = inputdlg(prompt,
        dlg_title,numlines);
end

if isempty(answer) == 1
    return;
end

% update information
dataRaw.sizeOfFrames =
   str2double(answer{1});
dataRaw.numFrames = str2double(
   answer{2});
dataRaw.kineticCycleTime =
   str2double(answer{3});
dataRaw.filepath = ...
    fullfile(datadirectory,
        filename);

% update basic info for app obj
app.appSizeOfFrames = str2double
   (answer{1});
app.appNumFrames = str2double(
   answer{2});
app.appKineticCycleTime =
```

```
        str2double(answer{3});
    app.appFilepath = fullfile(
        datadirectory,filename);
else
    % update dataRaw with loaded
        parameters from
    % app obj
    dataRaw.sizeOfFrames = app.
        appSizeOfFrames;
    dataRaw.numFrames = app.
        appNumFrames;
    dataRaw.kineticCycleTime = ...
        app.appKineticCycleTime;
    dataRaw.filepath = ...
        fullfile(datadirectory,
            filename);

end

% start multiWaitBar busy
multiWaitbar( 'Loading Image', 'Busy
    ' );

% Create the holder for image
data{:,:,dataRaw.numFrames} = [];

% Start loading the file by first
```

```
    opening the
% file and getting the file id. When
    finished
% close the wait window
fid = fopen(dataRaw.filepath,'r','b'
    );
i=0;
while feof(fid) ~=1
    tmp = fread(fid, dataRaw.
        sizeOfFrames^2, 'uint16');
    if feof(fid) ~= 1
        i=i+1;
        data{:,:,i}=reshape(tmp,
            dataRaw.sizeOfFrames,
            dataRaw.sizeOfFrames);
    end
end
fclose(fid);

% cell array into matrix and make
    single
data = cell2mat(data);
data = single(data);

% close wait bar
multiWaitbar( 'Loading Image', '
    Close' );
```

```
                % display the first frame
                imagesc(data(:,:,1),'Parent',app.
                   Axis);
                colormap(app.Axis,gray);
                colorbar(app.Axis);
                set(app.Axis,'YDir','normal',...
                    'DataAspectRatio',[1 1 1],...
                    'DataAspectRatioMode','manual');

                app.ZoomOrigInfo = get(app.Axis,...
                    {'Xlim','Ylim'});

                % Update Zmin Zmax
                set(app.ZminEdit, 'String',...
                    num2str(min(min(data(:,:,1)))));

                set(app.ZmaxEdit, 'String',...
                    num2str(max(max(data(:,:,1)))));

                %This for debugging make it only 10
                   frames
%                 data = data(:,:,1:1);
%                 dataRaw.numFrames = 1;

                % Update text and slider
                set(app.FilenameEdit,'String',
```

```
    filename);
if dataRaw.numFrames == 1
    set(app.ImageSlider, 'Min',0,...
        'Max',dataRaw.numFrames,'
            Value',1,...
        'SliderStep', [1 1]/dataRaw.
            numFrames);
    set(app.ImageSlider,...
        'Enable','on');
else
    set(app.ImageSlider, 'Min',1,...
        'Max',dataRaw.numFrames,'
            Value',1,...
        'SliderStep', [1 1]/dataRaw.
            numFrames);
    set(app.ImageSlider,...
        'Enable','on');
end

% change PlotChoiceDropDown to raw
set(app.PlotChoiceDropDown,'Value'
    ,1);

dataRaw.images = data;

app.ImageData = dataRaw;
% how get images
```

```
                % "app.ImageData.images(:,:,1);"

                % update CommentEdit
                msg = [' Acquisition ', filename, '
                   Loaded'];
                app.commentBoxUpdate(msg);

    end
end % end for loadImageCallback

%% Callback save work
function SaveWorkCallback(app,s,~)
    % Check to see if data is loaded, if not
       return
    if isempty(app.ImageData) == 1
        return
    end

    % get filename and remove the .sif or .dat
    filename = app.ImageData.filename;

    % split the filename from it's extension
    filename = strsplit(filename, '.');

    % switch case to save as a mat or csv
    switch s.Text
        case 'Save As *.mat'
```

```
            % Concatenate .mat to filename
            filename = strcat(filename{1},'.mat'
               );

            % save the .mat file in the same
               folder as the
            % data directory
            filepath = fullfile(app.ImageData.
               datadirectory,filename);

            % make the variable for ImageData
            data = app.ImageData;

            multiWaitbar( 'Hold on. Saving work
               ...', 'Busy');

            save(filepath,'data');

            multiWaitbar( 'Hold on. Saving work
               ...', 'Close' );
        case 'Save As *.csv'

            % create the cell carray
            DataCell = {};
            DataCell{1} = strcat(datestr(
               datetime('now')),',\n');
```

```
DataCell{2} = strcat('file name,',
   app.ImageData.filename,',\n');
DataCell{3} = strcat('Working
   Directory,',strrep(app.ImageData.
   datadirectory,'\','\\'),',\n');
DataCell{4} = strcat('Number of
   Frames,',num2str(app.ImageData.
   numFrames),',\n');
DataCell{5} = strcat('Size of Frames
   ,',num2str(app.ImageData.
   sizeOfFrames),',\n');
DataCell{6} = strcat('Kinetic Cycle
   Time,',num2str(app.ImageData.
   kineticCycleTime),',\n');
DataCell{7} = ',\n';
DataCell{8} = strcat('Image Bandpass
    Parameters',',\n');
DataCell{9} = strcat('Length of
   Noise,',num2str(app.ImageData.
   lengthOfNoiseRawEntire),',\n');
DataCell{10} = strcat('Offset for
   Bandpass,',num2str(app.ImageData.
   OffsetForBpassRawEntire),',\n');
DataCell{11} = strcat('Length of
   Particles,',num2str(app.ImageData
   .LengthOfParicleRawEntire),',\n')
   ;
```

```
lonerUse = get(app.
   lonerDiameterChkBox,'Value');
if lonerUse
    DataCell{12} = strcat('Loners
       used,','true,','Radius used,'
       ,get(app.lonerDiameterEdit,'
       String'),',\n');
else
    DataCell{12} = strcat('Loners
       used,','false',',\n');
end

DataCell{13} = ',\n';

% check each method, and if there is
    data make sure
% to save the data
if ~isempty(app.ImageData.
   PkThresholdMIP)

    % save MIP data
    DataCell{14} = strcat('Method
       Particle Find,','MIP',',\n');
    DataCell{15} = strcat('Particle
       Threshold,',num2str(app.
       ImageData.PkThresholdMIP),',\
```

```
    n');
DataCell{16} = strcat('Particle
    Diameter,',num2str(app.
    ImageData.PkDiameterMIP),',\n
    ');
DataCell{17} = strcat('
    Integration Diameter,',
    num2str(app.ImageData.
    IntDiameterMIP),',\n');
DataCell{18} = strcat('Total
    Number of Particles found,',
    num2str(size(app.ImageData.
    PksTrajMIP,2)),',\n');
DataCell{19} = strcat(',,
    Transient Data,',',\n');
DataCell{20} = strcat(',,Rows =
    Frame # | Column = Particle
    #,',',\n');

% start up multiWairBar
multiWaitbar( 'Hold on. Saving
    work...', 'Busy');

% writing output as csv
filenamePT = strcat(filename{1},
    '_MIP_ParametersANDTrajData',
    '.csv');
```

```
filenameT = strcat(filename{1},'
   _MIP_TrajData','.csv');
filenameGyration = strcat(
   filename{1},'
   _MIP_GyrationData','.csv');

% create the full path for the
   file
filepathPT = fullfile(app.
   ImageData.datadirectory,
   filenamePT);
filepathT = fullfile(app.
   ImageData.datadirectory,
   filenameT);
filepathGyration = fullfile(app.
   ImageData.datadirectory,
   filenameGyration);

fid = fopen(filepathPT, 'w') ;
for n=1:length(DataCell)
    fprintf(fid,DataCell{n});
end
fclose(fid);

dlmwrite(filepathPT,app.
   ImageData.PksTrajMIP,'-append
   ','delimiter',',','roffset'
```

```
        ,1,'coffset',3);
    dlmwrite(filepathT,app.ImageData
        .PksTrajMIP);
    dlmwrite(filepathGyration,app.
        ImageData.PkGyrationMIP);

    % close multiWairBar
    multiWaitbar( 'Hold on. Saving
        work...', 'Close' );
end

if ~isempty(app.ImageData.PksTrajPP)

    % get filename and remove the .
        sif or .dat
    filename = app.ImageData.
        filename;

    % split the filename from it's
        extension
    filename = strsplit(filename, '.
        ');

    % save PP data
    DataCell{14} = strcat('Method
        Particle Find,','Particle
        Pick',',\n');
```

```
DataCell{15} = strcat('Particle
   Threshold,','No threshold,
   user picked particles',',\n')
   ;
DataCell{16} = strcat('Particle
   Diameter,',num2str(app.
   ImageData.PkDiameterPP),',\n'
   );
DataCell{17} = strcat('
   Integration Diameter,',
   num2str(app.ImageData.
   IntDiameterPP),',\n');
DataCell{18} = strcat('Total
   Number of Particles picked,',
   num2str(size(app.ImageData.
   PksTrajPP,2)),',\n');
DataCell{19} = strcat(',,
   Transient Data,',',\n');
DataCell{20} = strcat(',,Rows =
   Frame # | Column = Particle
   #,',',\n');

% writing output as csv
filenamePT = strcat(filename{1},
   '
   _ParticlePicked_ParametersANDTrajData
   ','.csv');
```

```
filenameT = strcat(filename{1},'
    _ParticlePicked_TrajData','.
    csv');
filenameGyration = strcat(
    filename{1},'
    _ParticlePicked_GyrationData'
    ,'.csv');

% create the full path for the
    file
filepathPT = fullfile(app.
    ImageData.datadirectory,
    filenamePT);
filepathT = fullfile(app.
    ImageData.datadirectory,
    filenameT);
filepathGyration = fullfile(app.
    ImageData.datadirectory,
    filenameGyration);

fid = fopen(filepathPT, 'w') ;
for n=1:length(DataCell)
    fprintf(fid,DataCell{n});
end
fclose(fid);

dlmwrite(filepathPT,app.
```

```
            ImageData.PksTrajPP,'-append'
            ,'delimiter',',','roffset',1,
            'coffset',3);
        dlmwrite(filepathT,app.ImageData
            .PksTrajPP);
        dlmwrite(filepathGyration,app.
            ImageData.PkGyrationPP);
    end

    if ~isempty(app.ImageData.
        PksTrajNoise)

        % get filename and remove the .
            sif or .dat
        filename = app.ImageData.
            filename;

        % split the filename from it's
            extension
        filename = strsplit(filename, '.
            ');

        % save PP data
        DataCell{14} = strcat('Method
            Particle Find,','Noise
            Particles',',\n');
        DataCell{15} = strcat('
```

```
    Integration Diameter,',
    num2str(app.ImageData.
    IntDiameterNoise),',\n');
DataCell{16} = strcat('Total
    Number of Particles picked,',
    num2str(size(app.ImageData.
    PksTrajNoise,2)),',\n');
DataCell{17} = strcat(',,
    Transient Data,',',\n');
DataCell{18} = strcat(',,Rows =
    Frame # | Column = Particle
    #,',',\n');

% writing output as csv
filenamePT = strcat(filename{1},
    '_Noise_ParametersANDTrajData
    ','.csv');
filenameT = strcat(filename{1},'
    _Noise_TrajData','.csv');
filenameGyration = strcat(
    filename{1},'
    _Noise_GyrationData','.csv');

% create the full path for the
    file
filepathPT = fullfile(app.
    ImageData.datadirectory,
```

```
                filenamePT);
            filepathT = fullfile(app.
                ImageData.datadirectory,
                filenameT);
            filepathGyration = fullfile(app.
                ImageData.datadirectory,
                filenameGyration);

            fid = fopen(filepathPT, 'w') ;
            for n=1:length(DataCell)
                fprintf(fid,DataCell{n});
            end
            fclose(fid);

            dlmwrite(filepathPT,app.
                ImageData.PksTrajNoise,'-
                append','delimiter',',','
                roffset',1,'coffset',3);
            dlmwrite(filepathT,app.ImageData
                .PksTrajNoise);
            dlmwrite(filepathGyration,app.
                ImageData.PkGyrationNoise);
        end

    end
```

```
    % update CommentEdit
    msg = [' Saved to ', app.ImageData.
       datadirectory];
    app.commentBoxUpdate(msg);
end % end for SaveWorkCallback

%% Function checkEverything
function checkEverything(app,AutoScale,ROI,
   frameNumber,...
        imageChoice)
    % Check to see if data is loaded, if not
       return
    if isempty(app.ImageData) == 1
        return
    end

    % check to see if the current image choice
       is loaded
    switch imageChoice
        case 'All Raw'
            if isempty(app.ImageData.images) ==
               1
                return
            end
```

```
        theImageChoice = app.ImageData.
            images;

    case 'All Bpass'
        if isempty(app.ImageData.bpassImages
            ) == 1
            return
        end

        theImageChoice = app.ImageData.
            bpassImages;

    case 'Avg Raw'
        if isempty(app.ImageData.avgImage)
            == 1
            return
        end

        theImageChoice = app.ImageData.
            avgImage;

    case 'Avg Bpass'
        if isempty(app.ImageData.
            avgbpassImage) == 1
            return
        end
```

```
            theImageChoice = app.ImageData.
                avgbpassImage;
    end

    if AutoScale == 1
        Zmin = min(min(...
            theImageChoice(:,:,frameNumber)));
        Zmax = max(max(...
            theImageChoice(:,:,frameNumber)));
        set(app.ZminEdit, 'String',num2str(Zmin)
            );
        set(app.ZmaxEdit, 'String',num2str(Zmax)
            );
        range = [Zmin Zmax];
    else

        Zmin = str2double(get(app.ZminEdit,'
            String'));
        Zmax = str2double(get(app.ZmaxEdit,'
            String'));

        % Check to see if user set Zmax < Zmin
            or Zmin > Zmax
        if Zmax < Zmin
            msg = ' Cannot set Zmax less than
                Zmin. Try again.';
```

```
            app.commentBoxUpdate(msg);
            return;

        elseif Zmin > Zmax
            msg = ' Cannot set Zmin greater than
                Zmax. Try again.';
            app.commentBoxUpdate(msg);
            return;

        elseif Zmin == Zmax
            msg = ' Cannot set Zmin equal to
               Zmax. Try again.';
            app.commentBoxUpdate(msg);
            return;

        end
        range = [Zmin Zmax];
    end

        % Check to see if ROI should be used
        if ROI == 1
            % Plot image with the scale and ROI
            Im = theImageChoice(:,:,...
                frameNumber);
            imagesc(Im,'Parent',app.Axis,range);
            set(app.Axis,{'Xlim','Ylim'},app.
               ZoomNewInfo);
```

```
        colormap(app.Axis,gray);
        colorbar(app.Axis);
        set(app.Axis,'YDir','normal',...
            'DataAspectRatio',[1 1 1],...
            'DataAspectRatioMode','manual');
    else
        imagesc(theImageChoice(:,:,...
            frameNumber),'Parent',app.Axis,
               range);
        colormap(app.Axis,gray);
        colorbar(app.Axis);
        set(app.Axis,'YDir','normal',...
            'DataAspectRatio',[1 1 1],...
            'DataAspectRatioMode','manual');
    end

    % get PkFind plot XY of choice
    PkFindChoice = app.checkPkXYPlot;

    switch PkFindChoice
        case 'BPass Max Image Proj'
            pksChoice = app.ImageData.
               AvgCntrdxyMIP;
        case 'Pick Particle'
            pksChoice = app.ImageData.
               AvgCntrdxyPP;
        case 'CSV'
```

```
            pksChoice = app.ImageData.
                AvgCntrdxyCSV;
        case 'Choice Particles'
            pksChoice = app.ImageData.
                ListOfChosenPksPP;
        case 'Noise'
            pksChoice = app.ImageData.
                AvgCntrdxyNoise;
        case 'None'
            return
    end

    if ~isempty(pksChoice)
        % get the current pkNum
        PkNum = str2double(get(app.
            ParticleNum,'String'));

        % overlap pks XY
        % make the image actice axis to be
            safe
        axes(app.Axis);
        hold on;
        % plot pixel accuracy
        plot(app.Axis,...
            round(pksChoice(:,1)),...
            round(pksChoice(:,2)),...
            'ro','MarkerSize',5,'LineWidth'
```

```
                    ,2);

                % check to make sure PkNum isn't out
                    of the range
                if PkNum > size(pksChoice,1)
                    % Put it the last particle
                    PkNum = size(pksChoice,1);
                end

                % highlight current pk
                plot(app.Axis,...
                    round(pksChoice(PkNum,1)),...
                    round(pksChoice(PkNum,2)),...
                    'bo','MarkerSize',5,'LineWidth'
                       ,2);
                hold off;
            end
        end % end for checkEverything

        %% Callback image slider
        function ImageFigSliderCallback(app,~,~)

            % Get all needed info to check everything
            frameNumberSelected = int32(get(app.
               ImageSlider,'Value'));
            AutoScale = get(app.AutoScaleChkBx,'Value');
            ROI = get(app.ROIChkBx,'Value');
```

```
        imageChoice = app.imageChoiceCheck;

        if isempty(app.ImageData) ~= 1
            app.checkEverything(AutoScale,ROI,...
                frameNumberSelected,imageChoice)
        end

        % Update the frame number string
        set(app.CurrentFrameNumberEdit,...
            'String', num2str(frameNumberSelected));


    end % end for ImageFigSliderCallback

    %% Function update zoom information
    function UpdateZoomInfo(app,~,~)
        app.ZoomNewInfo = get(app.Axis,{'Xlim','Ylim
           '});
    end % end for UpdateZoomInfo

    %% Callback ROI
    function ROICallback(app,~,evd)
        if evd.Source.Value == 1
            if isempty(app.ZoomOrigInfo)
                % isZoomed = false;
                set(app.ROIChkBx,'Value',0);
                msg = ' No zoom, no ROI selected.';
```

```
                app.commentBoxUpdate(msg);
                return
            elseif isequal(get(app.Axis,'XLim'),...
                    app.ZoomOrigInfo(:,1:2)) && ...
                    isequal(get(app.Axis,'YLim'),...
                    app.ZoomOrigInfo(:,3:4))
                % isZoomed = false;
                msg = ' ROI has not changed from
                   last zoom.';
                app.commentBoxUpdate(msg);

                return
            else
                % isZoomed = true;
                % update ZoomNewInfo for replotting
                msg = ' Captured ROI';
                app.commentBoxUpdate(msg);
                app.ZoomOrigInfo = get(app.Axis,{'
                   Xlim','Ylim'});
                return;
            end
        else
        end
    end % end for ROICallback

    %% Callback image rescale Zmin Zmax
    function imageRescaleCallback(app,s,~)
```

```
% check to see if user is using the
   AutoScaleChkBx or
% either ZminEdit or ZmaxEdit
% AutoScaleChkbx 'String' == empty therefore
    isempty is 1
% ZminEdit or ZmaxEdit 'String' w/e they
   entered therefore
%   isempty is 0
if isempty(get(s,'String')) == 1

    if get(app.AutoScaleChkBx,'Value') == 0
        return
    end
    AutoScale = 1;
else
    set(app.AutoScaleChkBx,'Value',0);
    AutoScale = 0;
end

frameNumberSelected = int32(...
    get(app.ImageSlider,'Value'));

ROI = get(app.ROIChkBx,'Value');

imageChoice = app.imageChoiceCheck;
```

```
    if isempty(app.ImageData) ~= 1
        app.checkEverything(AutoScale,...
            ROI,frameNumberSelected,imageChoice)
    end
end % end for imageRescaleCallback

%% Callback for the frame number
function imageFnumCallback(app,~,~)
    if isempty(app.ImageData) ~= 1

        % get frame number
        frameNumberSelected = str2double(get(...
            app.CurrentFrameNumberEdit,'String')
               );

        % check to make you can plot the frame
           number
        if frameNumberSelected > app.ImageData.
           numFrames
            msg = [' You only have ',...
                num2str(app.ImageData.numFrames)
                   ,...
                ' frames!. I cannot plot frame '
                   ,...
                num2str(frameNumberSelected),'.'
                   ];
            app.commentBoxUpdate(msg);
```

```
    return;
elseif frameNumberSelected < 1
    msg = ['The smallest frame number '
       ...
        'I can plot is 1. Please try
           again.'];
    app.commentBoxUpdate(msg);
    return;
end

% Get all needed info to check
   everything
AutoScale = get(app.AutoScaleChkBx,'
   Value');
ROI = get(app.ROIChkBx,'Value');
imageChoice = app.imageChoiceCheck;

if isempty(app.ImageData) ~= 1
    app.checkEverything(AutoScale,...
        ROI,frameNumberSelected,
           imageChoice)
end

%Update slider
set(app.ImageSlider, 'Value',
   frameNumberSelected);
```

```
    end
end % end for imageFnumCallback

%% Callback for Pk Traj menu
function PkTrajPlotMenuCallback(app,~,e)
    % check which menu was alreay checked and
       turn it off
    currentMenuState = {get(app.TimePlotChoice,'
       Checked');...
        get(app.FramePlotChoice,'Checked');};
    % look at each TimePlotChoice and
       FramePlotChoice and
    % uncheck each one
    % loop through and turn every on off
    for n=1:length(currentMenuState)
        if strcmp(currentMenuState{n},'on')
            if n == 1
                set(app.TimePlotChoice,'Checked'
                   ,'off');
                break
            elseif n == 2
                set(app.FramePlotChoice,'Checked
                   ','off');
            end
        end
    end
    % set source to checked, on of the menu
```

```
        items in
    % PkTrajPlotMenu
    set(e.Source,'Checked','on');
end % end for PkTrajPlotMenuCallback

%% Callback for MaxImageProject image
function MaxImageProjectCallback(app,~,~)

    % check to make sure there is data to work
       on
    if isempty(app.ImageData) == 1
        % update comment the box
        msg = ' No data loaded';
        app.commentBoxUpdate(msg);
        return
    end

    % Step 2 in workflow.
    % max project the raw image
    app.ImageData.avgImage = app.ImageData.
       MaxImageProject(...
        app.ImageData,'Raw');

    % update CommentEdit
    msg = ' Max project raw images';
    app.commentBoxUpdate(msg);
```

```
% Step 3 in workflow.
% bpass the raw image
% get all the data
lnoise = str2double(get(app.lnoiseEdit,'
   String'));
lobject = str2double(get(app.lobjectEdit,'
   String'));
threshold = str2double(get(...
    app.thresholdEdit,'String'));

% create the holder for each frame
bpassIm = zeros(app.ImageData.sizeOfFrames,
   ...
    app.ImageData.sizeOfFrames,...
    app.ImageData.numFrames);

% sif data is single, keep everything the
   same
bpassIm = single(bpassIm);

% multiWaitBar
multiWaitbar( 'Band pass Filtering...', 0,
   ...
    'CancelFcn', @(a,b) set(app.CommentEdit,
       ...
    'String',['Canceled ',a]) );
```

```
% call bpass
for n=1:app.ImageData.numFrames
    abort = multiWaitbar( ...
        'Band pass Filtering...',...
        n/app.ImageData.numFrames );

    if abort
        multiWaitbar( 'Band pass Filtering
           ...',...
            'Close' );
        return
    else
        bpassIm(:,:,n) = app.ImageData.bpass
           (...
            app.ImageData.images(:,:,n),...
            lnoise,lobject,threshold);
    end

end

multiWaitbar( 'Band pass Filtering...', '
   Close' );

% update ImageData
app.ImageData.bpassImages = bpassIm;

% Update parameters
```

```
        app.ImageData.lengthOfNoiseRawEntire =
           lnoise;
        app.ImageData.LengthOfParicleRawEntire =
           lobject;
        app.ImageData.OffsetForBpassRawEntire =
           threshold;

        % update CommentEdit
        msg = [' Banded pass filtered. lnoise =',...
            num2str(lnoise),' lobject =', num2str(
               lobject),...
            ' threshold =', num2str(threshold)];
        app.commentBoxUpdate(msg);

        % Step 4 in workflow.
        app.ImageData.avgbpassImage = app.ImageData.
           MaxImageProject(...
            app.ImageData,'Bpass');

        % update CommentEdit
        msg = ' Max project banded pass images';
        app.commentBoxUpdate(msg);

    end % end for MaxImageProjectCallback

    %% Callback for plot choice dropdown
    function DisplayControlCallback(app,~,~)
```

```
% check to make sure you have data
if isempty(app.ImageData) == 1
    return
end

% get the choice the check boxes
whichPlot = app.imageChoiceCheck;

switch whichPlot
    % if All turn on slider and
       FrameNumberEdit
    % if Avg turn off slider,
       FrameNumberEdit, and set each
    % to 1
    case 'All Raw'

        set(app.ImageSlider,'Enable','on');
        set(app.CurrentFrameNumberEdit,'
           Enable','on');

        % plot All Raw images
        % Get all needed info to check
           everything
        frameNumberSelected = int32(get(app.
           ImageSlider,...
            'Value'));
```

```
            AutoScale = get(app.AutoScaleChkBx,'
               Value');
            ROI = get(app.ROIChkBx,'Value');
            imageChoice = whichPlot;

            if isempty(app.ImageData) ~= 1
                app.checkEverything(AutoScale,
                   ROI,...
                    frameNumberSelected,
                       imageChoice)
            end

            % update CommentEdit
            msg = ' Ploted Raw Images';
            app.commentBoxUpdate(msg);

        case 'All Bpass'

            % check to make sure you have data
               to work on
            if isempty(app.ImageData.bpassImages
               ) == 1

                % update CommentEdit
                msg = ' No Band Passed images to
                    plot.';
                app.commentBoxUpdate(msg);
```

```
    return
end

set(app.ImageSlider,'Enable','on');
set(app.CurrentFrameNumberEdit,'
   Enable','on');

% plot All Bpass images
% Get all needed info to check
   everything
frameNumberSelected = int32(get(app.
   ImageSlider,...
    'Value'));
AutoScale = get(app.AutoScaleChkBx,'
   Value');
ROI = get(app.ROIChkBx,'Value');
imageChoice = whichPlot;

if isempty(app.ImageData) ~= 1
    app.checkEverything(AutoScale,
       ROI,...
        frameNumberSelected,
           imageChoice)
end

% update CommentEdit
```

```
            msg = ' Ploted Bpass Images';
            app.commentBoxUpdate(msg);

        case 'Avg Raw'

            % check to make sure you have data
               to work on
            if isempty(app.ImageData.avgImage)
               == 1

                % update CommentEdit
                msg = ' No raw max image
                   projection to plot.';
                app.commentBoxUpdate(msg);

                return
            end

            set(app.ImageSlider,'Value',1);
            set(app.ImageSlider,'Enable','
               Inactive');
            set(app.CurrentFrameNumberEdit,...
                'Enable','Inactive');
            set(app.CurrentFrameNumberEdit,'
               String','1');

            % plot Avg Raw image
```

```
            % Get all needed info to check
               everything
            frameNumberSelected = int32(get(app.
               ImageSlider,...
                'Value'));
            AutoScale = get(app.AutoScaleChkBx,'
               Value');
            ROI = get(app.ROIChkBx,'Value');
            imageChoice = whichPlot;

            if isempty(app.ImageData) ~= 1
                app.checkEverything(AutoScale,
                   ROI,...
                    frameNumberSelected,
                       imageChoice)
            end

            % update CommentEdit
            msg = ' Ploted raw max image
               projection';
            app.commentBoxUpdate(msg);

        case 'Avg Bpass'

            % check to make sure you have data
               to work on
            if isempty(app.ImageData.
```

```
    avgbpassImage) == 1

    % update CommentEdit
    msg = ' No band passed max image
        projection to plot.';
    app.commentBoxUpdate(msg);

    return
end

set(app.ImageSlider,'Value',1);
set(app.ImageSlider,'Enable','
    Inactive');
set(app.CurrentFrameNumberEdit,...
    'Enable','Inactive');
set(app.CurrentFrameNumberEdit,'
    String','1');

% plot Avg Raw image
% Get all needed info to check
    everything
frameNumberSelected = int32(get(app.
    ImageSlider,...
    'Value'));
AutoScale = get(app.AutoScaleChkBx,'
    Value');
ROI = get(app.ROIChkBx,'Value');
```

```
            imageChoice = whichPlot;

            if isempty(app.ImageData) ~= 1
                app.checkEverything(AutoScale,
                   ROI,...
                     frameNumberSelected,
                        imageChoice)
            end

            % update CommentEdit
            msg = ' Plotted band passed max
               image projection';
            app.commentBoxUpdate(msg);

    end
end % end for DisplayControlCallback

%% Callback for particle find pushbutton
function pkFindPushCallback(app,~,~)

    % check to make sure you have data
    if isempty(app.ImageData) == 1
        return
    end

    % get PkFind method of choice
    PkFindChoice = get(app.pkFindDropDown,'
```

```
    String');
PkFindChoice = PkFindChoice{get(app.
    pkFindDropDown,'Value')};

% get loner information
lonerUse = get(app.lonerDiameterChkBox,'
    Value');
lonerPlot = get(app.lonerDiameterPlotChkBx,'
    Value');

% get diameter of pk
sz = str2double(get(app.pkSizeEdit,'String')
    );

% check to make sure diameter is odd
if mod(sz,2) == 0 || sz < 3

    % update CommentEdit
    msg = ' Particle diameter must be an odd
        value >= 3';
    app.commentBoxUpdate(msg);

    return;
end

% get diameter of integration window
intD = str2double(get(app.intDiameterEdit,'
```

```
    String'));

% check to make sure diameter is odd
if mod(intD,2) == 0 || intD < 3

    % update CommentEdit
    msg = ' Integration diameter must be an
       odd value >= 3';
    app.commentBoxUpdate(msg);

    return;
end

% get th value, if th is > 1 and != to 1 th
   will be
% used as is in counts
th = str2double(get(...
    app.pkThresholdEdit,'String'));

if th == 1

    % update CommentEdit
    msg = ' Cannot run with a threshold of
       1.';
    app.commentBoxUpdate(msg);

    return
```

```
end

% check which image to band pass
whichImage = get(app.pkFindDropDown,'String'
   );
whichImage = whichImage{get(app.
   pkFindDropDown,'Value')};

switch whichImage
    case 'BPass Frames'

        % check to make sure you have data
           to work on
        if isempty(app.ImageData.bpassImages
           ) == 1

            % update CommentEdit
            msg = ' No Band Passed frames.';
            app.commentBoxUpdate(msg);

            return
        end

    case 'BPass Max Image Proj'

        % check to make sure you have data
           to work on
```

```
                if isempty(app.ImageData.
                   avgbpassImage) == 1

                    % update CommentEdit
                    msg = ' No band passesd max
                       image projection.';
                    app.commentBoxUpdate(msg);

                    return
                end

        end
        % run pkfind with the choosen method
        app.PkFindChoiceMethod(...
            PkFindChoice,sz,th,intD,app.ImageData,
               whichImage,lonerUse,lonerPlot);
    end % end for pkFindPushCallback

    %% Function Particle find choice method
    function PkFindChoiceMethod(...
            app,whichMethod,sz,th,intD,ImageData,
               whichImage,lonerUse,lonerPlot)
        % run PkFind depending on the method checked

        switch whichMethod
            case 'BPass Max Image Proj'
                % run pkFind with no correction for
```

```
    overlapping pks
[PreListCntrdXYIntFrameNum,
    PreListXYPixel, STRING] = ...
     app.ImageData.findPksInital(
        ImageData,th,sz,intD,
        whichImage);

% check to see if there were too
    many pks
if isempty(STRING) == 0

     % update CommentEdit
     app.commentBoxUpdate(STRING);
     return
end

switch whichImage
     case 'BPass Frames'
          totalNumPks = 0;
          for n=1:ImageData.numFrames
               totalNumPks =
                  totalNumPks + ...
                   size(
                      PreListCntrdXYIntFrameNum
                      {:,:,n},1);
          end
```

```
% Create holder
allPksXY = zeros(totalNumPks
   ,2);
Index = 0;

% loop through each frame
   storing the pks XY
for n=1:ImageData.numFrames
    % get current size for
       Keepers and store
       data
    currentSize = size(
       PreListCntrdXYIntFrameNum
       {:,:,n},1);
    % get pkData from frame
       n
    tmpXYMaxInt =
       PreListCntrdXYIntFrameNum
       {:,:,n};
    % store only xy position
    allPksXY(Index+1:Index +
        currentSize,:) =
       tmpXYMaxInt(:,1:2);
    % move the index based
       on number of pks
    Index = Index +
       currentSize;
```

```
            end

            % bring back to pixel level
            allPksXY = round(allPksXY);

        case 'BPass Max Image Proj'
            % unBox cell
            tmpXYMaxInt =
               PreListCntrdXYIntFrameNum
               {:};
            % save only XY for each
               particle
            allPksXY = tmpXYMaxInt
               (:,1:2);

            % bring back to pixel level
            allPksXY = round(allPksXY);

            % save the XY data
            app.ImageData.
               RoundedCntrdXYMIP =
               allPksXY;
    end

    % check to see to run loner
    if lonerUse == 1
        % get loner radius
```

```
        lonerRadius = ...
            str2double(...
            get(app.lonerDiameterEdit,'
               String'));

        [allPksXY, ~] = ...
            app.ImageData.lonerpick(...
            ImageData.sizeOfFrames,...
            lonerRadius,...
            allPksXY,...
            lonerPlot);

        % save the XY data
        app.ImageData.
           RoundedCntrdXYMIP_Loners =
           allPksXY;
    end

    % Create Traj for pks
    [PostListCntrdXYIntFrameNumN,
       PksTrajMIP, ...
        pkListCntrdXYIntN, AvgCntrdxy,
           PkGyration, Warning] = ...
        app.ImageData.calcPksTraj(
           ImageData,allPksXY,intD);

    multiWaitbar( 'Finding Particles...'
```

```
    , 'Close' );

if ~isempty(Warning)
    % Display Warning information
       and exit
    app.commentBoxUpdate(Warning);
    return
end

% save data
app.ImageData.
   TotalEachFrameCntrdIntMIP = ...
    PostListCntrdXYIntFrameNumN;
app.ImageData.PksTrajMIP =
   PksTrajMIP;
app.ImageData.PkGyrationMIP =
   PkGyration;
app.ImageData.PksListCntrdXYIntMIP =
    ...
    pkListCntrdXYIntN;
app.ImageData.AvgCntrdxyMIP =
   AvgCntrdxy;
app.ImageData.
   PreTrajPksListPixelXYMIP =
   PreListXYPixel;
app.ImageData.
   PreListCntrdXYIntFrameNumMIP =
```

```
                PreListCntrdXYIntFrameNum;

            % Update parameters
            app.ImageData.PkThresholdMIP = th;
            app.ImageData.PkDiameterMIP = sz;
            app.ImageData.IntDiameterMIP = intD;

            % plot traj and pk position over
                image
            app.PlotAllPksTraj(PksTrajMIP);

            % Update text for PkNum's XY
            XY = round(AvgCntrdxy(1,:));
            XY = num2str(XY);
            set(app.ParticleTextXY,'String', XY)
                ;

        case 'Particle Pick'

            allPksXY = round(app.ImageData.
                ListOfChosenPksPP);

            % save the XY data
%               app.ImageData.RoundedCntrdXYPP =
    allPksXY;

            % check to see to run loner
```

```
if lonerUse == 1
    % get loner radius
    lonerRadius = ...
        str2double(...
        get(app.lonerDiameterEdit,'
           String'));

    [allPksXY, ~] = ...
        app.ImageData.lonerpick(...
        ImageData.sizeOfFrames,...
        lonerRadius,...
        allPksXY,...
        lonerPlot);
end

% Create Traj for pks
[PostListCntrdXYIntFrameNumPP,
   PksTrajPP, ...
    pkListCntrdXYIntPP, AvgCntrdxy,
       PkGyration, Warning] = ...
    app.ImageData.calcPksTraj(
       ImageData,allPksXY,intD);

multiWaitbar( 'Finding Particles...'
   , 'Close' );

if ~isempty(Warning)
```

```
        % Display Warning information
           and exit
        app.commentBoxUpdate(Warning);
        return
    end

    % save data
    app.ImageData.
       TotalEachFrameCntrdIntPP = ...
        PostListCntrdXYIntFrameNumPP;
    app.ImageData.PksTrajPP = PksTrajPP;
    app.ImageData.PkGyrationPP =
       PkGyration;
    app.ImageData.PksListCntrdXYIntPP =
       ...
        pkListCntrdXYIntPP;
    app.ImageData.AvgCntrdxyPP =
       AvgCntrdxy;

    % Update parameters
    app.ImageData.PkDiameterPP = sz;
    app.ImageData.IntDiameterPP = intD;

    % plot traj and pk position over
       image
    app.PlotAllPksTraj(PksTrajPP);
```

```
            % Update text for PkNum's XY
            XY = round(AvgCntrdxy(1,:));
            XY = num2str(XY);
            set(app.ParticleTextXY,'String', XY)
               ;

        case 'CSV'

    end
end % end for PkFindChoiceMethod

%% Callback for NoisePksPushCallback push button
function NoisePksPushCallback(app,~,~)

    % check to make sure there is data to work
       on
    if isempty(app.ImageData) == 1
        % update comment the box
        msg = ' No data loaded';
        app.commentBoxUpdate(msg);
        return
    end

    % get PkFind method of choice
    PkFindChoice = get(app.pkFindDropDown,'
       String');
    PkFindChoice = PkFindChoice{get(app.
```

```
    pkFindDropDown,'Value')};

switch PkFindChoice
    case 'BPass Max Image Proj'
        allPksXY = round(app.ImageData.
            AvgCntrdxyMIP);

        % check to make sure there is data
            to work on
        if isempty(allPksXY) == 1
            % update comment the box
            msg = ' No XY from MIP';
            app.commentBoxUpdate(msg);
            return
        end


    case 'Particle Pick'
        allPksXY = round(app.ImageData.
            ListOfChosenPksPP);

        % check to make sure there is data
            to work on
        if isempty(allPksXY) == 1
            % update comment the box
            msg = ' No XY from Particle
                Picker';
```

```
                app.commentBoxUpdate(msg);
                return
            end
        case 'CSV'

        otherwise
            return
    end

    % run the Noise Picker
    % get all the informatoin

    % size of the image
    imageSize = app.ImageData.sizeOfFrames;

    % get rejectRadius value, if th is > 1
       rejectRadius will be
    % used as is in counts
    rejectRadius = str2double(get(...
        app.NoiseRadiusEdit,'String'));

    if rejectRadius < 1

        % update CommentEdit
        msg = ' Cannot run with a reject radius
           < 1.';
        app.commentBoxUpdate(msg);
```

```
        return
    end

    % the total number of noise particles
    numNoisePks = str2double(get(...
        app.NoiseNumberEdit,'String'));

    if numNoisePks < 1

        % update CommentEdit
        msg = ' Cannot run calculate a negative
           number of Pks.';
        app.commentBoxUpdate(msg);

        return

    elseif floor(numNoisePks) ~= numNoisePks

        % update CommentEdit
        msg = ' The number must be an integer.';
        app.commentBoxUpdate(msg);

        return

    end
```

```
% create the list of noise particles
pkListNoise = app.ImageData.NoisePkPicke(...
    imageSize,rejectRadius,numNoisePks,
       allPksXY);

% calculate the noise particles trajs

% get diameter of integration window
intD = str2double(get(app.intDiameterEdit,'
   String'));

% check to make sure diameter is odd
if mod(intD,2) == 0 || intD < 3

    % update CommentEdit
    msg = ' Integration diameter must be an
       odd value >= 3';
    app.commentBoxUpdate(msg);

    return;
end

% Create Traj for pks
[PostListCntrdXYIntFrameNumNoise,
   PksTrajNoise, ...
    pkListCntrdXYIntNoise, AvgCntrdxy,
       PkGyration, Warning] = ...
```

```
    app.ImageData.calcPksTraj(app.ImageData,
       pkListNoise,intD);

multiWaitbar( 'Finding Particles...', 'Close
   ' );

if ~isempty(Warning)
    % Display Warning information and exit
    app.commentBoxUpdate(Warning);
    return
end

% save data
app.ImageData.TotalEachFrameCntrdIntNoise =
   ...
    PostListCntrdXYIntFrameNumNoise;
app.ImageData.PksTrajNoise = PksTrajNoise;
app.ImageData.PkGyrationNoise = PkGyration;
app.ImageData.PksListCntrdXYIntNoise = ...
    pkListCntrdXYIntNoise;
app.ImageData.AvgCntrdxyNoise = AvgCntrdxy;

% Update parameters
app.ImageData.IntDiameterNoise = intD;
app.ImageData.RadiusRejectNoise =
   rejectRadius;
```

```
% plot traj and pk position over image
app.PlotAllPksTraj(PksTrajNoise);

% Update text for PkNum's XY
XY = round(AvgCntrdxy(1,:));
XY = num2str(XY);
set(app.ParticleTextXY,'String', XY);

% turn on check for Choice particles in the
   menu
% get PkFind plot XY of choice
% First turn off the current one
PkFindChoice = app.checkPkXYPlot;

switch PkFindChoice
    case 'BPass Max Image Proj'
        set(app.PkPlotXY_MIP,'Checked','off'
           );
    case 'Pick Particle'
        set(app.PkPlotXY_PP,'Checked','off')
           ;
    case 'CSV'
        set(app.PkPlotXY_CSV,'Checked','off'
           );
    case 'Choice Particles'
        set(app.ChoicePksXY_PP,'Checked','
           off');
```

```
            case 'Noise'
                set(app.NoisePksXY,'Checked','on');
            case 'None'
                set(app.NoisePksXY,'Checked','on');
        end

        % make sure choice paritcle is checked in
           menu
        set(app.NoisePksXY,'Checked','on');

        % plot choosen particles
        % Get all needed info to check everything
        frameNumberSelected = int32(get(app.
           ImageSlider,'Value'));
        AutoScale = get(app.AutoScaleChkBx,'Value');
        ROI = get(app.ROIChkBx,'Value');
        imageChoice = app.imageChoiceCheck;

        app.checkEverything(AutoScale,ROI,...
            frameNumberSelected,imageChoice);


    end % for NoisePksPushCallback push button

    %% Function for check box Pk XY plot?
    function PkFindChoice = checkPkXYPlot(app)
```

```
% get all the current checked status
currentMenuState = {get(app.PkPlotXY_MIP,'
   Checked');...
    get(app.PkPlotXY_PP,'Checked');
    get(app.PkPlotXY_CSV,'Checked');
    get(app.ChoicePksXY_PP,'Checked');
    get(app.NoisePksXY,'Checked')};

% check to see if all choices are checked
   off
allOff = {'off';'off';'off';'off';'off'};

% if everything return the choice as none
if all(strcmp(allOff,currentMenuState)) == 1
    PkFindChoice = 'None';
    return
end

% loop through and figure which one is
   checked
for n=1:5
    if strcmp(currentMenuState{n},'on')
        if n == 1
            PkFindChoice = 'BPass Max Image
               Proj';
            break;
        elseif n == 2
```

```
                PkFindChoice = 'Pick Particle';
                break;
            elseif n == 3
                PkFindChoice = 'CSV';
                break;
            elseif n == 4
                PkFindChoice = 'Choice Particles
                    ';
                break
            else
                PkFindChoice = 'Noise';
            end
        end
    end

end % end for checkPkXYPlot

%% Function for check Pk traj plot menu?
function PkTrajPlotChoice = checkPkTrajPlot(app)
    % get all the current checked status
    currentMenuState = {get(app.TimePlotChoice,'
        Checked');...
         get(app.FramePlotChoice,'Checked');};
    % loop through and figure which one is
        checked
    for n=1:length(currentMenuState)
        if strcmp(currentMenuState{n},'on')
```

```
                if n == 1
                    PkTrajPlotChoice = 'TimePlot';
                    break
                elseif n == 2
                    PkTrajPlotChoice = 'FramePlot';
                end
            end
        end
    end % end for checkPkTrajPlot

    %% Callback for plot all pushbutton
    function PlotAllPksTrajPushCallback(app,~,~)

        % check to make sure there is data to work
           on
        if isempty(app.ImageData) == 1
            % update comment the box
            msg = ' No data loaded';
            app.commentBoxUpdate(msg);
            return
        end

        % get PkFind method of choice
        PkFindChoice = get(app.pkFindDropDown,'
           String');
        PkFindChoice = PkFindChoice{get(app.
           pkFindDropDown,'Value')};
```

```
        switch PkFindChoice
            case 'BPass Max Image Proj'
                PksTraj = app.ImageData.PksTrajMIP;
            case 'Particle Pick'
                PksTraj = app.ImageData.PksTrajPP;
            case 'CSV'
                PksTraj = app.ImageData.PksTrajCSV;
            case 'Noise'
                PksTraj = app.ImageData.PksTrajNoise
                    ;
        end

        % plot traj and pk position over image
        app.PlotAllPksTraj(PksTraj);

    end

    %% Function for plot all pushbutton
    function PlotAllPksTraj(app,PksTraj)
        % Plot all the trajectories with different
            color
        % on the same plot
        colorVec = hsv(size(PksTraj,2));

        % check to see what kineticCycleTime is
        if isa(app.ImageData.kineticCycleTime,'char'
```

```
    )
        % convert to double
        timestep = str2double(...
            app.ImageData.kineticCycleTime);
        % save it as a double
        app.ImageData.kineticCycleTime =
            timestep;

    elseif isa(app.ImageData.kineticCycleTime,'
        double')
        % your safe use as is
        timestep = app.ImageData.
            kineticCycleTime;
    end

    time = 0:timestep:(timestep*(app.ImageData.
        numFrames-1));
    cla(app.AnalyzeAxis);
    axes(app.AnalyzeAxis);
    hold on;
    for n=1:size(PksTraj,2)
        plot(app.AnalyzeAxis,...
            time,PksTraj(:,n),'Color',colorVec(n
                ,:));
    end
    xlabel(app.AnalyzeAxis,'Time (sec)');
    ylabel(app.AnalyzeAxis,'Intensity');
```

```
        hold off;

        % overlap pks XY with image
        % clear axis to be safe
        cla(app.Axis);

        % Get all needed info to check everything
        frameNumberSelected = int32(get(app.
           ImageSlider,'Value'));
        AutoScale = get(app.AutoScaleChkBx,'Value');
        ROI = get(app.ROIChkBx,'Value');
        imageChoice = app.imageChoiceCheck;

        app.checkEverything(AutoScale,ROI,...
            frameNumberSelected,imageChoice)

        % update CommentEdit
        msg = [' Found ', num2str(size(PksTraj,2)),'
            particles.'];
        app.commentBoxUpdate(msg);
    end % end for PlotAllPksTraj


    %% Function? for plot on pk traj
    function PlotOnePkTraj(app,PksTraj,PkNum,
       TimeOrFrame)
        % check to see which x-axis should be
```

```
    plotted
switch TimeOrFrame
    case 'TimePlot'
        timestep = app.ImageData.
            kineticCycleTime;
        xaxis = 0:timestep:(timestep*(app.
            ImageData.numFrames-1));
        Label = 'Time (sec)';
    case 'FramePlot'
        xaxis = 1:app.ImageData.numFrames;
        Label = 'Frame number';
end

% set axis
cla(app.AnalyzeAxis);
axes(app.AnalyzeAxis);

% plot PkNum's trajectory
plot(app.AnalyzeAxis,xaxis,PksTraj(:,PkNum))
    ;

% set x axis label
xlabel(app.AnalyzeAxis,Label);

% set y axis label
ylabel(app.AnalyzeAxis,'Intensity');
```

```
end % end for PlotOnePkTraj

%% Callback for particle up down button
function ParticleUpDownCallback(app,~,e)

    % check to make sure you have data
    if isempty(app.ImageData) == 1
        return
    end

    % get PkFind method of choice
    PkFindChoice = get(app.pkFindDropDown,'
       String');
    PkFindChoice = PkFindChoice{get(app.
       pkFindDropDown,'Value')};

    switch PkFindChoice
        case 'BPass Max Image Proj'
            PksTraj = app.ImageData.PksTrajMIP;
            PksAvgCntrdXY = app.ImageData.
               AvgCntrdxyMIP;
        case 'Particle Pick'
            PksTraj = app.ImageData.PksTrajPP;
            PksAvgCntrdXY = app.ImageData.
               AvgCntrdxyPP;
        case 'CSV'
            PksTraj = app.ImageData.PksTrajCSV;
```

```
            PksAvgCntrdXY = app.ImageData.
               AvgCntrdxyCSV;
        case 'Noise'
            PksTraj = app.ImageData.PksTrajNoise
               ;
            PksAvgCntrdXY = app.ImageData.
               AvgCntrdxyNoise;
    end

    UpOrDown = e.Source.String;

    % get the PkNum
    PkNum = str2double(get(app.ParticleNum,'
       String'));

    % get the total number of Pks
    TotalPkNum = size(PksTraj,2);

    switch UpOrDown
        case '>'
            % Check to see to loop back to the
               start of pks
            % otherwise PkNum = PkNum + 1;
            if PkNum+1 > TotalPkNum
                PkNum = 1;
            else
                PkNum = PkNum + 1;
```

```
            end

        case '<'
            % Check to see to loop back to the
               start of pks
            % otherwise PkNum = PkNum - 1;
            if PkNum-1 == 0
                PkNum = TotalPkNum;
            else
                PkNum = PkNum - 1;
            end
    end

    if ~isempty(PksTraj)
        % get the axis to plot
        TimeOrFrame = app.checkPkTrajPlot;

        % Plot PkNum's traj
        app.PlotOnePkTraj(PksTraj,PkNum,
           TimeOrFrame);

        % Update text for PkNum
        set(app.ParticleNum,'String',num2str(
           PkNum));

        % Update text for PkNum's XY
        XY = round(PksAvgCntrdXY(PkNum,:));
```

```
            XY = num2str(XY);
            set(app.ParticleTextXY,'String', XY);

            % Get all needed info to check
               everything
            frameNumberSelected = int32(get(app.
               ImageSlider,'Value'));
            AutoScale = get(app.AutoScaleChkBx,'
               Value');
            ROI = get(app.ROIChkBx,'Value');
            imageChoice = app.imageChoiceCheck;

            app.checkEverything(AutoScale,ROI,...
                frameNumberSelected,imageChoice);

            % update CommentEdit
            msg = [' Plotting PkNum ',num2str(PkNum)
               , ' of ', num2str(TotalPkNum)];
            app.commentBoxUpdate(msg);
        end

    end % end for ParticleUpDownCallback

    %% Callback for particle XY plot menu
    function PkXYPlotMenuCallback(app,~,e)

        currentStatus = get(e.Source,'Checked');
```

```
% make the list of the menu optiosn
currentMenuState = {get(app.PkPlotXY_MIP,'
   Checked');...
    get(app.PkPlotXY_PP,'Checked');...
    get(app.PkPlotXY_CSV,'Checked');
    get(app.NoisePksXY,'Checked')};

% look at each TimePlotChoice and
   FramePlotChoice and
% uncheck each one
% loop through and turn everything off
for n=1:length(currentMenuState)
    if strcmp(currentMenuState{n},'on')
        if n == 1
            set(app.PkPlotXY_MIP,'Checked','
               off');
            break
        elseif n == 2
            set(app.PkPlotXY_PP,'Checked','
               off');
            break
        elseif n == 3
            set(app.PkPlotXY_CSV,'Checked','
               off');
            break
        elseif n ==4
```

```
                set(app.NoisePksXY,'Checked','
                    Off');
                break
            else
                set(app.ChoicePksXY_PP,'Checked'
                    ,'off');
            end
        end
    end

    % set source to checked, on of the menu
       items in
    % PkTrajPlotMenu

    if strcmp(currentStatus, 'on')
        set(e.Source,'Checked','off');
    else
        set(e.Source,'Checked','on');
    end

    % plot choosen particles
    % Get all needed info to check everything
    frameNumberSelected = int32(get(app.
       ImageSlider,'Value'));
    AutoScale = get(app.AutoScaleChkBx,'Value');
    ROI = get(app.ROIChkBx,'Value');
    imageChoice = app.imageChoiceCheck;
```

```
        app.checkEverything(AutoScale,ROI,...
            frameNumberSelected,imageChoice);
    end % end for PkXYPlotMenuCallback

    %% Callback particle picker
    function ParticlePickerCallback(app,~,~)

        % check to make sure you have data
        if isempty(app.ImageData) == 1
            return
        end

        % set comment box to tell user how to use
        msg = ' Double Click on last particle to
           exit';
        app.commentBoxUpdate(msg);

        % call the function
        [x, y] = getpts(app.Axis);

        % turn on check for Choice particles in the
           menu
        % get PkFind plot XY of choice
        % First turn off the current one
        PkFindChoice = app.checkPkXYPlot;
```

```
switch PkFindChoice
    case 'BPass Max Image Proj'
        set(app.PkPlotXY_MIP,'Checked','off'
           );
    case 'Pick Particle'
        set(app.PkPlotXY_PP,'Checked','off')
           ;
    case 'CSV'
        set(app.PkPlotXY_CSV,'Checked','off'
           );
    case 'Choice Particles'
        set(app.ChoicePksXY_PP,'Checked','on
           ');
    case 'Noise'
        set(app.NoisePksXY,'Checked','off');
    case 'None'
        set(app.ChoicePksXY_PP,'Checked','on
           ');
end

% make sure choice paritcle is checked in
   menu
set(app.ChoicePksXY_PP,'Checked','on');

% it will always append to the list
app.ImageData.ListOfChosenPksPP = ...
    [app.ImageData.ListOfChosenPksPP; [x, y
```

```
            ]];

        % plot choosen particles
        % Get all needed info to check everything
        frameNumberSelected = int32(get(app.
           ImageSlider,'Value'));
        AutoScale = get(app.AutoScaleChkBx,'Value');
        ROI = get(app.ROIChkBx,'Value');
        imageChoice = app.imageChoiceCheck;

        app.checkEverything(AutoScale,ROI,...
            frameNumberSelected,imageChoice);

    end % end for ParticlePickerCallback

    %% Callback for delete from list
    function DeletePksFromListCallback(app,~,~)

        % check to make sure you have data
        if isempty(app.ImageData) == 1
            return
        end

        ListPks = app.ImageData.ListOfChosenPksPP;

        % check to see if list is empty
        if isempty(ListPks) == 1
```

```
    msg = ' No particles to delete. List is
       empty.';
    app.commentBoxUpdate(msg);
    return
end

ListPks_str = num2str(app.ImageData.
   ListOfChosenPksPP);

[PksToDelete,UserCancel_TorF] = listdlg('
   PromptString','Pick Pks XY to remove from
    list:',...
    'ListString',ListPks_str);

% check to see if user selected anything
%   1 for selected something
%   0 for selected nothing
if UserCancel_TorF == 0
    return
end

% remove the rows selected
ListPks(PksToDelete,:) = [];

% update the list
app.ImageData.ListOfChosenPksPP = ListPks;
```

```
        % plot choosen particles
        % Get all needed info to check everything
        frameNumberSelected = int32(get(app.
            ImageSlider,'Value'));
        AutoScale = get(app.AutoScaleChkBx,'Value');
        ROI = get(app.ROIChkBx,'Value');
        imageChoice = app.imageChoiceCheck;

        app.checkEverything(AutoScale,ROI,...
            frameNumberSelected,imageChoice);

        % update comment box
        str = [' Total particles removed from list:
            ', num2str(length(PksToDelete)), ' out of
            ', num2str(size(ListPks_str,1))];
        app.commentBoxUpdate(str);
    end % end for DeletePksFromListCallback

    %% check the display control
    function imageChoice = imageChoiceCheck(app)

        % get all the current checked status
        currentMenuState = [...
            get(app.DisplayControlBpassChkBx,'Value'
                ),...
            get(app.DisplayControlMaxProjImageChkBx,
                'Value')];
```

```
        currentMenuState = num2str(currentMenuState)
            ;

        switch currentMenuState
            case '0  0'
                imageChoice = 'All Raw';
            case '0  1'
                imageChoice = 'Avg Raw';
            case '1  0'
                imageChoice = 'All Bpass';
            case '1  1'
                imageChoice = 'Avg Bpass';
        end

    end

    %% Callback batch processing for spot density
    function batchSpotsCallback(app,~,~)

        % Check to see if data is loaded, if not
            return
        if isempty(app.ImageData) == 1
            return
        end

        % get the first file in the batch processing
        if nargin <= 3
```

```
        % check to see if the user has already
           picked a
        % directory
        if isempty(app.appDatadirectory)
            theFilePathStart = pwd;
        else
            theFilePathStart = app.
               appDatadirectory;
        end

        % The user picks a file
        % Browse for the image file
        [filename,workingDir]=uigetfile(...
            {'*.dat;*.sif',...
            'Accpted Files (*.dat, *.sif)';...
            '*.*', 'All Files (*.*)'},...
            'Choose a file from the batch',
               theFilePathStart);

        if filename == 0
            % User canceled
            return;
        end
    end

    % Check the file either for dat or sif
```

```
fileEnd = strsplit(filename,'.');
fileEnd = fileEnd{end};

switch fileEnd
    case 'sif'
        filePattern = fullfile(workingDir, '
           *.sif');
        BatchDatYorN = false;
    case 'dat'
        filePattern = fullfile(workingDir, '
           *.dat');
        BatchDatYorN = true;
end

theFiles = dir(filePattern);

% create the holder for particle total
   results
totalParticleResults = zeros(size(theFiles
   ,1),3);

% create the holder for the filenames
filenameList = {};
filenameList{size(theFiles,1)} = '';

% create the table for the results
totalParticleResultsTable = table;
```

```
% create the file name for the results
fileNameParticleResults = '
   Batch_Spot_Analysis.csv';
resultsFULLpath = fullfile(workingDir,
   fileNameParticleResults);


for k = 1 : length(theFiles)

    filename = theFiles(k).name;
    % store the filename
    filenameList{k} = filename;

    % Now do whatever you want with this
       file name
    % loadImageCallback(app,~,~,filename,
       datadirectory)
    % call the load function, the callback
       should have 4
    % argumets
    e = [];
    app.loadImageCallback(app,e,filename,
       workingDir,BatchDatYorN);

    % call MIP and Bpass
    app.MaxImageProjectCallback(app);
```

```
% find particles without loners
% turn off the loner check box
set(app.lonerDiameterChkBox,'Value',0);
% call find particles
app.pkFindPushCallback(app);
% should check which method to do first
% find the number of particles
numOfParticlesWithOutLoners = ...
    size(app.ImageData.PksTrajMIP,2);

% run the noise machine
numOfTrials = 2;

for n=1:numOfTrials
    app.NoisePksPushCallback(app);
end

% find particles with loners
% turn on the loner check box
set(app.lonerDiameterChkBox,'Value',1);
% call find particles
app.pkFindPushCallback(app);
% should check which method to do first
% find the number of particles
numOfParticlesWithLoners = ...
    size(app.ImageData.PksTrajMIP,2);
```

```
        % calculate the percent rejection
        percentReject = ...
            100*...
            ((...
            numOfParticlesWithOutLoners - ...
            numOfParticlesWithLoners)...
            /numOfParticlesWithOutLoners);

        % populate the results array
        totalParticleResults(k,:) = [...
            numOfParticlesWithOutLoners,...
            numOfParticlesWithLoners,...
            percentReject];

        % create the 's'ource variable for save
           function
        s = struct;
        s.Text = 'Save As *.csv';

        % save the work
        app.SaveWorkCallback(s);
    end

    TheTrajData = [];
    TheNoiseData = [];
    TheGyrationData = [];
```

```
TheGyrationNoise = [];

% loop over the files again to concat them
   together
for k = 1 : length(theFiles)

    % create the file end for the data
    fileEndForTrajData = '_MIP_TrajData.csv'
       ;
    fileEndForTrajNoise = '_Noise_TrajData.
       csv';
    fileEndForGyrationData = '
       _MIP_GyrationData.csv';
    fileEndforGyrationNoise = '
       _Noise_GyrationData.csv';

    filename = theFiles(k).name;
    pathToFile = theFiles(k).folder;


    % Check the file either for dat or sif
    fileStart = strsplit(filename,'.');
    fileStart = fileStart{1};

    filenameTrajData = strcat(fileStart,
       fileEndForTrajData);
    filenameNoiseData = strcat(fileStart,
```

```
    fileEndForTrajNoise);
filenameGyrationData = strcat(fileStart,
    fileEndForGyrationData);
filenameGyrationNoise = strcat(fileStart
    ,fileEndforGyrationNoise);

% read the current data into the
    workspace
currentTrajData = csvread(fullfile(
    pathToFile,filenameTrajData));
currentNoiseData = csvread(fullfile(
    pathToFile,filenameNoiseData));
currentGyrationData = csvread(fullfile(
    pathToFile,filenameGyrationData));
currentGyrationNoise = csvread(fullfile(
    pathToFile,filenameGyrationNoise));

% concat the data
TheTrajData = [TheTrajData,
    currentTrajData];
TheNoiseData = [TheNoiseData,
    currentNoiseData];
TheGyrationData = [TheGyrationData,
    currentGyrationData];
TheGyrationNoise = [TheGyrationNoise,
    currentGyrationNoise];
```

```
end

% created the file ends for the final data
saveToFileTrajData = '_Trajs.csv';
saveToFileTrajNoise = '_TrajNoise.csv';
saveToFileGyrationData = '_GyrationTrajs.csv
    ';
saveToFileGyrationNoise = '
    _GyrationTrajNoise.csv';

% save both the traj data and noise as csv
csvwrite(fullfile(workingDir,strcat(
    fileStart,saveToFileTrajData)),
    TheTrajData)
csvwrite(fullfile(workingDir,strcat(
    fileStart,saveToFileTrajNoise)),
    TheNoiseData)
csvwrite(fullfile(workingDir,strcat(
    fileStart,saveToFileGyrationData)),
    TheGyrationData)
csvwrite(fullfile(workingDir,strcat(
    fileStart,saveToFileGyrationNoise)),
    TheGyrationNoise)

% finish up with particle spot analysis
totalParticleResultsTable.FileNames =
    filenameList';
```

```
                totalParticleResultsTable.
                   TotalNumOfParticles ...
                    = totalParticleResults(:,1);
                totalParticleResultsTable.
                   TotalNumOfParticlesWithLoners...
                    = totalParticleResults(:,2);
                totalParticleResultsTable.PercentReject ...
                    = totalParticleResults(:,3);

                % write the csvfile for the spot analysis
                writetable(totalParticleResultsTable,
                   resultsFULLpath);

            end
    end % end for methods (private)
end % end for SME_ImageUI (classdef)
```

```
classdef acquisitionImage < handle

    properties

        % basic information
        filename
        datadirectory
        numFrames
        sizeOfFrames
        kineticCycleTime
        filepath

        % image data
        images
        bpassImages
        avgImage
        avgbpassImage

        % parameters used for particle picking
        lengthOfNoiseRawEntire
        lengthOfNoiseRawAvg
        OffsetForBpassRawEntire
        OffsetForBpassRawAvg
        LengthOfParicleRawEntire
        LengthOfParicleRawAvg

        % parameters used for method MIP = 'M'ax 'I'mage 'P'
```

```
   roject
PkThresholdMIP
PkDiameterMIP
IntDiameterMIP

% parameters used for method PP = 'P'article 'P'
   icker
ListOfChosenPksPP
PkDiameterPP
IntDiameterPP

% parameters used for method CSV = 'C'omma-'S'
   eparated 'V'alues
PkThresholdCSV
PkDiameterCSV
IntDiameterCSV

% parameters used for method Noise
IntDiameterNoise
RadiusRejectNoise

% For each method store each frame list of pks and
   their centroid
TotalEachFrameCntrdIntMIP
TotalEachFrameCntrdIntPP
TotalEachFrameCntrdIntCSV
TotalEachFrameCntrdIntL
```

```
TotalEachFrameCntrdIntNoise

% For each method store the pre traj pixel XY
PreTrajPksListPixelXYMIP
PksTrajListPixelXYCSV
PksTrajListPixelXYL
PksTrajListPixelXYNoise

% For each method store the pre traj cntrd XY
PreListCntrdXYIntFrameNumMIP
PreListCntrdXYIntFrameNumCSV
PreListCntrdXYIntFrameNumXYL
PreListCntrdXYIntFrameNumXYNoise

% For each method store the rounded XY Cntrd from
   the pre list
RoundedCntrdXYMIP
RoundedCntrdXYCSV
RoundedCntrdXYL
RoundedCntrdXYNoise

% For each method store the loners rounded XY Cntrd
   from the
% pre list
RoundedCntrdXYMIP_Loners
RoundedCntrdXYCSV_Loners
```

```
% For each method store cntrd XY and integreated
   brithness aka
% 'Int'ensity
PksListCntrdXYIntMIP
PksListCntrdXYIntPP
PksListCntrdXYIntCSV
PksListCntrdXYIntL
PksListCntrdXYIntNoise

% For each method store the list of pks 'Traj'
   ectories
PksTrajMIP
PksTrajPP
PksTrajCSV
PksTrajL
PksTrajNoise

% For each method store the list of pks gyration
PkGyrationMIP
PkGyrationPP
PkGyrationCSV
PkGyrationL
PkGyrationNoise

% For each method store the list of pks Average
   Centroid XY
AvgCntrdxyMIP
```

```
    AvgCntrdxyPP
    AvgCntrdxyCSV
    AvgCntrdxyL
    AvgCntrdxyNoise

end % end for properties (public)

methods
    %% Constructor
    function acquisitionImage_obj = acquisitionImage(
       NameOfFile,DataDir)
        acquisitionImage_obj.filename = NameOfFile;
        acquisitionImage_obj.datadirectory = DataDir;
    end % end for acquisitionImage_obj (Constructor)

    %% Initialize fields
    function InitializeImageField(acquisitionImage_obj)
        % Check to see if numFrames is defined
        if isempty(acquisitionImage_obj.numFrames) == 1
            return;
        end

        % Initialize the field that will hold raw image
        acquisitionImage_obj.images.frame = [];
        acquisitionImage_obj.images(acquisitionImage_obj
           .numFrames).frame
    end % end for InitializeImageField
```

```
end % end for methods (public)

methods(Static)

    %% Function average image
    function mip = MaxImageProject(Data,whichImage)

        switch whichImage
            case 'Raw'

                % Making this the brightest average now
                   - DJW7-15-18
                mip = max(Data.images, [], 3);

            case 'Bpass'

                % Making this the brightest average now
                   - DJW7-15-18
                mip = max(Data.bpassImages, [], 3);

        end % end for avgImage

    end

    %% Function WhoIsOverLapping Is this still in use
       ????
    function [Loners,keepers] = ...
```

```
    WhoIsOverlapping(rejectRadius,pkListMaxInt)

pkListMaxInt = sortrows(pkListMaxInt,3,'ascend')
   ;

% create the multiWaitBar
multiWaitbar( 'Calculating distances between pks
   ...',...
    0, 'CancelFcn', @(a,b) disp( ['Cancel ',a] )
        );

% Calculate the distance of every particle to
   each other particle,
% including itself
distancePkList = zeros(size(pkListMaxInt,1),size
   (pkListMaxInt,1));
for jj=1:size(pkListMaxInt,1)
    % create the abort for mulitWaitbar
    abort = multiWaitbar( ...
        'Calculating distances between pks...',
           ...
        jj/size(pkListMaxInt,1) );

    % leave function if user cancels
    if abort
        multiWaitbar('CLOSEALL');
        return
```

```
    else
        tmp = sqrt((pkListMaxInt(jj,1)-
           pkListMaxInt(:,1)).^2+...
            (pkListMaxInt(jj,2)-pkListMaxInt
               (:,2)).^2);
        distancePkList(jj,:) = tmp;
    end
end
clear tmp;

% create the multiWaitBar
multiWaitbar( 'Finding isolated pks...',...
    0, 'CancelFcn', @(a,b) disp( ['Cancel ',a] )
        );

% find particles that are not overlapping
keepers = zeros(size(pkListMaxInt,1),3);
keepersIndex = 1;
for n=1:size(distancePkList,1)
    % create the abort for mulitWaitbar
    abort = multiWaitbar( 'Finding isolated pks
       ...',...
        n/size(pkListMaxInt,1) );

    % leave function if user cancels
    if abort
        multiWaitbar('CLOSEALL');
```

```
                    return
                else

                    if isempty(find(distancePkList(n,:) <
                       (2*rejectRadius + 1) &...
                            distancePkList(n,:) ~= 0,1)) ==
                               1
                        keepers(keepersIndex,:) =
                           pkListMaxInt(n,:);
                        keepersIndex = keepersIndex + 1;
                    end

                end

            end

            % clean up zeros from the storage
            keepers(keepers(:,1) == 0 & keepers(:,2) == 0,:)
                = [];

%             % remove the lower half of the distancePkList
%             % This will ensure you don't inspect every
   particle pair twice
%             distancePkList=triu(distancePkList);
%             [rr, cc] = find(distancePkList < (2*
   rejectRadius + 1) & ...
%                 distancePkList ~= 0);
```

```
%
%             RowCol = [rr, cc];
%
%             % create the multiWaitBar
%             multiWaitbar( 'Picking brightest pks...',...
%                 0, 'CancelFcn', @(a,b) disp( ['Cancel ',a]
    ) );
%
%             % create the Leader Board
%             LB = zeros(size(pkListMaxInt,1),3);
%             LBindex = 1;
%             for n=1:size(RowCol,1)
%                 % create the abort for mulitWaitbar
%                 abort = multiWaitbar( 'Picking brightest
   pks...',...
%                     n/size(pkListMaxInt,1) );
%
%                 % leave function if user cancels
%                 if abort
%                     multiWaitbar('CLOSEALL');
%                     return
%                 else
%
%                     players = RowCol(n,:);
%                     % compete
%                     if pkListMaxInt(players(1),3) >
   pkListMaxInt(players(2),3)
```

```
%
%                           [LB, LBindex ] = CheckList(...
%                               LB,pkListMaxInt(players(1),:),
   pkListMaxInt(players(2),:),LBindex);
%
%                       elseif pkListMaxInt(players(1),3) <
   pkListMaxInt(players(2),3)
%
%                           [LB, LBindex ] = CheckList(...
%                               LB,pkListMaxInt(players(2),:),
   pkListMaxInt(players(1),:),LBindex);
%                       else
%
%                           % flip a coin and decide the
   winner
%                           coin = randi([0 1],1);
%                           disp(['Coin tossed, ', num2str(
   coin)]);
%                           if coin == 1
%                               [LB, LBindex ] = CheckList(...
%                                   LB,pkListMaxInt(players(1)
   ,:),pkListMaxInt(players(2),:),LBindex);
%                           else
%                               [LB, LBindex ] = CheckList(...
%                                   LB,pkListMaxInt(players(2)
   ,:),pkListMaxInt(players(1),:),LBindex);
%                           end
```

```
%
%                     end
%                 end
%             end
%
%             function [LeaderBoard, index ] = CheckList(...
%                     LeaderBoard,winner,loser,index)
%
%                 if isempty(find(LeaderBoard(:,1) == winner
   (1) & ...
%                         LeaderBoard(:,2) == winner(2),1))
   == 0
%                     return
%                 end
%
%                 if isempty(find(LeaderBoard(:,1) == loser
   (1) & ...
%                         LeaderBoard(:,2) == loser(2),1))
%                     % Loser wasn't on the list put winner
   on the list
%                     LeaderBoard(index,:) = winner;
%                     index = index + 1;
%                 else
%                     % Loser was on list replace with
   winner
%                     rows = find(LeaderBoard(:,1) == loser
   (1) & ...
```

```
%                           LeaderBoard(:,2) == loser(2));
%                       LeaderBoard(rows,:) = winner;
%                   end
%               end
%
%               LB(LB(:,1) == 0 & LB(:,2) == 0,:) = [];

            Loners = keepers;

%               keepers = [keepers; LB];

            % close multiWaitBar
            multiWaitbar( 'Calculating distances between pks
                ...',...
                'Close' );
            multiWaitbar( 'Finding isolated pks...',...
                'Close' );
%               multiWaitbar( 'Picking brightest pks...',...
%                   'Close' );


        end % end for WhoIsOverlapping

        %% Function find initial pk positions
        function [PreListCntrdXYIntFrameNum, PreListXYPixel,
            STRING] = ...
                findPksInital(ImageData,th,sz,intD,
```

```
        whichImage)
    % create the holder for each frame
    PreListCntrdXYIntFrameNum = zeros(900000,5);

    % create the holder for each frame pixel level
       xy coord
    PreListXYPixel = zeros(900000,2);

%       PreListCntrdXYIntFrameNum = zeros(5,900000);

    % sif data is single, keep everything the same
    PreListCntrdXYIntFrameNum = single(...
        PreListCntrdXYIntFrameNum);
    PreListXYPixel = single(PreListXYPixel);



    i = 1;
    numPks = 0;
    Index = 1;
    IndexPixelXY = 1;

    switch whichImage
        case 'BPass Frames'
            numOfSteps = ImageData.numFrames;
            Images = ImageData.bpassImages;
        case 'BPass Max Image Proj'
            numOfSteps = 1;
```

```
        Images = ImageData.avgbpassImage;
end

% multiWaitBar
multiWaitbar( 'Finding Particles...', 0,...
    'CancelFcn', @(a,b) disp( ['Cancel ',a] ) );

% call bpass
for n=1:numOfSteps
    abort = multiWaitbar( ...
        'Finding Particles...',...
        n/numOfSteps );

    if abort
        multiWaitbar( 'Finding Particles...',...
            'Close' );
        return
    else

        if th < 1
            maxCount = max(max(Images(:,:,n)));
            th = th * maxCount;
        end

        pkListPixelXY = ImageData.pkfnd(...
            Images(:,:,n),...
            th,sz);
```

```
pkListCntrdXY_andInt = ImageData.cntrd(
   ...
    Images(:,:,n),...
    pkListPixelXY, intD);

pkListCntrdXY_andInt(:,5) = i;

numPks = numPks + ...
    size(pkListCntrdXY_andInt,1);

if numPks > 900000

    % update CommentEdit
    STRING = 'Too many particles!
       Increase the threshold.';
    PreListCntrdXYIntFrameNum = [];
    return
end


PreListXYPixel(IndexPixelXY:IndexPixelXY
   +size(pkListPixelXY,1)-1,:)...
    = pkListPixelXY;

PreListCntrdXYIntFrameNum(Index:Index+
   size(pkListCntrdXY_andInt,1)-1,:)...
```

```
            = pkListCntrdXY_andInt;

        i = i + 1;
        Index = Index + ...
            size(pkListCntrdXY_andInt,1);

        IndexPixelXY = IndexPixelXY + ...
            size(pkListPixelXY,1);


    end

end

% Get rid of any empty zeros
PreListCntrdXYIntFrameNum(...
    PreListCntrdXYIntFrameNum(:,1) == 0 &...
    PreListCntrdXYIntFrameNum(:,2) == 0 &...
    PreListCntrdXYIntFrameNum(:,3) == 0 &...
    PreListCntrdXYIntFrameNum(:,4) == 0 &...
    PreListCntrdXYIntFrameNum(:,5) == 0,:)...
    = [];

% sperate the information per frame
Cellarray{:,:,numOfSteps} = [];
for n=1:numOfSteps
    Cellarray{:,:,n} = ...
```

```
            PreListCntrdXYIntFrameNum(...
            PreListCntrdXYIntFrameNum(:,5) == n,:);
    end

    PreListCntrdXYIntFrameNum = Cellarray;

    clear Cellarray;

    % not adding a method to account for per frame
       pixel location
    % b/c I added this after we resigned the per
       frame analysis
    % Get rid of any empty zeros
    PreListXYPixel(PreListXYPixel(:,1) == 0 &...
        PreListXYPixel(:,2) == 0,:)= [];

    STRING = [];



end % end for findPksInital

%% Function for calculating particle trajs
function [PostListCntrdXYIntFrameNum, PksTraj, ...
        pkListCntrdXYInt, AvgCntrdxy, PkGyration,
           Warning] = calcPksTraj(ImageData,PkArray,
           sz)
```

```
% PostListCntrdXYIntFrameNum
% m x n where:
%   m = numPks * numFrames
%   n = 5
% PostListCntrdXYIntFrameNum(:,1) is the x-
   coordinates
% PostListCntrdXYIntFrameNum(:,2) is the y-
   coordinates
% PostListCntrdXYIntFrameNum(:,3) is the
   brightnesses
% PostListCntrdXYIntFrameNum(:,4) is the square
   of the radius
%   of gyration
% PostListCntrdXYIntFrameNum(:,5) is the Frame
   at which the
%   particle was on
%
% PksTraj
% m x n where:
%   m = numFrames
%   n = numPks
% PksTraj(:,PkNum) is the brigtnesses per frame,
    where the row
%   number is equal to the frame number
%   PkNum is the particle number, each column
   identifies the
%       PkNum
```

```
%
% pkListCntrdXYInt
% - pkListCntrdXYInt will always be the
   information listed
%   below for the last frame
% m x n where:
%   m = numPks
%   n = 5
% pkListCntrdXYInt(:,1) is the x-coordinates
% pkListCntrdXYInt(:,2) is the y-coordinates
% pkListCntrdXYInt(:,3) is the brightnesses
% pkListCntrdXYInt(:,4) is the square of the
   radius
%   of gyration
% pkListCntrdXYInt(:,5) is the Frame at which
   the
%   particle was on
%
% AvgCntrdxy
% m x n where:
%   m = numPks
%   n = 2
% - Each row refers to a unique particle
% AvgCntrdxy(:,1) is the average Cntrd X for
   each particle
% AvgCntrdxy(:,2) is the average Cntrd Y for
   each particle
```

```
% check which image to work with
numOfSteps = ImageData.numFrames;
Images = ImageData.bpassImages;

% Double check to make sure the bpass on each
   frame has been
% run first

if isempty(Images)
    % app.CommentEdit
    Warning = ['Run bpass on each frame'...
        ' before finding particles.'];
    PostListCntrdXYIntFrameNum = [];
    PksTraj = [];
    pkListCntrdXYInt = [];
    AvgCntrdxy = [];
    return
end


% if PkArray is 3D reshape as shown below else
   take as is
if length(size(PkArray,1)) > 2
    % First reshape:
    %   PkArray(numPks,3,numFrames) to PkArray(
       TotalnumPks,2)
```

```
% PkArray has order for a single frame
   PkArray{:,:,FrameN}:
%  column #:     x    |     y    | Integrated
   Intensity
%
   ----------------------------------------


%    row #1: Pk1's x | Pk1's y | Pk1's Int
%    row #2: Pk2's x | Pk2's y | Pk2's Int
%
% After PkArray will be 2D:
%  column #:     x    |     y
%           --------------------
%    row #1: Pk1's x | Pk1's y
%    row #2: Pk2's x | Pk2's y

% get the total number of particles
totalNumPks = 0;
for n=1:numOfSteps
    totalNumPks = totalNumPks + size(PkArray
       {:,:,n},1);
end

% make the holder for the total number of
   pks
allPksXY = zeros(totalNumPks,2);
```

```
        % start the index at 0 to store pks XY
        Index = 0;

        % loop through each frame storing the pks XY
        for n=1:numOfSteps
            % get current size for Keepers and store
                data
            currentSize = size(PkArray{:,:,n},1);
            % get pkData from frame n
            tmpXYMaxInt = PkArray{:,:,n};
            % store only xy position
            allPksXY(Index+1:Index + currentSize,:)
               = tmpXYMaxInt(:,1:2);
            % move the index based on number of pks
            Index = Index + currentSize;
        end

        % remove any duplicates
        allPksXY = unique(allPksXY,'rows');

    else
        % PkArray is 2D
        allPksXY = PkArray;
        % remove any duplicates
        allPksXY = unique(allPksXY,'rows');

    end
```

```
% clear up tmps including input
clear PkArray tmpXYMaxInt

% create the storage for each frame
PostListCntrdXYIntFrameNum = zeros(size(allPksXY
   ,1)*numOfSteps,5);

% sif data is single, keep everything the same
PostListCntrdXYIntFrameNum = single(
   PostListCntrdXYIntFrameNum);

Index = 1;
i = 1;
for n=1:numOfSteps
    % create mulitWaitbar
    abort = multiWaitbar( 'Calculating
       trajectories...',...
        n/numOfSteps );

    if abort
        multiWaitbar( 'Calculating trajectories
           ...','Close' );
        return
    else
        pkListCntrdXYInt = ImageData.cntrd(
           Images(:,:,n),...
```

```
                allPksXY, sz);

            pkListCntrdXYInt(:,5) = i;

            PostListCntrdXYIntFrameNum(Index:Index+
               size(pkListCntrdXYInt,1)-1,:)=
               pkListCntrdXYInt;

            i = i + 1;
            Index = Index + size(pkListCntrdXYInt,1)
               ;
        end
    end

    % Get rid of any empty zeros
    PostListCntrdXYIntFrameNum(...
        PostListCntrdXYIntFrameNum(:,1) == 0 &...
        PostListCntrdXYIntFrameNum(:,2) == 0 &...
        PostListCntrdXYIntFrameNum(:,3) == 0 &...
        PostListCntrdXYIntFrameNum(:,4) == 0 &...
        PostListCntrdXYIntFrameNum(:,5) == 0,:)...
        = [];

    % create the holder for PkTrajs where the format
        of PksTraj is
    % as followed:
    % column #:            Pk1           |
```

```
                Pk2
%
    ------------------------------------------------

%     row #1: Integrated Int @ Frame1 |
    Integrated Int @ Frame1
%     row #2: Integrated Int @ Frame2 |
    Integrated Int @ Frame2
%
% PksTraj(numFrames,numPks);
PksTraj = zeros(ImageData.numFrames,size(
    pkListCntrdXYInt,1));
% loop through each particle
for pkNum = 1:size(pkListCntrdXYInt,1)
    % increment frame number
    frameNum = 1;
    % Collect each Integrated Int for pkNum by
        looping through
    %   the master list
    % For pkNum = 1; Frame1 Int k = 1; Frame2
        Int k = 1 + numPks . . .
    for k = pkNum:size(pkListCntrdXYInt,1):size(
        PostListCntrdXYIntFrameNum,1)
         PksTraj(frameNum,pkNum) = ...
             PostListCntrdXYIntFrameNum(k,3);
         frameNum = frameNum + 1;
    end
```

```
end

% save the square of the gyration
% Get data from the big list

% create the holder for PkGyration where the
   format of
% PkGyration is as followed:
% column #:             Pk1           |
             Pk2
%
   ----------------------------------------------------

%    row #1: gyration Int @ Frame1 | gyration
   Int @ Frame1
%    row #2: gyration Int @ Frame2 | gyration
   Int @ Frame2
%
% PkGyration(numFrames,numPks);
PkGyration = zeros(ImageData.numFrames,size(
   pkListCntrdXYInt,1));
% loop through each particle
for pkNum = 1:size(pkListCntrdXYInt,1)
    % increment frame number
    frameNum = 1;
    % Collect each Integrated Int for pkNum by
       looping through
```

```
        %   the master list
        % For pkNum = 1; Frame1 Int k = 1; Frame2
           Int k = 1 + numPks . . .
        for k = pkNum:size(pkListCntrdXYInt,1):size(
           PostListCntrdXYIntFrameNum,1)
            PkGyration(frameNum,pkNum) = ...
                PostListCntrdXYIntFrameNum(k,4);
            frameNum = frameNum + 1;
        end
    end

    % replace NaNs with zeros in the gyration data
    PkGyration(isnan(PkGyration))=0;

    % calculate the average cntrd position
    % create the holder and convert to single
    AvgCntrdxy = zeros(size(pkListCntrdXYInt,1),2);
    AvgCntrdxy = single(AvgCntrdxy);

    % loop through the number of pks and calculate
       the mean Cntrd X
    % and Y at each frame
    for pkNum=1:size(pkListCntrdXYInt,1)
        % rowNumbers = current pkNum's row indices
           at each frame
        rowNumbers = pkNum:size(pkListCntrdXYInt,1):
           size(PostListCntrdXYIntFrameNum,1);
```

```
            % - calculate the mean cntrd X and cntrd Y
               store in
            %       AvgCntrdxy
            % - store in tmp to check for NaN. NaN come
               up from the cntrd
            %       calculation. If the pk is off on
               that frame the cntrd
            %       cannot be calculated
            tmp = PostListCntrdXYIntFrameNum(rowNumbers
               ,:);
            tmp(isnan(tmp(:,1)),:) = [];
            AvgCntrdxy(pkNum,:) = [mean(tmp(:,1)),mean(
               tmp(:,2))];
        end

        % close multiWaitbar
        multiWaitbar( 'Calculating trajectories...','
           Close' );
        Warning = [];
    end % end for calcPksTraj

    %% Function lonerpick, remove overlapping particles
    function [keepers, overlapXY] = ...
            lonerpick(imageSize,rejectRadius,pkList,
               PlotYorN)
```

```
% Create the grid
[xx, yy] = meshgrid(1:imageSize);

% C = false(imageSize,imageSize);
C = zeros(imageSize,imageSize);

% Create the circles on the grid
tic;
parfor n=1:size(pkList,1)

    C = C + ((xx-pkList(n,1)).^2+(yy-pkList(n,2)
       ).^2<=rejectRadius^2);
end
toc;
% list of pks that overlap
overlapXY = zeros(size(pkList,1),2);

keepers = zeros(size(overlapXY,1),2);

tic;
% Check to find particles which overlap if not
   put them on a keepers list
parfor n=1:size(pkList,1)

        try OverLapImage = C(...
                pkList(n,2)-rejectRadius:pkList(
                   n,2)+rejectRadius,...
```

```
                    pkList(n,1)-rejectRadius:pkList(
                        n,1)+rejectRadius);
            catch
                OverLapImage = 2;
            end

            if isempty(OverLapImage(OverLapImage >
                1)) == 0
                overlapXY(n,:) = pkList(n,:);
            else
                keepers(n,:) = pkList(n,:);
            end
    end
    toc;
    % clean up overlapXY and keeprs
    overlapXY(overlapXY(:,1) == 0 & overlapXY(:,2)
        == 0,:) = [];
    keepers(keepers(:,1) == 0 & keepers(:,2) == 0,:)
        = [];

    % clean up
    overlapXY(overlapXY(:,1) == 0 & overlapXY(:,2)
        == 0,:) = [];
    keepers(keepers(:,1) == 0 & keepers(:,2) == 0,:)
        = [];

    % display image
```

```
    if PlotYorN == 1
        figure;
        imshow(C);
        hold on;
        scatter(overlapXY(:,1),overlapXY(:,2),20,'r'
           );
        scatter(keepers(:,1),keepers(:,2),8,'g');
        title(['Circles of original pkList.' ...
            'Reds are overlaps and green are keepers
               . ']);
        hold off;

        % round the result of the keepers
        keepers=round(keepers);
    end
end % end for lonerpick

%% Function NoisePkPicke
function pkListNoise = NoisePkPicke(imageSize,
   rejectRadius,numNoisePks,pkList)

    % create the list of noise particles
    pkListNoise = [...
        randi(...
        [1+round(rejectRadius*5),imageSize-round(
           rejectRadius*5)],numNoisePks,1),...
        randi(...
```

```
    [1+round(rejectRadius*5),imageSize-round(
       rejectRadius*5)],numNoisePks,1)];

% get current size of the Noise list
OldSizeOfpkListNoise = length(pkListNoise);
CurrentSizeOfpkListNoise = length(pkListNoise)
   -1;

% combine the orginal list with the noise
CombinedpkList = [pkList; pkListNoise];

numIter = 0;

while CurrentSizeOfpkListNoise <
   OldSizeOfpkListNoise

    OldSizeOfpkListNoise = length(pkListNoise);

    for n=1:size(pkList,1)
        % calculate the distances from the
           current particle and all other
        % particles
        distance = sqrt((CombinedpkList(n,1)-
           CombinedpkList(:,1)).^2+ ...
            (CombinedpkList(n,2)-CombinedpkList
               (:,2)).^2);
```

```
        % create a list particles cloest to the
           current within the reject
        % radius
        ListOfclosestPks = ...
            CombinedpkList(distance < (2*
               rejectRadius + 1) & distance ~=
               0, :);

        % loop through the list and delete this
           particles
        for j=1:size(ListOfclosestPks,1)

            pkListNoise(...
                pkListNoise(:,1) ==
                   ListOfclosestPks(j,1) & ...
                pkListNoise(:,2) ==
                   ListOfclosestPks(j,2),:) =
                   [];

        end

    end

    CurrentSizeOfpkListNoise = length(
       pkListNoise);

    numIter = numIter + 1;
```

```
        if numIter > 50
            break
        end

    end

    % call the particle find with the noise XY

end

%% Function bpass
function res = bpass(image_array,lnoise,lobject,
   threshold)
    %
    % NAME:
    %               bpass
    % PURPOSE:
    %               Implements a real-space bandpass
        filter that suppresses
    %               pixel noise and long-wavelength
       image variations while
    %               retaining information of a
       characteristic size.
    %
    % CATEGORY:
    %               Image Processing
```

```
% CALLING SEQUENCE:
%               res = bpass( image_array, lnoise
   , lobject )
% INPUTS:
%               image:  The two-dimensional
   array to be filtered.
%               lnoise: Characteristic
   lengthscale of noise in pixels.
%                       Additive noise averaged
   over this length should
%                       vanish. May assume any
   positive floating value.
%                       May be set to 0 or false
   , in which case only the
%                       highpass "background
   subtraction" operation is
%                       performed.
%               lobject: (optional) Integer
   length in pixels somewhat
%                       larger than a typical
   object. Can also be set to
%                       0 or false, in which
   case only the lowpass
%                       "blurring" operation
   defined by lnoise is done,
%                       without the background
   subtraction defined by
```

```
%                       lobject.  Defaults to
   false.
%               threshold: (optional) By default
   , after the convolution,
%                       any negative pixels are
   reset to 0.  Threshold
%                       changes the threshhold
   for setting pixels to
%                       0.  Positive values may
   be useful for removing
%                       stray noise or small
   particles.  Alternatively, can
%                       be set to -Inf so that
   no threshholding is
%                       performed at all.
%
% OUTPUTS:
%               res:    filtered image.
% PROCEDURE:
%               simple convolution yields
   spatial bandpass filtering.
% NOTES:
% Performs a bandpass by convolving with an
   appropriate kernel.  You can
% think of this as a two part process.  First, a
    lowpassed image is
% produced by convolving the original with a
```

```
   gaussian.  Next, a second
% lowpassed image is produced by convolving the
   original with a boxcar
% function. By subtracting the boxcar version
   from the gaussian version, we
% are using the boxcar version to perform a
   highpass.
%
% original - lowpassed version of original =>
   highpassed version of the
% original
%
% Performing a lowpass and a highpass results in
    a bandpassed image.
%
% Converts input to double.  Be advised that
   commands like 'image' display
% double precision arrays differently from UINT8
    arrays.

% MODIFICATION HISTORY:
%               Written by David G. Grier, The
   University of Chicago, 2/93.
%
%               Greatly revised version DGG
   5/95.
%
```

```
%               Added /field keyword JCC 12/95.
%
%               Memory optimizations and fixed
   normalization, DGG 8/99.
%               Converted to Matlab by D.Blair
   4/2004-ish
%
%               Fixed some bugs with conv2 to
   make sure the edges are
%               removed D.B. 6/05
%
%               Removed inadvertent image shift
   ERD 6/05
%
%               Added threshold to output.  Now
   sets all pixels with
%               negative values equal to zero.
   Gets rid of ringing which
%               was destroying sub-pixel
   accuracy, unless window size in
%               cntrd was picked perfectly.  Now
    centrd gets sub-pixel
%               accuracy much more robustly ERD
   8/24/05
%
%               Refactored for clarity and
   converted all convolutions to
```

```
%               use column vector kernels for
   speed.  Running on my
%               macbook, the old version took
   ~1.3 seconds to do
%               bpass(image_array,1,19) on a
   1024 x 1024 image; this
%               version takes roughly half that.
    JWM 6/07
%
%       This code 'bpass.pro' is copyright 1997,
    John C. Crocker and
%       David G. Grier.  It should be considered
    'freeware'- and may be
%       distributed freely in its original form
   when properly attributed.

if nargin < 3, lobject = false; end
if nargin < 4, threshold = 0; end

normalize = @(x) x/sum(x);


image_array = double(image_array);

if lnoise == 0
    gaussian_kernel = 1;
else
```

```
    gaussian_kernel = normalize(...
        exp(-((-ceil(5*lnoise):ceil(5*lnoise))
           /(2*lnoise)).^2));
end

if lobject
    boxcar_kernel = normalize(...
        ones(1,length(-round(lobject):round(
           lobject))));
end

% JWM: Do a 2D convolution with the kernels in
   two steps each.  It is
% possible to do the convolution in only one
   step per kernel with
%
% gconv = conv2(gaussian_kernel',gaussian_kernel
   ,image_array,'same');
% bconv = conv2(boxcar_kernel', boxcar_kernel,
   image_array,'same');
%
% but for some reason, this is slow.  The whole
   operation could be reduced
% to a single step using the associative and
   distributive properties of
% convolution:
%
```

```
% filtered = conv2(image_array,...
%   gaussian_kernel'*gaussian_kernel -
   boxcar_kernel'*boxcar_kernel,...
%   'same');
%
% But this is also comparatively slow (though
   inexplicably faster than the
% above).  It turns out that convolving with a
   column vector is faster than
% convolving with a row vector, so instead of
   transposing the kernel, the
% image is transposed twice.

gconv = conv2(image_array',gaussian_kernel','
   same');
gconv = conv2(gconv',gaussian_kernel','same');

if lobject
    bconv = conv2(image_array',boxcar_kernel','
       same');
    bconv = conv2(bconv',boxcar_kernel','same');

    filtered = gconv - bconv;
else
    filtered = gconv;
end
```

```
    % Zero out the values on the edges to signal
       that they're not useful.
    lzero = max(lobject,ceil(5*lnoise));

    filtered(1:(round(lzero)),:) = 0;
    filtered((end - lzero + 1):end,:) = 0;
    filtered(:,1:(round(lzero))) = 0;
    filtered(:,(end - lzero + 1):end) = 0;

    % JWM: I question the value of zeroing out
       negative pixels.  It's a
    % nonlinear operation which could potentially
       mess up our expectations
    % about statistics.  Is there data on 'Now
       centroid gets subpixel accuracy
    % much more robustly'?  To choose which approach
        to take, uncomment one of
    % the following two lines.
    % ERD: The negative values shift the peak if the
        center of the cntrd mask
    % is not centered on the particle.

    % res = filtered;
    filtered(filtered < threshold) = 0;
    res = filtered;
end % end for bpass
```

```
%% Function pkfnd
function out=pkfnd(im,th,sz)
    % finds local maxima in an image to pixel level
       accuracy.
    %  this provides a rough guess of particle
    %  centers to be used by cntrd.m.  Inspired by
       the lmx subroutine of Grier
    %  and Crocker's feature.pro
    % INPUTS:
    % im: image to process, particle should be
       bright spots on dark background with little
       noise
    %   ofen an bandpass filtered brightfield image
       (fbps.m, fflt.m or bpass.m) or a nice
    %   fluorescent image
    % th: the minimum brightness of a pixel that
       might be local maxima.
    %   (NOTE: Make it big and the code runs faster
    %   but you might miss some particles.  Make it
       small and you'll get
    %   everything and it'll be slow.)
    % sz:  if your data's noisy, (e.g. a single
       particle has multiple local
    % maxima), then set this optional keyword to a
       value slightly larger than the diameter of
       your blob.  if
    % multiple peaks are found withing a radius of
```

```
   sz/2 then the code will keep
% only the brightest.  Also gets rid of all
   peaks within sz of boundary
%OUTPUT:  a N x 2 array containing, [row,column]
    coordinates of local maxima
%           out(:,1) are the x-coordinates of
   the maxima
%           out(:,2) are the y-coordinates of
   the maxima
%CREATED: Eric R. Dufresne, Yale University, Feb
    4 2005
%MODIFIED: ERD, 5/2005, got rid of ind2rc.m to
   reduce overhead on tip by
%  Dan Blair;  added sz keyword
% ERD, 6/2005: modified to work with one and
   zero peaks, removed automatic
%  normalization of image
% ERD, 6/2005: due to popular demand, altered
   output to give x and y
%  instead of row and column
% ERD, 8/24/2005: pkfnd now exits politely if
   there's nothing above
%  threshold instead of crashing rudely
% ERD, 6/14/2006: now exits politely if no
   maxima found
% ERD, 10/5/2006:  fixed bug that threw away
   particles with maxima
```

```
%  consisting of more than two adjacent points



%find all the pixels above threshold
%im=im./max(max(im));
ind=find(im > th);
[nr,nc]=size(im);
tst=zeros(nr,nc);
n=length(ind);
if n==0
    out=[];
%                 display('nothing above threshold');
    return;
end
mx=[];
%convert index from find to row and column
rc=[mod(ind,nr),floor(ind/nr)+1];
for i=1:n
    r=rc(i,1);c=rc(i,2);
    %check each pixel above threshold to see if
       it's brighter than it's neighbors
    %  THERE'S GOT TO BE A FASTER WAY OF DOING
       THIS.  I'M CHECKING SOME MULTIPLE TIMES,
    %  BUT THIS DOESN'T SEEM THAT SLOW COMPARED
       TO THE OTHER ROUTINES, ANYWAY.
    if r>1 & r<nr & c>1 & c<nc
```

```
        if im(r,c)>=im(r-1,c-1) & im(r,c)>=im(r,
           c-1) & im(r,c)>=im(r+1,c-1) & ...
                im(r,c)>=im(r-1,c)  & im(r,c)>=
                   im(r+1,c) &   ...
                im(r,c)>=im(r-1,c+1) & im(r,c)>=
                   im(r,c+1) & im(r,c)>=im(r+1,c
                   +1)
            mx=[mx,[r,c]'];
            %tst(ind(i))=im(ind(i));
        end
    end
end
%out=tst;
mx=mx';

[npks,crap]=size(mx);

%if size is specified, then get ride of pks
   within size of boundary
if nargin==3 & npks>0
    %throw out all pks within sz of boundary;
    ind=find(mx(:,1)>sz & mx(:,1)<(nr-sz) & mx
       (:,2)>sz & mx(:,2)<(nc-sz));
    mx=mx(ind,:);
end

%prevent from finding peaks within size of each
```

```
   other
[npks,crap]=size(mx);
if npks > 1
    %CREATE AN IMAGE WITH ONLY PEAKS
    nmx=npks;
    tmp=0.*im;
    for i=1:nmx
        tmp(mx(i,1),mx(i,2))=im(mx(i,1),mx(i,2))
           ;
    end
    %LOOK IN NEIGHBORHOOD AROUND EACH PEAK, PICK
       THE BRIGHTEST
    for i=1:nmx
        roi=tmp( (mx(i,1)-floor(sz/2)):(mx(i,1)
           +(floor(sz/2)+1)),(mx(i,2)-floor(sz
           /2)):(mx(i,2)+(floor(sz/2)+1))) ;
        [mv,indi]=max(roi);
        [mv,indj]=max(mv);
        tmp( (mx(i,1)-floor(sz/2)):(mx(i,1)+(
           floor(sz/2)+1)),(mx(i,2)-floor(sz/2))
           :(mx(i,2)+(floor(sz/2)+1)))=0;
        tmp(mx(i,1)-floor(sz/2)+indi(indj)-1,mx(
           i,2)-floor(sz/2)+indj-1)=mv;
    end
    ind=find(tmp>0);
    mx=[mod(ind,nr),floor(ind/nr)+1];
end
```

```
    if size(mx)==[0,0]
        out=[];
    else
        out(:,2)=mx(:,1);
        out(:,1)=mx(:,2);
    end
end % end for pkfnd

%% Function cntrd
function out=cntrd(im,mx,sz,interactive)
    % out=cntrd(im,mx,sz,interactive)
    %
    % PURPOSE:  calculates the centroid of bright
       spots to sub-pixel accuracy.
    %  Inspired by Grier & Crocker's feature for IDL
       , but greatly simplified and optimized
    %  for matlab
    %
    % INPUT:
    % im: image to process, particle should be
       bright spots on dark background with little
       noise
    %   ofen an bandpass filtered brightfield image
       or a nice fluorescent image
    %
    % mx: locations of local maxima to pixel-level
```

```
   accuracy from pkfnd.m
%
% sz: diamter of the window over which to
   average to calculate the centroid.
%     should be big enough
%     to capture the whole particle but not so
   big that it captures others.
%     if initial guess of center (from pkfnd) is
    far from the centroid, the
%     window will need to be larger than the
   particle size.  RECCOMMENDED
%     size is the long lengthscale used in bpass
    plus 2.
%
%
% interactive:  OPTIONAL INPUT set this variable
    to one and it will show you the image used
   to calculate
%    each centroid, the pixel-level peak and the
    centroid
%
% NOTE:
%  - if pkfnd, and cntrd return more then one
   location per particle then
%  you should try to filter your input more
   carefully.  If you still get
%  more than one peak for particle, use the
```

```
   optional sz parameter in pkfnd
%  - If you want sub-pixel accuracy, you need to
    have a lot of pixels in your window (sz>>1).
%    To check for pixel bias, plot a histogram
   of the fractional parts of the resulting
   locations
%  - It is HIGHLY recommended to run in
   interactive mode to adjust the parameters
   before you
%    analyze a bunch of images.
%
% OUTPUT:  a N x 4 array containing, x, y and
   brightness for each feature
%           out(:,1) is the x-coordinates
%           out(:,2) is the y-coordinates
%           out(:,3) is the brightnesses
%           out(:,4) is the sqare of the radius
   of gyration
%
% CREATED: Eric R. Dufresne, Yale University,
   Feb 4 2005
%  5/2005 inputs diamter instead of radius
%  Modifications:
%  D.B. (6/05) Added code from imdist/dist to
   make this stand alone.
%  ERD (6/05) Increased frame of reject
   locations around edge to 1.5*sz
```

```
%  ERD 6/2005  By popular demand, 1. altered
   input to be formatted in x,y
%  space instead of row, column space  2. added
   forth column of output,
%  rg^2
%  ERD 8/05  Outputs had been shifted by
   [0.5,0.5] pixels.  No more!
%  ERD 8/24/05  Woops!  That last one was a red
   herring.  The real problem
%  is the "ringing" from the output of bpass.  I
    fixed bpass (see note),
%  and no longer need this kludge.  Also, made
   it quite nice if mx=[];
%  ERD 6/06  Added size and brightness output ot
    interactive mode.  Also
%   fixed bug in calculation of rg^2
%  JWM 6/07  Small corrections to documentation


if nargin==3
    interactive=0;
end

if sz/2 == floor(sz/2)
    warning('sz must be odd, like bpass');
end
```

```
if isempty(mx)
    warning('there were no positions inputted
       into cntrd. check your pkfnd theshold')
    out=[];
    return;
end


r=(sz+1)/2;
%create mask - window around trial location over
    which to calculate the centroid
m = 2*r;
x = 0:(m-1) ;
cent = (m-1)/2;
x2 = (x-cent).^2;
dst=zeros(m,m);
for i=1:m
    dst(i,:)=sqrt((i-1-cent)^2+x2);
end


ind=find(dst < r);

msk=zeros([2*r,2*r]);
msk(ind)=1.0;
%msk=circshift(msk,[-r,-r]);
```

```
dst2=msk.*(dst.^2);
ndst2=sum(sum(dst2));

[nr,nc]=size(im);
%remove all potential locations within distance
   sz from edges of image
ind=find(mx(:,2) > 1.5*sz & mx(:,2) < nr-1.5*sz)
   ;
mx=mx(ind,:);
ind=find(mx(:,1) > 1.5*sz & mx(:,1) < nc-1.5*sz)
   ;
mx=mx(ind,:);

[nmx,crap] = size(mx);

%inside of the window, assign an x and y
   coordinate for each pixel
xl=zeros(2*r,2*r);
for i=1:2*r
    xl(i,:)=(1:2*r);
end
yl=xl';

pts=[];
%loop through all of the candidate positions
for i=1:nmx
    %create a small working array around each
```

```
    candidate location, and apply the window
    function
tmp=msk.*im((mx(i,2)-r+1:mx(i,2)+r),(mx(i,1)
    -r+1:mx(i,1)+r));
%calculate the total brightness
norm=sum(sum(tmp));
%calculate the weigthed average x location
xavg=sum(sum(tmp.*xl))./norm;
%calculate the weighted average y location
yavg=sum(sum(tmp.*yl))./norm;
%calculate the radius of gyration^2
%rg=(sum(sum(tmp.*dst2))/ndst2);
rg=(sum(sum(tmp.*dst2))/norm);

%concatenate it up
pts=[pts,[mx(i,1)+xavg-r,mx(i,2)+yavg-r,norm
    ,rg]'];

%OPTIONAL plot things up if you're in
    interactive mode
if interactive==1
    imagesc(tmp)
    axis image
    hold on;
    plot(xavg,yavg,'x')
    plot(xavg,yavg,'o')
    plot(r,r,'.')
```

```
                    hold off
                    title(['brightness ',num2str(norm),'
                        size ',num2str(sqrt(rg))])
                    pause(0.1)
                end


            end
            out=pts';

        end % end for cntrd

    end % end for methods (Static)
end % end for acquisitionImage (classdef)
```

REFERENCES

(1) Köhler, A.; Bässler, H. *Mater. Sci. Eng. R Rep.* **2009**, *66*, 71–109.

(2) Schueppel, R.; Uhrich, C.; Pfeiffer, M.; Leo, K.; Brier, E.; Reinold, E.; Baeuerle, P. *Chemphyschem* **2007**, *8*, 1497–1503.

(3) Hedley, G. J.; Ruseckas, A.; Samuel, I. D. W. *Chem. Rev.* **2017**, *117*, 796–837.

(4) Kozlov, O. V.; de Haan, F.; Kerner, R. A.; Rand, B. P.; Cheyns, D.; Pshenichnikov, M. S. *Phys. Rev. Lett.* **2016**, *116*, 057402.

(5) Baryshnikov, G.; Minaev, B.; Ågren, H. Theory and Calculation of the Phosphorescence Phenomenon., 2017.

(6) Popp, J.; Kaiser, W.; Gagliardi, A. *Adv. Theory Simul.* **2019**, *2*, 1800114.

(7) Brédas, J.-L.; Norton, J. E.; Cornil, J.; Coropceanu, V. *Acc. Chem. Res.* **2009**, *42*, 1691–1699.

(8) Bittner, E. R.; Lankevich, V.; Gélinas, S.; Rao, A.; Ginger, D. A.; Friend, R. H. *Phys. Chem. Chem. Phys.* **2014**, *16*, 20321–20328.

(9) Veldman, D.; Meskers, S. C. J.; Janssen, R. A. J. *Adv. Funct. Mater.* **2009**, *19*, 1939–1948.

(10) Köhler, A.; Wilson, J. S.; Friend, R. H. *Adv. Mater.* **2002**, *14*, 701–707.

(11) Zhang, Y.; Forrest, S. R. Triplets Contribute to Both an Increase and Loss in Fluorescent Yield in Organic Light Emitting Diodes., 2012.

(12) Rao, A.; Chow, P. C. Y.; Gélinas, S.; Schlenker, C. W.; Li, C.-Z.; Yip, H.-L.; Jen, A. K.-Y.; Ginger, D. S.; Friend, R. H. *Nature* **2013**, *500*, 435–439.

(13) Lunt, R. R.; Giebink, N. C.; Belak, A. A.; Benziger, J. B.; Forrest, S. R. *J. Appl. Phys.* **2009**, *105*, 053711.

(14) González, D. M.; Körstgens, V.; Yao, Y.; Song, L.; Santoro, G.; Roth, S. V.; Müller-Buschbaum, P. *Advanced Energy Materials* **2015**, *5*, 1401770.

(15) Meier, H.; Stalmach, U.; Kolshorn, H. Effective conjugation length and UV/vis spectra of oligomers., 1997.

(16) Yu, J.; Hu, D.; Barbara, P. F. *Science* **2000**, *289*, 1327–1330.

(17) Collini, E.; Scholes, G. D. *Science* **2009**, *323*, 369–373.

(18) Hwang, I.; Scholes, G. D. *Chem. Mater.* **2011**, *23*, 610–620.

(19) Traub, M. C.; Lakhwani, G.; Bolinger, J. C.; Vanden Bout, D.; Barbara, P. F. *J. Phys. Chem. B* **2011**, *115*, 9941–9947.

(20) Hooley, E. N.; Tilley, A. J.; White, J. M.; Ghiggino, K. P.; Bell, T. D. M. *Phys. Chem. Chem. Phys.* **2014**, *16*, 7108–7114.

(21) Hu, Z.; Adachi, T.; Haws, R.; Shuang, B.; Ono, R. J.; Bielawski, C. W.; Landes, C. F.; Rossky, P. J.; Vanden Bout, D. A. *J. Am. Chem. Soc.* **2014**, *136*, 16023–16031.

(22) Hu, Z.; Shao, B.; Geberth, G. T.; Vanden Bout, D. A. *Chem. Sci.* **2018**, *9*, 1101–1111.

(23) Park, H.; Kwon, Y.; Kaufman, L. J. *J. Phys. Chem. C* **2019**, *123*, 1960–1965.

(24) Lee, Y. J.; Kim, D. Y.; Barbara, P. F. *J. Phys. Chem. B* **2006**, *110*, 9739–9742.

(25) Barbara, P. F.; Gesquiere, A. J.; Park, S.-J.; Lee, Y. J. *Acc. Chem. Res.* **2005**, *38*, 602–610.

(26) Gesquiere, A. J.; Lee, Y. J.; Yu, J.; Barbara, P. F. *J. Phys. Chem. B* **2005**, *109*, 12366–12371.

(27) Wöll, D.; Braeken, E.; Deres, A.; De Schryver, F. C.; Uji-i, H.; Hofkens, J. *Chem. Soc. Rev.* **2009**, *38*, 313–328.

(28) Vacha, M.; Habuchi, S. *Npg Asia Materials* **2010**, *2*, 134.

(29) Vanden Bout, D. A.; Yip, W.-T.; Hu, D.; Fu, D.-K.; Swager, T. M.; Barbara, P. F. *Science* **1997**, *277*, 1074–1077.

(30) Grey, J. K.; Kim, D. Y.; Norris, B. C.; Miller, W. L.; Barbara, P. F. *J. Phys. Chem. B* **2006**, *110*, 25568–25572.

(31) Traub, M. C.; Vogelsang, J.; Plunkett, K. N.; Nuckolls, C.; Barbara, P. F.; Vanden Bout, D. A. *ACS Nano* **2012**, *6*, 523–529.

(32) Huser, T.; Yan, M. *J. Photochem. Photobiol. A Chem.* **2001**, *144*, 43–51.

(33) Sarzi Sartori, S.; De Feyter, S.; Hofkens, J.; Van der Auweraer, M.; De Schryver, F.; Brunner, K.; Hofstraat, J. W. *Macromolecules* **2003**, *36*, 500–507.

(34) Liang, J.-J.; White, J. D.; Chen, Y. C.; Wang, C. F.; Hsiang, J. C.; Lim, T. S.; Sun, W. Y.; Hsu, J. H.; Hsu, C. P.; Hayashi, M.; Fann, W. S.; Peng, K. Y.; Chen, S. A. *Phys. Rev. B Condens. Matter* **2006**, *74*, 085209.

(35) Habuchi, S.; Onda, S.; Vacha, M. Molecular weight dependence of emission intensity and emitting sites distribution within single conjugated polymer molecules., 2011.

(36) Yip, W.-T.; Hu, D.; Yu, J.; Vanden Bout, D. A.; Barbara, P. F. Classifying the Photophysical Dynamics of Single- and Multiple-Chromophoric Molecules by Single Molecule Spectroscopy., 1998.

(37) Hu, D.; Yu, J.; Wong, K.; Bagchi, B.; Rossky, P. J.; Barbara, P. F. *Nature* **2000**, *405*, 1030–1033.

(38) Huser, T.; Yan, M.; Rothberg, L. J. *Proc. Natl. Acad. Sci. U. S. A.* **2000**, *97*, 11187–11191.

(39) English, D. S.; Harbron, E. J.; Barbara, P. F. *J. Phys. Chem. A* **2000**, *104*, 9057–9061.

(40) English, D. S.; Furube, A.; Barbara, P. F. *Chem. Phys. Lett.* **2000**, *324*, 15–19.

(41) Ishitobi, H.; Kai, T.; Fujita, K.; Sekkat, Z.; Kawata, S. *Chem. Phys. Lett.* **2009**, *468*, 234–238.

(42) Yu, J.; Lammi, R.; Gesquiere, A. J.; Barbara, P. F. *J. Phys. Chem. B* **2005**, *109*, 10025–10034.

(43) Palacios, R. E.; Barbara, P. F. *J. Fluoresc.* **2007**, *17*, 749–757.

(44) Pensack, R. D.; Song, Y.; McCormick, T. M.; Jahnke, A. A.; Hollinger, J.; Seferos, D. S.; Scholes, G. D. *J. Phys. Chem. B* **2014**, *118*, 2589–2597.

(45) Burrows, H. D.; Seixas de Melo, J.; Serpa, C.; Arnaut, L. G.; Monkman, A. P.; Hamblett, I.; Navaratnam, S. *J. Chem. Phys.* **2001**, *115*, 9601–9606.

(46) Seixas de Melo, J.; Burrows, H. D.; Svensson, M.; Andersson, M. R.; Monkman, A. P. *J. Chem. Phys.* **2003**, *118*, 1550–1556.

(47) Burrows, H. D.; Seixas de Melo, J.; Serpa, C.; Arnaut, L. G.; Miguel, M. d. G.; Monkman, A. P.; Hamblett, I.; Navaratnam, S. *Chem. Phys.* **2002**, *285*, 3–11.

(48) Thomas, A. K.; Brown, H. A.; Datko, B. D.; Garcia-Galvez, J. A.; Grey, J. K. *J. Phys. Chem. C* **2016**, *120*, 23230–23238.

(49) Gillespie, D. T. *Physica A: Statistical Mechanics and its Applications* **1992**, *188*, 404–425.

(50) Gillespie, D. T. *J. Phys. Chem.* **1977**, *81*, 2340–2361.

(51) McQuarrie, D. A. *J. Chem. Phys.* **1963**, *38*, 433–436.

(52) McQuarrie, D. A.; Jachimowski, C. J.; Russell, M. E. *J. Chem. Phys.* **1964**, *40*, 2914–2921.

(53) Laurenzi, I. J. *J. Chem. Phys.* **2000**, *113*, 3315–3322.

(54) Tachiya, M. *Kinetics of Nonhomogeneous Processes* **1987**, 575–670.

(55) Gruber, J. M.; Chmeliov, J.; Krüger, T. P. J.; Valkunas, L.; van Grondelle, R. *Phys. Chem. Chem. Phys.* **2015**, *17*, 19844–19853.

(56) Barzykin, A. V.; Tachiya, M. *J. Phys. Chem. B* **2006**, *110*, 7068–7072.

(57) Jiang, X. M.; Österbacka, R.; Korovyanko, O.; An, C. P.; Horovitz, B.; Janssen, R. A. J.; Vardeny, Z. V. *Adv. Funct. Mater.* **2002**, *12*, 587–597.

(58) Cook, S.; Furube, A.; Katoh, R. *Energy Environ. Sci.* **2008**, *1*, 294–299.

(59) Guo, J.; Ohkita, H.; Benten, H.; Ito, S. *J. Am. Chem. Soc.* **2009**, *131*, 16869–16880.

(60) Banerji, N.; Cowan, S.; Vauthey, E.; Heeger, A. J. *J. Phys. Chem. C* **2011**, *115*, 9726–9739.

(61) Turro, N. J. In *Molecular photochemistry*; W. A. Benjamin, Inc.: New York, New York, 1967, p 21.

(62) Gilbert, A.; Baggott, J. E. In *Essentials of molecular photochemistry*, 1991.

(63) Monkman, A. P.; Burrows, H. D.; Hartwell, L. J.; Horsburgh, L. E.; Hamblett, I.; Navaratnam, S. *Phys. Rev. Lett.* **2001**, *86*, 1358–1361.

(64) Monkman, A. P.; Burrows, H. D.; Hamblett, I.; Navarathnam, S.; Svensson, M.; Andersson, M. R. The effect of conjugation length on triplet energies, electron de-localization and electron–electron correlation in soluble polythiophenes., 2001.

(65) Beljonne, D.; Cornil, J.; Friend, R. H.; Janssen, R. A. J.; Brédas, J. L. *J. Am. Chem. Soc.* **1996**, *118*, 6453–6461.

(66) Zheldakov, I. L.; Wasylenko, J. M.; Elles, C. G. *Phys. Chem. Chem. Phys.* **2012**, *14*, 6211–6218.

(67) Tatchen, J.; Gilka, N.; Marian, C. M. *Phys. Chem. Chem. Phys.* **2007**, *9*, 5209–5221.

(68) Rodriguez-Serrano, A.; Rai-Constapel, V.; Daza, M. C.; Doerr, M.; Marian, C. M. *Photochem. Photobiol. Sci.* **2012**, *11*, 1860–1867.

(69) Hall, D. B.; Underhill, P.; Torkelson, J. M. *Polym. Eng. Sci.* **1998**, *38*, 2039–2045.

(70) Lee, Y. J.; Kim, D. Y.; Grey, J. K.; Barbara, P. F. *Chemphyschem* **2005**, *6*, 2404–2409.

(71) Crocker, J. C.; Grier, D. G. *J. Colloid Interface Sci.* **1996**, *179*, 298–310.

(72) The Matlab Particle Tracking Code Repository., `https://site.physics.georgetown.edu/matlab/`, Accessed: 2019-6-6.

(73) Scharsich, C.; Lohwasser, R. H.; Sommer, M.; Asawapirom, U.; Scherf, U.; Thelakkat, M.; Neher, D.; Köhler, A. Control of aggregate formation in poly(3-hexylthiophene) by solvent, molecular weight, and synthetic method., 2012.

(74) Gavrilenko, A. V.; Matos, T. D.; Bonner, C. E.; Sun, S.-S.; Zhang, C.; Gavrilenko, V. I. *J. Phys. Chem. C* **2008**, *112*, 7908–7912.

(75) Niles, E. T.; Roehling, J. D.; Yamagata, H.; Wise, A. J.; Spano, F. C.; Moulé, A. J.; Grey, J. K. *J. Phys. Chem. Lett.* **2012**, *3*, 259–263.

(76) Placzek, G. *Zeitschrift für Physik* **1931**, *70*, 84–103.

(77) Long, D. A., *The Raman Effect: A Unified Treatment of the Theory of Raman Scattering by Molecules*; Wiley: 2002.

(78) Hester, E. R. In *Molecular spectroscopy : Volume 2*; Royal Society of Chemistry: Cambridge, 1974, p 439.

(79) Tang, J.; Albrecht, A. C. In *Raman Spectroscopy: Theory and Practice*, Szymanski, H. A., Ed.; Springer US: Boston, MA, 1970, pp 33–68.

(80) Asher, S. A. *Annu. Rev. Phys. Chem.* **1988**, *39*, 537–588.

(81) Heller, E. J.; Sundberg, R.; Tannor, D. *J. Phys. Chem.* **1982**, *86*, 1822–1833.

(82) Tannor, D. J.; Heller, E. J. *J. Chem. Phys.* **1982**, *77*, 202–218.

(83) Lee, S.-Y.; Heller, E. J. Time-dependent theory of Raman scattering., 1979.

(84) Shin, K. S. K.; Zink, J. I. *Inorg. Chem.* **1989**, *28*, 4358–4366.

(85) Zink, J. I.; Shin, K.-S. K. In *Advances in Photochemistry*, Volman, D. H., Hammond, G. S., Neckers, D. C., Eds.; Advances in Photochemistry, Vol. 28; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1991, pp 119–214.

(86) Heller, E. J. In *Potential Energy Surfaces and Dynamics Calculations: for Chemical Reactions and Molecular Energy Transfer*, Truhlar, D. G., Ed.; Springer US: Boston, MA, 1981, pp 103–131.

(87) Petrenko, T.; Neese, F. *J. Chem. Phys.* **2007**, *127*, 164319.

(88) Petrenko, T.; Neese, F. *J. Chem. Phys.* **2012**, *137*, 234107.

(89) Schwartz, B. J. *Annu. Rev. Phys. Chem.* **2003**, *54*, 141–172.

(90) Spano, F. C.; Silva, C. *Annu. Rev. Phys. Chem.* **2014**, *65*, 477–500.

(91) Parkinson, P.; Müller, C.; Stingelin, N.; Johnston, M. B.; Herz, L. M. *J. Phys. Chem. Lett.* **2010**, *1*, 2788–2792.

(92) Cadby, A. J.; Partee, J.; Shinar, J.; Martin, S. J.; Spangler, C. W.; Bradley, D. D. C.; Lane, P. A. *Phys. Rev. B Condens. Matter* **2002**, *65*, 245202.

(93) Busby, E.; Carroll, E. C.; Chinn, E. M.; Chang, L.; Moulé, A. J.; Larsen, D. S. *J. Phys. Chem. Lett.* **2011**, *2*, 2764–2769.

(94) Beljonne, D.; Shuai, Z.; Pourtois, G.; Bredas, J. L. *J. Phys. Chem. A* **2001**, *105*, 3899–3907.

(95) Barford, W.; Bursill, R. J.; Makhov, D. V. *Phys. Rev. B Condens. Matter* **2010**, *81*, 035206.

(96) Marian, C. M. *WIREs Comput Mol Sci* **2012**, *2*, 187–203.

(97) Rybicki, J.; Nguyen, T. D.; Sheng, Y.; Wohlgenannt, M. *Synth. Met.* **2010**, *160*, 280–284.

(98) Monkman, A.; Burrows, H. D. *Synth. Met.* **2004**, *141*, 81–86.

(99) Smith, M. B.; Michl, J. *Annu. Rev. Phys. Chem.* **2013**, *64*, 361–386.

(100) Wohlgenannt, M.; Graupner, W.; Österbacka, R.; Leising, G.; Comoretto, D.; Vardeny, Z. V. *Synth. Met.* **1999**, *101*, 267–268.

(101) Busby, E.; Xia, J.; Wu, Q.; Low, J. Z.; Song, R.; Miller, J. R.; Zhu, X.-Y.; Campos, L. M.; Sfeir, M. Y. *Nat. Mater.* **2015**, *14*, 426–433.

(102) Thomas, A. K.; Garcia, J. A.; Ulibarri-Sanchez, J.; Gao, J.; Grey, J. K. *ACS Nano* **2014**, *8*, 10559–10568.

(103) King, S. M.; Matheson, R.; Dias, F. B.; Monkman, A. P. *J. Phys. Chem. B* **2008**, *112*, 8010–8016.

(104) Steiner, F.; Vogelsang, J.; Lupton, J. M. *Phys. Rev. Lett.* **2014**, *112*, 137402.

(105) Heeney, M.; Zhang, W.; Crouch, D. J.; Chabinyc, M. L.; Gordeyev, S.; Hamilton, R.; Higgins, S. J.; McCulloch, I.; Skabara, P. J.; Sparrowe, D.; Tierney, S. Regioregular poly(3-hexyl)selenophene: a low band gap organic hole transporting polymer., 2007.

(106) Reid, O. G.; Pensack, R. D.; Song, Y.; Scholes, G. D.; Rumbles, G. *Chem. Mater.* **2014**, *26*, 561–575.

(107) Paquin, F.; Latini, G.; Sakowicz, M.; Karsenti, P.-L.; Wang, L.; Beljonne, D.; Stingelin, N.; Silva, C. *Phys. Rev. Lett.* **2011**, *106*, 197401.

(108) Ballantyne, A. M.; Chen, L.; Nelson, J.; Bradley, D. D. C.; Astuti, Y.; Maurano, A.; Shuttle, C. G.; Durrant, J. R.; Heeney, M.; Duffy, W.; McCulloch, I. *Adv. Mater.* **2007**, *19*, 4544–4547.

(109) Wang, Y.; Liu, X.; Peng, J.; Qiu, F. *RSC Adv.* **2015**, *5*, 107970–107976.

(110) Clark, J.; Silva, C.; Friend, R. H.; Spano, F. C. *Phys. Rev. Lett.* **2007**, *98*, 206406.

(111) Yamagata, H.; Spano, F. C. *J. Phys. Chem. Lett.* **2014**, *5*, 622–632.

(112) Spano, F. C. *Acc. Chem. Res.* **2010**, *43*, 429–439.

(113) Brinkmann, M. *J. Polym. Sci. B Polym. Phys.* **2011**, *49*, 1218–1233.

(114) Köhler, A.; Hoffmann, S. T.; Bässler, H. *J. Am. Chem. Soc.* **2012**, *134*, 11594–11601.

(115) Wu, E. C.; Stubbs, R. E.; Peteanu, L. A.; Jemison, R.; McCullough, R. D.; Wildeman, J. *J. Phys. Chem. B* **2017**, *121*, 5413–5421.

(116) Hu, Z.; Adachi, T.; Lee, Y.-G.; Haws, R. T.; Hanson, B.; Ono, R. J.; Bielawski, C. W.; Ganesan, V.; Rossky, P. J.; Vanden Bout, D. A. *Chemphyschem* **2013**, *14*, 4143–4148.

(117) Englman, R.; Jortner, J. *Mol. Phys.* **1970**, *18*, 145–164.

(118) Barford, W.; Trembath, D. *Phys. Rev. B Condens. Matter* **2009**, *80*, 165418.

(119) Sanders, S. N.; Kumarasamy, E.; Pun, A. B.; Appavoo, K.; Steigerwald, M. L.; Campos, L. M.; Sfeir, M. Y. *J. Am. Chem. Soc.* **2016**, *138*, 7289–7297.

(120) Basché, T.; Kummer, S.; Bräuchle, C. *Nature* **1995**, *373*, 132–134.

(121) Donati, G.; Lingerfelt, D. B.; Petrone, A.; Rega, N.; Li, X. *J. Phys. Chem. A* **2016**, *120*, 7255–7261.

(122) Samiullah, M.; Moghe, D.; Scherf, U.; Guha, S. *Phys. Rev. B Condens. Matter* **2010**, *82*, 205211.

(123) Andernach, R.; Utzat, H.; Dimitrov, S. D.; McCulloch, I.; Heeney, M.; Durrant, J. R.; Bronstein, H. *J. Am. Chem. Soc.* **2015**, *137*, 10383–10390.

(124) Cekli, S.; Winkel, R. W.; Schanze, K. S. *J. Phys. Chem. A* **2016**, *120*, 5512–5521.

(125) Goswami, S.; Gish, M. K.; Wang, J.; Winkel, R. W.; Papanikolas, J. M.; Schanze, K. S. *ACS Appl. Mater. Interfaces* **2015**, *7*, 26828–26838.

(126) Etinski, M.; Rai-Constapel, V.; Marian, C. M. *J. Chem. Phys.* **2014**, *140*, 114104.

(127) Datko, B. D.; Thomas, A. K.; Fei, Z.; Heeney, M.; Grey, J. K. *Phys. Chem. Chem. Phys.* **2017**, *19*, 28239–28248.

(128) Steiner, F.; Lupton, J. M.; Vogelsang, J. *J. Am. Chem. Soc.* **2017**, *139*, 9787–9790.

(129) Xue, L.; Yang, Y.; Xu, J.; Zhang, C.; Bin, H.; Zhang, Z.-G.; Qiu, B.; Li, X.; Sun, C.; Gao, L.; Yao, J.; Chen, X.; Yang, Y.; Xiao, M.; Li, Y. *Adv. Mater.* **2017**, *29*.

(130) Scholes, G. D. *Annu. Rev. Phys. Chem.* **2003**, *54*, 57–87.

(131) Clauset, A.; Shalizi, C.; Newman, M. *SIAM Rev.* **2009**, *51*, 661–703.

(132) Hoogenboom, J. P.; den Otter, W. K.; Offerhaus, H. L. *J. Chem. Phys.* **2006**, *125*, 204713.

(133) Goldstein, M. L.; Morris, S. A.; Yen, G. G. *The European Physical Journal B - Condensed Matter and Complex Systems* **2004**, *41*, 255–258.

(134) Smolinsky, L. *Journal of the Association for Information Science and Technology* **2017**, *68*, 1792–1795.

(135) Kozankiewicz, B.; Orrit, M. *Chem. Soc. Rev.* **2014**, *43*, 1029–1043.

(136) Barbara, P. F. *Acc. Chem. Res.* **2005**, *38*, 503–503.

(137) Scurlock, R. D.; Wang, B.; Ogilby, P. R.; Sheats, J. R.; Clough, R. L. *J. Am. Chem. Soc.* **1995**, *117*, 10194–10202.

(138) Yeow, E. K. L.; Melnikov, S. M.; Bell, T. D. M.; De Schryver, F. C.; Hofkens, J. *J. Phys. Chem. A* **2006**, *110*, 1726–1734.

(139) Jin, H.; Heller, D. A.; Kim, J.-H.; Strano, M. S. *Nano Lett.* **2008**, *8*, 4299–4304.

(140) Abdou, M. S. A.; Orfino, F. P.; Son, Y.; Holdcroft, S. *J. Am. Chem. Soc.* **1997**, *119*, 4518–4524.

(141) Sperlich, A.; Kraus, H.; Deibel, C.; Blok, H.; Schmidt, J.; Dyakonov, V. *J. Phys. Chem. B* **2011**, *115*, 13513–13518.

(142) Burling, F. T.; Goldstein, B. M. *J. Am. Chem. Soc.* **1992**, *114*, 2313–2320.

(143) Janssen, R. A. J.; Nelson, J. *Adv. Mater.* **2013**, *25*, 1847–1858.

(144) Xiao, S.; Zhang, Q.; You, W. *Adv. Mater.* **2017**, *29*.

(145) Tayebjee, M. J. Y.; McCamey, D. R.; Schmidt, T. W. *J. Phys. Chem. Lett.* **2015**, *6*, 2367–2378.

(146) Zhai, Y.; Sheng, C.; Vardeny, Z. V. *Philos. Trans. A Math. Phys. Eng. Sci.* **2015**, *373*, 20140327–20140327.

(147) Rao, A.; Friend, R. H. *Nature Reviews Materials* **2017**, *2*, 17063.

(148) Xia, J.; Sanders, S. N.; Cheng, W.; Low, J. Z.; Liu, J.; Campos, L. M.; Sun, T. *Adv. Mater.* **2017**, *29*.

(149) Lin, Y. L.; Fusella, M. A.; Kozlov, O. V.; Lin, X.; Kahn, A.; Pshenichnikov, M. S.; Rand, B. P. *Adv. Funct. Mater.* **2016**, *26*, 6489–6494.

(150) Hu, J.; Xu, K.; Shen, L.; Wu, Q.; He, G.; Wang, J.-Y.; Pei, J.; Xia, J.; Sfeir, M. Y. *Nat. Commun.* **2018**, *9*, 2999.

(151) Bange, S.; Scherf, U.; Lupton, J. M. *J. Am. Chem. Soc.* **2012**, *134*, 1946–1949.

(152) Penfold, T. J.; Gindensperger, E.; Daniel, C.; Marian, C. M. *Chem. Rev.* **2018**, *118*, 6975–7025.

(153) Scholes, G. D.; Rumbles, G. *Nat. Mater.* **2006**, *5*, 683–696.

(154) Buchanan, E. A.; Michl, J. *J. Am. Chem. Soc.* **2017**, *139*, 15572–15575.

(155) Hu, D.; Yu, J.; Padmanaban, G.; Ramakrishnan, S.; Barbara, P. F. *Nano Lett.* **2002**, *2*, 1121–1124.

(156) Vallée, R. A. L.; Cotlet, M.; Van der Auweraer, M.; Hofkens, J.; Müllen, K.; De Schryver, F. C. *J. Am. Chem. Soc.* **2004**, *126*, 2296–2297.

(157) Yu, J.; Hu, D.-H.; Barbara, P. F. In *Single Molecule Spectroscopy: Nobel Conference Lectures*, Rigler, R., Orrit, M., Basché, T., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2001, pp 114–129.

(158) Wang, C. F.; White, J. D.; Lim, T. L.; Hsu, J. H.; Yang, S. C.; Fann, W. S.; Peng, K. Y.; Chen, S. A. *Phys. Rev. B Condens. Matter* **2003**, *67*, 035202.

(159) Schindler, F.; Lupton, J. M.; Feldmann, J.; Scherf, U. *Proc. Natl. Acad. Sci. U. S. A.* **2004**, *101*, 14695–14700.

(160) Wohlgenannt, M.; Vardeny, Z. V. *J. Phys. Condens. Matter* **2003**, *15*, R83.

(161) Hofkens, J.; Schroeyers, W.; Loos, D.; Cotlet, M.; Köhn, F.; Vosch, T.; Maus, M.; Herrmann, A.; Müllen, K.; Gensch, T.; De Schryver, F. C. *Spectrochim. Acta A Mol. Biomol. Spectrosc.* **2001**, *57*, 2093–2107.

(162) Kraabel, B.; Moses, D.; Heeger, A. J. *J. Chem. Phys.* **1995**, *103*, 5102–5108.

(163) O'toole, J. T. *J. Appl. Polym. Sci.* **1965**, *9*, 1291–1297.

(164) Hawkett, B. S.; Napper, D. H.; Gilbert, R. G. *J. Chem. Soc. Lond. Faraday Trans. 1* **1977**, *73*, 690–698.

(165) Birtwistle, D. T.; Blackley, D. C. *J. Chem. Soc. Lond. Faraday Trans. 1* **1981**, *77*, 1351–1358.

(166) Ballard, M. J.; Gilbert, R. G.; Napper, D. H. *J. Polym. Sci. B Polym. Lett. Ed.* **1981**, *19*, 533–537.

(167) Barzykin, A. V.; Tachiya, M. *J. Phys. Chem. B* **2003**, *107*, 2953–2957.

(168) Janssen, R. A. J.; Sariciftci, N. S.; Heeger, A. J. *J. Chem. Phys.* **1994**, *100*, 8641–8645.

(169) Hoffmann, S. T.; Athanasopoulos, S.; Beljonne, D.; Bässler, H.; Köhler, A. *J. Phys. Chem. C* **2012**, *116*, 16371–16383.

(170) Tempelaar, R.; Stradomska, A.; Knoester, J.; Spano, F. C. *J. Phys. Chem. B* **2013**, *117*, 457–466.

(171) Shi, T.; Li, H.; Tretiak, S.; Chernyak, V. Y. *J. Phys. Chem. Lett.* **2014**, *5*, 3946–3952.

(172) Hoffman, D. P.; Leblebici, S. Y.; Schwartzberg, A. M.; Mathies, R. A. *J. Phys. Chem. Lett.* **2015**, *6*, 2919–2923.

(173) Girotto, C.; Cheyns, D.; Aernouts, T.; Banishoeib, F.; Lutsen, L.; Cleij, T. J.; Vanderzande, D.; Genoe, J.; Poortmans, J.; Heremans, P. *Org. Electron.* **2008**, *9*, 740–746.

(174) Henckens, A.; Knipper, M.; Polec, I.; Manca, J.; Lutsen, L.; Vanderzande, D. *Thin Solid Films* **2004**, *451-452*, 572–579.

(175) Jiang, Y.; Peng, Q.; Gao, X.; Shuai, Z.; Niu, Y.; Lin, S. H. *J. Mater. Chem.* **2012**, *22*, 4491–4501.

(176) Smith, A. P.; Smith, R. R.; Taylor, B. E.; Durstock, M. F. *Chem. Mater.* **2004**, *16*, 4687–4692.

(177) Nguyen, L. H.; Günes, S.; Neugebauer, H.; Sariciftci, N. S.; Colladet, K.; Fourier, S.; Cleij, T. J.; Lutsen, L.; Gelan, J.; Vanderzande, D. *Eur. Phys. J. Appl. Phys.* **2006**, *36*, 219–223.

(178) Huo, L.; Chen, T. L.; Zhou, Y.; Hou, J.; Chen, H.-Y.; Yang, Y.; Li, Y. *Macromolecules* **2009**, *42*, 4377–4380.

(179) Lafalce, E.; Toglia, P.; Zhang, C.; Jiang, X. *Appl. Phys. Lett.* **2012**, *100*, 213306.

(180) Meng, K.; Ding, Q.; Wang, S.; He, Y.; Li, Y.; Gong, Q. *J. Phys. Chem. B* **2010**, *114*, 2602–2606.

(181) Olejnik, E.; Pandit, B.; Basel, T.; Lafalce, E.; Sheng, C.-X.; Zhang, C.; Jiang, X.; Vardeny, Z. V. *Phys. Rev. B Condens. Matter* **2012**, *85*, 235201.

(182) Liess, M.; Jeglinski, S.; Lane, P. A.; Vardeny, Z. V. *Synth. Met.* **1997**, *84*, 891–892.

(183) Golovnin, I. V.; Paraschuk, D. Y.; Pan, X. Y.; Chigarev, N. V.; Knize, R. J.; Zhdanov, B. V.; Kobryanskii, V. M. *Synth. Met.* **2001**, *116*, 53–56.

(184) Lane, P. A.; Wei, X.; Vardeny, Z. V. *Phys. Rev. Lett.* **1996**, *77*, 1544–1547.

(185) Ozaki, M.; Ehrenfreund, E.; Benner, R. E.; Barton, T. J.; Yoshino, K.; Vardeny, Z. V. *Phys. Rev. Lett.* **1997**, *79*, 1762–1765.

(186) Musser, A. J.; Al-Hashimi, M.; Maiuri, M.; Brida, D.; Heeney, M.; Cerullo, G.; Friend, R. H.; Clark, J. *J. Am. Chem. Soc.* **2013**, *135*, 12747–12754.

(187) Bardeen, C. J. *Annu. Rev. Phys. Chem.* **2014**, *65*, 127–148.

(188) Diliën, H.; Marin, L.; Botek, E.; Champagne, B.; Lemaur, V.; Beljonne, D.; Lazzaroni, R.; Cleij, T. J.; Maes, W.; Lutsen, L.; Vanderzande, D.; Adriaensens, P. J. *J. Phys. Chem. B* **2011**, *115*, 12040–12050.

(189) Adachi, T.; Brazard, J.; Ono, R. J.; Hanson, B.; Traub, M. C.; Wu, Z.-Q.; Li, Z.; Bolinger, J. C.; Ganesan, V.; Bielawski, C. W.; Vanden Bout, D. A.; Barbara, P. F. *J. Phys. Chem. Lett.* **2011**, *2*, 1400–1404.

(190) Paquin, F.; Yamagata, H.; Hestand, N. J.; Sakowicz, M.; Bérubé, N.; Côté, M.; Reynolds, L. X.; Haque, S. A.; Stingelin, N.; Spano, F. C.; Silva, C. *Phys. Rev. B Condens. Matter* **2013**, *88*, 155202.

(191) Frolov, S.; Leng, J. M.; Vardeny, Z. V. *Mol. Cryst. Liq. Cryst. Sci. Technol. Sect. A* **1994**, *256*, 473–479.

(192) Frolov, S. V.; Vardeny, Z. V. *Synth. Met.* **1997**, *84*, 905–906.

(193) Guoshun, Y.; Keda, H.; Yang, Q. *J. Polym. Sci. Part A: Polym. Chem.* **2014**, *52*, 591–595.

(194) Nguyen, T.-Q.; Doan, V.; Schwartz, B. J. *J. Chem. Phys.* **1999**, *110*, 4068–4078.

(195) Gao, J.; Thomas, A. K.; Yang, J.; Aldaz, C.; Yang, G.; Qin, Y.; Grey, J. K. *J. Phys. Chem. C* **2015**, *119*, 8980–8990.

(196) Bittner, E. R.; Karabunarliev, S.; Herz, L. M. *J. Chem. Phys.* **2007**, *126*, 191102.

(197) Spano, F. C. *J. Chem. Phys.* **2005**, *122*, 234701.

(198) Zhao, Z.; Spano, F. C. *J. Phys. Chem. C* **2007**, *111*, 6113–6123.

(199) Roehling, J. D.; Arslan, I.; Moulé, A. J. *J. Mater. Chem.* **2012**, *22*, 2498–2506.

(200) Louarn, G.; Mévellec, J. Y.; Lefrant, S.; Buisson, J. P.; Fichou, D.; Teulade-Fichou, M. P. *Synth. Met.* **1995**, *69*, 351–352.

(201) Mevellec, J. Y.; Buisson, J. P.; Lefrant, S.; Eckhard, H.; Jen, K. Y. *Synth. Met.* **1990**, *35*, 209–213.

(202) Reber, C.; Zink, J. I. *J. Phys. Chem.* **1992**, *96*, 571–576.

(203) Chen, L.; Zhu, L.; Shuai, Z. *J. Phys. Chem. A* **2006**, *110*, 13349–13354.

(204) Rosspeintner, A.; Lang, B.; Vauthey, E. *Annu. Rev. Phys. Chem.* **2013**, *64*, 247–271.

(205) Ohkita, H.; Cook, S.; Astuti, Y.; Duffy, W.; Tierney, S.; Zhang, W.; Heeney, M.; McCulloch, I.; Nelson, J.; Bradley, D. D. C.; Durrant, J. R. *J. Am. Chem. Soc.* **2008**, *130*, 3030–3042.

(206) Fazzi, D.; Barbatti, M.; Thiel, W. *Phys. Chem. Chem. Phys.* **2015**, *17*, 7787–7799.

(207) Siegert, S.; Vogeler, F.; Marian, C. M.; Weinkauf, R. *Phys. Chem. Chem. Phys.* **2011**, *13*, 10350–10363.

(208) Kölle, P.; Schnappinger, T.; de Vivie-Riedle, R. *Phys. Chem. Chem. Phys.* **2016**, *18*, 7903–7915.

(209) Mai, S.; Marquetand, P.; González, L. *Int. J. Quantum Chem.* **2015**, *115*, 1215–1231.

(210) Renaud, N.; Grozema, F. C. *J. Phys. Chem. Lett.* **2015**, *6*, 360–365.

(211) Datko, B. D.; Livshits, M. Y.; Zhang, Z.; Portlock, D.; Qin, Y.; Rack, J. J.; Grey, J. K. *Phys. Chem. Chem. Phys.* **2018**, *20*, 22159–22167.

(212) DiCésare, N.; Belletête, M.; Marrano, C.; Leclerc, M.; Durocher, G. *J. Phys. Chem. A* **1999**, *103*, 795–802.

(213) Elfers, N.; Lyskov, I.; Spiegel, J. D.; Marian, C. M. *J. Phys. Chem. C* **2016**, *120*, 13901–13910.

(214) Gierschner, J.; Mack, H.-G.; Lüer, L.; Oelkrug, D. *J. Chem. Phys.* **2002**, *116*, 8596–8609.

(215) Lin, J. B.; Jin, Y.; Lopez, S. A.; Druckerman, N.; Wheeler, S. E.; Houk, K. N. *J. Chem. Theory Comput.* **2017**, *13*, 5624–5638.

(216) Millefiori, S.; Alparone, A.; Millefiori, A. *J. Heterocycl. Chem.* **2000**, *37*, 847–853.

(217) Zimmerman, A. A.; Orlando, C. M.; Gianni, M. H.; Weiss, K. *J. Org. Chem.* **1969**, *34*, 73–77.

(218) Mai, S.; Marquetand, P.; González, L. *J. Phys. Chem. Lett.* **2016**, *7*, 1978–1983.

(219) Schnappinger, T.; Kölle, P.; Marazzi, M.; Monari, A.; González, L.; de Vivie-Riedle, R. *Phys. Chem. Chem. Phys.* **2017**, *19*, 25662–25670.

(220) Yuan, J.; Zhang, Y.; Zhou, L.; Zhang, G.; Yip, H.-L.; Lau, T.-K.; Lu, X.; Zhu, C.; Peng, H.; Johnson, P. A.; Leclerc, M.; Cao, Y.; Ulanski, J.; Li, Y.; Zou, Y. *Joule* **2019**, *3*, 1140–1151.

(221) Shockley, W.; Queisser, H. J. *J. Appl. Phys.* **1961**, *32*, 510–519.

(222) Dexter, D. L. *J. Lumin.* **1979**, *18-19*, 779–784.

(223) Hanna, M. C.; Nozik, A. J. *J. Appl. Phys.* **2006**, *100*, 074510.

(224) Nelson, C. A.; Monahan, N. R.; -Y. Zhu, X. *Energy Environ. Sci.* **2013**, *6*, 3508–3519.

(225) Blackley, D. C. In *Emulsion Polymers and Emulsion Polymerization*; ACS Symposium Series, Vol. 165; AMERICAN CHEMICAL SOCIETY: 1981, pp 437–454.

(226) Steinfeld, J. I.; Francisco, J. S.; Hase, W. L., *Chemical Kinetics and Dynamics*; Prentice Hall: 1999.

(227) Van Kampen, N. G., *Stochastic Processes in Physics and Chemistry*; Elsevier: 2011.

(228) Ash, C., *The probability tutoring book: an intuitive course for engineers and scientists (and everyone else!)*; IEEE Press: 1993.

(229) multiWaitbar(label, varargin) - File Exchange - MATLAB Central., https://www.mathworks.com/matlabcentral/fileexchange/26589-multiwaitbar-label-varargin, Accessed: 2019-6-10.

(230) Andor MATLAB Sif library example- File Exchange., http://www.andor.com/exchange/file?id=54, Accessed: 2019-6-10.