

IT'S GOOD TO BE FLEXIBLE: ENERGY TRANSPORT FACILITATED BY  
CONFORMATIONAL FLUCTUATIONS IN LIGHT-HARVESTING POLYMERS

Leah M. Rader Bowers

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Chemistry.

Chapel Hill  
2020

Approved by:  
John Papanikolas  
Joanna Atkin  
Andy Moran  
Marcey Waters  
Alex Miller

©2020  
Leah M. Rader Bowers  
ALL RIGHTS RESERVED

## ABSTRACT

Leah M. Rader Bowers: It's Good to be Flexible: Energy transport facilitated by conformational fluctuations in light-harvesting polymers  
(Under the Direction of John M. Papanikolas)

We investigate the mechanism of energy transfer between Ruthenium (II) (Ru) and Osmium (II) (Os) polypyridyl complexes affixed to a polyfluorene backbone (PF-RuOs) using a combination of time-resolved emission spectroscopy and coarse-grained molecular dynamics (CG MD). While we can determine the *total* energy transfer rate within an ensemble of solvated PF-RuOs from time-dependent Os\* emission spectra, system heterogeneity and flexibility give rise to highly multi-exponential kinetics. We developed a three-part computational kinetic model to supplement our spectroscopic results: 1) CG MD model of PF-RuOs that simulates molecular motions out to 700 ns and reduces computational time by ~400x compared to all atom MD, 2) Energy transfer kinetic simulations in CG MD PF-RuOs, and 3) Computational experiments that interrogate the mechanisms by which motion aids energy transfer. Good agreement between simulated and experimental emission transients reveals our kinetic model accurately simulates the molecular motion of PF-RuOs during energy transfer. We find that pendant flexibility allows 81% of the excited state to sensitize an Os trap compared to a 48% occupation when we treat pendants statically. Static pendants are only able to engage in local energy transfer. While flexibility enables pendants to swing in and out of the original domain spreading the excited state out to  $\pm 30$  away from the initial excitation. This work was made possible through extensive collaborations with the groups of Dr. John Reynolds and Dr. Kirk Schanze.

*To my loving parents and siblings...*

## ACKNOWLEDGEMENTS

First, I would like to thank my advisor John Papanikolas for his support and guidance throughout my tenure at UNC. I have learned so much about science, communication, and teamwork as a part of the Papanikolas group and I will take these lessons with me throughout my career. I have had the privilege of working with amazing people throughout my time in the Papanikolas group. In particular, I would like to thank Dr. Melissa Gish, Dr. Lenzi Williams, Dr. Emma Cating-Subramanian, and Dr. Erika Van Goethem for being amazing mentors to me at my time at UNC. You heard me, supported me and showed through your own actions how chemistry was very much a part of your lives but not all of it. I am grateful for all the lessons of self-worth and validating my mantra of keeping mental health as my number one priority. You all were so patient and understanding throughout and guided me in a way that was empathetic to the anti-Black and anti-Femme pandemic that followed me into our group's lab every day. I appreciate you for your friendship and familial nature more than you know. Anyone would be lucky to have y'all as their mentors and friends.

My family is my world. To my mom, Miriam, dad, Richard, brother and sister – Eve and Levi, I love you all so much. You have been there by my side every step of the way even though you were a 7-hour drive away. You constantly cheered me on even though you often had no clue what I was so happy about – a code that finally went right or getting up and doing the bare minimum on hard days. You lifted me up when I faced difficult decisions or had to have tough conversations or when I did not feel like I belonged. I loved your random calls to tell me some

joke or story that only you thought was funny or the ones you'd tell me that had us laughing until I couldn't breathe. Thank you for visiting often and always checking in on me. I loved every time I got to go back to Pittsburgh – those visits always helped me recharge. I am the most grateful to have you all as parents and siblings.

To my grandparents who weren't able to be here to see me cross the finish line but who I know are always cheering me on. I inherited so much from each of you. Grandpa Grandpa – your ease and willingness to make life into a game made my tenure here a lot more fun. Grandma Janet – my ability to sing with the voice you, my mom and dad gave me got me through all of this program's challenges. Grammy – I thank you for teaching me how to stick up for myself and others with grace and strength. Grandpa – your humor and lightness has carried me a long way in making great friends I care about deeply.

Friends – there are too many of you to acknowledge here but I will try. Taylor Moot and Melissa Gish were two of the first people who I didn't feel afraid to be my goofy self around. They provided a safe space in which I was able to express myself freely without being perceived as less intelligent. Taylor and Melissa were also so ready to hang outside of our labs to grab coffee and chat about the realities of being a woman in a PhD program. They made me feel less alone and taught me that it was ok to make mistakes, do my best and that how to set my own hours and be just as or more productive. From them I learned that there was no one way I had to be a scientist and for that freedom I will be forever grateful. Big ups go to my friends in the department – (soon to be Dr.) David Abraham, (soon to be Dr.) Olivia Sanchez-Felix, (soon to be Dr.) Taylor Teitsworth, Dr. Olivia Williams, Dr. Jimmy Custer and Dr. Aaron Taggart. Y'all showed me what it was like to hold multiple interests while pursuing a graduate career. We didn't just talk Chemistry all the time. The times we were able to just hang out are memories I'll

cherish forever. Taylor, Olivia, and Olivia specifically, I am grateful for our multitude of conversations surrounding racism and sexism in academia. We spoke on its existence, held each other up and I know y'all will forever have my backs just as I'll have yours. Thank you Jimmy, Aaron, and David for being our accomplices to fight against discrimination that happens regularly, yes even in the UNC Chemistry department. I trust that y'all will continue that fight when you move to the next stage in your lives.

Big ups also for my great friends outside the department – Lauren, Cassie, Sarah Margaret, Kara, Ben, Dempsey, Leslie, CC and Sydney. Y'all make me whole! Talks with you have had the power of turning bad days into great ones. I've had some of my hardest laughs and greatest adventures with y'all and your help in restructuring the system we live in makes my soul sing to every single Lizzo song. You get me. You believe me. You're there for me and I cannot express how grateful I truly am for each and every one of you.

## TABLE OF CONTENTS

LIST OF TABLES.....	x
LIST OF FIGURES.....	xii
LIST OF ABBREVIATIONS AND SYMBOLS.....	xvii
Chapter 1: Introduction.....	1
1.1. Overview.....	7
Chapter 2: Experimental Methods.....	9
2.1. Synthesis and Material.....	9
2.2. Spectroscopic Experiments.....	10
2.2.1. Steady-State Absorption and Emission Methods.....	10
2.2.2. Time-Resolved Emission Methods.....	11
2.3. Experimental Results and Discussion.....	13
2.3.1. Structural Characterization of PF-RuOs.....	13
2.3.2. Evidence of Energy Transfer.....	19
Chapter 3 : Molecular Dynamic (MD) Simulations.....	23
3.1. Introduction.....	23
3.2. System Preparation.....	25
3.2.1. All-Atom (AA) System.....	25
3.2.2. Motion Group (MG)-Defined System.....	30



3.2.3.	System Parameterization.....	35
3.2.4.	Coarse-Grain (CG) System.....	43
3.3.	Results and Discussion.....	44
3.3.1.	CG MD Simulation Results.....	45
3.3.2.	Nearest Neighbors (NNs) Analysis.....	46
3.3.3.	Follow Original NN and NNN Analysis.....	47
Chapter 4: CG MD Energy Transfer (EnT) Simulations .....		51
4.1.	Ru Coordinate Extraction.....	51
4.2.	EnT Simulations.....	52
4.3.	Comparing Simulation to Experiment.....	59
4.3.1.	Model Accuracy.....	59
4.3.2.	Sample Preparation for Transient Absorption.....	59
4.4.	Computational Experiments and Discussion.....	64
4.4.1.	Diffusion.....	64
4.5.	Conclusion.....	73
Appendix A: MD Trajectory Code .....		74
A.1.	Job Submission.....	74
A.2.	AA and CG MD Trajectory.....	75
Appendix B: EnT Simulations Code.....		86
B.1.	Ru Coordinate Extraction.....	86

B.1.1.    EnT Simulation Code.....	89
Appendix C: Computational Experiments Code.....	99
C.1. Ru Coordinate Extraction. ....	99
C.1.1.    Follow Original NN and NNN Analysis.....	102
C.1.2.    Excited State Diffusion. ....	105
REFERENCES. ....	110

## LIST OF TABLES

<b>Table 3.1.</b> Bead type description of atoms. ....	31
<b>Table 3.2.</b> Bead type set equilibrium bond lengths, $R_0$ , and force constants, $k_0$ , for bond stretching potential.....	38
<b>Table 3.3.</b> Bead type charges. ....	39
<b>Table 3.4.</b> Bead type sigmas, $\sigma_{ij}$ . ....	40
<b>Table 3.5.</b> Bead type set equilibrium angles, $\theta_0$ , and force constants, $k_A$ , that provide best fits for angle bending potential. ....	41
<b>Table 3.6.</b> Bead type sigmas, $\sigma_{ij}$ . for Lennard-Jones 12-6 potential.....	42
<b>Table 3.7.</b> Bead type interaction energies, $\varepsilon_{ij}$ , for Lennard-Jones 12-6 potential.....	42
<b>Table 4.1.</b> EnT simulation rate constants at closest approach ( $k_0^{Ru}$ and $k_0^{Os}$ ) that yield best agreement between simulation and experiment for different attenuation parameter values, $\beta$ ...	60
<b>Table A.1.</b> Files required to run a 40 ns AA MD trajectory. ....	76
<b>Table A.2.</b> Files required to run a histogram analysis angles between a specific set of beads. ...	79
<b>Table A.3.</b> Files required to run the first part of a 700 ns CG MD trajectory over 350 ns. ....	80
<b>Table A.4.</b> Files required to run the second part of a 700 ns CG MD trajectory over 350 ns. ....	83

<b>Table B.1.</b> Files required to extract the XYZ coordinates of the Ru complexes in the 80mer CG MD model. ....	86
<b>Table B.2.</b> Main output files of the extractXYZ.pl script. ....	89
<b>Table B.3.</b> Files required to run EnT simulations on 70mer CG MD models. ....	90
<b>Table C.1.</b> Files required to run NN distance calculations on 70mer CG MD models.....	99
<b>Table C.2.</b> Files required to run original NN and NNN distance calculations on 70mer CG MD models. ....	102

## LIST OF FIGURES

- Figure 1.1.** UV-visible absorption spectrum of Ru ( $[\text{Ru}(\text{bpy})_3]^{2+}$ ) in acetonitrile. The electronic transitions are labeled, along with a depiction of the metal-to-ligand charge transfer (MLCT) transition in  $[\text{Ru}(\text{bpy})_3]^{2+}$ . ..... 2
- Figure 1.2.** Simplified molecular orbital diagram for  $[\text{Ru}(\text{bpy})_3]^{2+}$ . ..... 3
- Figure 1.3.** (A) Chemical structure of Polystyrene-Ruthenium (PS-Ru). (B) Coarse-grained 20 complex representation of the polymeric assembly,  $[\text{PS-Ru}_{17}\text{Os}_3]^{40+}$  (PS-RuOs) where Ru (red) and Os (purple) complexes are bound by a pendant linkage to the polystyrene backbone. .... 4
- Figure 1.4.** (A) Chemical structures of the Polyfluorene-Ruthenium (PF-Ru), Ru and Os Models. (B) A 20-complex representation of the polymeric assembly,  $[\text{PF-Ru}_{60}\text{Os}_{10}]^{140+}$  (PF-RuOs) tested experimentally, where Ru (red) and Os (purple) complexes are bound by a pendant linkage to a conjugated polyfluorene backbone (pink). ..... 6
- Figure 2.1.** Synthetic routes to polymers PF-RuOs, PF-Ru, PF-Os, Ru and Os Models. In this study, PF-Ru, PF-Os, Ru and Os models serve as control systems. .... 10
- Figure 2.2.** Schematic of the time-resolved emission set-up. Important optical components are labeled as follows M = mirror, BS = beamsplitter, AOM = acousto-optic modulator, L = lens, BBO =  $\beta$ -Barium Borate crystal, WP = waveplate, PC = polarizing cube, S = sample, HPF = high pass filters. .... 11
- Figure 2.3.** Absorption spectra of the PF-RuOs assembly; PF, Ru, and Os Model systems in room temperature acetonitrile. Spectra of Ru and Os Model systems are weighted by 0.85 and 0.15, respectively, to represent the relative concentration of each complex within the assembly. The arrow identifies the excitation wavelength used for time-resolved emission experiments. .. 13
- Figure 2.4.** Jablonski diagram of an EnT pathway in PF-RuOs. .... 14

**Figure 2.5.** Steady-state emission spectra of PF-RuOs (black), Ru Model (PF-Ru, red), and Os Model (PF-Os, purple) after continuous excitation at 450 nm at 450 W. Shaded regions under PF-Ru and PF-Os curves indicate the spectral ranges at which time-resolved Ru\* and Os\* emissions in PF-RuOs are integrated. .... 17

**Figure 2.6.** Time-resolved Ru\* and Os\* emission transients observed in the Ru Model (red) and PF-RuOs (purple) around 670 and 800 nm, respectively. All peak intensities are normalized to 1. .... 19

**Figure 2.7.** (A) Time-resolved Ru\* (red) and Os\* (purple) emission transients observed in PF-RuOs around 670 and 800 nm, respectively, with peak intensities normalized to 1. (B) Os\* emission plotted over the first 20 ns and overlaid with the instrument response function (inset). (C) Residuals from single and biexponential fits of delayed rise in Os\* emission. .... 20

**Figure 3.1.** (A) Coarse-graining procedure for the polyfluorene-ruthenium repeat unit (FRu<sub>2</sub>). (B) Coarse-graining procedure for the acetonitrile (MeCN) solvent. (C) Coarse-graining procedure for the hexafluorophosphate (PF<sub>6</sub><sup>-</sup>) counterion. From left to right, each panel depicts each component as a chemical structure, all-atom (AA), motion group (MG) and coarse-grained (CG) model. .... 24

**Figure 3.2.** All-atom (AA) depiction of Polyfluorene-Ruthenium monomer unit (AA FRu<sub>2</sub>), acetonitrile (AA MeCN), and hexafluorophosphate (AA PF<sub>6</sub><sup>-</sup>). Grey atoms represent C atoms, Ru atoms are shown in teal, blue atoms are N, O atoms are represented by red, pink represents P atoms, F atoms are represented in light blue, and white indicates H atoms. .... 26

**Figure 3.3.** Motion Group (MG) depiction of polyfluorene-ruthenium monomer unit (MG FRu<sub>2</sub>), acetonitrile (MG MeCN), and hexafluorophosphate (MG PF<sub>6</sub><sup>-</sup>). MGs are identified by their type A – I. Red transparent spheres illustrate atoms belonging to Ru complexes, darker blue spheres represent atoms belonging to the pendant ligands, and spheres in light blue illustrate polymer backbone atoms. Pink and green spheres represent the methyl and nitrile groups, respectively, and PF<sub>6</sub><sup>-</sup> is represented by a grey sphere. No atom belongs to two different MGs. .... 30

**Figure 3.4.** MG-depiction of PF-Ru<sub>20</sub> (MG 20mer) after polymerizing FRu<sub>2</sub> out ten units. .... 32

**Figure 3.5.** Simulation unit cell constructed with MG 20mer and MG PF<sub>6</sub><sup>-</sup> counterions. .... 34

**Figure 3.6.** Four-unit simulation cell with MG-defined PF-Ru<sub>80</sub> (MG 80mer) and MG PF<sub>6</sub><sup>-</sup> counterions. .... 34

<b>Figure 3.7.</b> Four-unit simulation cell with the MG 80mer and PF <sub>6</sub> <sup>-</sup> counterions and packed with MG MeCN solvent molecules. ....	35
<b>Figure 3.8.</b> (A) MG-defined AA simulation cell. (B) Process for measuring equilibrium bond length between two MGs. (C) Process for measuring equilibrium angle between three MGs. (D) Process for measuring bead size across bonds an MG are shown. ....	37
<b>Figure 3.9.</b> Angle Analysis Fitting. The population distribution of the equilibrium angles (teal) is fit by the angle bending interaction potential function (black). ....	41
<b>Figure 3.10.</b> Coarse grained (CG) depiction of FRu <sub>2</sub> , MeCN, and PF <sub>6</sub> <sup>-</sup> with beads identified by their type A-I. ....	43
<b>Figure 3.11.</b> Histogrammed distances of all NN (green) and NNN (blue) pairs in the CG MD PF-Ru 70mer over 350 ns. ....	46
<b>Figure 3.12.</b> Histogrammed distances of original NN and NNN pairs in six different CG MD PF-Ru 70mer trajectories and multiple starting structures (T <sub>0</sub> s) recorded at times t. ....	47
<b>Figure 3.13.</b> Total bin counts of original NN and NNN distances above 60 Å over time. ....	48
<b>Figure 3.14.</b> Molecular diffusion of NN (orange) and NNN (green) of Ru <sub>35</sub> (yellow) in Trajectory 7 at times t. ....	49
<b>Figure 4.1.</b> The 80mer CG PF-Ru model with sets “A” labeled. The cell is shown without solvent for clarity. ....	51
<b>Figure 4.2.</b> Description of EnT simulation steps. ....	54
<b>Figure 4.3.</b> Flow chart for running EnT simulations on an ensemble of CG MD PF-RuOs 70mer models to accurately fit experimental EnT kinetic traces. ....	58
<b>Figure 4.4.</b> Overlaid fitting with flexible CG MD PF-RuOs simulated kinetics for parameter set with $\beta = 1.5 \text{ \AA}^{-1}$ . Plot depicts Os* emission (purple) fit with the simulated Os* time-dependent population (black) overlaid with Ru emission (red) fit with the simulated Ru* time-dependent population (grey). The simulated ground state Os population intensity after 700 ns reaches 81% (wine). ....	61

<b>Figure 4.5.</b> Overlaid fitting with static CG MD PF-RuOs simulated kinetics for parameter set with $\beta = 1.5 \text{ \AA}^{-1}$ . Plot depicts Os* emission (purple) fit with the simulated Os* time-dependent population (black) overlaid with Ru emission (red) fit with the simulated Ru* time-dependent population (grey). The simulated ground state Os population intensity after 700 ns reaches 48% (green).....	62
<b>Figure 4.6.</b> (A) ES diffusion after initial excitation of Ru <sub>35</sub> in Trajectories 1-6 with flexible pendants at t = 350 ns. (B) ES diffusion after initial excitation of Ru <sub>35</sub> in a system with static pendants at t = 350 ns. ....	65
<b>Figure 4.7.</b> (A) Averaged time-dependent ES diffusion plots after initially exciting Ru <sub>35</sub> in Trajectory 5 with flexible pendants at times t. (B) Averaged time-dependent ES diffusion plots after initially exciting Ru <sub>35</sub> in Trajectory 5 with static pendants at times t. ....	66
<b>Figure 4.8.</b> Molecular diffusion of primary domain (orange) after excitation of Ru <sub>35</sub> (yellow) in Trajectory 5 at times t. ....	67
<b>Figure 4.9.</b> (A) Averaged time-dependent ES diffusion plots after initially exciting Ru <sub>35</sub> in Trajectory 2 with flexible pendants at times t. (B) Averaged time-dependent ES diffusion plots after initially exciting Ru <sub>35</sub> in Trajectory 2 with static pendants at times t. ....	68
<b>Figure 4.10.</b> Molecular diffusion of primary (orange) and secondary (green) domains after excitation of Ru <sub>35</sub> (yellow) in Trajectory 2 at times t. ....	69
<b>Figure 4.11.</b> (A) Averaged time-dependent ES diffusion plots on Ru out to $\pm 30$ away from the initial excited Ru at time t in systems with flexible pendants. (B) Distribution ( $\sigma^2$ ) plotted over times t in systems with flexible pendants. ....	71
<b>Figure 4.12.</b> (A) Averaged time-dependent ES diffusion plots on Ru out to $\pm 30$ away from the initial excited Ru at time t in systems with static pendants. (B) Distribution ( $\sigma^2$ ) plotted over times t in systems with static pendants. ....	72



## LIST OF ABBREVIATIONS AND SYMBOLS

AA	all-atom
AOM	acusto-optic modulator
atm	atmospheres
BBO	$\beta$ -Barium Borate
bpy	bipyridine
ca.	circa
°C	Celsius
cm	centimeters
CG	coarse-grain
$D_{\text{coeff}}$	diffusion coefficient
EnT	energy transfer
ES	excited state
fs	femtoseconds
fwhm	full width half max
g	grams
GS	ground state

GUI graphical user interface

HPC high performance computer

HPF high pass filter

ns<sup>-1</sup> inverse nanoseconds

iTMC ionic transition metal complex

K Kelvin

kcal kilocalories

kHz kilohertz

LJ Lennard-Jones

MS Materials Studio

MHz megahertz

mol moles

MC Monte Carlo

MD Molecular Dynamics

MG motion group

nJ nanojoules

ns nanoseconds

NPA Natural Population Analysis

NIR near-infrared

OD optical density

Os Osmium

PMDETA Pentamethyldiethylenetriamine

ps picoseconds

PF polyfluorene

FRu<sub>2</sub> polyfluorene-ruthenium repeat unit

PS polystyrene

Ru Ruthenium

<sup>1</sup>MLCT singlet metal-to-ligand charge transfer

<sup>3</sup>MLCT triplet metal-to-ligand charge transfer

TTA triplet-triplet annihilation

UV ultraviolet

Vis visible

W watts

20mer 20-complex polymer

70mer 70-complex polymer

80mer 80-complex polymer

$k_A$  angle bending force constant

Å angstrom

$\beta$  attenuation parameter

$\langle \rangle$	average
$\sigma_{ij}$	bead diameter
$k_B$	bond stretching force constant
$q_i$	charge
[ ]	concentration
D	Debye
°	degrees
$P$	eigenvector of the diagonalized rate matrix
$\theta_0$	equilibrium angle
$R_0$	equilibrium bond length
$\chi_{EnT}$	energy transfer efficiency
*	excited state
$X_n$	excited state population on complex n
→	excited state transfer to
$\tau^{Ru}$	excited state Ru to Ru transfer lifetime
$\tau^{Os}$	excited state Ru to Os transfer lifetime
$k^{Ru}$	excited state Ru to Ru transfer rate
$k^{Os}$	excited state Ru to Os transfer rate
$\delta$	gamma
$k_{mn}$	hopping rates between complexes n and n

$k_{NN}$	hopping rates located along the rate matrix diagonal
$\varepsilon_{ij}$	interaction energy
$\text{\AA}^{-1}$	inverse angstrom
$P^{-1}$	inverse eigenvector of the diagonalized rate matrix
$\tau_{em}^*$	lifetime of an initially excited donor complex
$\mu\text{s}$	microseconds
$\tau_{\max}^{Os}$	most probable excited state Ru to Os transfer lifetime
$N_{Os}$	number of emitted Os photons
N	number of Os that can be loaded onto the polymer backbone
$k_{em}^{Os}$	Os excited state decay rate
$\phi_{Os}$	Os quantum yield
%	percent
$\pm$	plus and minus
$P_{Ru}$	probability a given site is Ru
$P_{Os}$	probability a given site is Os
$A_0$	probability of creating a PF-Os homopolymer
$P_n$	probability of loading n Os complexes onto the polymer backbone
$k_0^{Ru}$	rate constant at closest approach for two Ru complexes
$k_0^{Os}$	rate constant at closest approach for a Ru and Os complex
$k_{em}^{Ru}$	Ru excited state decay rate

$\phi_{Ru}$	Ru quantum yield
$\sigma$	sigma
$\sigma^2$	sigma squared
$\varepsilon_1$	solution permittivity
$\sum$	sum
$\tau$	tau
t	time
$(t)$	time-dependent
$\theta_{ijk}$	time-dependent angle between three bound beads
$D(t)$	time-dependent diagonalized kinetic matrix
$R(t)$	time-dependent distance matrix
$K(t)$	time-dependent kinetic matrix
$X(t)$	time-dependent population vector for all complexes on polymer
$R_{Os}$	time-dependent separation between a Ru and Os complex
$R_{Ru}$	time-dependent separation between two Ru complexes
$R_{ij}$	time-dependent separation between two complexes
$\underline{X}(\delta(t))$	time-dependent sum of updated Ru* and Os* final populations
N	total number of complexes in the assembly
$N_T$	total number of emitted photons
$\lambda$	wavelength

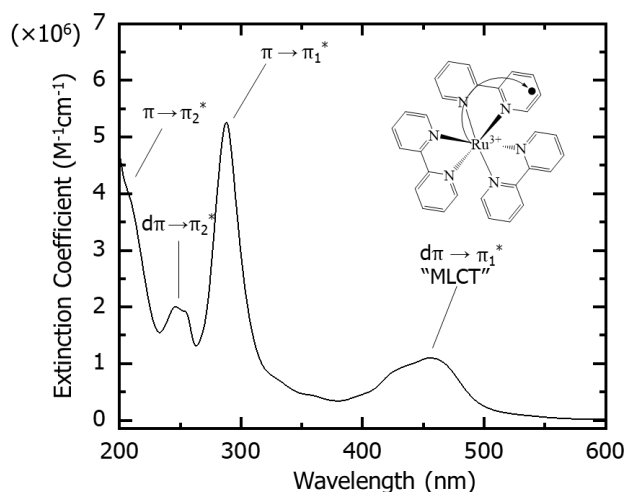
$y_0$     y-intercept

## Chapter 1: Introduction

One approach to realizing artificial light-harvesting assemblies is the development of rigid multi-component systems where orientations and distances of the subunits are precisely controlled. Numerous dendritic designs rely on the rigidity of the dendrimer scaffold to pointedly study the directionality of energy transport (EnT) from higher energy peripheral chromophores to energy-sinks at the core.<sup>1-3</sup> The shape<sup>4</sup> and spatial arrangement of chromophores in most dendritic assemblies are critical to transfer energy efficiently along a desired energy gradient, or pathway.<sup>5-7</sup> These pathways are best predicted in static structures that were synthesized with great control. Although these dendritic assemblies are well-defined, they are often difficult to synthesize. As more chromophores are added to the structure, their synthesis becomes even more demanding and investigatory challenges into their energy transport mechanisms result.

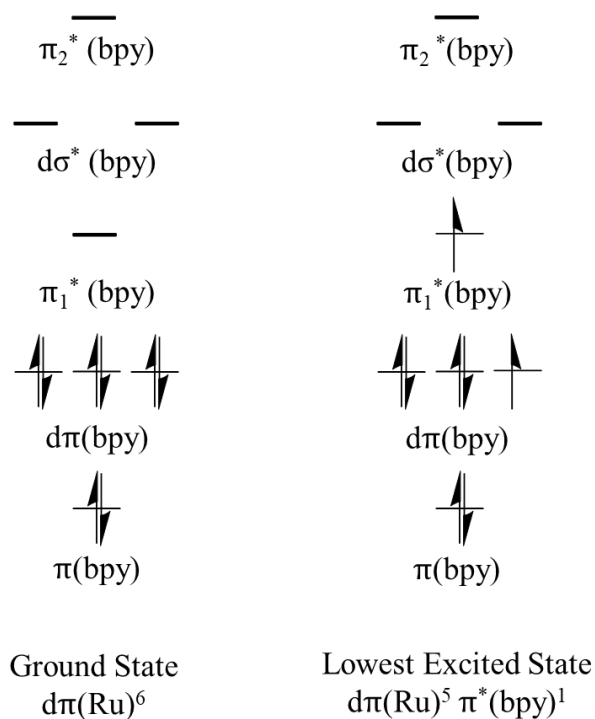
Balzani et. al. was able to precisely control energy gradients in dendrimers containing  $\text{Ru}(\text{bpy})_3]^{2+}$  (Ru) and  $\text{Os}(\text{bpy})_3]^{2+}$  (Os) chromophores via a “small-upward” synthetic approach.<sup>2</sup> The absorption spectrum of  $[\text{Ru}(\text{bpy})_3]^{2+}$  is shown below in Figure 1.1.





**Figure 1.1.** UV-visible absorption spectrum of Ru ( $[\text{Ru}(\text{bpy})_3]^{2+}$ ) in acetonitrile. The electronic transitions are labeled, along with a depiction of the metal-to-ligand charge transfer (MLCT) transition in  $[\text{Ru}(\text{bpy})_3]^{2+}$ .

Excitation of the metal-to-ligand charge transfer ( $^1\text{MLCT}$ ) transition in  $[\text{Ru}(\text{bpy})_3]^{2+}$  results in a  $d\pi \rightarrow \pi^*$  transition of an electron at the metal center to a bpy ligand (Figure 1.2).



**Figure 1.2.** Simplified molecular orbital diagram for  $[\text{Ru}(\text{bpy})_3]^{2+}$ .

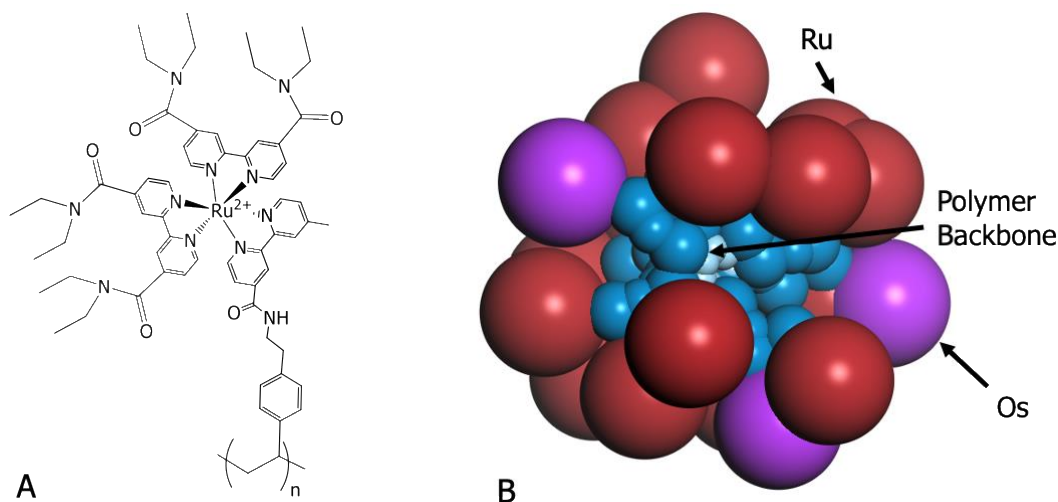
The populated  $^1\text{MLCT}$  excited states (ES) rattle down to the lowest energy  $^3\text{MLCT}$  within hundreds of fs.<sup>8-10</sup> When Ru and Os chromophores are linked together in a supra molecular array, energy can be transferred from an excited chromophore to an unexcited one with a directionality that flows from the higher to lower level  $^3\text{MLCT}$  state. Given that the  $^3\text{MLCT}$  state of  $\text{Ru}^*$  is greater than that of  $\text{Os}^*$ , Balzani and coworkers are able to direct the excited state from Ru to Os with a high degree of control.<sup>2</sup> The group is able to retain this control as they build their array out to incorporate more chromophores on the periphery. Since Ru complexes have an excited state lifetime of  $\sim 1 \mu\text{s}$ , relaxation to the ground state does not limit EnT across multiple complexes in these systems.

An alternative approach to Ru and Os-based dendritic assemblies is to create light-harvesting assemblies based on readily functionalized polymeric scaffolds where many chromophores can be loaded onto the backbone in a more facile manner.<sup>11-31</sup> The relative ease of creating polymeric assemblies comes at the expense of reduced control over structural parameters and a greater degree of structural heterogeneity both of which provide additional challenges when studying energy transport.

Polymeric assemblies present a particular challenge since the structure of the polymer backbone often affects overall system rigidity. When chromophores are affixed to the backbone via flexible pendant linkages, sterics, pendant length, structure of the backbone and its ability to twist and alleviate strain determines how proximal chromophores are to one another. This proximity largely determines whether the distance distribution between chromophore pairs is quasi-static or

dynamic. Both cases result in multiexponential energy transport kinetics that make spectroscopy-based EnT investigations difficult to assess without use of computational modeling.

Early work in our group focused on light-harvesting assemblies comprised of multiple Ru (II) and Os (II) polypyridyl complexes affixed to a polystyrene (PS) backbone (PS-RuOs) shown in Figure 1.3.<sup>15,14</sup>



**Figure 1.3.** (A) Chemical structure of Polystyrene-Ruthenium (PS-Ru). (B) Coarse-grained 20 complex representation of the polymeric assembly,  $[\text{PS-Ru}_{17}\text{Os}_3]^{40+}$  (PS-RuOs) where Ru (red) and Os (purple) complexes are bound by a pendant linkage to the polystyrene backbone.

In these investigations, the non-conjugated PS backbone acts solely as a scaffold. The backbone's ability to twist in conjunction with the short tether that holds Ru and Os chromophores to it, pulls the metal complexes into close proximity to one another (Figure 1.3). This dense packing creates a quasi-static system where it is reasonable to neglect molecular

motion. As a result of its molecular immobility, EnT pathways in PS-RuOs are largely predictable and can be simulated using Monte Carlo modeling. Supplemented with experimental results, Monte Carlo simulations give us insight into the most probable hopping times between donor and acceptor complex pairs (1.6 and 0.4 ns<sup>-1</sup>), average peripheral distances between nearest neighbors and next nearest neighbors (1 and 16 Å, respectively) and the efficiency of EnT within the system (95%)<sup>15</sup> Through-space Dexter EnT is the dominant mechanism in this system (Equation 1.1)

$$k^{ij} = k_0^{ij} \exp(-\beta R_{ij}) \quad (1.1)$$

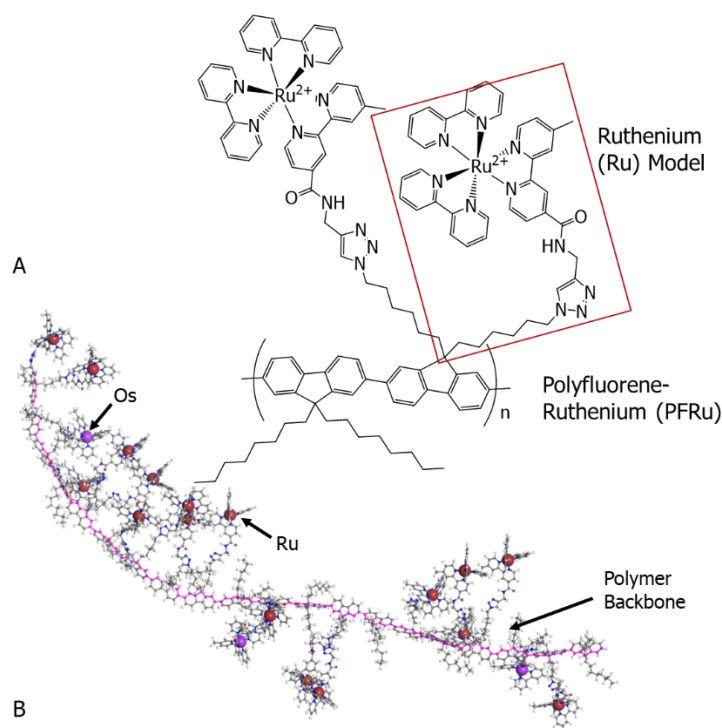
, where  $k^{ij}$  is the hopping rate between any donor and acceptor,  $k_0^{ij}$  represents the fastest possible hopping rate when the orbital pairs are perfectly overlapped,  $\beta$  signifies the donor and acceptor orbital fall off rate in space and  $R_{ij}$  represents the distance between any pair.

Given the small separation between chromophores and the long-lived excited state lifetime of Ru, it is not surprising that hopping times are fast and overall EnT efficiency is high. Compared to the *static* distance distribution observed in PS-RuOs, assemblies with more rigid polymer scaffolds and longer pendants introduce a *dynamical* conformational heterogeneity. Inherently flexible pendant chromophores affixed to rigid scaffolds allow those pendants to elongate and interact with the surrounding solvent<sup>25</sup>, resulting in a distribution of distances that changes on the ps timescale.

Since pendant motions occur on same timescale as transfer rates,<sup>32</sup> studies of flexible polymer-pendant assemblies force investigation of rate constants that are more determined by motions than by electronic couplings, reorganization energies and driving forces.<sup>25</sup> Flexibility presents an

especially difficult investigatory problem since intrinsic and extrinsic properties of the system are difficult to deconvolve from the experimental data without use of detailed simulations and dynamic modeling.

In this work, we focus on the ultrafast dynamics of polymeric assemblies consisting of multiple Ru(II) and Os(II) polypyridyl complexes affixed to a polyfluorene (PF) backbone by flexible pendant linkages (PF-RuOs) (Figure 1.4).



**Figure 1.4.** (A) Chemical structures of the Polyfluorene-Ruthenium (PF-Ru), Ru and Os Models. (B) A 20-complex representation of the polymeric assembly,  $[\text{PF-Ru}_{60}\text{Os}_{10}]^{140+}$  (PF-RuOs) tested experimentally, where Ru (red) and Os (purple) complexes are bound by a pendant linkage to a conjugated polyfluorene backbone (pink).

Similar to studies conducted on PS-RuOs, the PF backbone acts only as a scaffold. The conjugated nature of the backbone causes it to remain relatively rigid and open, allowing the long pendants to separate from each other as they elongate into the surrounding solvent. Although highly conjugated, PF monomer units absorb far to the blue of the 450 nm pulse used to initiate EnT at a Ru complex. If treated quasi-statically, nearest neighbor metal complexes in PF-RuOs would be peripherally separated by  $\sim 12$  Å on average, almost 12 times that average separations in PS-RuOs. Complex separations this large would result in near negligible hopping times and EnT kinetics (Equation 1.1); however, we estimate an 80-85% EnT efficiency for this system. Considering the potential aid of the inherent flexibility of PF-RuOs and its pendant chromophores in EnT, we develop a joint experimental and computational model that allows us to explore the role of these motions those transfer processes.

## 1.1. Overview.

This work is divided into three chapters. In *Chapter 2*, we focus on the experimental spectroscopy techniques used to characterize PF-RuOs and probe its EnT kinetics. By exciting one Ru complex on each assembly and monitoring the time-resolved Os\* emission, we begin to explore the *average total* EnT kinetics of a PF-RuOs ensemble of assemblies. The resulting highly multi-exponential kinetic traces make intrinsic properties of the system difficult to deconvolve without addition of computational simulations that accurately treat the molecular motions of PF-RuOs. *Chapter 3* details how we model those motions through coarse-grained (CG) molecular dynamic (MD) simulations. Compared to an all-atom approach, coarse-graining the PF-RuOs system reduces the number of particles involved in pairwise interaction calculations thus significantly reduces the computational cost of simulating motion. *Chapter 4* discusses the EnT simulations we conduct on the CG MD PF-RuOs model resulting from Chapter 3. Our EnT

simulations probe kinetics in many structures of the CG MD PF-RuOs model in which the number of Os loaded (*Os loading*), how the complexes are distributed across the 70 sites (*Ru/Os configurations*) vary and which Ru complex is initially excited to accurately model the EnT kinetics occurring in our experimental ensemble of PF-RuOs assemblies.

The combination of these chapters results in a description of a robust kinetic model that utilizes time-resolved emission spectroscopy with CG MD and EnT kinetic simulations to elucidate a microscopic view of how motions influence EnT kinetics in PF-RuOs.

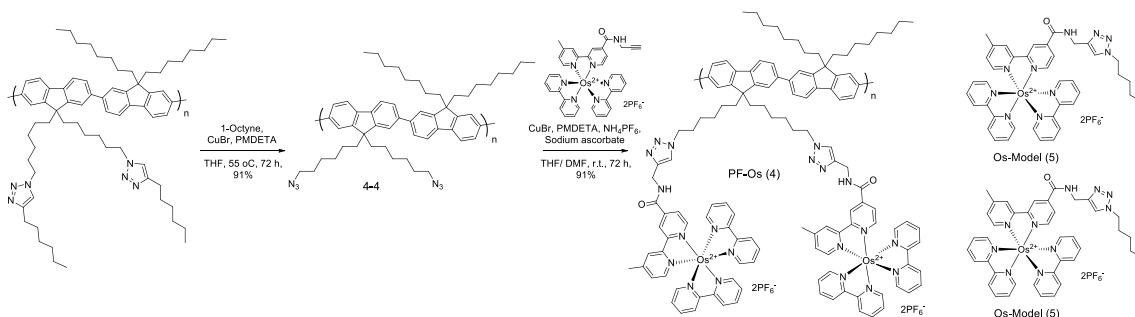
## Chapter 2: Experimental Methods

### 2.1. Synthesis and Material.

Our spectroscopic experiments focus on the polyfluorene-based system with mixed Ru and Os loading, PF-Ru<sub>60</sub>Os<sub>10</sub> (PF-RuOs). The polymeric arrays of PF-RuOs and homopolymers PF-Ru<sub>70</sub> (PF-Ru) and PF-Os<sub>70</sub> (PF-Os) (Figure 2.1) were synthesized via a Cu(I)-catalyzed “click” chemistry approach by reacting an azide-functionalized polyfluorene with an alkynyl containing Ru(II) and/or Os(II) polypyridyl complex. To obtain quantitatively functionalized hybrid assemblies, CuBr, PMDETA, and sodium ascorbate were employed as the catalyst, base/Cu-coordinating ligand, and reducing agent, respectively. Additionally, an excess of NH<sub>4</sub>PF<sub>6</sub> was added to the reaction mixtures to prevent iTMC counterion exchange and maintain solubility of the polymeric intermediates. Notably, PF-RuOs was designed to contain ca. 15 mol% of low energy Os(II) chromophores to report on excited state migration along the pendant iTMC units. The average 15% Os loading ration in PF-RuOs was obtained via a one-pot procedure using a



mixture of reactive Ru(II) and Os(II) chromophores (85%:15%) rather than their sequential addition.



**Figure 2.1.** Synthetic routes to polymers PF-RuOs, PF-Ru, PF-Os, Ru and Os Models. In this study, PF-Ru, PF-Os, Ru and Os models serve as control systems.

$^1\text{H}$  NMR measurements were used to determine the degree of metal site vacancies. This analytic technique along with UV-Vis absorption measurements confirmed the *average* 85%:15% loading ratio of Ru:Os an ensemble of PF-RuOs assemblies.<sup>30</sup>  $^1\text{H}$  NMR and UV-Vis absorption measurements were also used to characterize the homopolymers. More or less Os complexes can be loaded onto any assembly, but details of this statistical loading will be discussed later in the text.

## 2.2. Spectroscopic Experiments

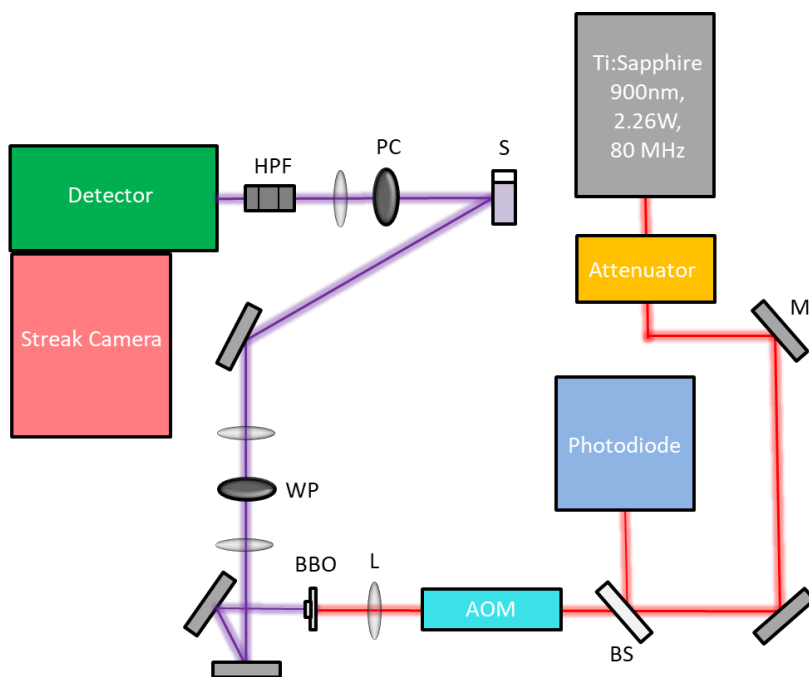
### 2.2.1. Steady-State Absorption and Emission Methods.

Steady-state absorption spectra were recorded using a UV-Vis-NIR absorption spectrophotometer. Steady-state emission spectra were collected using a photon-counting Edinburgh FLS920 spectrofluorimeter, where the Xe lamp light source was operating at 450 W. Optically dilute samples (less than 0.12 OD at the excitation wavelength) were purged with

Argon for 35 min prior to use. To confirm the absence of photoproducts, we took solution absorbances before and after all emission measurements.

### 2.2.2. Time-Resolved Emission Methods.

Time-resolved emission measurements were taken using a mode-locked Ti:Sapphire laser (Coherent-Chameleon) paired with a Hamamatsu streak camera.



**Figure 2.2.** Schematic of the time-resolved emission set-up. Important optical components are labeled as follows M = mirror, BS = beamsplitter, AOM = acusto-optic modulator, L = lens, BBO =  $\beta$ -Barium Borate crystal, WP = waveplate, PC = polarizing cube, S = sample, HPF = high pass filters.

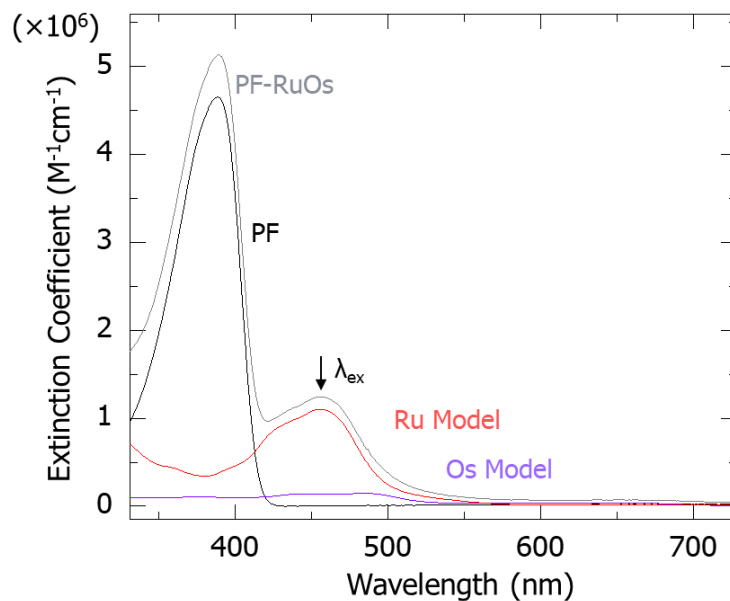
As shown in Figure 2.2, we attenuate the 80 MHz laser fundamental to 28 nJ/pulse and use a Conoptics acusto-optic modulator (AOM) to pulse-pick the beam down to 127 kHz ( $\tau = 7.87 \mu\text{s}$ ) - a frequency which is 3 times the natural lifetime of the emission detected at 670 nm and 5 times that of the emission detected at 800 nm. The beam goes through a  $\beta$ -barium borate (BBO)

crystal that frequency doubles the 900 nm output to 450 nm. Two lenses collimate the beam before it reaches the 1 cm quartz cuvette containing our sample of PF-RuOs. The light emitted at 45° from the incident beam travels through a polarizing cube set to magic angle (54.7°) before being focused and passed through a series of neutral density and high pass filters that decrease signal from ambient light and the 450 nm pump.

The emitted photons focus into a single grating monochromator centered on wavelengths of either 670 or 800 nm ( $\pm 50$  nm) to observe Ru or Os emission, respectively. The monochromator connects to the streak camera, which is coupled to a multichannel plate-photomultiplier tube (MCP). For each experiment, we count 50,000 photons centered around the set observation wavelength over time windows of 5, 20, 200 ns, 2 and 10  $\mu$ s to get time-dependent spectra of photon intensity. To get Ru emission transients, we integrate photons emitted across the entire wavelength window (620 to 720 nm) over time. For Os emission transients, we only integrate Os photons emitted between 790 to 850 nm to minimize contributions from Ru emissions. We repeat the experiment ten times and average the emission transients for each time window to increase the statistical accuracy of our results. We stitch an average of the last few points in the 5 ns window to those around the 5 ns mark in the 20 ns window and subsequently stitch in transients from the 200 ns, 2 and 10  $\mu$ s time windows in order to observe kinetics at early and late times with optimal temporal resolution. To measure the instrument response for each time window, we reflect our pump off a scatter cell and monitor the scattered light at 450 nm as it hits our detector. The temporal resolution is 72 ps at full width half max (fwhm) for a 5 ns time window.

## 2.3. Experimental Results and Discussion.

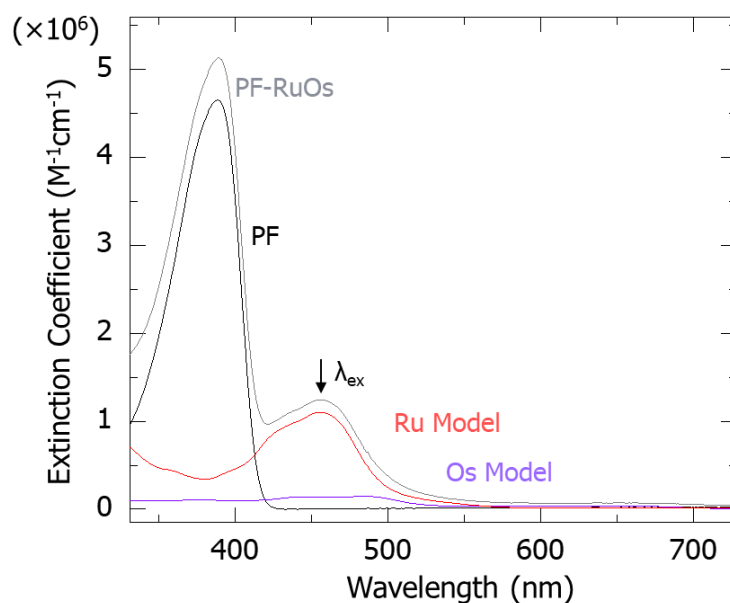
### 2.3.1. Structural Characterization of PF-RuOs.



**Figure 2.3.** Absorption spectra of the PF-RuOs assembly; PF, Ru, and Os Model systems in room temperature acetonitrile. Spectra of Ru and Os Model systems are weighted by 0.85 and 0.15, respectively, to represent the relative concentration of each complex within the assembly. The arrow identifies the excitation wavelength used for time-resolved emission experiments.

The absorption spectrum of PF-RuOs is shown in Figure 2.3 as a superposition of composite PF, Ru and Os model spectra. The summation of the weighted component parts to comprise the PF-RuOs spectra indicates that the metal complexes are weakly coupled and retain their electronic properties after being loaded onto the polymer backbone; a characteristic that is not always present in light-harvesting arrays.<sup>13</sup> Retention of these properties is important to tracking energy transport by observation of separate Ru\* and Os\* emissions (discussed later). Although previous studies follow EnT from the initially excited conjugated PF backbone to a pendant

ligand,<sup>23</sup> this study focuses on excitation at 450 nm to incite metal-to-ligand charge transfer (<sup>1</sup>MLCT) state in Ru or Os. The probability of exciting Ru or Os at 450 nm is based only on the relative *average* mole fraction (85%:15%, Ru:Os) of the complexes in PF-RuOs, since the complexes have the same extinction coefficient at the excitation wavelength. Since the PF absorbs at 390 nm (Figure 2.3) the backbone acts solely as a scaffold.<sup>15,30</sup>



**Figure 2.4.** Jablonski diagram of an EnT pathway in PF-RuOs.

Figure 2.4 illustrates one example of an EnT pathway that can occur in PF-RuOs. This pathway follows the solid black arrows where the Ru complex at site 1 is photoexcited ( $\text{Ru}_1^*$ ) at 450 nm to create a <sup>1</sup>MLCT state that rapidly cools ( $\tau \sim 400$  fs) to a long-lived <sup>3</sup>MLCT state ( $\tau \sim 1$   $\mu$ s).<sup>33</sup> The excited state (ES) hops from isoenergetic  $\text{Ru}_1^*$  to  $\text{Ru}_2^*$  ( $\text{Ru}_1^* \rightarrow \text{Ru}_2^*$ ), with rates of  $k^{\text{Ru}}$ , by a dominant through-space Dexter-style EnT mechanism<sup>25</sup> and continues on to either sensitize a lower-energy Os trap ( $\text{Ru}_4^* \rightarrow \text{Os}_5$ ), with a rate of  $k^{\text{Os}}$ , or decay back to the ground state with a rate of  $k_{em}^{\text{Ru}}$  (dotted red arrows) before reaching the trap. Since an  $\text{Os}^* \rightarrow \text{Ru}$  transition is

energetically forbidden, the excited state decays after becoming trapped with a rate of  $k_{em}^{Os}$  (solid purple arrow). The total EnT pathway is complete once the excited state relaxes to the ground state and is the sum of all hopping and decay rates in the pathway.

Figure 2.4 illustrates just one EnT pathway that can occur in this system. Given that the Os loading, Ru/Os configuration and initial excitation varies for each assembly in an ensemble, the excited state can travel by many other EnT pathways. Generally, time-dependent Dexter hopping rates between  $Ru^* \rightarrow Ru$  ( $k^{Ru}$ ) or  $Ru^* \rightarrow Os$  ( $k^{Os}$ ) are described here

$$k^{Ru} = k_0^{Ru} \exp(-\beta R_{Ru}) \quad (2.1A)$$

$$k^{Os} = k_0^{Os} \exp(-\beta R_{Os}) \quad (2.1B)$$

, where  $k_0^{Ru}$  or  $k_0^{Os}$  is the rate constant at optimal orbital overlap between any two Ru complexes or a Ru complex and Os trap ( $ns^{-1}$ );  $\beta$  is an attenuation parameter ( $\text{\AA}^{-1}$ ) that is approximated to be independent of complex identity,<sup>15</sup> and time-dependent pendant separations between any two Ru complexes or a Ru complex and Os trap are represented by  $R_{Ru}$  or  $R_{Os}$ , respectively ( $\text{\AA}$ ).

Equations 2.1 indicates hopping rates,  $k^{Ru}$  and  $k^{Os}$ , are exponentially dependent on  $\beta$  and the time-dependent distance between two complexes,  $R_{ij}$ . To achieve fast hopping kinetics and high EnT efficiency, a donor and acceptor complex must be in close proximity to one another, especially if  $\beta$  is large. Equation 2.2 illustrates the relationship between hopping rates and efficiency,

$$\chi_{EnT} = \frac{\sum \tau^{Ru} + \sum \tau^{Os}}{\tau_{em}^*} \quad (2.2A)$$

, where  $\chi_{EnT}$  is the transfer efficiency and  $\tau_{em}^*$  is the lifetime of an initially excited donor complex,

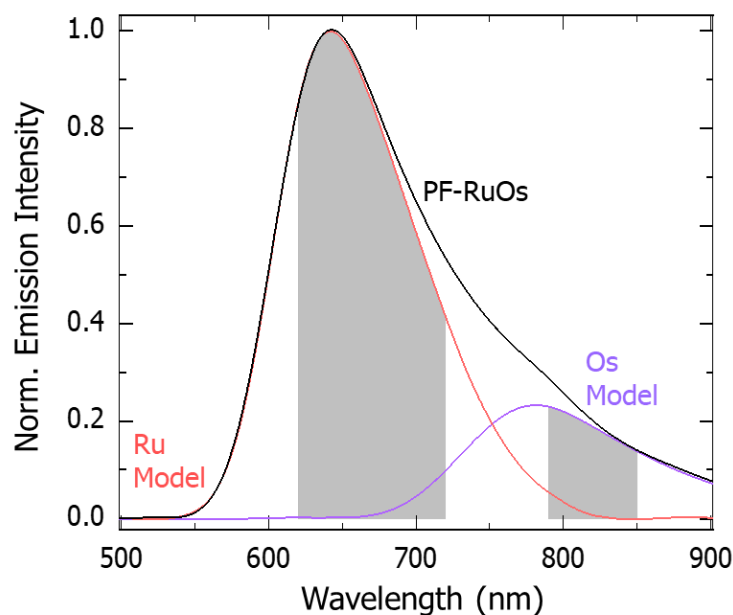
$$\sum \tau^{Ru} = \sum \left( \frac{1}{k^{Ru}} \right) \quad (2.2B)$$

$$\sum \tau^{Os} = \sum \left( \frac{1}{k^{Os}} \right) \quad (2.2C)$$

, and  $\sum \tau^{Ru}$  and  $\sum \tau^{Os}$  represent the sum of hopping lifetimes between any donor and acceptor within a total EnT pathway.

Given that the lifetime of most Ru complexes is  $\sim 1 \mu s$ , the excited state may hop from site to site numerous times within its lifetime if hopping rates are fast compared to the rate of decay. The closely packed quasi-static complexes in PS-RuOs (average  $\sim 2-3 \text{ \AA}$  separation), give rise to hopping rates that are hundreds of ps and high EnT efficiencies. Conversely, nearest neighbor complexes in PF-RuOs that move about the solvent freely (static average  $\sim 12 \text{ \AA}$  separation) would lead to a slow hopping kinetics and small EnT efficiency overall. However, the system's inherent flexibility may bring those complexes into close enough proximity to one another, to rapidly transfer energy.

We can better approximate EnT efficiency by probing steady-state  $Ru^*$  and  $Os^*$  emission in PF-RuOs after largely exciting Ru complexes.



**Figure 2.5.** Steady-state emission spectra of PF-RuOs (black), Ru Model (PF-Ru, red), and Os Model (PF-Os, purple) after continuous excitation at 450 nm at 450 W. Shaded regions under PF-Ru and PF-Os curves indicate the spectral ranges at which time-resolved Ru\* and Os\* emissions in PF-RuOs are integrated.

Steady-state excitation of PF-RuOs, or mixed polymer, at 450 nm gives rise to continuous Ru\* and Os\* emission centered around 670 nm and 800 nm, respectively (Figure 2.5). Shaded regions under the spectra for the Ru Model (PF-Ru) and Os Model (PF-Os), or Ru and Os homopolymers, indicate the spectral ranges in which Ru and Os time-resolved emissions in PF-RuOs are integrated. Emission observed around 670 nm (620 – 720 nm) is mostly Ru\* emission, and emission observed around 800 nm (790 – 850 nm) is mostly Os\* emission. The steady-state emission spectra of PF-RuOs is overlaid here with superimposed spectra of Ru and Os Models scaled to reflect the relative amounts of Ru and Os in PF-RuOs. Given that the total area under each curve reflects the number of Ru and Os *excited states* in the mixed assembly, we can



estimate EnT efficiency, or how much of excited state makes it to an Os trap after selectively exciting Ru (Equation 2.3)

$$\chi_{EnT} = \frac{N_{Os}}{N_T} \quad (2.3)$$

, where  $\chi_{EnT}$ ,  $N_{Os}$ , and  $N_T$  represent energy transfer efficiency, the number of emitted Os photons that result from Ru\*  $\rightarrow$  Os sensitization, and the total number of emitted photons, respectively.

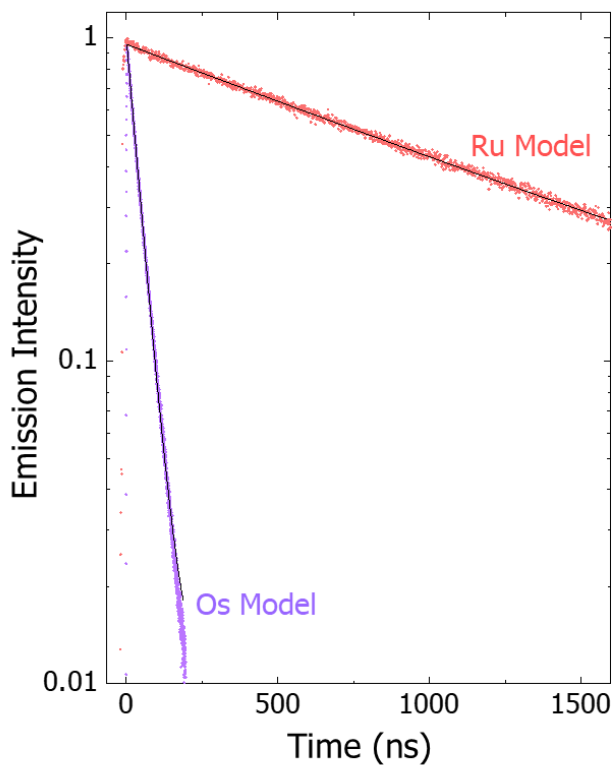
From the steady-state spectra of PF-RuOs, PF-Ru and PF-Os, it appears that Ru emission is five times more intense than that of Os, which would suggest a largely inefficient EnT process.

However, when we scale Os\* emission intensity by 30 to account for differences between Ru and Os quantum yields ( $\phi_{Ru} = 7\%$  and  $\phi_{Os} = 0.32\%$ ),<sup>15</sup> the area under the curves now represents the number of *photons*, an intrinsic property of PF-RuOs. Based on relative integrated intensities we estimate  $\chi_{EnT}$  to be  $\sim 80\text{-}85\%$ .

Since the hopping rate between donor and acceptor complexes is exponentially dependent on distance and orbital overlap between donor and acceptor complexes (Equation 2.1), it may be surprising that EnT in PF-RuOs yields such a high efficiency. To gain a better understanding of this system and of how flexibility plays a role in this its EnT processes, we employ time-resolved emission spectroscopy to determine excited state transfer kinetics in the homopolymer and mixed polymer assemblies. Comparing the dynamical differences between the two systems gives us good spectroscopic evidence that energy transfer occurs in PF-RuOs.

### 2.3.2. Evidence of Energy Transfer.

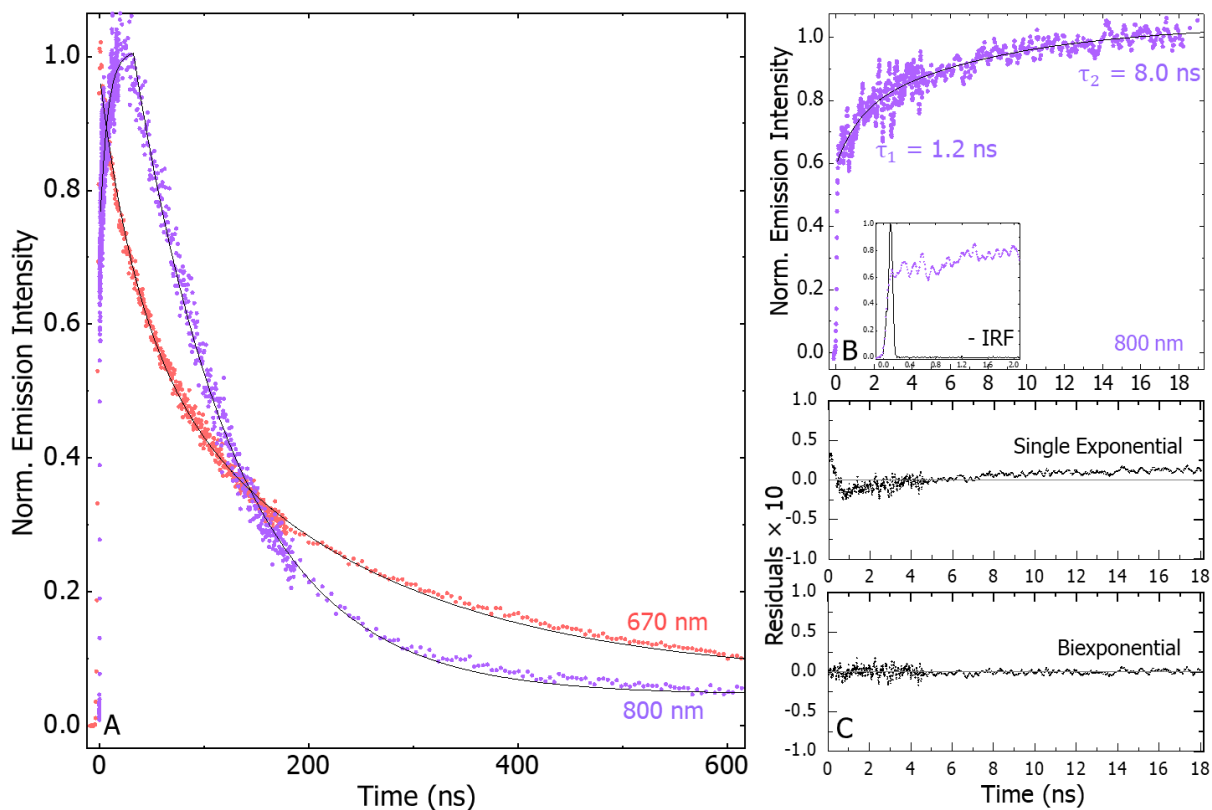
Using the setup illustrated in Figure 2.2, we photoexcite a sample of solvated Ru Models with a single pulse at 450 nm, and observe the single-exponential decay of  $\text{Ru}^*$  with a lifetime of 1112 ns ( $\tau_{em}^{\text{Ru}}$ ). We repeat the experiment for a sample of solvated Os Models and observe the single-exponential decay of  $\text{Os}^*$  with a lifetime of 40 ns ( $\tau_{em}^{\text{Os}}$ ) (Figure 2.6).



**Figure 2.6.** Time-resolved  $\text{Ru}^*$  and  $\text{Os}^*$  emission transients observed in the Ru Model (red) and PF-RuOs (purple) around 670 and 800 nm, respectively. All peak intensities are normalized to 1.

The linear relationship between emission intensity and time reveals the uniformity of  $\text{Ru}^* \rightarrow \text{Ru}$  and  $\text{Os}^* \rightarrow \text{Os}$  hopping kinetics in PF-Ru and PF-Os, respectively. Since these homopolymers lack reporting molecules, we cannot get EnT kinetics from their emissions.

Unlike the linear homopolymer traces, we do not see the same kinetic uniformity for  $\text{Ru}^* \rightarrow \text{Ru}$  and  $\text{Ru}^* \rightarrow \text{Os}$  hopping kinetics in PF-RuOs (Figure 2.7).



**Figure 2.7.** (A) Time-resolved  $\text{Ru}^*$  (red) and  $\text{Os}^*$  (purple) emission transients observed in PF-RuOs around 670 and 800 nm, respectively, with peak intensities normalized to 1. (B)  $\text{Os}^*$  emission plotted over the first 20 ns and overlaid with the instrument response function (inset). (C) Residuals from single and biexponential fits of delayed rise in  $\text{Os}^*$  emission.

Focusing solely on PF-RuOs, Figure 2.7A highlights the multi-exponential kinetic behavior in the mixed polymer system. Unlike the homopolymer systems, PF-RuOs has both isoenergetic Ru complexes and Os traps randomly affixed to sites along the polymer chain. After a single pulse, 450 nm excitation excites a Ru complex in each assembly in the ensemble. The excited state can

proceed via any pathway outlined in Figure 2.4 to either yield Ru\* (decay) or Os\* emission (sensitization and decay). When the streak camera monochromator is centered around 670 nm, we observe Ru as the sole emitter (red trace). Emission detected around 800 nm (purple trace) reflects the appearance of Os excited states mostly by Ru\* sensitization. The decay kinetics we observe in Figure 2.7A at 670 nm is the result Ru\* emission after several Ru\* → Ru hops which we fit to a biexponential to guide the eye. We observe Os sensitization and decay kinetics at 800 nm with the delayed rise and decay emission kinetics, which are both fit to a single exponential. The delayed rise coincides with the decay of Ru\* emissions in PF-RuOs and is not present in PF-Os emission kinetics, which confirms Os sensitization by Ru\*.

The sensitization rate is determined by fitting the slow Os\* growth that occurs after the ultrafast rise in emission (Figure 2.7B). The instrument response limited ultrafast component accounts for 54% of the maximum intensity and arises from instantaneous Os emission (Figure 2.7B inset). Although an attempt was made to fit the growth to a single exponential, a biexponential function best fits the delayed Os growth (Figure 2.7C). This indicates that EnT largely occurs by two processes: immediate EnT from Ru to an Os one or two hops away ( $\tau_1^{Os} = 1.2$  ns,  $k_1^{Os} = 0.83$  ns<sup>-1</sup>) and EnT where the excited state must hop between multiple Ru before reaching an Os trap ( $\tau_2^{Os} = 8.0$  ns,  $k_2^{Os} = 0.125$  ns<sup>-1</sup>). Os traps are capable of being sensitized by ‘far away’ Ru\* complexes after many Ru\* → Ru hops as illustrated by the slow Os\* decay in PF-RuOs ( $\tau_3^{Os} = 120$  ns) compared to that in PF-Os ( $\tau_{em}^{Os} = 40$  ns).

When we compare the PF-RuOs multiple hopping time ( $\tau_2 = 8.0$  ns) to the most probable Ru\* → Os hopping time in PS-RuOs ( $\tau_{max}^{Os} = 0.4$  ns),<sup>15</sup> the hopping rate is almost an order of magnitude faster in PS-RuOs. The difference here arises from the slightly larger nearest neighbor

separations we observe in PF-RuOs compared to those in PS-RuOs. Cavan et. al. performed a radial distribution of the nearest neighbor peak and estimated that each complex has 4-5 nearest neighbors with more than half being separated at the peripheries by less than 2 Å. We conduct our own distance distribution experiments and find average separations between nearest neighbors in PF-RuOs are 2.34 Å.

The traces in Figure 2.7 represent the *average* kinetics for a variety of *total* EnT pathways that can occur within the PF-RuOs ensemble. Any given EnT pathway depends on the assembly's Os loading, Ru/Os configuration, initially excited complex and how close that complex is to an Os trap. Fluctuations occurring on timescales similar to  $\text{Ru}^* \rightarrow \text{Ru}$  and  $\text{Ru}^* \rightarrow \text{Os}$  hopping times also affect  $R_{ij}$  in a way that can instantaneously change an EnT pathway to contain more or less hops. There are various extrinsic system properties affecting the pathway of excited state transport. Intrinsic properties such as hopping rates and pendant motions are difficult to deconvolve from the average total EnT kinetics we observe through time-resolved emission spectroscopy of the PF-RuOs ensemble. In order to observe the system's conformational motions and understand how they affect EnT in in PF-RuOs, we turn toward computational modeling.

## Chapter 3: Molecular Dynamic (MD) Simulations

### 3.1. Introduction.

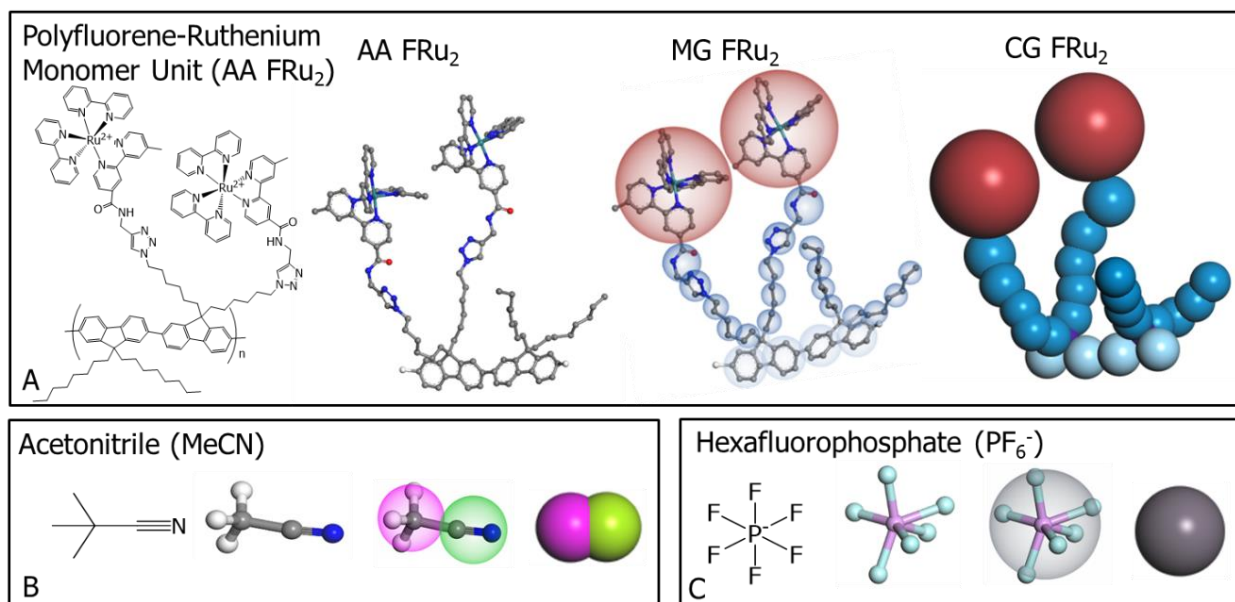
The inherent flexibility of PF-RuOs drastically complicates EnT kinetics in a way that was not present in studies of PS-RuOs. The ability of the polystyrene backbone to fold over on itself forces Ru and Os pendants to pack closely such that the larger system is quasi-static. Motions in PS-RuOs are irrelevant and Monte Carlo modeling can be used to track EnT kinetics in the assembly.<sup>15</sup>

The fluctuations present in PF-RuOs pendant complexes force us to augment our experimental results with a time-dependent model that uses Newton's Equations to describe the motion allowed by the system's sparse chromophore packing. To obtain a microscopic view of how motions influence EnT dynamics we must use molecular dynamic (MD) modeling to accurately treat PF-RuOs as a flexible system.

The majority of the cost incurred by MD simulations comes from calculating pairwise interactions between particles.<sup>19, 34</sup> Simulating motion in a 20-complex version of all-atom (AA) PF-Ru (AA PF-Ru<sub>20</sub>) in a cell with its surrounding solvent and counterions requires that we calculate pairwise interactions between 15660 total atoms. Running AA MD simulations over the full extent of energy transport events in this system (700 ns) would take ~13 years on two nodes running in parallel on a high-performance computer (HPC). This steep computational cost forces us to reconsider our approach. Grouping atoms into beads, or coarse-graining (CG) the system,

greatly reduces the number of pairwise interaction calculations and, as a result, computational time required to simulate motion.<sup>35-41</sup>

Figure 3.1 highlights the process for coarse-graining each of the system components into models that serve to accurately simulate the structure's inherent motions at a low computational cost.



**Figure 3.1.** (A) Coarse-graining procedure for the polyfluorene-ruthenium repeat unit (FRU<sub>2</sub>). (B) Coarse-graining procedure for the acetonitrile (MeCN) solvent. (C) Coarse-graining procedure for the hexafluorophosphate (PF<sub>6</sub><sup>-</sup>) counterion. From left to right, each panel depicts each component as a chemical structure, all-atom (AA), motion group (MG) and coarse-grained (CG) model.



We create all-atom models of the components from their chemical structures (Figure 3.1). Groups of atoms are selected and designated as motion groups (MGs), which are displayed in Figure 3.1 as transparent spheres. The coarse-graining procedure converts each MG to a bead containing that same grouping of atoms. We group the solvent atoms into two MGs to treat MeCN as a dielectric solvent (Figure 3.1B). All atoms in the  $\text{PF}_6^-$  counterion are grouped into one MG (Figure 3.1C). All motion groups are then converted into beads, or coarse-grained.

Coarse-graining the Polyfluorene-Ruthenium repeat unit ( $\text{FRu}_2$ ), acetonitrile (MeCN) solvent and hexafluorophosphate ( $\text{PF}_6^-$ ) counterions results in a CG PF-Ru<sub>20</sub> system with 4506 beads total – 71% fewer particles than the all-atom system. Running CG MD simulations over 700 ns takes ~12 days on two nodes running in parallel on an HPC – a near 400x speedup compared to the AA MD calculation.




## 3.2. System Preparation.

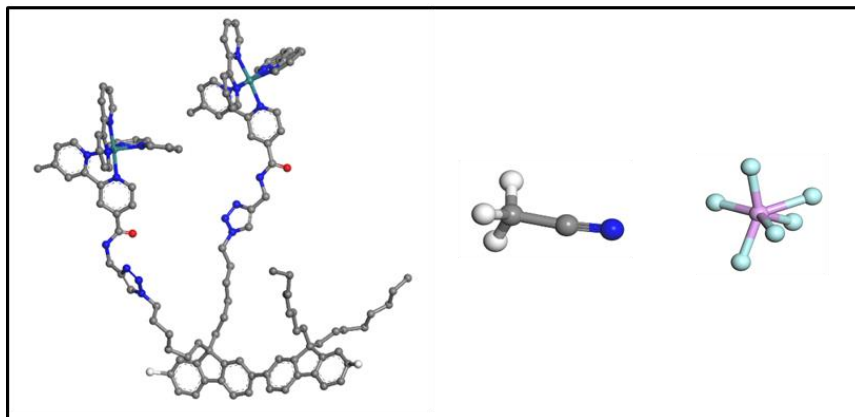
Here, we go into great detail about how each component of the all-atom (AA) model system is constructed, partitioned into motion groups (MGs), coarse-grained (CG) and subsequently parameterized in preparation to run MD simulations on the CG PF-Ru<sub>80</sub> assembly. Materials Studio is the primary modeling program we use to create AA and CG structures, run molecular dynamics simulations and conduct analyses on system structures and trajectories

### 3.2.1. All-Atom (AA) System.

We navigate first to the toolbox area and open a new 3D Atomistic Document () in the visualization window. Materials Studio's suite of drawing tools () allows us to create an AA model of the repeat unit, solvent and counter ion (Figure 3.5). We can change the



drawn carbon atoms to other atoms (  ), add Hs where necessary (  ), and clean up the structure (  ) using Materials Studio tools.



**Figure 3.2.** All-atom (AA) depiction of Polyfluorene-Ruthenium monomer unit (AA FRu<sub>2</sub>), acetonitrile (AA MeCN), and hexafluorophosphate (AA PF<sub>6</sub><sup>-</sup>). Grey atoms represent C atoms, Ru atoms are shown in teal, blue atoms are N, O atoms are represented by red, pink represents P atoms, F atoms are represented in light blue, and white indicates H atoms.

Currently, all molecules in Figure 3.2 are comprised of atoms with no partial charges assigned. Each FRu<sub>2</sub> monomeric repeat unit has a charge of +4 overall (+2 for each Ru), PF<sub>6</sub><sup>-</sup> has a charge of -1 and MeCN has a net neutral charge with electronegativity directed toward the nitrile group. We use a Gaussian calculation to assign partial charges in all three molecules by first navigating to ‘Modules’ then ‘Gaussian’ and ‘Calculation’ in the dropdown menu at the top of the display (Figure 3.3). The Gaussian calculation to determine charges on the FRu<sub>2</sub> repeat unit contains a mixed basis set with B3LYP/6-311g for the light atoms and /LANL2DZ for the heavy Ru transition metal. We can input some of the basics in the MS calculation window, copy and paste the calculation into a text editor to make specific changes. For detail, the .gnif file is shown here with appropriate line spacing:

```
%chk=FRu2.chk
%Mem=20GB
%nprocshared=12
#p RHF/B3LYP/GEN
SCF=(MaxCycle=65)
Density Pop(Regular,NPA)
NoSymm
```

FRu2 Energy

4 1

(Structure coordinates)

H C N O 0

6-311g

\*\*\*\*

Ru 0

LANL2DZ

\*\*\*\*

Ru 0

LANL2

(At least four spaces)

The Gaussian calculation to determine charges on the MeCN repeat unit contains a B3LYP/6-311g basis set for all atoms. For detail, the .gnif file is shown here with all appropriate spacing:

```
%chk=MeCN.chk
%Mem=20GB
%nprocshared=12
```

```
#p RHF/B3LYP/6-311G
SCF=(MaxCycle=65)
Density Pop(Regular,NPA)
NoSymm
```

MeCN Energy

0 1

(Structure coordinates)

(At least four spaces)

The Gaussian calculation to determine charges on the MeCN repeat unit contains a B3LYP/6-311g basis set for all atoms. For detail, the .gnif file is shown here with all appropriate spacing:

```
%chk=PF6.chk
%Mem=20GB
%nprocshared=12
#p RHF/B3LYP/6-311G
SCF=(MaxCycle=65)
Density Pop(Regular,NPA)
NoSymm
```

PF6 Energy

-1 1

(Structure coordinates)

(At least four spaces)

Running a Gaussian job on Longleaf requires the gaussian/16a03 module added, the FRu2.log and FRu2.gnif files in the same folder. Once we change the directory to the path in which those files exist, the following code is required to run the calculation:

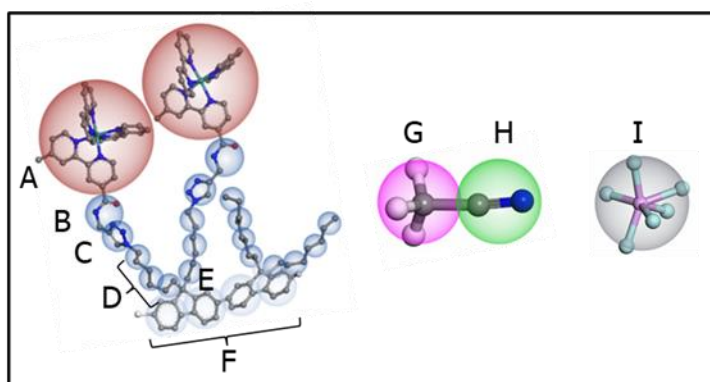
```
srun -p general -N 1 n 1 -mem20g -t 05-00:00:00 -o FRu2.log g16 FRu2.gnif
```

The .log file shows up in home window after the run is complete and at that point we can download it to workspace. We change the .log file to a .gof file and copy it into the Materials Studio project folder. The next step is to open the .xsd file used as input for the Gaussian energy calculation and go to 'Modules' then 'Gaussian' and 'Analysis' in the dropdown menu at the top of the display. Under 'population analysis', the results file should give the option of using the .gof file. Click on 'assign NPA' to assign charges to the atoms in the .xsd file. The final step is to check that the total charge is accurate. To do this, we select the entire structure and go to 'Modify' then 'charges' and click 'edit' to make sure total charge is as expected (i.e. +4 for FRu<sub>2</sub>). Running the Gaussian job on Longleaf for the MeCN and PF<sub>6</sub><sup>-</sup> molecules follow the same steps as outlined above. The partial charges we assign to the atoms are most important for AA MD simulations.

### 3.2.2. Motion Group (MG)-Defined System.

To lower the computational cost of MD simulations, we group together atoms that are considered relatively static (i.e. Ru complexes, triazole groups, C<sub>2</sub>H<sub>3</sub>NO and C<sub>2</sub>H<sub>4</sub> segments, fluorene backbone units, and etc.). Modeling these groups as motion groups (MGs) is the first step toward decreasing the number of particles that participate in pairwise interactions. MGs are transparent spheres that offer a preview of the unique ‘beads’ that result from a coarse-graining process.

We create a MG-defined AA FRu<sub>2</sub> model by selecting the atoms we want to group together, choosing ‘Modify’ then ‘Motion Groups’ from the top menu and clicking on ‘create’ from selection. This is repeated for all atom groupings so that the repeat unit, solvent, and counter ion structure have their component atoms grouped into MGs of type A – I (Figure 3.3).



**Figure 3.3.** Motion Group (MG) depiction of polyfluorene-ruthenium monomer unit (MG FRu<sub>2</sub>), acetonitrile (MG MeCN), and hexafluorophosphate (MG PF<sub>6</sub><sup>-</sup>). MGs are identified by their type A – I. Red transparent spheres illustrate atoms belonging to Ru complexes, darker blue spheres represent atoms belonging to the pendant ligands, and spheres in light blue illustrate polymer backbone atoms. Pink and green spheres represent the methyl and nitrile groups, respectively, and PF<sub>6</sub><sup>-</sup> is represented by a grey sphere. No atom belongs to two different MGs.

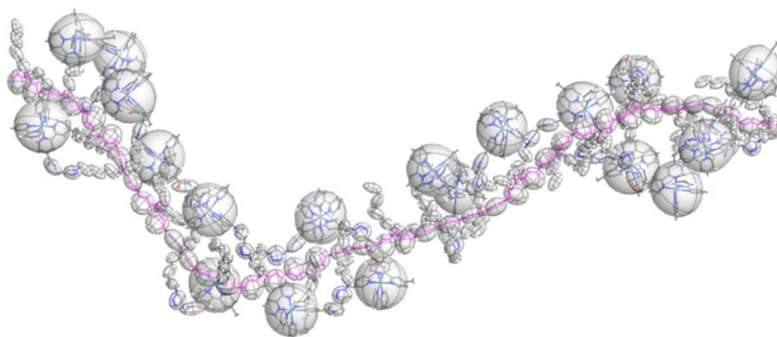
To treat MeCN as a dielectric solvent, we use two motion groups to define the solvent molecule. Atoms in the methyl segment (Me) will be converted to a bead carrying a partial positive charge while those in the nitrile segment (CN) will become a bead that carries an equal but opposite partial negative charge. For the counterion  $\text{PF}_6^-$ , all atoms are selected into one MG, which will become one bead with a -1 charge after the coarse-graining process. The  $\text{PF}_6^-$  counterions are represented as single beads because its electronegativity has no net directionality and serves only to maintain electric neutrality with the solvated PF-Ru<sub>20</sub> and its net charge of +40. Two counterions are required to balance the +2 charge at the center of each Ru complex in every FRu<sub>2</sub> repeat unit.

The MG FRu<sub>2</sub> model depicted in Figure 3.6 is the result of multiple attempts to create a final CG PF-Ru<sub>20</sub> model that most accurately simulates motions in a 40 ns trajectory of our AA PF-Ru<sub>20</sub> model at a low computational cost. The selected groupings are small enough that they do not restrict the inherent flexibility of FRu<sub>2</sub> nor are there too many groupings that the CG model resembles the AA structure. Table 3.1 describes the atoms in each *type* of MG. These types remain the same for eventual model beads.

**Table 3.1.** Bead type description of atoms.

<b>Bead Type</b>	<b>Atoms</b>
<b>A</b>	C <sub>31</sub> H <sub>25</sub> N <sub>6</sub> Ru
<b>B</b>	C <sub>2</sub> H <sub>3</sub> NO
<b>C</b>	C <sub>2</sub> HN <sub>3</sub>
<b>D</b>	C <sub>2</sub> H <sub>4</sub> or C <sub>2</sub> H <sub>5</sub>
<b>E</b>	C
<b>F</b>	Ph or PhC
<b>G</b>	CH <sub>3</sub>
<b>H</b>	CN
<b>I</b>	PF <sub>6</sub>


With all MGs defined for the system, we polymerize the  $\text{FRu}_2$  segment. We first define the repeat unit by going to ‘Build’ then ‘Build Polymer’ in the top menu then click on ‘Repeat Unit’ to select head and tail carbons. After defining the repeat unit, we again use the ‘Build’ then ‘Build Polymer’ menus in the dropdown to create a ten-unit isotactic ‘Homopolymer’ with 20 Ru complexes (MG 20mer), as shown in Figure 3.4.



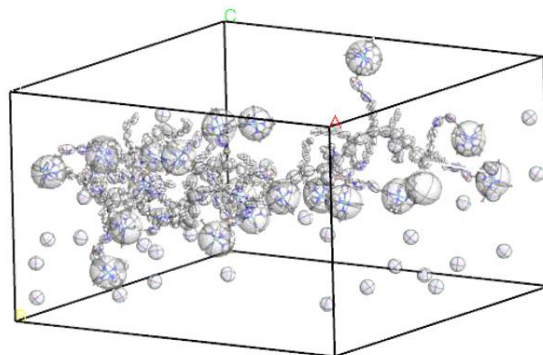
**Figure 3.4.** MG-depiction of  $\text{PF-Ru}_{20}$  (MG 20mer) after polymerizing  $\text{FRu}_2$  out ten units.

The polymerization setup includes random ‘Torsion’ to give the structure freedom to relax. We relax the structure further by running optimization calculations on the MG 20mer model. Optimization and annealing calculations are run in the Forcite module of Materials Studio. The optimization calculation is setup with medium quality, ‘smart’ algorithm, with energy and force set to 0.001 kcal/mol and 0.5 kcal/mol/Å, respectively, and ‘current charges’ set on the atoms. As soon as the structure is optimized, we anneal the MG 20mer model with medium quality, apply five annealing cycles with a 300 K to 700 K ramp. There are five heating ramps per cycle each with 200 dynamic steps. The calculation optimizes the structure after each cycle. For each calculation, we apply a Universal energy forcefield with Ewald electrostatic and atom-based van der Waals interactions to the system with the calculation run on all eight cores of a desktop computer. The MG MeCN solvent and  $\text{PF}_6^-$  counterion are optimized and annealed by the same process.

Relaxing all components of our MG-defined AA system is necessary step in setting up a cell for Molecular Dynamics calculations. Without the proper relaxation steps, it is challenging to pack the MG polymer in the simulation cell with solvent and counter molecules and equilibrate the system without exceeding energy limitations.

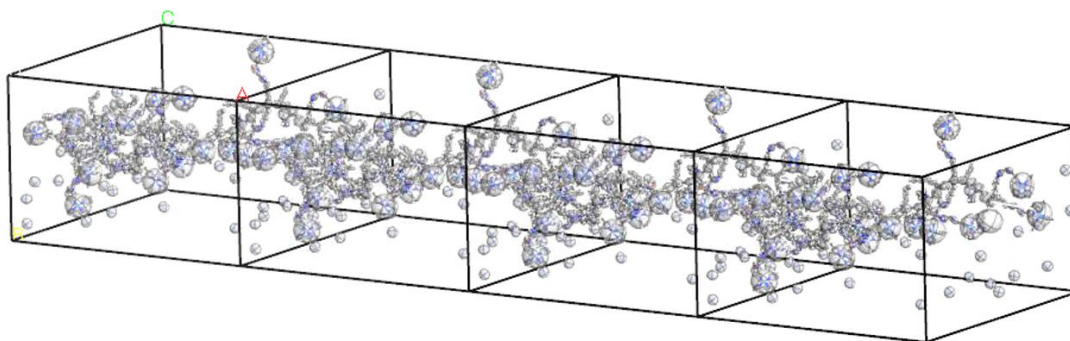
With all components of the MG-defined AA system sufficiently relaxed, we construct a simulation cell to hold 40 counterions (Cell 1). We achieve this by going to ‘Modules’ then ‘Amorphous Cell’ and ‘Calculation’ in the top menu. To run the construction calculation, we set the density to 1 g/cm<sup>3</sup> and apply a Universal force field to the system. After annealing Cell 1 under the same conditions as above, we make a copy of Cell 1, and label it Cell 2. In Cell 2 we delete the PF<sub>6</sub><sup>-</sup> counterions and place a copy of the annealed MG 20mer model into the empty cell. The polymer is oriented to align with cell length along axis A using translational keys (  ). We make room for the MG 20mer, 40 counterions, and 8412 solvent molecules by changing the dimensions of Cell 2 (89 x 58 x 61 Å). When changing the ‘lattice parameters’ we uncheck ‘keep fractional coordinates’ under the ‘advanced’ tab (Figure 3.4C). The 40 annealed MG PF<sub>6</sub><sup>-</sup> counterions from Cell 1 are transferred into Cell 2 so that Cell 2 contains our annealed MG 20mer model and 40 PF<sub>6</sub><sup>-</sup> counterions. Figure 3.5 illustrates our system after the first stage of constructing our simulation cell is complete.





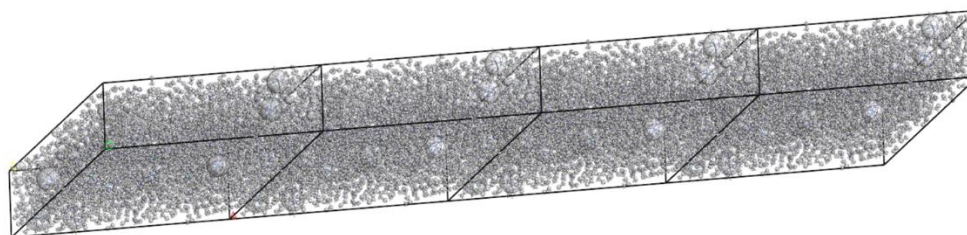
**Figure 3.5.** Simulation unit cell constructed with MG 20mer and MG  $\text{PF}_6^-$  counterions.

The polymer in Figure 3.5 does not yet represent the 70-complex system that we probed experimentally. From the periodic boundary conditions of the simulation cell, we can infinitely propagate the cell and its contents. We use the lattice display style window (Figure 3.4B) to propagate the lattice two units along the A axis - the axis in which our polymer is oriented. We connect the terminal Carbon atom of the polymer backbone in the first cell to the head Carbon atom of the polymer backbone in the second cell. The cell and its contents are further propagated to create a four-unit simulation cell that contains a MG-defined PF-Ru<sub>80</sub> (MG 80mer) and 160  $\text{PF}_6^-$  counterion models (Figure 3.6).



**Figure 3.6.** Four-unit simulation cell with MG-defined PF-Ru<sub>80</sub> (MG 80mer) and MG  $\text{PF}_6^-$  counterions.

We navigate back to ‘Modules’ then ‘Amorphous Cell’ and ‘Calculation’ to pack the lattice with MeCN molecules. The ‘packing’ calculation is setup with a density of 0.786 g/cm<sup>3</sup>, which follows that of acetonitrile at 25 °C,<sup>42</sup> and an applied universal force field. The size of the unit cell lattice defined under the construction calculation ensures that 8412 MG-defined solvent molecules can be packed into the four-unit lattice (Figure 3.7).



**Figure 3.7.** Four-unit simulation cell with the MG 80mer and PF<sub>6</sub><sup>-</sup> counterions and packed with MG MeCN solvent molecules.

The simulation cell containing one MG 80mer, 8412 acetonitrile solvent molecules and 160 [PF<sub>6</sub>]<sup>-</sup> counterions undergoes optimization and annealing calculations under the same conditions previously described. We propagate our polymer in space using the periodic boundary conditions of the cell. The symmetry of the extended lattice simplifies calculations by mirroring motions that occur in a single unit cell to the remaining three. This results in a solvated 80mer MD simulation that will have the same computational cost as a solvated 20mer MD simulation.

### 3.2.3. System Parameterization.

UV-Visible absorption spectra were collected using a UV-Visible-NIR absorption spectrophotometer (Agilent Technologies, model 8453A) operated with tungsten and deuterium lamps lit. Air was used as the baseline and samples were placed perpendicular to the beam path. An integration time of 0.5 s was used.

Before we CG this system and perform MD simulations on the cell, we run a 40 ns MG-defined AA MD simulation on the cell (details in Section 5.2). The MGs visible in the resulting MD trajectory act as the centers of mass for each group of atoms, from which we are able accurately parameterize, or map, values for each interaction potential type onto our CG MD model.<sup>34, 35, 43-46</sup> Potentials used in our eventual CG MD simulations include bond stretching, angle bending, electrostatic, and van der Waals interactions (Equations 3.1A-D, respectively). The two bonding interactions are constrained by harmonic and cosine harmonic potentials, respectively, while the two non-bonding are defined by Coulombic and Lennard-Jones 12-6 potentials, respectively.<sup>38</sup>

$$U_B(R_{ij}) = \frac{1}{2} k_B (R_{ij} - R_0)^2 \quad (3.1A)$$

$$U_A(\theta_{ijk}) = \frac{1}{2} k_A [\cos(\theta_{ijk}) - \cos(\theta_0)]^2 \quad (3.1B)$$

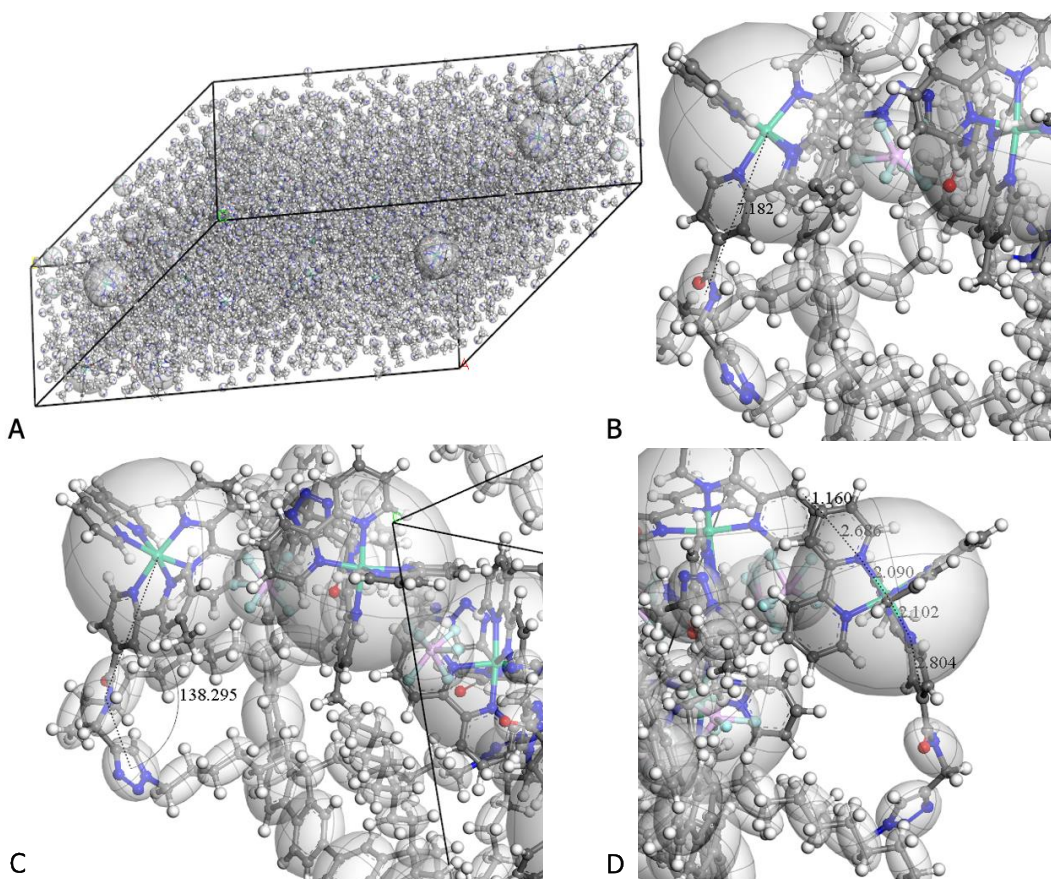
$$U_{el}(r_{ij}) = \frac{q_i q_j}{\epsilon_1 r_{ij}} \quad (3.1C)$$

$$U_{LJ}(r_{ij}) = 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] \quad (3.1D)$$

Equations 3.1 describe potential functions for (A) bond stretching, where  $R_{ij}$  is the time-dependent distance between two bound beads,  $R_0$  represents the equilibrium bond length and  $k_b$  is the bond bending force constant; (B) angle bending, where  $\theta_{ijk}$  represents the time-dependent angle between three bound beads,  $\theta_0$  is the equilibrium angle, and  $k_A$  represents the angle bending force constant; (C) electrostatic interactions, where  $q_i$  and  $q_j$  are charges on a pair of beads,  $\epsilon_1$  is the solution permittivity, resp.,  $r_{ij}$  is the time-dependent distance between two non-bonding charged beads; and (D) van der Waals interactions, where  $\epsilon_{ij}$  is interaction energy,  $\sigma_{ij}$  is


the bead diameter under a soft sphere approximation, and  $r_{ij}$  is the time-dependent distance between two neutral non-bonding beads.

From the MGs in our AA cell (Figure 3.8A), we can determine equilibrium bond lengths ( $R_0$ ) and angles ( $\theta_0$ ) by measuring distances from one MG to its neighbor (Figure 3.8B) and angles between three MGs (Figure 3.8C), respectively. Individual bead sizes ( $\sigma_{ii}$ ) are determined by measuring distances across MGs (Figure 3.8D).<sup>47</sup>



**Figure 3.8.** (A) MG-defined AA simulation cell. (B) Process for measuring equilibrium bond length between two MGs. (C) Process for measuring equilibrium angle between three MGs. (D) Process for measuring bead size across bonds an MG are shown.

Taking measurements of the MG defined AA system allows us to effectively ‘map’ the AA structure onto our CG model to ensure the accuracy of our CG model. We complete these measurements for individual MGs and sets of MGs that are set to become beads. To explain how these measurements are conducted, we focus on sets A – B and A – B – C to describe the equilibrium bond length and angle determination, respectively, and type A to describe how bead size is determined.

We measure the equilibrium bond length A – B by clicking on the distance measuring () tool in the top menu then selecting MG of types A and B on the first monomer unit to measure the distance between them (Figure 3.8B). To get a statistically accurate measurement, all A – B distances are selected, averaged across the polymer in that frame and all others in the 40 ns AA MD trajectory. The average equilibrium bond length for set A – B is 7.72 Å. This type of measurement is repeated for each bead type set. Bond stretching force constants were already determined by Marrink et.al. and left alone.<sup>38</sup> Table 3.2 lists equilibrium bond lengths,  $R_0$ , and force constants,  $k_0$ , for each set.

**Table 3.2.** Bead type set equilibrium bond lengths,  $R_0$ , and force constants,  $k_0$ , for bond stretching potential.

<b>Bead Type Set</b>	<b>Equilibrium Bond Length, <math>R_0</math>, (Å)</b>	<b>Force Constant, <math>k_B</math>, (kcal/mol/Å<sup>2</sup>)</b>
<b>A – B</b>	7.72	2.99
<b>B – C</b>	4.06	2.99
<b>C – D</b>	3.26	2.99
<b>D – E</b>	3.99	2.99
<b>D – D</b>	3.08	2.99
<b>E – F</b>	4.53	2.99
<b>F – F</b>	4.22	2.99
<b>G – H</b>	2.04	2.99

The charges on the Ru complex and counterion MGs are +2 and -1, respectively. Acetonitrile is defined by two MGs, where the partial charge on each ( $q_{ij}$ ) is determined from the dipole moment of the solvent ( $u$ ) and the equilibrium separation between MGs ( $R_0$ ).

$$u = q_{ij} * R_0 \quad (3.2)$$

Given that acetonitrile has a dipole moment of 3.84 D at room temperature<sup>48</sup> and its MGs (G and H) separated by 2.04 Å (Table 3.3), we calculate a partial charge of  $\pm 0.406$  using Equation 3.2. Partial charges on MeCN, the charges on Ru complex bead and  $\text{PF}_6^-$  counterion are listed in Table 3.3.


**Table 3.3.** Bead type charges.

Bead Type	Charge, q
A	+2
G	+0.406
H	-0.406
I	-1

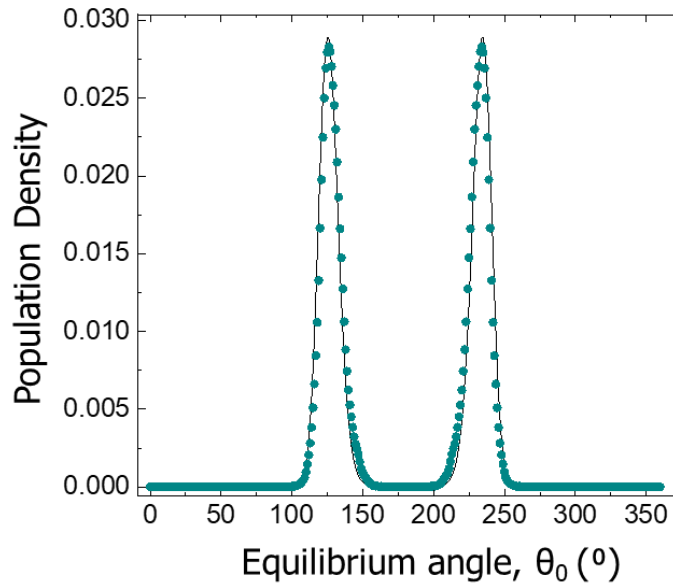
For all beads except those that comprise the solvent (G and H), we measure bead sizes by physically summing the lengths across bonds, through cyclic structures, and adding the Lennard-Jones radii of atoms at each end of that MG (Figure 3.11D). Sigma for each bead ( $\sigma_{ii}$ ) are listed in Table 3.4.

**Table 3.4.** Bead type sigmas,  $\sigma_{ij}$ .

Bead Type Set	Sigma, $\sigma_{ij}$ (Å)
A	13.06
B	6.22
C	5.46
D	4.92
E	3.34
F	6.10
G	4.18
H	4.18
I	6.36

We measure the equilibrium angles between A – B – C by clicking on the angle measuring tool (  ) in the top menu then selecting MG of types A, B, and C on the first pendant of the first monomer unit to measure the angle between them (Figure 3.11B). To get a statistically accurate measurement of A – B – C, we select all like angles along the polymer in that frame. When we propagate through the 40 ns AA MD trajectory, we collect angles A – B – C across all frames and save them as the excel file ABC.xlsx. This angle measurement is repeated for each bead type set and the angles of each set are stored in their own BeadTypeSet.xlsx file. We input each excel file into a home-written script (*AngleAnalysis.m*) that histograms these angles (Section 3.1.5).

*AngleAnalysis.m* outputs an array of equilibrium angle population densities for any given bead type set. The script loops through all six bead type sets of interest to produce a 6 x 1 matrix of equilibrium angle population densities. For each bead type set, we adjust the equilibrium angle ( $\theta_0$ ) and force constant ( $k_A$ ) values in the angle bending interaction potential function (Equation 3.1B) until we best fit our population density of angles. An example of the fit for A – B – C is shown in Figure 3.12.



**Figure 3.9.** Angle Analysis Fitting. The population distribution of the equilibrium angles (teal) is fit by the angle bending interaction potential function (black).

**Table 3.5.** Bead type set equilibrium angles,  $\theta_0$ , and force constants,  $k_A$ , that provide best fits for angle bending potential.

Bead Type Set	Equilibrium Angle, $\theta_0$ , (°)	Force Constant, $k_A$ , (kcal/mol)
A – B – C	126	60
B – C – D	154	8
C – D – E	140	7
D – E – D	120	8
D – D – E	180	60
D – D – D	180	12

Finally, we set parameters for Lennard-Jones 12-6 (LJ 12-6) interaction potential parameters between each non-bonded, neutral bead. Each bead type size,  $\sigma_{ii}$ , has already been determined and is listed in Table 3.5. When one bead interacts with a bead of different type, LJ sigma is the average sigma of those beads,  $\sigma_{ij}$ . For example, the LJ 12-6 potential between A and B has an interaction energy of 1.20 kJ/mol and a LJ 12-6 sigma of 9.64 Å as calculated from the average



bead sizes of bead A ( $\sigma_{ij}$ , 13.06 Å) and B ( $\sigma_{ij}$ , 6.22 Å). Interaction energies between each bead type pairing were already determined by Marrink et. al. and left alone.<sup>38</sup> Table 3.6 lists the interaction potential parameters for the LJ 12-6 sigma ( $\sigma_{ij}$ ) and interaction energy ( $\epsilon_{ij}$ ) for all possible bead-bead interactions.

**Table 3.6.** Bead type sigmas,  $\sigma_{ij}$ , for Lennard-Jones 12-6 potential.

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
<b>A</b>	13.06								
<b>B</b>	9.64	6.22							
<b>C</b>	9.26	5.84	5.46						
<b>D</b>	8.98	5.56	5.18	4.92					
<b>E</b>	8.20	4.78	4.40	4.12	3.34				
<b>F</b>	9.58	6.16	5.78	5.50	4.72	6.10			
<b>G</b>	8.62	5.20	4.82	4.54	3.76	5.14	4.18		
<b>H</b>	8.62	5.20	4.82	4.54	3.76	5.14	4.18	4.18	
<b>I</b>	9.72	6.28	5.90	5.64	4.78	6.22	5.26	5.26	6.36

**Table 3.7.** Bead type interaction energies,  $\epsilon_{ij}$ , for Lennard-Jones 12-6 potential

	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
<b>A</b>	1.20								
<b>B</b>	1.20	1.08							
<b>C</b>	0.96	1.08	0.96						
<b>D</b>	0.48	0.65	0.65	0.84					
<b>E</b>	0.55	0.65	0.65	0.84	0.84				
<b>F</b>	0.65	0.74	0.74	0.74	0.84	0.84			
<b>G</b>	1.34	1.20	1.20	0.48	0.55	0.65	0.60		
<b>H</b>	0.48	0.65	0.65	0.84	0.84	0.74	0.24	0.42	
<b>I</b>	1.08	0.96	0.96	0.48	0.55	0.65	1.08	0.48	0.84

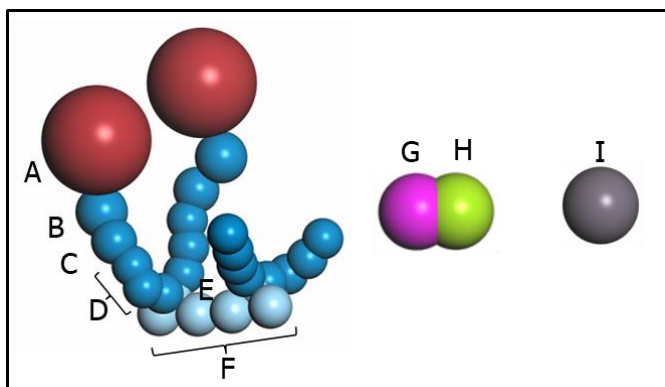
Values from Tables 3.2 – 3.7 are later entered into the Forcefield Document (*newbeads\_ang.off*) we use to simulate motion in our CG model.

### 3.2.4. Coarse-Grain (CG) System.

To convert MG versions of the polymer, solvent and counterion into CG structures, Materials Studio requires that we first create a *Typing Document* from our MGs. This document correlates each unique group of atoms to a bead *type*. From the top menu, we navigate to ‘Build’ then ‘Build Mesostructure’ and ‘Coarse Grain’. We select ‘Motion Groups’ under Method and click on ‘Create’ to make the typing document that corresponds with our MG-defined cell.

From our typing document, we can convert the MG-defined cell to a CG system by once again navigating from the top menu to ‘Build’ then ‘Build Mesostructure’ and ‘Coarse Grain’ then selecting ‘Build’ a coarse grain document with the typing document as our input. We can change the view of our coarse-grained (CG) lattice to space filling spheres (CPK) in the Display Style window. We can adjust the size of the beads to represent those calculated in Table 3.5 by navigating to ‘Build’ then ‘Build Mesostructure’ and ‘Bead Types’ in the top menu.

The CG simulation lattice contains one CG 80mer, 8412 CG acetonitrile solvent molecules and 160 CG  $[\text{PF}_6]^-$  counterions, or 18024 total beads. We assign bead types and add charges in the Materials Studio properties panel (Figure 3.2C).



**Figure 3.10.** Coarse grained (CG) depiction of  $\text{FRu}_2$ , MeCN, and  $\text{PF}_6^-$  with beads identified by their type A-I.

In the panel we assign all beads their unique type. As an example, we focus first on Ru complex beads, type A (Figure 3.13). We select all Ru complex beads, go to the properties panel, label that set of beads as type A, and add a +2 charge to the center of each bead. The only beads that require a charge besides the Ru complexes are  $\text{PF}_6^-$  counterions (-1), the methyl (+0.406) and nitrile groups (-0.406) of acetonitrile. We make sure all charges balance out by going to ‘Modify’ then ‘charges’ from the top menu, selecting all beads in the cell, and setting all charges to zero. If we select all Ru beads and set all of those charges to +2, the total charge in for the cell should be +40. Giving each bead a ‘type’, or identity, helps the forcefield document recognize the beads and will be used in our MD calculations to apply some combination of potentials to each bead based on that identity.

Unlike how we determined size for all other beads in the system (Section 3.1.3), the bead size of MeCN depends on its dielectric partial charges and density. To determine the size of our MeCN beads, we place 2103 fully parameterized MeCN molecules into a single unit cell. The same number of solvent molecules fill each unit cell of our system lattice. With the correct partial charges applied the solvent beads, we run a series of 10 ns MD simulations on the MeCN cell with radii set from 3 to 4.4 Å. After each run, we check the cell density to see if it matches that of MeCN at room temperature (0.786 g/cm<sup>3</sup>). We find that a sigma of 4.18 Å achieves this density (Tables 3.5 and 3.8).

### **3.3. Results and Discussion.**

Before running a full MD trajectory on the CG cell, we must first equilibrate the cell. Since there exist no other ways to relax the CG system like there were for the AA cell (i.e. geometry optimization and annealing), we must equilibrate the cell for a longer period of time. After navigating to ‘Modules’ then ‘Mesocite’ and ‘calculation, we set the calculation to run for at

least 30 ns with 1 fs steps under the same conditions used for the AA MD equilibration. This can be done on a desktop computer. We must go to the final frame in the CG MD equilibration output and use it to set our ‘initial velocities’ as ‘current’. This way, we can either start our full CG MD calculation trajectory from the newly relaxed structure or run another equilibration to further relax the system.

For both types of coarse-grained MD simulations, Verlet Integration summates the forces from bond stretching and angle bending interactions are constrained by harmonic and cosine harmonic potentials imposed on each bead every 8 fs. A cubic spline switching function truncates long-range, non-bonding potentials at 12.5 Å, where interactions between bead pairs separated by more than this distance are not calculated. A Molecular Dynamics (MD) simulation cell is coupled to a Nose thermostat set to 298 K and a Berendsen barostat set to 1 atm, both of which operate with a relaxation time of 1 ps. Time steps occur every 8 fs and are saved every 5000<sup>th</sup> frame such that 8751 frames are produced for each 350 ns trajectory. Running the calculation on UNC’s high-performance computing cluster takes about 6 days to complete – 12 days to run the full 700 ns trajectory.

To increase the statistic accuracy of our eventual kinetic model, we make five more copies of the MG-defined AA system, re-anneal, CG, parameterize, and run MD simulations on each system using the same procedures described above. Code for the AA and MD simulation calculations can be found in section 5.1.

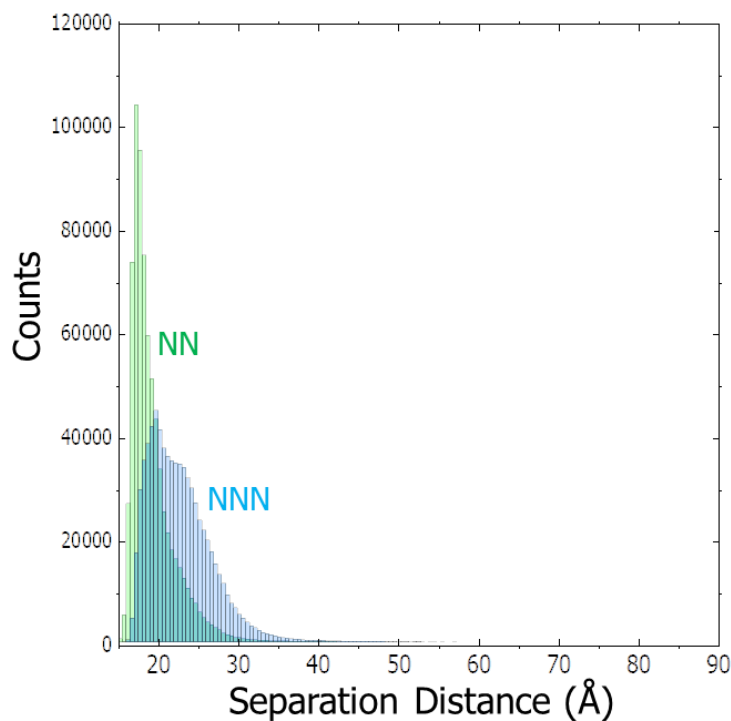
### **3.3.1. CG MD Simulation Results.**

All six 700 ns CG MD trajectories that result highlight the prominence of pendant complex motions. The flexible pendant linkages elongate away from the backbone and make rapid stochastic fluctuations as the pendant beads come into contact with the surrounding solvent. The

backbone of the CG assembly, on the other hand, remains relatively rigid. We can probe the structural evolution of the assembly, with particular focus paid to the pendants by tracking pendant complex-complex distances over time.

### 3.3.2. Nearest Neighbors (NNs) Analysis.

Most of these distances result in near negligible EnT kinetics. Since we are most concerned with how energy is transferred within the system, we focus on structural fluctuations that affect distances between complexes and their nearest neighbors (NN) and next nearest neighbors (NNN). In Figure 3.14, the distances between each complex and its NN and NNN are plotted over time.

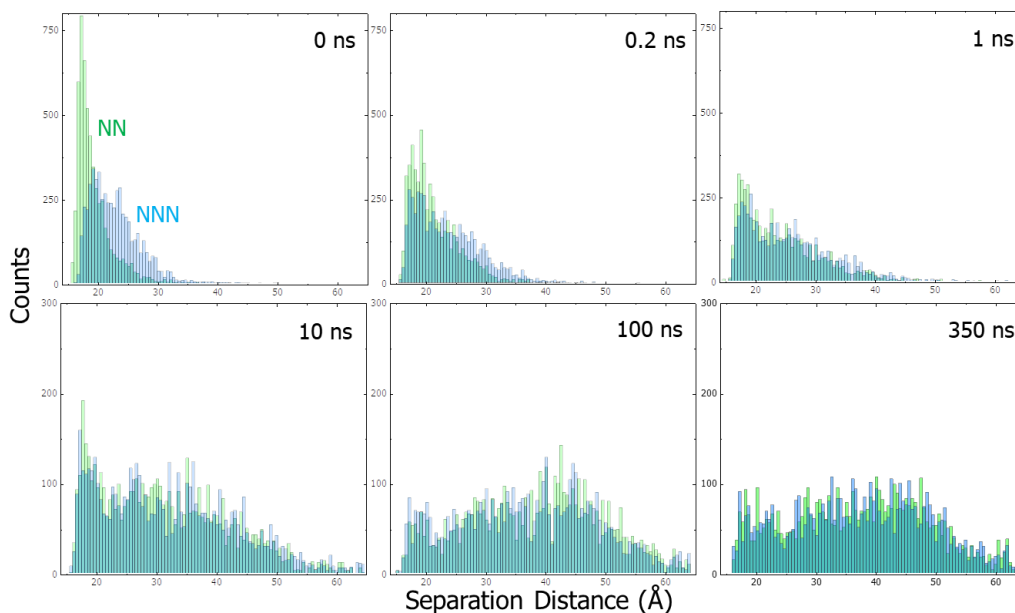


**Figure 3.11.** Histogrammed distances of all NN (green) and NNN (blue) pairs in the CG MD PF-Ru 70mer over 350 ns.

The plot indicates NN and NNN have average center-center separations of 17 and 23 Å, respectively. Given that the Lennard-Jones radius of the Ru complex bead is 7.33 Å, NNs and NNNs are peripherally separated by an average of 2.34 and 7.34 Å, respectively. The average NN separation would yield a  $0.027 \text{ ns}^{-1}$  hopping rate by Dexter EnT. However, given that 0.8 % of NN peripheral distances are under 1.34 Å, hopping rates could reach  $0.12 \text{ ns}^{-1}$  or up to  $48.6 \text{ ns}^{-1}$  at a minimum NN separation of  $-2.66 \text{ Å}$ , since orbitals are permitted to overlap. These rates agree with the EnT lifetimes we observe for the Os sensitization for a single hop ( $\tau_1^{Os} = 1.2 \text{ ns}$ ,  $k_1^{Os} = 0.83 \text{ ns}^{-1}$ ) and multiple hops ( $\tau_2^{Os} = 8.0 \text{ ns}$ ,  $k_2^{Os} = 0.125 \text{ ns}^{-1}$ ) in Figure 2.7.

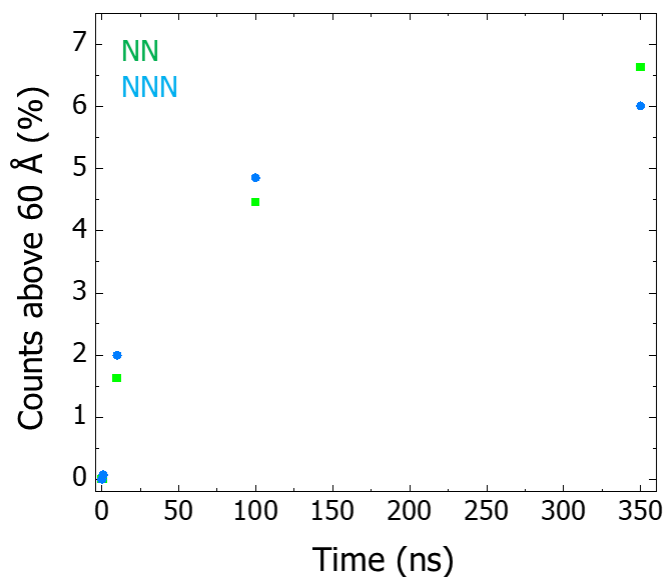
### 3.3.3. Follow Original NN and NNN Analysis.

To determine how the flexible structure evolves, we label the original NN and NNN of each complex at  $t = 0 \text{ ns}$  and track the distances between those pairs over time (Figure 3.12).



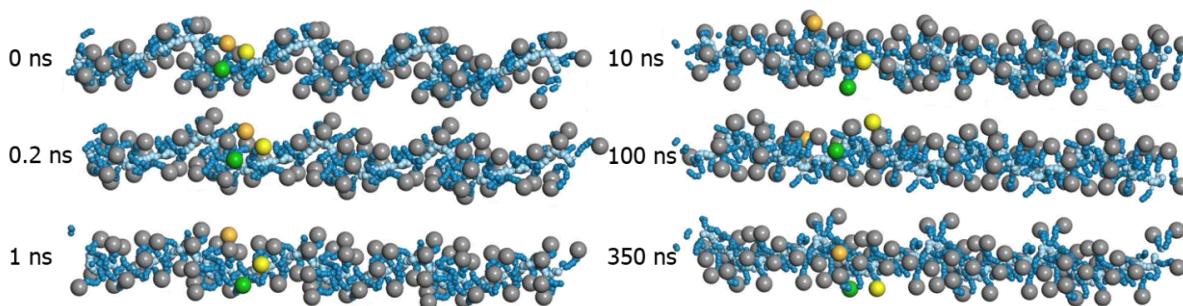
**Figure 3.12.** Histogrammed distances of original NN and NNN pairs in six different CG MD PF-Ru 70mer trajectories and multiple starting structures ( $T_{0s}$ ) recorded at times  $t$ .

We can quantify the evolution of this distribution by tracking the percentage of bin counts over  $60 \text{ \AA}$  over time (Figure 3.13).



**Figure 3.13.** Total bin counts of original NN and NNN distances above  $60 \text{ \AA}$  over time.

The percentage of original NN and NNN pairs separated by more than  $60 \text{ \AA}$  increases exponentially over 350 ns. At  $t = 350 \text{ ns}$ , 6 – 6.5 % of distances are separated by  $60 \text{ \AA}$  or more and the average distance a NN pair surpasses that of a NNN pair. The 1068% and 340% increase in respective NN and NNN pair average distances we observe indicates the presence of large-scale pendant fluctuations. We can get a sense for what these motions look like by tracking a single complex and its NN and NNN from one of our CG MD models (Figure 3.14).



**Figure 3.14.** Molecular diffusion of NN (orange) and NNN (green) of Ru<sub>35</sub> (yellow) in Trajectory 7 at times t.

A central complex, Ru<sub>35</sub> (yellow), and its NN (orange) and NNN (yellow) are separated by 18.9 and 23 Å, respectively, at t = 0 ns. Out to t = 1 ns, the group remains in close proximity. The NN and NNN drift to become a respective 48.3 and 25.0 Å away from Ru<sub>35</sub> by t = 10 ns. At t = 350 ns, Ru<sub>35</sub> and its NN and NNN are separated by 40.1 and 40.2 Å, respectively. When we visualize the simulated motion of this labeled CG MD trajectory with a 0.04 ns temporal resolution, the fluctuations we observe directly correlate the trends in Figure 3.15. Motions out to 1 ns are localized about the Ru<sub>35</sub>, its NN and NNN. At short timescales, few fluctuations cause Ru<sub>35</sub> to venture far from its original grouping and thus it ‘remembers’ its original NN and NNN. This is evidenced by the distinct distance distributions for the NN and NNN pairs in Figure 3.14. Over time, many more fluctuations occur and move Ru<sub>35</sub> further from its original NN and NNN such that Ru<sub>35</sub> ‘forgets’ its original grouping by 100 ns. It is at this point that NN and NNN distance distributions become indistinguishable.

Structural analyses of the CG MD PF-Ru model uncover the characteristic motions of the system. Distances between complexes and their NN ranging from -2.66 to 1.34 Å (Figure 3.14) would give rise to Os sensitization kinetics similar to those we observed in our time-resolved emission experiments. These pendant complexes diffuse via small and rapid fluctuations such

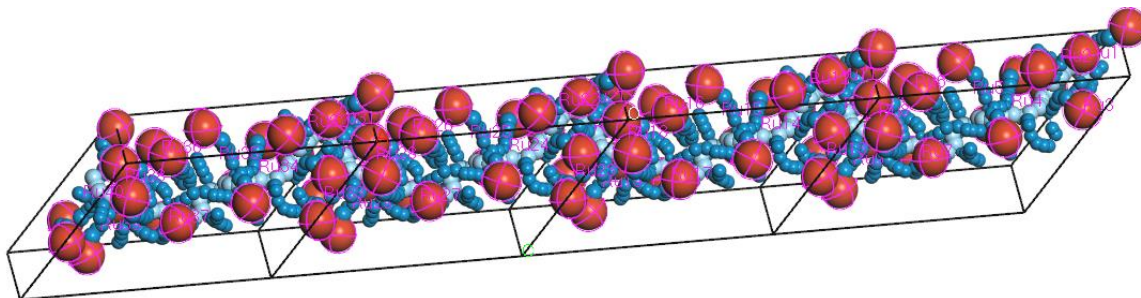


that the average complex becomes separated from its NN and NNN by 25 Å over 350 ns. TO discern how these structural fluctuations influence EnT dynamics, we must simulate that EnT in our model system.

## Chapter 4: CG MD Energy Transfer (EnT) Simulations

### 4.1. Ru Coordinate Extraction.

Since our main focus is the through-space transfer of the excited state from complex to complex, we only need to extract the time-dependent XYZ coordinates of the Ru complexes from our CG MD trajectories to incorporate our model into the EnT simulations. To extract these coordinates, we select Ru complex pairs in order of monomer fragments in the first half of the 700 ns CG MD trajectory and label the pairs as sets “A1-40”, or simply as sets “A” (Figure 4.1).



**Figure 4.1.** The 80mer CG PF-Ru model with sets “A” labeled. The cell is shown without solvent for clarity.

We repeat the same selection and set identification for the second half of the 700 ns trajectory. These trajectories with defined sets “A” serve as input files for a script that extracts the XYZ complex coordinates of Ru beads in sets “A”. We create an Excel document from the coordinates-only text files for both halves of each six 700 ns trajectory. The Ru XYZ coordinates we extract from the 700 ns CG MD PF-Ru<sub>80</sub> trajectories are used as inputs for the EnT simulation.

## 4.2. EnT Simulations.

Experimentally, EnT kinetics in an ensemble of PF-RuOs assemblies are observed via time-resolved emission spectroscopy through the rapid decay of Ru\* intensity and subsequent growth and eventual decay of Os\* intensity. Although the extended CG MD PF-Ru model can replicate the size of the experimental assembly, it does not yet represent a statistically accurate ensemble of PF-RuOs assemblies. It would be too computationally costly to simulate motions out to 700 ns in extended CG PF-RuOs assemblies that modeled all possible Ru/Os configurations for each Os loading scenario. Instead, we incorporate statistical Os loading into our EnT simulation script so we can more accurately model the experimental PF-RuOs ensemble of assemblies and retain the computational efficiency gained upon coarse-graining the system.

The statistical Os loading we use here comes from the synthetic process of PF-RuOs. Egle et al. synthesized the 70mer experimental [PF-Ru<sub>60</sub>Os<sub>10</sub>]<sup>140+</sup> assemblies with a 85%:15% Ru to Os ratio, on average. Os complexes are loaded onto the polyfluorene backbone followed by an excess of Ru complexes to generate an ensemble of [PF-Ru<sub>(N-n)</sub>Os<sub>n</sub>]<sup>140+</sup> assemblies, where N represents the total number of complexes on the assembly (N = 70, 70mer) and n is the number of Os that can be loaded onto the backbone (n = 0-17). On average, n = 10 ([PF-Ru<sub>60</sub>Os<sub>10</sub>]<sup>140+</sup>), but more or less Os complexes can be loaded onto any given assembly such that n = 15 ([PF-Ru<sub>55</sub>Os<sub>15</sub>]<sup>140+</sup>), n = 2 ([PF-Ru<sub>68</sub>Os<sub>2</sub>]<sup>140+</sup>) or n = 0 ([PF-Ru<sub>70</sub>]<sup>140+</sup>), to provide a few examples. Newton's binominal formula allows us to calculate the probability of any *Os loading scenario*:

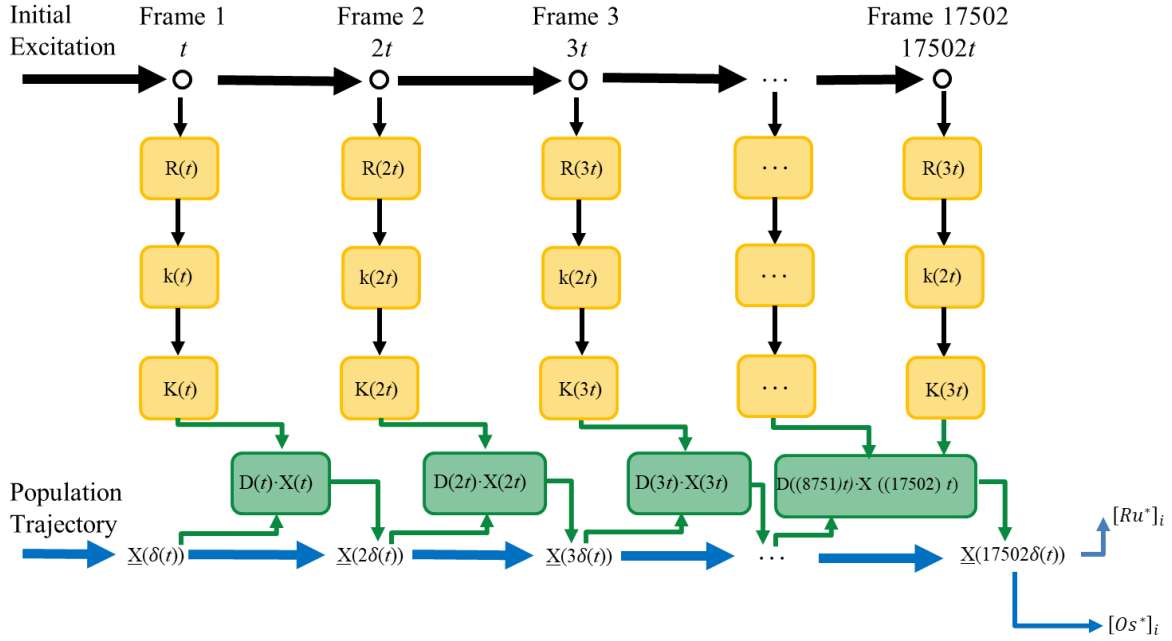
$$P_n = \frac{N!}{(N-n)!n!} \times P_{Ru}^{(N-n)} \times P_{Os}^n \quad (4.1A)$$

$$P_{Os} = \frac{1}{(1-A_0) \times 7} \quad (4.1B)$$

Where  $P_n$  represents the probability of loading  $n$  Os complexes onto the polymer backbone,  $P_{Ru}$  (85%) and  $P_{Os}$  (15%) are the set probabilities that a given site is Ru or Os, respectively, and  $A_0$  represents the probability of creating a PF-Os homopolymer. From Equation 4.1, there are 3.9%, 0.0009%, and 0.0003% probabilities  $[PF-Ru_{55}Os_{15}]^{140+}$ ,  $[PF-Ru_{68}Os_2]^{140+}$  and  $[PF-Ru_{70}]^{140+}$ , or the PF-Ru homopolymer, are created, respectively. For every Os loading scenario, many different *Ru/Os configurations* can result since  $n$  Os complexes are randomly loaded onto any site along the backbone.

Our EnT simulations incorporate Os loading by extracting 80 time-dependent, Ru complex coordinates from the extended CG MD PF-Ru model (CG MD PF-Ru 80mer). Since we probe a 70-complex system experimentally, we select only the center 70 complex coordinates to run EnT simulations on a true 70mer. We then select an Os loading scenario and artificially “load” those Os complexes onto random sites of the CG MD PF-Ru 70mer to create the simulation’s first Ru/Os configuration (CG MD PF-RuOs 70mer).

In a solution of PF-RuOs assemblies, our low energy high rep rate laser only excites one complex per assembly (450 nm, 28 nJ/pulse), which suppresses triplet-triplet annihilation (TTA) processes eliminating higher-order processes and allows for a network of first-order EnT events to occur in PF-RuOs. To simulate EnT under these experimental conditions, the first metal complex in the selected Ru/Os configuration is excited to initiate an EnT simulation (Figure 4.2).



**Figure 4.2.** Description of EnT simulation steps.

Unlike in PS-RuOs where static complex-complex distances allow rate constants to be averaged over the entire EnT process, inherent fluctuations in PF-RuOs result in time-dependent changes in Ru – Ru and Ru – Os separations. To reflect the system’s flexibility, we update the coordinates of all complexes at each Frame, separated by  $t$  (40 ps).

We use the distance formula to calculate separations between all complex pairs at each Frame and compile them into  $70 \times 70$  time-dependent distance matrices,  $R(t)$ . Dexter style EnT functions (Equation 2.1) are then used to calculate hopping rates between any two complexes at each Frame,  $k(t)$ , which are organized into  $70 \times 70$  time-dependent rate matrices,  $K(t)$ .

The kinetic matrices reflect the probability of the excited state hopping to any other site  $Ru^* \rightarrow Ru$  or  $Ru^* \rightarrow Os$ , at rates given by  $k^{Ru}$  or  $k^{Os}$  or decaying to its ground state; a rate that follows the emission lifetime of excited state Ru,  $k_{em}^{Ru}$ . Once energy is transferred to an Os site, the

excited state is trapped and  $Os^*$  decays to the ground state at a rate given by its emission lifetime,  $k_{em}^{Os}$ . The parameters  $k_{em}^{Ru}$  and  $k_{em}^{Os}$  are derived from the PF-Ru and PF-Os homopolymer lifetimes and are constants of value  $1.01 \times 10^{-3}$  and  $2.00 \times 10^{-2} \text{ ns}^{-1}$ , respectively (Figure 2.6).

By closely following the procedure outlined by Berbaran-Santos et. al.,<sup>49</sup> we construct a general set of rate equations that describe these events.

$$\frac{dX_n}{dt} = k_{NN} X_n \quad (4.2A)$$

$$\frac{dX_{GS}^{Ru}}{dt} = \sum k_{em}^{Ru} X_{Ru} \quad (4.2B)$$

$$\frac{dX_{GS}^{Os}}{dt} = \sum k_{em}^{Os} X_{Os} \quad (4.2C)$$

$X_n$  is the excited state population on complex,  $n$  and  $k_{NN}$  is located along the diagonal and is the sum of the excited state decay to all other complexes and ground states

$$k_{NN} = -\sum_{n=1}^{70} (k_{nn} + k_{em}^{Ru} + k_{em}^{Os}) \quad (4.3)$$

, where  $k_{nn}$  hopping rate between metal complexes  $n$  and  $n$ .

We express these rate equations in matrix form

$$\frac{dX}{dt} = K(t) X(t) \quad (4.4)$$

Where  $X(t)$  (state vector) and  $K(t)$  (kinetic matrix) are

$$X(t) = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ \dots \\ X_n \\ X_{GS}^{Ru} \\ X_{GS}^{Os} \end{bmatrix} \quad K(t) = \begin{bmatrix} k_{11} & k_{12} & \dots & \dots & k_{1n} & 0 & 0 \\ k_{21} & k_{22} & \dots & \dots & k_{2n} & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ k_{n1} & k_{n2} & \dots & \dots & k_{nn} & 0 & 0 \\ k_{Em}^{Ru} & k_{Em}^{Ru} & \dots & \dots & k_{Em}^{Ru} & 0 & 0 \\ k_{Em}^{Os} & k_{Em}^{Os} & \dots & \dots & k_{Em}^{Os} & 0 & 0 \end{bmatrix} \quad (4.5)$$

and  $t$  represents the time between steps (40 ps);  $X(0)$  are the initial excited state concentrations for the metal complexes ( $X = Ru$  or  $Os$ ) at sites  $n$  at  $t = 0$  ns.

$K(t)$  is a 72x72 time-dependent kinetic matrix where EnT rates occupy the first 70 rows and columns and decay from excited state Ru or Os occupy the final two rows. We diagonalize the kinetic matrix at each time frame,  $D(t)$ , to calculate the sum of rates for a given initial excitation of  $X_n$  in a selected Ru/Os configuration.

$$\frac{dX}{dt} = D(t) X(t) \quad (4.6)$$

The solution of the differential in Equation 4.6 is similar to the corresponding scalar interpretation

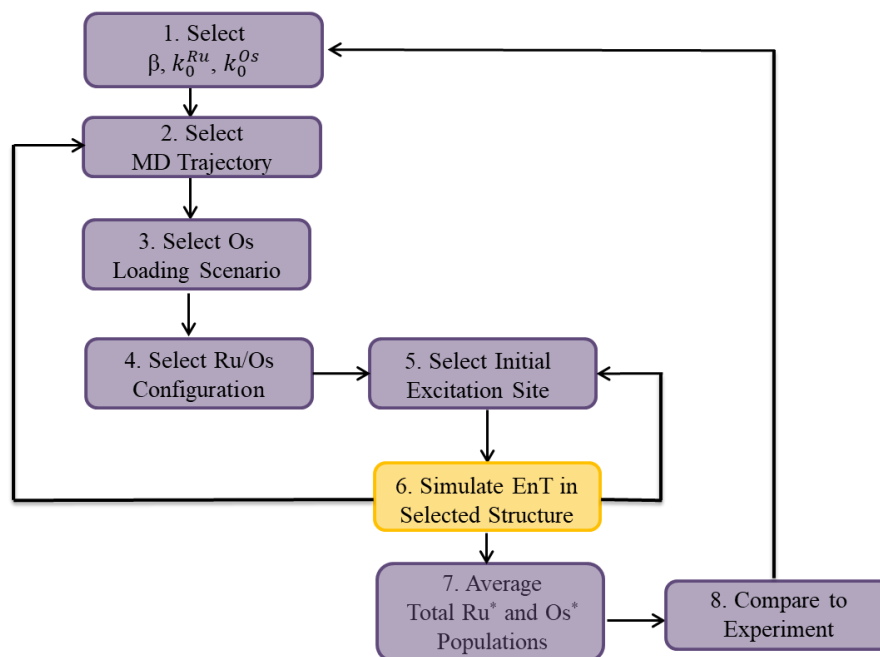
$$\underline{X}(\delta(t)) = P \times \exp(D(t) * t) \times P^{-1} X(t) \quad (4.7)$$

, where  $P$  and  $P^{-1}$  are the eigenvector and inverse eigenvector of the diagonalized rate matrix, respectively,  $X(t)$  is the updated initial population vector at time  $t$ , and  $\underline{X}(\delta(t))$  represents the sum of updated  $Ru^*$  and  $Os^*$  final populations at time  $t$ .

If complex  $n = 1$  were initially excited in our selected Ru/Os configuration, the excited state would fully occupy  $X_1$ , such that  $X_1 = 1$ , at  $t = 0$  ns. From the derivative of the first order rate law (Equation 4.7), we use our initial populations and the diagonalized rate matrix at  $t = 0$  ns,  $X(t)$  and  $D(t)$  respectively, to calculate sum of  $Ru^*$  and  $Os^*$  final population at  $t = 0$  ns,  $\underline{X}(\delta(t))$ . We use this final population vector as our new initial population vector and calculate a new final population for Frame 2. We continue to update the initial and final population vectors like this at each frame out to 700 ns (17502 frames) for the selected Ru/Os configuration and initial excitation. The result are population vectors for total  $Ru^*$  and  $Os^*$  population over time after initial excitation of  $X_1$  in the selected Ru/Os configuration.

Running an EnT simulation on just one Ru/Os configuration is not a statistically accurate representation of EnT dynamics that occur in an ensemble of PF-RuOs assemblies. No matter what  $\beta$ ,  $k_0^{Ru}$ , or  $k_0^{Os}$  values we choose for our rate equations (Figure 4.3, Step 1), our simulated EnT kinetics would yield a poor fit to the experimental results.





**Figure 4.3.** Flow chart for running EnT simulations on an ensemble of CG MD PF-RuOs 70mer models to accurately fit experimental EnT kinetic traces.

To increase the accuracy of our EnT simulations, incorporate five additional CG MD PF-Ru trajectories (Figure 4.3, Step 2). EnT simulations are looped over each MD trajectory, where each system can be loaded with 0 – 17 Os complexes (Figure 4.3, Step 3). EnT simulations are run on each Os loading scenario, each of which have 20 different Ru/Os configurations (Figure 4.3, Step 4). Finally, the EnT simulations loop over an initial excitation of each complex in all 20 Ru/Os configurations (Figure 4.3, Step 5). EnT simulations are run on each initial excitation in the selected Ru/Os configuration (Figure 4.3, Step 6) and the resulting Ru and Os excited and ground state populations are averaged across initial excitations and Ru/Os configurations. A weighted average is applied to the Ru and Os excited and ground state populations resulting from EnT simulations run for each Os loading scenario based on probabilities calculated in Equations 4.1 (Figure 4.3, Step 7). We run a total of 133,000 EnT simulations. This increases the statistical accuracy of our calculations, and allows our averaged simulated Ru and Os excited and ground

state population kinetics over time,  $\langle \underline{X}(\delta(t)) \rangle$  to more closely resemble the experimental traces (Figure 4.3, Step 8).

### 4.3. Comparing Simulation to Experiment.

#### 4.3.1. Model Accuracy.

Atomic layer under ambient conditions at 450°C for 30 minutes with a 45 minute ramp to temperature and left to cool for multiple hours.

#### 4.3.2. Sample Preparation for Transient Absorption.

We design our EnT simulations to accurately represent the ensemble of PF-RuOs assemblies we probe by time-resolved emission spectroscopy. In order to fit our experimental results properly, we must parameterize the kinetic constants of the simulated data (Figure 4.3, Step 1).

The through-space Dexter EnT functions in Equation 2.1 explicitly define kinetics for single EnT hops from  $\text{Ru}^* \rightarrow \text{Ru}$  or  $\text{Ru}^* \rightarrow \text{Os}$ . Five parameters from these functions are utilized in our EnT simulations:  $k_{em}^{Ru}$ ,  $k_{em}^{Os}$ ,  $\beta$ ,  $k_0^{Ru}$ , and  $k_0^{Os}$ . We give parameters  $k_{em}^{Ru}$  and  $k_{em}^{Os}$  their inherent values of  $1.01 \times 10^{-3}$  and  $2.00 \times 10^{-2} \text{ ns}^{-1}$ , respectively, and make guesses for the attenuation parameter ( $\beta$ ), the rate constant at closest approach for two Ru complexes ( $k_0^{Ru}$ ), and that for a Ru complex and Os trap ( $k_0^{Os}$ ) in an effort to best fit our experimental results. Good initial guesses for these parameters come in part from the set of  $\beta$  and corresponding  $k_0$  values Fleming et. al. used to simulate EnT kinetics in PS-RuOs and from our fitting of time-resolved emission traces at early times (Figure 2.7B).<sup>15</sup>

Since we cannot identify a single value for the attenuation parameter, we complete simulations for a range of  $\beta$  values between 1.2 and  $1.9 \text{ \AA}^{-1}$ . For a given  $\beta$  value, we adjust  $k_0^{Ru}$

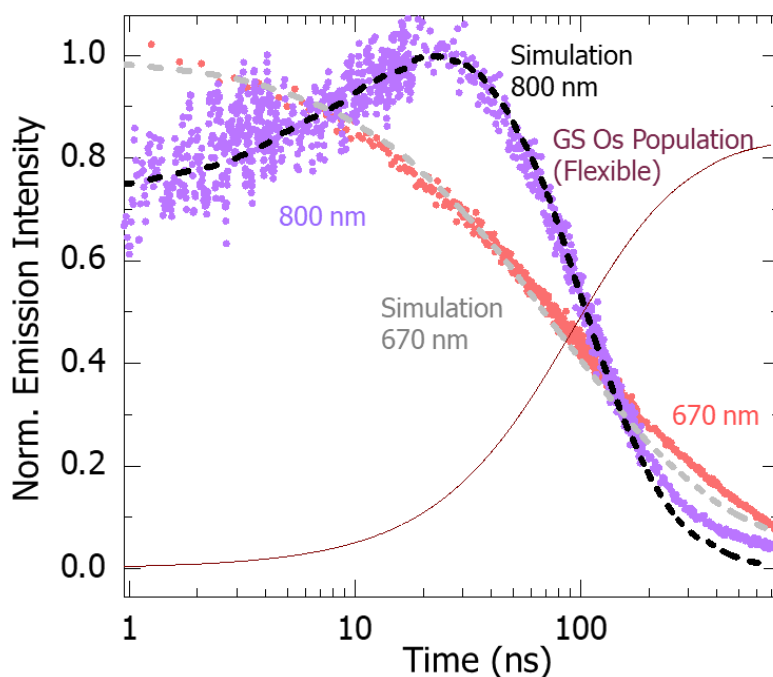
and  $k_0^{Os}$  to simultaneously fit simulated  $Ru^*$  and  $Os^*$  transients to those of the experiment. We keep  $k_0$  values for a given  $\beta$  when the simulated and experimental results align and run the entire EnT simulation script again with different  $k_0$  values when the fitting is poor. Table 4.1 lists the  $k_0$  values that provide the best fits for each  $\beta$ .

**Table 4.1.** EnT simulation rate constants at closest approach ( $k_0^{Ru}$  and  $k_0^{Os}$ ) that yield best agreement between simulation and experiment for different attenuation parameter values,  $\beta$ .

$\beta$ ( $\text{\AA}^{-1}$ )	$k_0^{Ru}$ , (ns $^{-1}$ )	$k_0^{Os}$ , (ns $^{-1}$ )	Efficiency
<b>1.2</b>	0.37	2.86	0.83
<b>1.3</b>	0.3	4.0	0.81
<b>1.5</b>	0.9	6.6	0.81
<b>1.7</b>	1.6	9.5	0.82
<b>1.9</b>	6.6	16	0.82

There is good agreement between our experimental data and simulations when  $\beta$  is set to 1.6 - 2.0  $\text{\AA}^{-1}$ . Simulations using  $\beta$  values outside of this range do not elicit good fits no matter which  $k_0$  values are used in the combination. As  $\beta$  increases,  $k_0$  values, or rate at closest approach, must also increase to compensate for the faster falloff rate between donor and acceptor complexes. We are unable to eliminate any of the  $\beta$  values listed in Table 4.1 since the fitting obtained with the other  $\beta$  values are all qualitatively similar to each other. Here, we present simulation data only for  $\beta = 1.5 \text{\AA}^{-1}$ , since it is the center of the range.

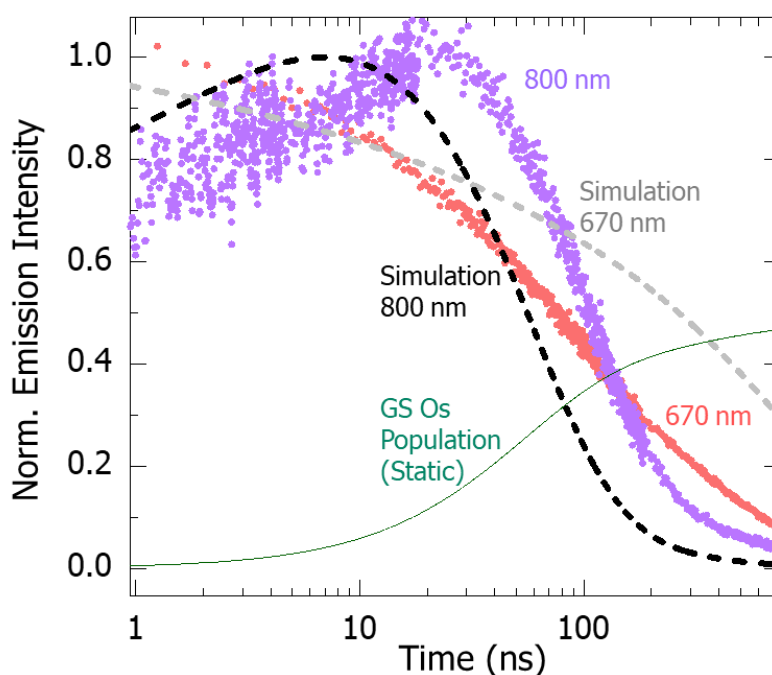
Figure 4.4 features a plot that overlays our 70mer simulation results with the experimental data.



**Figure 4.4.** Overlaid fitting with flexible CG MD PF-RuOs simulated kinetics for parameter set with  $\beta = 1.5 \text{ \AA}^{-1}$ . Plot depicts Os\* emission (purple) fit with the simulated Os\* time-dependent population (black) overlaid with Ru emission (red) fit with the simulated Ru\* time-dependent population (grey). The simulated ground state Os population intensity after 700 ns reaches 81% (wine).

The “goodness of fit” was determined by eye with strong emphasis placed on the rise and rollover part of the transient since a large portion of sensitization events occurs within this time window. The time axis is plotted on a log scale to better represent the short- and long-time components of the data. Simulation-generated Ru and Os EnT kinetics show excellent qualitative agreement with the experimental results. The agreement shown in Figure 4.5, along with the similarity between simulated and experimental EnT efficiencies (81% and 80-85% respectively), gives us enough confidence in our kinetic model to use it to investigate the role of flexibility in EnT within PF-RuOs.

The fits in Figure 4.5 result from EnT simulations in which pendants in our CG MD 70mer model remain flexible (flexible case). Using the same  $\beta = 1.5 \text{ \AA}^{-1}$  parameter set and similar ensemble sampling method, we simulate EnT in a CG MD 70mer model where pendants are forced to be static (static case). We attempt to fit our experimental data with simulated transients produced by the same range of  $\beta$  values (Table 4.4). Unlike in the flexible case, we find that simulated kinetics in systems with static pendants do not agree well with the experimental results no matter how small we set  $\beta$  (Figure 4.5).



**Figure 4.5.** Overlaid fitting with static CG MD PF-RuOs simulated kinetics for parameter set with  $\beta = 1.5 \text{ \AA}^{-1}$ . Plot depicts Os\* emission (purple) fit with the simulated Os\* time-dependent population (black) overlaid with Ru emission (red) fit with the simulated Ru\* time-dependent population (grey). The simulated ground state Os population intensity after 700 ns reaches 48% (green).

Although rates of simulated Ru excited state decay and Os sensitization remain concurrent at early times, the simulated EnT that does occur between static pendants gives rise to faster Ru excited state decay at short times and slower Ru decay at long times while the simulated growth and decay of the Os kinetics are faster overall. In systems with static pendants, the hopping rates depend exponentially on *fixed* distances between pendants,  $R_{ij}$ . Ru complexes already positioned close enough to Os in static CG MD PF-RuOs have excited states that decay rapidly to sensitize those traps, giving rise to the fast Os sensitization and emission kinetics we observe.

Far away Ru\* complexes in static systems do not possess the motion necessary to overcome the large attenuation parameter and/or hopping distances necessary to sensitize an Os trap. Instead of being quenched by Os, excited states on far away Ru will more often emit, resulting in smaller population in the Os ground state after 700 ns (48%). Rapid Os excited state decay and relatively small GS Os population buildup suggests that Os is a less ineffective quencher of far-away excited state Ru in static systems compared to systems where pendants are flexible.

Our kinetic method – CG MD and EnT simulations – allows us to incorporate the inherent motion of our 70mer by periodically updating the rate matrix to simulate Ru\* and Os\* populations over time and fit them to our experimental results. Our ability to turn the motion of the system's pendants off or on enables us to probe the role of these fluctuations in EnT. Given that only simulated kinetics from the flexible model agree well with experimental kinetics and that there exists a large difference in Os GS populations for the flexible and static cases (81% and 48%, respectively), pendant flexibility seems to aid EnT in this system.

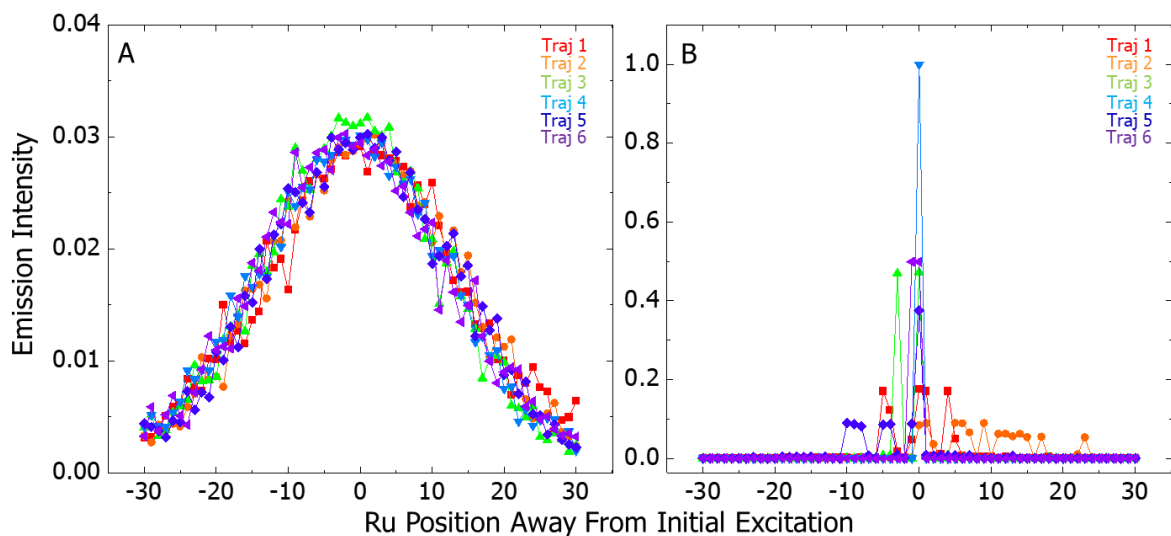
#### **4.4. Computational Experiments and Discussion.**

The fitting agreement we achieve in the flexible case gives us enough confidence in our kinetic model and a good indication that flexibility is beneficial for the system. From that confidence, we can validate our predictions that structural change or immobility influence EnT kinetics through several computational experiments that utilize our kinetic model as a framework.

The structural NN and NNN analyses we conducted on our CG MD PF-Ru model (Section 3.2) and comparisons of the fitting and GS Os population between the flexible and static cases gives us some indication that these motions aid EnT in PF-RuOs. However, we must utilize computational experiments that compare excited state diffusion in systems where pendants are flexible to those that are static to confirm this.

##### **4.4.1. Diffusion.**

Excited state diffusion is initiated by exciting a central Ru complex, Ru<sub>35</sub>. We can observe the population of the excited state on Ru out to  $\pm 30$  away after 350 ns in flexible and static systems for six different 350 ns trajectories in Figure 4.6.

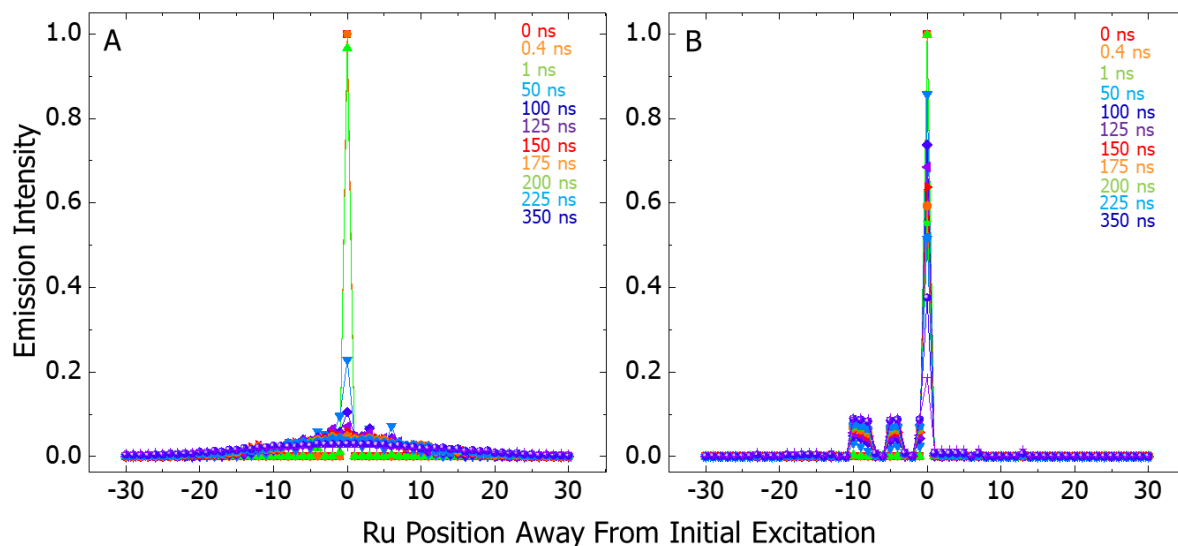


**Figure 4.6.** (A) ES diffusion after initial excitation of Ru<sub>35</sub> in Trajectories 1-6 with flexible pendants at t = 350 ns. (B) ES diffusion after initial excitation of Ru<sub>35</sub> in a system with static pendants at t = 350 ns.

For all trajectories, ES diffusion in the flexible case is largely Gaussian after 350 ns (Figure 4.6A). This is not the case when pendants are static (Figure 4.6B). Each trajectory is ‘frozen’ at t = 0 ns, positioning Ru<sub>35</sub> close to some complexes and far from others. When we simulate excitation of Ru<sub>35</sub>, the excited state cannot rely on molecular diffusion of the pendants to venture out along the backbone but must rely on its proximity to other complexes to achieve EnT. For example, the ES diffusion plot for Trajectory 5 at 350 ns reports significant population on Ru<sub>35</sub> and on neighboring complexes Ru<sub>34</sub>, Ru<sub>31</sub>, Ru<sub>30</sub>, Ru<sub>27</sub>, Ru<sub>26</sub> and Ru<sub>25</sub>. Over 350 ns, the initially excited Ru<sub>35</sub> distributes 62% of its ES population equally amongst these six neighboring complexes so that each carries ~9% of the ES population.

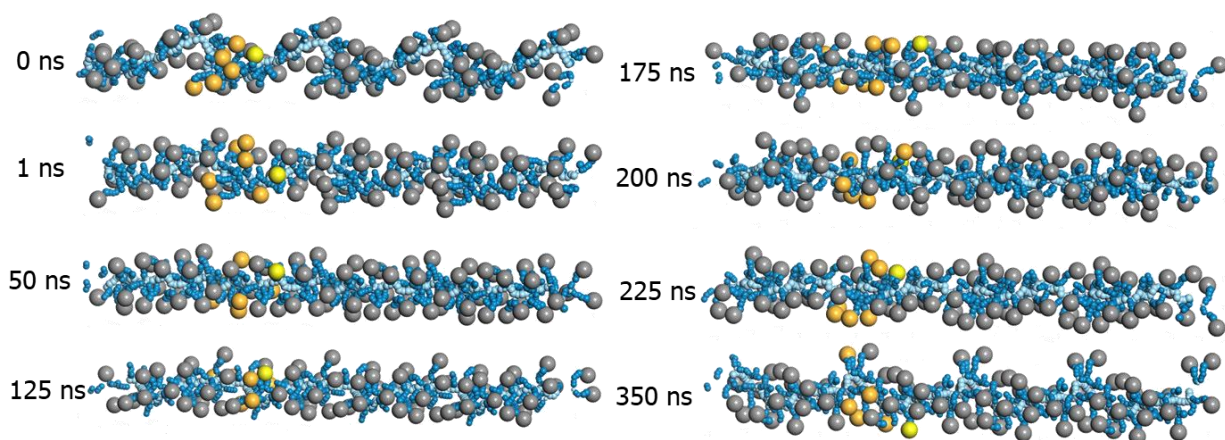
We initially excite Ru<sub>35</sub> and plot the ES diffusion in Trajectory 5 at times t in flexible and static systems to track the time-dependent distribution of the ES population (Figure 4.7).





**Figure 4.7.** (A) Averaged time-dependent ES diffusion plots after initially exciting Ru<sub>35</sub> in Trajectory 5 with flexible pendants at times  $t$ . (B) Averaged time-dependent ES diffusion plots after initially exciting Ru<sub>35</sub> in Trajectory 5 with static pendants at times  $t$ .

At early times, the flexible case possesses a sharp peak similar to that in the static case. The sharp peak of the ES diffusion trace slowly evolves into broad band over time and shows full Gaussian behavior by 200 ns as shown in Figure 4.7A. The static case maintains a spiky profile with a majority of the ES population settled on the initially excited Ru<sub>35</sub> and the remaining fraction dispersed evenly amongst its neighboring complexes. Over time, the ES transfers from Ru<sub>35</sub> to its neighboring complexes, eventually matching the profile for Trajectory 5 observed in Figure 4.6. We can observe the molecular diffusion of Ru<sub>35</sub> and its neighboring complexes, or its *primary domain*, from this CG MD PF-Ru trajectory over time (Figure 4.8).

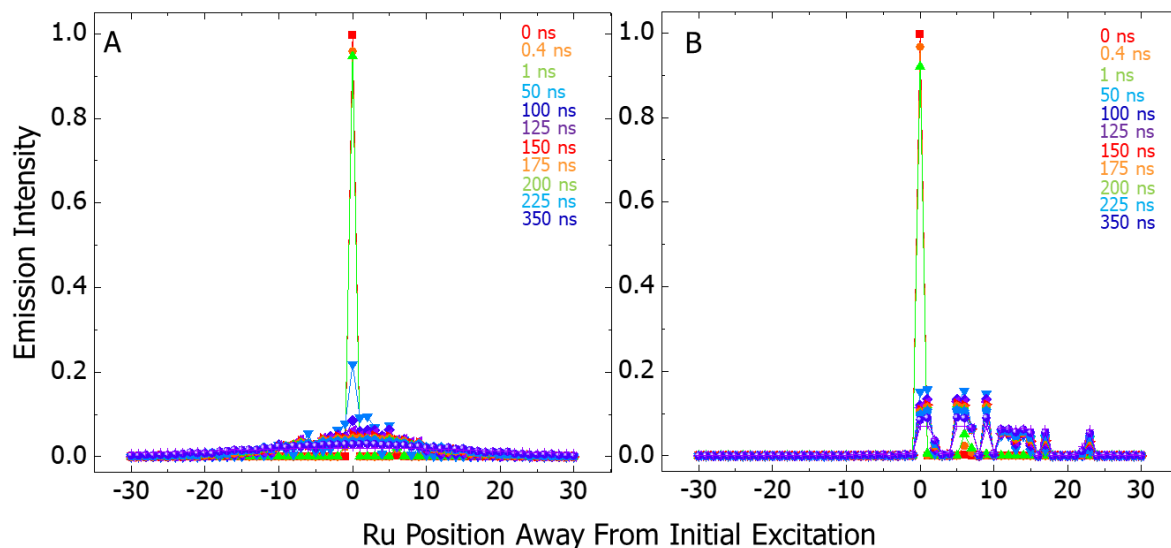


**Figure 4.8.** Molecular diffusion of primary domain (orange) after excitation of Ru<sub>35</sub> (yellow) in Trajectory 5 at times  $t$ .

The primary domain of Ru<sub>35</sub> at  $t = 0$  ns is compact ( $\sim 18.28$  Å average separation) and will remain that way as the ‘frozen’ structure for the static case. Without pendant motion, the ES is only able to populate complexes in its primary domain. The ES population growth is only evidenced on complexes based in this domain (Figure 4.7B).

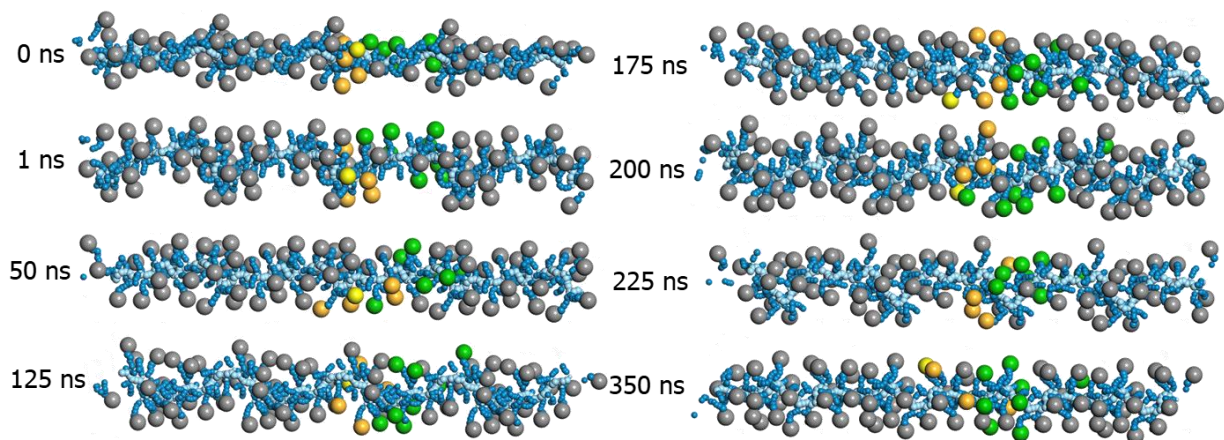
In the systems with flexible pendants, the primary domain molecularly diffuses in a way that is similar to that of a single pendant. Over time and after many naturally-occurring motions, the primary domain falls apart into its individual component pieces with pendants separated by an average of 35.31 Å by 350 ns. At short times, the pendants in the flexible case have not undergone enough motions to break out of their primary domain, restricting the ES to those complexes much like in the static case. The sharp peak in the ES diffusion plot broadens as pendants molecularly diffuse away from their original domain and move as individual pendants that can transport the ES across all polymer pendants. This diffusive behavior ultimately results in a gaussian ES diffusion curve by  $\sim t = 200$  ns (Figure 4.7A).

We investigate a different trajectory, Trajectory 2, to see how starting structure affects time-dependent ES diffusion from an initially excited Ru<sub>35</sub> in flexible and static systems (Figure 4.9).



**Figure 4.9.** (A) Averaged time-dependent ES diffusion plots after initially exciting Ru<sub>35</sub> in Trajectory 2 with flexible pendants at times  $t$ . (B) Averaged time-dependent ES diffusion plots after initially exciting Ru<sub>35</sub> in Trajectory 2 with static pendants at times  $t$ .

Similar to Trajectory 5, the sharp peak at  $t = 0$  ns in the ES diffusion plot for the flexible Trajectory 2 case evolves into a broad peak over 350 ns (Figure 4.9A). The diffusion plot for the static case has multiple peaks like it did for Trajectory 5. The peaks in Trajectory 2, however, are two-tiered. At  $t = 50$  ns, the initially excited Ru<sub>35</sub> and Ru<sub>36</sub>, Ru<sub>40</sub>, Ru<sub>41</sub>, Ru<sub>44</sub> each carry about 15% of the ES population. By  $t = 350$  ns, the ES population is distributed more broadly amongst Ru<sub>42</sub>, Ru<sub>46</sub>, Ru<sub>47</sub>, Ru<sub>48</sub>, Ru<sub>49</sub>, Ru<sub>50</sub>, Ru<sub>52</sub>, Ru<sub>58</sub>. Since the ES first flows from Ru<sub>35</sub> to the group containing Ru<sub>36</sub>, it is the *primary domain* of Ru<sub>35</sub> and the latter is its *secondary domain*. From Trajectory 2 of the CG MD PF-Ru model, we can observe the molecular diffusion of these domains over time, as shown in Figure 4.10.



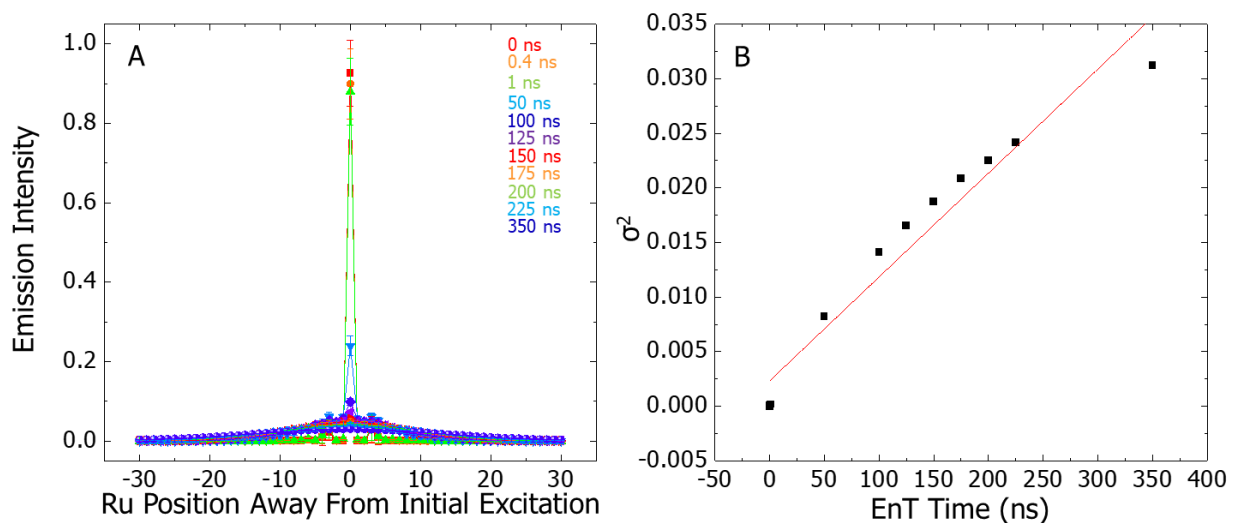
**Figure 4.10.** Molecular diffusion of primary (orange) and secondary (green) domains after excitation of Ru<sub>35</sub> (yellow) in Trajectory 2 at times  $t$ .

At  $t = 0$  ns, the primary and secondary domains of Ru<sub>35</sub> each have small average separations between complexes in their respective groups (17.27 Å and 16.97 Å, respectively) with the primary domain as a whole situated closer to Ru<sub>35</sub> than the secondary domain. Absent pendant motion, the ES is only able to travel amongst pendants in its primary domain. The ES must reach the periphery of the first domain before it can diffuse into the second. According to the plot in Figure 4.9B, the ES population in the primary domain begins to populate that of the secondary domain after ~50 ns.

The mobility of the pendants in the flexible system causes both domains to break down over time. At  $t = 50$  ns, we already observe heavy overlap between primary and secondary domains (Figure 4.10). These domains disassociate into pendants separated by an average of 42.51 Å and 49.30 Å, respectively, by 350 ns. Compared to ES diffusion in Trajectory 5 (Figure 4.7A), there exist subtle differences in which complexes the ES populates around  $t = 1$  ns. In both cases, the ES populates complexes that were close to the initially excited Ru<sub>35</sub> at  $t = 0$  ns, where the static system is ‘frozen’. This indicates that the flexible system retains a somewhat strong memory of

its original domains at 1 ns. This memory fades as molecular fluctuations allow pendants to diffuse further from those groups, resulting in the slow broadening of the sharp ES diffusion peak over time (Figure 4.9A). At around  $\sim t = 175$  ns, the pendants have fluctuated to the extent that the system is no longer confined by its original domains. This gives the ES freedom to move across the entire system and results in an entirely gaussian ES diffusion curve. In this work, we only elaborate on the domain structures of one Ru complex in the first 350 ns of Trajectories 2 and 5. Multiple domains could exist for any given trajectory and/or initially excited Ru that may give the ES more ability to bleed out and explore a greater length of the polymer.

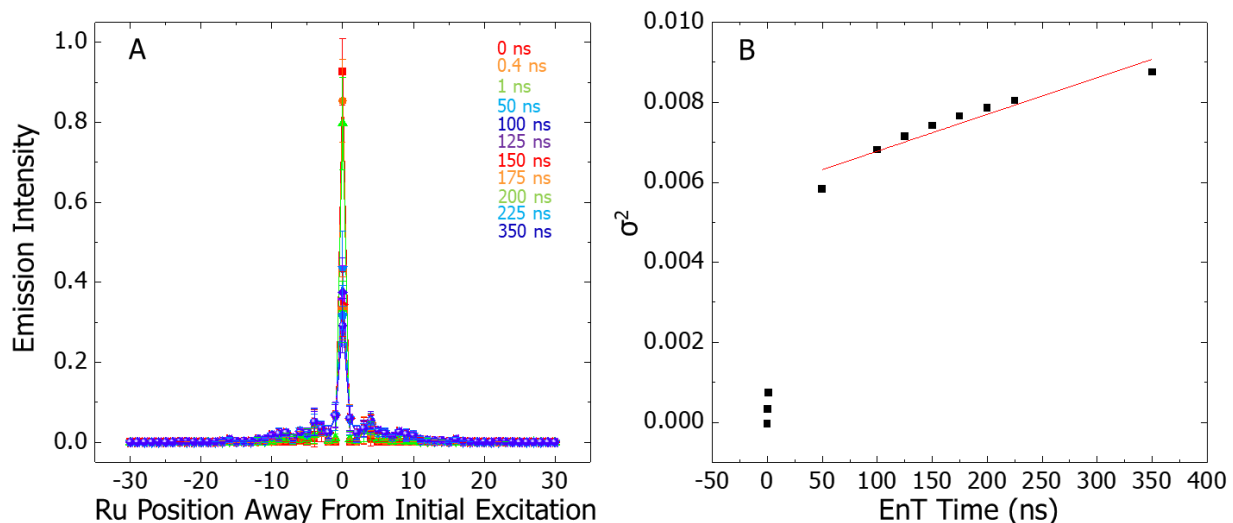
To get a better understanding of the ES diffusion within the ensemble, we can excite each centrally localized Ru (Ru<sub>31</sub> – Ru<sub>40</sub>), average the ES populations for all ten different initial excitations and six 350 ns CG MD trajectories and overlay those plots after the ES is allowed to diffuse over times  $t$  to extract the ES diffusion coefficient ( $D_{\text{coeff}}$ ) of the flexible system (Figure 4.11).



**Figure 4.11.** (A) Averaged time-dependent ES diffusion plots on Ru out to  $\pm 30$  away from the initial excited Ru at time  $t$  in systems with flexible pendants. (B) Distribution ( $\sigma^2$ ) plotted over times  $t$  in systems with flexible pendants.

The ES Diffusion in Figure 4.11A looks very similar to that in Figures 4.7A and 4.9A. By increasing the statistical accuracy of the plot we get a gaussian curve that is slightly more smooth at long times, while the same sharp peak at early time remains. This indicates that the fluctuations and domain evolution across trajectories and initial excitations result in very similar ES diffusions. Plotting  $\sigma^2$  of the average ES diffusion over time results in a  $D_{\text{coeff}}$  of  $9.52 \times 10^{-5}$  complexes<sup>2</sup>/ns for the flexible case (Figure 4.11B). The ES illustrates diffusive behavior throughout the trajectory given that the trend line passes through  $y_0 = 0$ .

We can compare these results to those in static systems by similarly averaging ES populations in all initial excitations and six 350 ns CG MD trajectories and plotting over times  $t$ . We calculate  $\sigma^2$  at each time to extract the ES diffusion coefficient,  $D_{\text{coeff}}$  of the static system (Figure 4.12).



**Figure 4.12.** (A) Averaged time-dependent ES diffusion plots on Ru out to  $\pm 30$  away from the initial excited Ru at time  $t$  in systems with static pendants. (B) Distribution ( $\sigma^2$ ) plotted over times  $t$  in systems with static pendants.

Averaging ES populations across static systems and initial excitations results in a slightly more gaussian profile; however, we lose the description of the unique domain(s) each individual trajectory provides (Figure 4.12A). Plotting  $\sigma^2$  of the average ES diffusion over time, alludes to an ES that only shows diffusive behavior after 50 ns (Figure 4.12B). If we fit only these values,  $y_0 = 0$  and  $D_{\text{coeff}} = 9.20 \times 10^{-6} \text{ complexes}^2/\text{ns}$  – a diffusion coefficient an order of magnitude less than that of the flexible case. The non-diffusive behavior observed before 50 ns can likely be explained by the restriction that ES feels in its primary domain shortly after initial excitation. No matter how large the primary domain is or how many domains are accessible to the ES, the ES seems unable to populate any of them fully until after 50 ns. At  $t = 50$  ns and beyond, the ES can occupy a substantial portion of a large primary domain or bleed into its secondary domain – in either case, taking on a modest ES diffusion compared to that observed in the flexible case.

#### 4.5. Conclusion.

Averaging ES populations across static systems and initial excitations results in a slightly more gaussian profile; however, we lose the description of the unique domain(s) each individual trajectory provides (Figure 4.12A). Plotting  $\sigma^2$  of the average ES diffusion over time, alludes to an ES that only shows diffusive behavior after 50 ns (Figure 4.12B). If we fit only these values,  $y_0 = 0$  and  $D_{\text{coeff}} = 9.20 \times 10^{-6}$  complexes<sup>2</sup>/ns – a diffusion coefficient an order of magnitude less than that of the flexible case. The non-diffusive behavior observed before 50 ns can likely be explained by the restriction that ES feels in its primary domain shortly after initial excitation. No matter how large the primary domain is or how many domains are accessible to the ES, the ES seems unable to populate any of them fully until after 50 ns. At  $t = 50$  ns and beyond, the ES can occupy a substantial portion of a large primary domain or bleed into its secondary domain – in either case, taking on a modest ES diffusion compared to that observed in the flexible case.



## APPENDIX A: MD TRAJECTORY CODE

### A.1. Job Submission.

Full length AA and CG MD simulations are run on 2 parallel nodes of Dogwood, one of UNC's high-performance computing (HPC) clusters. The following programs and accounts are required in order to submit jobs to Dogwood: 1) A Dogwood account that can be obtained through UNC's (ITS), 2) SSH Secure Shell Client or another terminal program, which allows user to type necessary commands, and 3) SSH Secure File Transfer Client or another file transfer program, which holds all input and output files of job. The file transfer program will ask for the connection type, hostname, port number, and username which are SFTP (SSH File Transfer Protocol), `longleaf.unc.edu`, 21 (Mac) or 22 (PC), and the user's onyen, respectively. To submit a job, users may log into their dogwood account with the following command: `ssh onyen@dogwood.unc.edu` and enter onyen password when prompted. The user can navigate to their home directory (`cd /nas/longleaf/home/onyen`) or to whichever path their submission files exist.

All coded scripts and functions for this project are saved on the AD backup under the following paths:

1) `ad.unc.edu/cas/chemistry/share/PapanikolasGroup/Workspace (Current)/Bowers(LMRB)/PF-RuOs_MDSimulations/Code_LMRBMac/Matlab`

2) *ad.unc.edu/cas/chemistry/share/PapanikolasGroup/Workspace (Current)/Bowers(LMRB)/PF-RuOs\_MDSimulations/Code\_LMRBMac/Cluster.*

3) */nas/longleaf/home/mbalu*

Path 1 holds a copy of all the original files and make edits to the code. The finalized files used to run those jobs are kept on Path 2. Path 3 is the longleaf home directory that contains scripts ready to be run.

## **A.2. AA and CG MD Trajectory.**

We run a 10 ps molecular dynamics (MD) equilibration with a 1 fs timestep on the system to relax the system components after packing them together into the simulation cell. All Molecular Dynamics (MD) simulations are performed using a set of programs in Biovia's Materials Studio, version 8 [Marrink, 2007], on a high-performance computing cluster. To setup this calculation, we go to 'Modules' then 'Forcite' and 'Calculation' in the top menu.

For equilibration calculations, the simulation cell is coupled to a Velocity Scale thermostat set to 298 K and a Berendsen barostat set to 1 atm, both of which operate with a relaxation time of 1 ps. For the all atom MD simulation, we utilize a universal forcefield that summates forces imposed on each bead every 1 fs via Verlet Integration. A cubic spline switching function truncates long-range, non-bonding potentials at 12.5 Å, where interactions between bead pairs separated by more than this distance are not calculated. The equilibration calculation is run on all 8 cores of a desktop computer. Once the cell equilibrates, we run our full-length molecular dynamics calculation under mostly the same conditions. Unlike the equilibration calculation, however, we couple the cell to a Nose thermostat, take the trajectory out to 40 ns, and run the calculation on an HPC.

Table A.1 defines the filenames required to run a 40 ns MG-defined AA MD trajectory (Chapter 3.1.3).

**Table A.1.** Files required to run a 40 ns AA MD trajectory.

Filename	Description
AAPFRu80.xtd	MG-defined AA MD Trajectory
<i>ForciteScript.pl</i>	Copied MS MD script that simulates motion in CG-defined AA model out to 40 ns
<i>submit.sh</i>	Submission command designating specific Dogwood partition to run MD script
<i>script.x</i>	Sets up communication between Dogwood and Materials Studio 18.1

Files for editing can be found under the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/C*

*luster/CGMDTraj(1)\_Dogwood.* Files to run can be found under the following path:

*/nas/longleaf/hom/mbalu/AA.*

The equilibrated 80mer MG-defined AA model (AAPFRu80.xtd) serves as input for the

*ForciteScript.pl* script defined below.

*ForciteScript.pl*

```
#!/perl
```

```
use strict;
```

```
use Getopt::Long;
```

```
use MaterialsScript qw(:all);
```

```
my $doc = $Documents{"AAPFRu70_1.xtd"};
```

```
my $results = Modules->Forcite->Dynamics->Run($doc, Settings(
```

```
    ChargeAssignment => "Use current",
```

```
Ensemble3D => "NPT",
Pressure => 0.0001,
NumberOfSteps => 10000000,
TimeStep => 1,
Thermostat => "Nose",
EnergyDeviation => 1e+006,
InitialVelocities => "Current",
StressXX => -0.0001,
StressYY => -0.0001,
StressZZ => -0.0001));
my $outTrajectory = $results->Trajectory;
```

### *submit.sh*

```
#!/bin/sh

#SBATCH -J AA70_1
#SBATCH -o my_output.%j
#SBATCH -n 80
#SBATCH -N 2
#SBATCH -t 7-00:00:00
#SBATCH --mem=200g
#SBATCH -p skylake
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=mbalu@live.unc.edu
./script.x ForciteScript
```

### *script.x*

```
#!/bin/sh

export WORKDIR=`pwd`
```

```
export DSD_QUEUED_RUN=1
export TASKS_PER_NODE=`expr $SLURM_NTASKS / $SLURM_NNODES`
scontrol show hostnames $SLURM_JOB_NODELIST|sed
s/\$/:$TASKS_PER_NODE/g > machines.LINUX
export DSD_MachineList=$WORKDIR/machines.LINUX
export DSD_NumProc=$SLURM_NTASKS
/nas/longleaf/apps/materialstudio/2018/MaterialsStudio18.1/etc/Scripti
ng/bin/RunMatScript.sh -np $DSD_NumProc $1
```

To submit the job, the following command prompts are required:

```
cd /nas/longleaf/home/CGMDTraj/AA (change directory)
chmod a+x submit.sh script.x (allow Dogwood to read scripts)
sbatch submit.sh (submit job)
squeue -u mbalu (check on job progress)
```

Table A.2 describes the files required to histogram angles selected for a particular bead set type in a 40 ns MG-defined AA MD trajectory (Section 3.1.3)

**Table A.2.** Files required to run a histogram analysis angles between a specific set of beads.

Filename	Description
BeadTypeSet.xlsx	All angles of BeadTypeSet saved across 40 ns MG-defined AA MD trajectory
<i>AngleAnalysis.m</i>	Matlab script that provides population densities as a function of equilibrium for all bead set types
<i>Histogram.m</i>	Matlab function that histograms angles extracted from AA MD simulation and plots population density as a function of wavelengths 0 – 360° (AngA)

Files to edit and run can be found in the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/AngleAnalysis.*

The Signal Processing Toolbox for Matlab must be installed before running. Codes for the files listed in Table A.2 are outlined below.

### *AngleAnalysis.m*

```
clear;
%Read each excel file containing Bead Set Type angles across the 40 ns
AA MD trajectoryprominence = .001;
x = {xlsread('ABC.xlsx') xlsread('BCD.xlsx') xlsread('CDE.xlsx')
xlsread('DED.xlsx') xlsread('DDE.xlsx') xlsread('DDD.xlsx')};
files = numel(x);

POPANG = cell(files,1);
PEAKNUM = cell(files,1);
ANGA = cell(files,1);
BEGANG = cell(files,1);
ENDANG = cell(files,1);
BW = cell(files,1);
DATA = cell(files,1);
%loop over all excel files
for i = 1:files
    Data = x{i};
    [POPAng,peaknum,AngA,BegAng,EndAng,bw] =
Histogram(Data,prominence);
    POPANG{i} = POPAng;
    PEAKNUM{i} = peaknum;
```

```

    ANGA{i} = AngA;
    BEGANG{i} = BegAng;
    ENDANG{i} = EndAng;
    BW{i} = bw;
    DATA{i} = Data;
end

```

### *Histogram.m*

```

function
[POPAng,peaknum,AngA,BegAng,EndAng,bw]=Histogram(Data,prominence)
    [y,ang,bw] = ksdensity(Data,0:1:180);
    [peaks,POPAng,~,~] =
findpeaks(y,ang,'MinPeakProminence',prominence,'Annotate','extents');
    peaknum = numel(peaks);
    histogram(Data);
    histfit(Data,[],'kernel');
    y = y';
    angl = ang';
    BegAng = angl(1);
    EndAng = angl(end);
    ang2 =360-angl;
    ANG = vertcat(angl,ang2);
    Y = vertcat(y,y);
    AngA = horzcat(ANG,Y);
end

```

The final frame of the equilibrated CG MD model (CGPFRu\_M\_1.xtd) serves as input for the first part of the 700 ns MD trajectory. Table A.3 defines the filenames required to run our first 350 ns CG MD trajectory.

**Table A.3.** Files required to run the first part of a 700 ns CG MD trajectory over 350 ns.

<b>Filename</b>	<b>Description</b>
CGPFRu_M_1.xtd (M = 1 – 7)	Part 1 of 80mer CG MD Trajectory M
<i>MesociteScript.pl</i>	MD script from MS with equilibrium trajectory and forcefield document as inputs
<i>newbeads_ang.off</i>	Forcefield document
<i>submit.sh</i>	Submission command designating specific Dogwood partition to run MD script
<i>script.x</i>	Sets up communication between Dogwood and Materials Studio 18.1

Files for editing can be found in the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Cluster*

*/CGMDTraj (1)\_Dogwood*. Files to run can be found under the following path:

*/nas/longleaf/hom/mbalu/N\_1*.

Codes for the files listed in Table A.3 are defined below.

*MesociteScript.pl*

```
#!/perl

use strict;
use Getopt::Long;
use MaterialsScript qw(:all);

my $doc = $Documents{"CGPFRu70_M_1.xtd"};

my $results = Modules->Mesocite->Dynamics->Run($doc, Settings(
    CurrentForcefield => "/newbeads_ang",
    Ensemble3D => "NPT",
    Pressure => 0.0001,
    NumberOfSteps => 43750000,
    TimeStep => 8,
    EnergyDeviation => 100000,
    InitialVelocities => "Current",
    StressXX => -0.0001,
    StressYY => -0.0001,
    StressZZ => -0.0001));
my $outTrajectory = $results->Trajectory;
```

*newbeads\_ang.off*

[See Tables 3.3 – 3.7]



### *submit.sh*

```
#!/bin/sh

#SBATCH -J CG70_M_1
#SBATCH -o my_output.%j
#SBATCH -n 80
#SBATCH -N 2
#SBATCH -t 7-00:00:00
#SBATCH --mem=200g
#SBATCH -p skylake
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=mbalu@live.unc.edu
```

```
./script.x MesociteScript
```

### *script.x*

[same as above]

To submit the job, the following command prompts are required:

```
cd /nas/longleaf/home/CGMDTraj/M_1 (change directory)
chmod a+x submit.sh script.x (allow Dogwood to read scripts)
sbatch submit.sh (submit job)
squeue -u onyen (check on job progress)
```

The resulting .xtd file represents the first half of the 700 ns motion simulation

(CGPFRu\_M\_1.xtd). We add this file to our Material Studio project, save the trajectory at its last frame (CGPFRu\_M\_2.xtd), and use it as input for the second part of the MD Trajectory. Table A.4 outlines the filenames required to run the second 350 ns CG MD trajectory.

**Table A.4.** Files required to run the second part of a 700 ns CG MD trajectory over 350 ns.

Filename	Description
CGPFRu_M_2.xtd (M = 1 – 6)	Part 2 of 80mer CG MD Trajectory M
<i>MesociteScript.pl</i>	MD script from MS with end of Part 1 trajectory and forcefield document as inputs
<i>newbeads_ang.off</i>	Forcefield document
<i>submit.sh</i>	Submission command designating specific Dogwood partition to run MD script
<i>script.x</i>	Sets up communication between Dogwood and Materials Studio 18.1

The files for editing can be found in the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Cluster/CGMDTraj(1)\_Dogwood.* Files to run can be found under the following path:

*/nas/longleaf/hom/mbalu/N\_2.*

Scripts from Table A.4 are defined below.

*MesociteScript.pl*

```
#!/perl

use strict;
use Getopt::Long;
use MaterialsScript qw(:all);

my $doc = $Documents{"CGPFRu70_M_1.xtd"};

my $results = Modules->Mesocite->Dynamics->Run($doc, Settings(
    CurrentForcefield => "/newbeads_ang",
    Ensemble3D => "NPT",
    Pressure => 0.0001,
    NumberOfSteps => 43750000,
    TimeStep => 8,
```

```
    EnergyDeviation => 100000,  
    InitialVelocities => "Current",  
    StressXX => -0.0001,  
    StressYY => -0.0001,  
    StressZZ => -0.0001));  
my $outTrajectory = $results->Trajectory;
```

*newbeads\_ang.off*

[See Tables 3.3 – 3.7]

*submit.sh*

```
#!/bin/sh  
  
#SBATCH -J CG70_M_2  
#SBATCH -o my_output.%j  
#SBATCH -n 80  
#SBATCH -N 2  
#SBATCH -t 7-00:00:00  
#SBATCH --mem=200g  
#SBATCH -p skylake  
#SBATCH --mail-type=END,FAIL  
#SBATCH --mail-user=mbalu@live.unc.edu  
  
./script.x MesociteScript
```

*script.x*

[Same as above]

To submit the job, the following command prompts are required:

```
cd /nas/longleaf/home/CGMDTraj/M_1 (change directory)  
chmod a+x submit.sh script.x (allow Dogwood to read scripts)
```

*sbatch submit.sh* (submit job)

*squeue -u onyen* (check on job progress)

## APPENDIX B: ENT SIMULATIONS CODE

### B.1. Ru Coordinate Extraction.

Table 6.1 outlines the files required to extract the desired XYZ complex coordinates.

**Table B.1.** Files required to extract the XYZ coordinates of the Ru complexes in the 80mer CG MD model.

Filename	Description
CGPFRu_M_N.xtd (M = 1 – 6, N = 1 or 2)	Part N of 80mer CG MD Trajectory N with sets “A” defined
<i>extractXYZ.pl</i>	Extracts XYZ coordinates of Ru bead sets “A” in order
<i>script.x</i>	Sets up communication between Dogwood and Materials Studio 18.1
<i>submit.sh</i>	Submission command designating specific Dogwood partition to run MD script

Perl files for editing can be found under the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/Custer/ExtractCoord (2)\_Dogwood/80mer\_Ru*. Files for running on Dogwood can be found under the following path: */nas/longleaf/home/mbalu/ExtractCoord*.

The scripts and embedded functions from Table B.1 are defined below.

```
extractXYZ.pl
```

```
#!/perl
```

```
use strict;
```

```

use MaterialsScript qw(:all);

#open the multiframe trajectory structure file or die
my $doc = $Documents{"./CGPFRu_M_N.xtd"};

if (!$doc) {die "no document";}

my $trajectory = $doc->Trajectory;

if ($trajectory->NumFrames>1) {
    # Open new report file
    my $report=Documents->New("CGPFRu70_Ru_M_N.txt");
    $report->Append("Found ".$trajectory->NumFrames." frames in the
trajectory\n");
    $report->Close;
    # Open new xmol trajectory file
    my $xmolFile=Documents->New("CGPFRu70_Ru_M_N_coords.txt");
    my $xmolFile_simple=Documents-
>New("CGPFRu70_Ru_7_2_coords_simple.txt");

    # loops over the frames
    for (my $frame=1; $frame<=$trajectory->NumFrames; ++$frame){
        $trajectory->CurrentFrame = $frame;
        #get beads in the structure
        for (my $q=0; $q<=39; ++$q){
            my $beads=$doc->UnitCell->Sets->Item($q);
            my $Nbeads=@$beads;
            #write header xyz
            $xmolFile->Append(sprintf "%i \n\n", $Nbeads);
            # write bead symbol and x-y-z- coordinates
            foreach my $bead (@$beads){

```

```

        $xmlFile->Append(sprintf "%s %f %f %f \n", $bead-
>BeadTypeName, $bead->X, $bead->Y, $bead->Z);
        $xmlFile_simple->Append(sprintf "%s %f %f \n", $bead-
>X, $bead->Y, $bead->Z);
    }
}
}
#close trajectory file
$xmlFile->Close;

}
else {
    print "The " . $doc->Name . " is not a multiframe trajectory file
\n";
}

```

### *submit.sh*

```

#!/bin/sh

#SBATCH -J XYZ70_M_N
#SBATCH -o my_output.%j
#SBATCH -n 88
#SBATCH -N 2
#SBATCH -t 3-00:00:00
#SBATCH --mem=200g
#SBATCH -p 528_queue
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=mbalu@live.unc.edu

./script.x extractXYZ

```

### *script.x*

[same as above]

To submit the job, the following command prompts are required:

```
cd /nas/longleaf/home/CGMDTraj/M_1 (change directory)
chmod a+x submit.sh script.x (allow Dogwood to read scripts)
sbatch submit.sh (submit job)
squeue -u onyen (check on job progress)
```

The output text files contain the XYZ coordinates of Ru beads in sets “A” across all frames in each of our CG MD trajectories (Table B.2).

**Table B.2.** Main output files of the *extractXYZ.pl* script.

Filename	Description
CGPFRu80_Ru_M_N.txt (N = 1 – 7, M = 1 or 2)	Number of frames in Part N of 80mer CG MD trajectory M
CGPFRu80_Ru_M_N_coords.txt	Lists coordinates of all Ru beads, identifies bead type and number that exist in in each frame (incl. spaces between frames)
CGPFRu80_Ru_M_N_coords_simple.txt	Lists coordinates of all Ru beads in each frame (no spaces between frames)

Files can be found under the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/Custer/ExtractCoord (2)\_Dogwood/80mer\_Ru.*

### **B.1.1. EnT Simulation Code.**

EnT simulations are run on Longleaf, another one of UNC’s HPCs. A Longleaf account that can be obtained through ITS. Similar to jobs submitted to Dogwood, users may use SSH Secure Shell Client or another terminal program to log into their longleaf account with the following command: *ssh onyen@longleaf.unc.edu* and enter user onyen password when prompted. SSH Secure File Transfer Client holds all input and output files of a job. The user’s home directory still exists at the following path: */nas/longleaf/home/onyen.*



Along with the other files listed in Table B.3, our *ENT\_SIMULATION\_PARFOR.m* function simulates EnT between those complex beads as they move around from frame to frame (Section 4.2).

**Table B.3.** Files required to run EnT simulations on 70mer CG MD models.

Filename	Description
CGPFRu_M_N.xlsx (n = 1 – 7)	Extracted XYZ Ru coordinates from Part N of the 80mer CG MD Trajectory M
<i>ENT_SIMULATION_PARFOR.m</i>	Parallel-computing enabled EnT Simulation function
<i>DISTANCE_MATRIX.m</i>	Uses distance function to calculate time-dependent distances between all Ru-Ru and Ru-Os complex pairs
<i>KINPOP.m</i>	Function containing <i>KINETIC_MATRIX.m</i> , <i>POPULATION.m</i> , and <i>STATIC_POPULATION.m</i>
<i>KINETIC_MATRIX.m</i>	Calculates time-dependent kinetic matrices from the distance matrices and diagonalizes each kinetic matrix
<i>POPULATION.m</i>	Calculates time-dependent populations of Ru* and Os* in systems where pendants remain flexible
<i>STATIC_POPULATION.m</i>	Calculates time-dependent populations of Ru* and Os* in systems where pendants are treated statically
<i>OS_WEIGHT.m</i>	Calculates probability of each Os loading scenario
<i>submit.sh</i>	Submission command designating specific Dogwood partition to run EnT Simulation function

Matlab files for editing can be found on the following path:

*PapanikolasGroup/Workspace/Bowers/PFRuOs\_CGMDSimulations/Code\_LMRBMac/Cluster/E*

*nTSim (3)\_Longleaf*. Files for running on Longleaf can be found under the following path:

*/nas/longleaf/home/mbalu/EnTSim*.

The scripts and embedded functions from Table B.3 are defined below.

*ENT\_SIMULATION\_PARFOR.m*.

```

clear;
tic;

%set up calculation with 24 processors on parallel computer
myCluster = parcluster('local');
myCluster.NumWorkers = 24;
saveProfile(myCluster);
parpool(24);
%-----USER DEFINED PARAMETERS-----%
%%select files of different PF-RuOs CG MD trajectories%%
TrajRu = transpose({xlsread('CGPF-Ru70_3_1.xlsx') xlsread('CGPF-
Ru70_3_2.xlsx') xlsread('CGPF-Ru70_4_1.xlsx') xlsread('CGPF-
Ru70_4_2.xlsx') xlsread('CGPF-Ru70_5_1.xlsx') xlsread('CGPF-
Ru70_5_2.xlsx') xlsread('CGPF-Ru70_6_1.xlsx') xlsread('CGPF-
Ru70_6_2.xlsx') xlsread('CGPF-Ru70_7_1.xlsx') xlsread('CGPF-
Ru70_7_2.xlsx') xlsread('CGPF-Ru70_10_1.xlsx') xlsread('CGPF-
Ru70_10_2.xlsx')});
trajRu = {cat(1,TrajRu{1:2}) cat(1,TrajRu{3:4}) cat(1,TrajRu{5:6})
cat(1,TrajRu{7:8}) cat(1,TrajRu{9:10}) cat(1,TrajRu{11:12})}';
%number of Ru coordinates actually extracted (4 unit cells, each
containing one 20mer)
PolyCount = 80;
%Ru coordinates in 70mer
RuCount = 70;
%each trajectory in trajRu has 17502 frames
FramesPerTraj = 17502;
%each trajectory in trajRu is 700 ns
TrajTime = 700;
%number of Ru/Os Configurations for each Os loading scenario
configurations = 10;
%rates of closest approach and decay (ns^-1) and attenuation parameter
(A^-1)
ConstantArray = [k0Ru k0Os kemRu kemOs B]';
%probability Os will be loaded with given fraction of homopolymer
p_os = [0.144300144 0.146520147 0.148809524 0.151975684 0.158730159
0.162337662];
%Fraction PF-Ru70 homopolymer..Mostly at 0.06 but may readjust
f_hp = [0.01 0.025 0.04 0.06 0.1 0.12];
p_os = p_os(4);
f_hp = f_hp(4);
%number of Os loading scenarios...up to this many Os can be loaded
Trials = 17;
%LJ Ru bead radius used for calculating periphery-periphery separation
RuRadius = (2^(1/6))*6.53325;
% %----END USER DEFINED PARAMETERS-----%
r = length(trajRu);
TotalFrameCount = FramesPerTraj*r;
T = TrajTime*r;
timestep = T/TotalFrameCount;
times = 0:timestep:T-timestep;
times = times';

```

```

%create Time variable with 40 ps time steps (MS has 8 fs steps saved
every 5000th frame)
Time = times(1:FramesPerTraj);
%structure loaded with N Os (N = 0-17)
x = [0:Trials]';
%%Variable Set up%%
S = cell(length(x),1);
I_o = cell(length(x),1);
GS_Os = cell(length(x),1);
I_o_static = cell(length(x),1);
GS_Os_static = cell(length(x),1);
I_r = cell(length(x),1);
I_r_static = cell(length(x),1);
IO = cell(length(x),r);
IR = cell(length(x),r);
GSO = cell(length(x),r);
GSR = cell(length(x),r);
IO_static = cell(length(x),r);
IR_static = cell(length(x),r);
GSO_static = cell(length(x),r);
DIFF3D_ = cell(length(x),r);
%loop over six 700 ns CG MD trajectories
for j = 1:r
    Ru = trajRu{j};
    Coord2DRu = mat2cell(Ru,PolyCount*ones(FramesPerTraj,1),3);
    %calculate time-dependent distance matrices and discount 5 complex
coordinates at each end to create true 70mer %%
    [Dist3D,Coord2D_Ru_R] =
DISTANCE_MATRIX(Coord2DRu,RuRadius,FramesPerTraj,RuCount);
    %use time-dependent distance matrices to calculate time dependent
kinetic matrices
    %for all possible initial excitations for one Ru/Os Configuration
in one
    %Os loading scenario...Loop over all Os loading scenarios%%
    for i = 1:18
        [~,Ios,Iru,GSOs,GSRu,Ios_static,Iru_static,GSOs_static,DIFF3D]
=
KINPOP(RuCount,j,i,x,configurations,Dist3D,FramesPerTraj,timestep,Cons
tantArray);
        IO{i,j} = mean(horzcat(Ios{:}),2);
        IR{i,j} = mean(horzcat(Iru{:}),2);
        GSO{i,j} = mean(horzcat(GSOs{:}),2);
        IO_static{i,j} = mean(horzcat(Ios_static{:}),2);
        IR_static{i,j} = mean(horzcat(Iru_static{:}),2);
        GSO_static{i,j} = mean(horzcat(GSOs_static{:}),2);
        DIFF3D_{i,j} = mean(horzcat(DIFF3D{:}),2);
    end
end
%calculate weighting for each Os loading scenario
[x_,check] = OS_WEIGHT(RuCount,Trials,p_os,f_hp);
%%average excited state (ES) populations across trajectories for each
Os loading scenario

```

```

%%- avg across rows in opened cells from above%%
for m = 1:18
    I_o{m} = mean(horzcat(IO{m,:}),2);
    I_r{m} = mean(horzcat(IR{m,:}),2);
    GS_Os{m} = mean(horzcat(GSO{m,:}),2);

    I_o_static{m} = mean(horzcat(IO_static{m,:}),2);
    I_r_static{m} = mean(horzcat(IR_static{m,:}),2);
    GS_Os_static{m} = mean(horzcat(GSO_static{m,:}),2);
end
%%Weighted average of ES populations across Os loading scenarios with
%%weights x%%
I_o__ = horzcat(I_o{:});
I_o_w_ = sum(x_'.*I_o__,2)./sum(x_',2);
I_o_w_ = real(I_o_w_);

I_r__ = horzcat(I_r{:});
I_r_w_ = sum(x_'.*I_r__,2)./sum(x_',2);
I_r_w_ = real(I_r_w_);

GS_Os__ = horzcat(GS_Os{:});
GS_Os_w_ = sum(x_'.*GS_Os__,2)./sum(x_',2);
GS_Os_w_ = real(GS_Os_w_);
%normalize peaks to 1
I_o_wnorm = I_o_w_ ./ max(I_o_w_);
I_r_wnorm = (I_r_w_ ./ max(I_r_w_));
%%MAIN OUTPUT VAR: Measure Os ES population in flexible case
OsPOP_norm = horzcat(Time,I_o_wnorm);
RuPOP_norm = horzcat(Time,I_r_wnorm);
GSOsPOP = horzcat(Time,GS_Os_w_);

%%STATIC%%
I_o_static__ = horzcat(I_o_static{:});
I_o_static_w_ = sum(x_'.*I_o_static__,2)./sum(x_',2);
I_o_static_w_ = real(I_o_static_w_);

I_r_static__ = horzcat(I_r_static{:});
I_r_static_w_ = sum(x_'.*I_r_static__,2)./sum(x_',2);
I_r_static_w_ = real(I_r_static_w_);

GS_Os_static__ = horzcat(GS_Os_static{:});
GS_Os_static_w_ = sum(x_'.*GS_Os_static__,2)./sum(x_',2);
GS_Os_static_w_ = real(GS_Os_static_w_);
%normalize peaks to 1
I_o_static_w_norm = I_o_static_w_ ./ max(I_o_static_w_);
I_r_static_w_norm = (I_r_static_w_ ./ max(I_r_static_w_));
%%MAIN OUTPUT VAR: Measure Ru ES population in static case
RuPOP_static_norm = horzcat(Time,I_r_static_w_norm);
OsPOP_static_norm = horzcat(Time,I_o_static_w_norm);
GSOsPOP_static = horzcat(Time,GS_Os_static_w_);

%save all main output variables

```

```
save('70mer_B15_RuSets.mat','OsPOP_norm','OsPOP_static_norm','GSOsPOP'
,'GSOsPOP_static','RuPOP_norm','RuPOP_static_norm','Time','check','DIF
F3D_');
```

```
clear;
%document calculation time
toc;
timerVal = tic;
```

### *DISTANCE\_MATRIX.m.*

```
function [Dist3D,Coord2D_Ru_R] =
DISTANCE_MATRIX(Coord2DRu,RuRadius,FramesPerTraj,RuCount)
Dist3D = cell(FramesPerTraj,1);
Coord2D_Ru_R = cell(FramesPerTraj,1);
RuRu = zeros(RuCount,RuCount);
%loop over all frames
for m = 1:FramesPerTraj
    Coord2D_Ru_f = Coord2DRu{m};
    %choose coordinates to create true 70mer CG MD model
    Coord2D_Ru_frame = Coord2D_Ru_f(5:74,:);
    %loop over all complex pairs
    for k = 1:RuCount
        for i = 1:k
            Xk = Coord2D_Ru_frame(k,1);
            Xi = Coord2D_Ru_frame(i,1);
            Yk = Coord2D_Ru_frame(k,2);
            Yi = Coord2D_Ru_frame(i,2);
            Zk = Coord2D_Ru_frame(k,3);
            Zi = Coord2D_Ru_frame(i,3);
            %distance formula
            distance = sqrt((Xk-Xi)^2+(Yk-Yi)^2+(Zk-Zi)^2);
            if Xk == Xi&&Yk == Yi&&Zk == Zi
                distance = inf;
            end
            %calculate periphery-periphery distances between complexes
            RuRu(k,i) = distance-(2*RuRadius);
            RuRu(i,k) = distance-(2*RuRadius);
        end
    end
    %save 70mer coordinates and distances
    Coord2D_Ru_R{m} = Coord2D_Ru_frame;
    Dist3D{m} = RuRu;
end
end
```

### *KINPOP.m.*

```
function
[ST,Ios,Iru,GSOs,GSRu,Ios_static,Iru_static,GSOs_static,DIFF3D] =
```

```

KINPOP(RuCount,j,i,x,structures,Dist3D,FramesPerTraj,TS_,ConstantArray
)
InitialArray = zeros(RuCount+2,1);
%excite all complexes
InitialArray(1:RuCount) = 1;
%%variable setup%%
Iru = cell(structures,1);
Ios = cell(structures,1);
GSOs = cell(structures,1);
GSRu = cell(structures,1);
Ios_static = cell(structures,1);
Iru_static = cell(structures,1);
GSOs_static = cell(structures,1);
DIFF3D = cell(structures,1);
%loop over number of Ru/Os configurations for each Os loading scenario
for s = 1:structures
    ST = zeros(RuCount,1);
    %x = Os loading scenario and x indices randomly given 1 are "Os"
    and 0 are "Ru"
    ST(randperm(numel(ST),x(i))) = 1;
    %calculate time-dependent, diagonalized kinetic matrices from
    time-dependent distance matrices
    [khop3D_d,Q_,khop3D_dM,Q_M] =
KINETIC_MATRIX(Dist3D,FramesPerTraj,RuCount,TS_,ST,ConstantArray);
    %display calculation status - which trajectory, Os loading
    scenario and Ru/Os configuration
    A = [j i s];
    disp(A)
    %calculate ES population over time for given Ru/Os configuration
    in systems with flexible pendants
    [Iru_a,Ios_a,GSRu_a,GSOs_a,Diff3D] =
POPULATION(FramesPerTraj,RuCount,InitialArray,ST,khop3D_d,Q_);
    %calculate ES population over time for given Ru/Os configuration
    in systems with static pendants
    [Iru_a_static,Ios_a_static,~,GSOs_a_static] =
STATIC_POPULATION(FramesPerTraj,RuCount,InitialArray,ST,khop3D_dM,Q_M)
;
    %collect time-dependent Ru and Os ES and GS populations for each
    Ru/Os
    %configuration in flexible and static cases
    Iru{s} = Iru_a;
    Ios{s} = Ios_a;
    GSOs{s} = GSOs_a;
    GSRu{s} = GSRu_a;
    DIFF3D{s} = Diff3D;

    Ios_static{s} = Ios_a_static;
    Iru_static{s} = Iru_a_static;
    GSOs_static{s} = GSOs_a_static;
end
end

```

### *KINETIC\_MATRIX.m.*

```
function [khop3D_d,Q_,khop3D_dM,Q_M] =
KINETIC_MATRIX(Dist3D,FramesPerTraj,RuCount,TS_,ST,ConstantArray)
%PendantCount+2 to include GSRu and GSOs
khop = zeros(RuCount+2,RuCount+2);
khop3D_d = cell(FramesPerTraj,1);
Q_ = cell(FramesPerTraj,1);
%%loop over distance matrices at each time frame and input into Dexter
EnT equation%%
for i = 1:FramesPerTraj
    Dist = Dist3D{i};
    %Ru to Ru, k0Ru and B
    khop(ST==0,ST==0) = ConstantArray(1)*exp(-
ConstantArray(5)*(Dist(ST==0,ST==0)));
    %Ru to Os, k0Os and B
    khop(ST==1,ST==0) = ConstantArray(2)*exp(-
ConstantArray(5)*(Dist(ST==1,ST==0)));
    %Os ES cant go to Ru
    khop(ST==0,ST==1) = 0;
    %no ES transfer between any Os
    khop(ST==1,ST==1) = 0;
    %GS population by Ru, ES decay
    khop(end-1,ST==0) = ConstantArray(3);
    %GS population by Os, ES decay
    khop(end,ST==1) = ConstantArray(4);
    %loop over diagonal elements - each Ru* can decay by
multiple pathways, Os* decays to GS
    for x = 1:RuCount
        khop(x,x) = -(sum(khop(:,x)));
    end
    %Q returns the eigenvector matrix, D is the eigenvalue matrix
    [Q,D] = eig(khop);
    %Calculate matrix exponential of diagonalized kinetic matrix *
timestep (40 ps) at each frame
    khop3D_d{i} = expm(D*TS_);
    %Save eigenvector matrix at each frame
    Q_{i} = Q;
end
%Save diagonalized kinetic matrix and eigenvector matrix at Frame 1
for static population calculations
khop3D_dM = khop3D_d{1};
Q_M = Q_{1};
end
```

### *POPULATION.m.*

```
function [Iru_a,Ios_a,GSRu_a,GSOs_a,Diff3D] =
POPULATION(FramesPerTraj,RuCount,InitialArray,ST,khop3D_d,Q_)
    Iru_a = zeros(FramesPerTraj,1);
    Ios_a = zeros(FramesPerTraj,1);
```

```

GSRu_a = zeros(FramesPerTraj,1);
GSOs_a = zeros(FramesPerTraj,1);
Diff3D = zeros(FramesPerTraj,1);
%loop over frames to open kinetic matrix
for j = 1:FramesPerTraj
    Q = Q_{j};
    khop3D = khop3D_d{j};
    %khop3D is diagonalized,D,72x72 matrix at Frame
j...expm(D*TS_)
    soln = (Q*khop3D/Q);
    %kinetics * initial population gives final population at each
time step
    pop_f = soln*InitialArray;
    %along 72 rows (single column), get sum of all Ru and Os ES
and GS populations at each time point
    %divide by RuCount because exciting all 70complexes at once
    Iru_a(j) = sum(pop_f(ST==0,1))/RuCount;
    Ios_a(j) = sum(pop_f(ST==1,1))/RuCount;
    GSRu_a(j) = pop_f(RuCount+1,1)/RuCount;
    GSOs_a(j) = pop_f(RuCount+2,1)/RuCount;
    %propagate ES population - let final population become new
initial population
    InitialArray = pop_f;
    %total population should be conserved (1)...checking to see
how
    %far off the sum is at the end and difference should be ~0
    if sum(pop_f) ~= RuCount
        Diff3D(j) = abs(RuCount - sum(pop_f));
    else
        Diff3D(j) = 0;
    end
end
end
end

```

### *STATIC\_POPULATION.m.*

```

function [Iru_a_static,Ios_a_static,GSRu_a_static,GSOs_a_static] =
STATIC_POPULATION(FramesPerTraj,RuCount,InitialArray,ST,khop3D_dM,Q_M)
    Iru_a_static = zeros(FramesPerTraj,1);
    Ios_a_static = zeros(FramesPerTraj,1);
    GSRu_a_static = zeros(FramesPerTraj,1);
    GSOs_a_static = zeros(FramesPerTraj,1);
    %use kinetic matrix at first frame of selected range and use that
for all time steps...static
    %khop3D is diagonalized,D,72x72 matrix at Frame 1
    soln = (Q_M*khop3D_dM/Q_M);
    %loop over all frames
    for j = 1:FramesPerTraj
        %kinetics * initial population gives final population at each
time step
        pop_f = soln*InitialArray;

```



```

        %propagate ES population - let final population become new
initial population
        InitialArray = pop_f;
        %along 72 rows (single column), get sum of all Ru and Os ES
and GS populations at each time point
        %divide by RuCount because exciting all 70 complexes at once
        Iru_a_static(j) = sum(pop_f(ST==0,1))/RuCount;
        Ios_a_static(j) = sum(pop_f(ST==1,1))/RuCount;
        GSRu_a_static(j) = pop_f(RuCount+1,1)/RuCount;
        GSOs_a_static(j) = pop_f(RuCount+2,1)/RuCount;
    end
end

```

*submit.sh.*

```
#!/bin/sh
```

```
#SBATCH -J PEnT_Bkk
```

```
#SBATCH -o my_output.%j
```

```
#SBATCH -n 24
```

```
#SBATCH -N 1
```

```
#SBATCH -t 3-00:00:00
```

```
#SBATCH --mem=250g
```

```
#SBATCH -p general
```

```
#SBATCH --mail-type=END,FAIL
```

```
#SBATCH --mail-user=mbalu@live.unc.edu
```

```
matlab -nodesktop -nosplash -singleCopThread -r ENT_SIMULATION_PARFOR
-logfile ENT_SIMULATION_PARFOR.out
```

To submit the job, the following command prompts are required:

```
cd /nas/longleaf/home/CGMDTraj/M_1 (change directory)
```

```
chmod a+x submit.sh script.x (allow Dogwood to read scripts)
```

```
sbatch submit.sh (submit job)
```

```
squeue -u onyen (check on job progress)
```

## APPENDIX C: COMPUTATIONAL EXPERIMENTS CODE

### C.1. Ru Coordinate Extraction.

Along with the other files listed in Table C.1, our *NNsAnalysis.m* function histograms distances between all complexes and their updated NNs and NNNs over 350 ns (Section 3.2.1).

**Table C.1.** Files required to run NN distance calculations on 70mer CG MD models.

Filename	Description
<i>NNsAnalysis.m</i>	Histograms distances between all complexes and their updated NNs and NNNs over 350 ns
<i>NNsDIST.m</i>	Calculates NN and NNN distances every 0.2 ns and averages over 350 ns intervals in six different 700 ns MD trajectories

Matlab files can be found on the following path: *PapanikolasGroup/Workspace/Bowers/PF-RuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/NNsAnalysis*.

Scripts and embedded functions from Table C.1 are described below.

#### *NNsAnalysis.m*

```
clear;
tic;
%-----USER DEFINED PARAMETERS-----%
%%%select files of different PF-Ru CG MD trajectories%%
TrajRu = transpose({xlsread('CGPF-Ru70_3_1.xlsx') xlsread('CGPF-
Ru70_3_2.xlsx') xlsread('CGPF-Ru70_4_1.xlsx') xlsread('CGPF-
Ru70_4_2.xlsx') xlsread('CGPF-Ru70_5_1.xlsx') xlsread('CGPF-
Ru70_5_2.xlsx') xlsread('CGPF-Ru70_6_1.xlsx') xlsread('CGPF-
Ru70_6_2.xlsx') xlsread('CGPF-Ru70_7_1.xlsx') xlsread('CGPF-
Ru70_7_2.xlsx') xlsread('CGPF-Ru70_10_1.xlsx') xlsread('CGPF-
Ru70_10_2.xlsx')});
```

```

trajRu = {cat(1,TrajRu{1:2}) cat(1,TrajRu{3:4}) cat(1,TrajRu{5:6})
cat(1,TrajRu{7:8}) cat(1,TrajRu{9:10}) cat(1,TrajRu{11:12})}';
PolyCount = 80;
RuCount = 70;
FramesPerTraj = 17502;
TrajTime = 700;
timestep = round((TrajTime/FramesPerTraj),2,'significant');
times = [0:timestep:TrajTime]';
%skip timeframes to get 0.2 ns spacing between frames
timeframes = 5;
timeintervals = timeframes*timestep;
FramesPerSection=350/timeintervals;
endFPS = timeframes*FramesPerSection;
TimeSection = times(1:timeframes:endFPS);
T0 = 5;
% %----END USER DEFINED PARAMETERS-----%
r = length(trajRu);
M = [T0:timeframes:(endFPS+T0)-timeframes]';
%%Calculate time dependent NN and NNN distances
[DIST3D_NN_traj] =
NNsDIST(trajRu,FramesPerTraj,FramesPerSection,M,r,PolyCount,RuCount);
NNs = 30;
DIST3D_NNs = cell(NNs,1);
%loop over all NN#
for i = 1:NNs
    %rows = NNs, columns = #t0s*FramesPerSection
    DIST3D_NNs{i} = horzcat(DIST3D_NN_traj{i,:});
end
DIST3D_NNS = zeros(735000,NNs);
%loop over all trajectories
for j = 1:NNs
    DIST3D_NNs_ = DIST3D_NNs{j};
    DIST3D_NNS(:,j) = reshape(DIST3D_NNs_,[735000,1]);
end
save('NNsAnalysis_6T_350ns.mat','timeframes','TimeSection','m','DIST3D
_NNS');

toc;
timerVal = tic;

```

### *NNsDIST.m*

```

function [DIST3D_NN_traj] =
NNsDIST(trajRu,FramesPerTraj,FramesPerSection,M,r,PolyCount,RuCount)
RuRu = zeros(RuCount,RuCount);
Dist3D = cell(FramesPerTraj,1);
Dist3D_nn_ = zeros(RuCount,FramesPerSection);
DIST3D_NN_ = cell(RuCount,1);
DIST3D_NN= cell(r,1);
%%loop over all trajectories%%
for j = 1:r

```

```

Ru = trajRu{j};
%organize 70mer coordinates into frames
Coord2DRu = mat2cell(Ru, PolyCount*ones(FramesPerTraj,1),3);
%loop over all frames
for m = 1:FramesPerTraj
    Coord2D_Ru_f = Coord2DRu{m};
    %choose coordinates to create true 70mer CG MD model
    Coord2D_Ru_frame = Coord2D_Ru_f(5:74,:);
    %loop over all complex pairs
    for k = 1:RuCount
        for i = 1:k
            Xk = Coord2D_Ru_frame(k,1);
            Xi = Coord2D_Ru_frame(i,1);
            Yk = Coord2D_Ru_frame(k,2);
            Yi = Coord2D_Ru_frame(i,2);
            Zk = Coord2D_Ru_frame(k,3);
            Zi = Coord2D_Ru_frame(i,3);
            distance = sqrt((Xk-Xi)^2+(Yk-Yi)^2+(Zk-Zi)^2);
            if Xk == Xi&&Yk == Yi&&Zk == Zi
                distance = inf;
            end
            %calculate center-center distances between complexes
            RuRu(k,i)= distance;
            RuRu(i,k)= distance;
        end
        Dist3D{m} = RuRu;
    end
end
%loop over all Ru complexes
for NN = 1:RuCount
    %loop over all starting times, T0, in M
    for frame = 1:length(M)
        RuRu = Dist3D{M(frame)};
        %loop over each complex pair of NN
        for NNPair = 1:RuCount
            %sort RuRu distances for each complex from least to
greatest
            M_ = sort(RuRu(:,NNPair));
            %save NN in first row, NNN in second,etc for each
frame
            %(columns)
            Dist3D_nn_(NNPair,frame)= M_(NN);
        end
    end
    %save all NN# at each frame for each complex
    DIST3D_NN_{NN} = Dist3D_nn_;
end
%save for each trajectory
DIST3D_NN{j} = DIST3D_NN_;
end
DIST3D_NN_traj = horzcat(DIST3D_NN{:});
End

```

### C.1.1. Follow Original NN and NNN Analysis.

Along with the other files listed in Table C.2, our *FollowOrigNN.m* function histograms distances between all complexes and their original NNs and NNNs over 350 ns (Section 3.2.2).

**Table C.2.** Files required to run original NN and NNN distance calculations on 70mer CG MD models.

Filename	Description
<i>FollowOrigNN.m</i>	Histograms distances between all complexes and their original NNs and NNNs over 350 ns
<i>FollowOrigNN_DIST.m</i>	Calculates NN and NNN distances every 0.2 ns. Averages over 350 ns intervals and 13 different starting structures in six different 700 ns MD trajectories

Matlab files can be found on the following path: *PapanikolasGroup/Workspace/Bowers/PF-RuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/FollowOrigNN.*

Scripts and embedded functions from Table C.2 are described below

#### *FollowOrigNN.m*

```
clear;
tic;
%-----USER DEFINED PARAMETERS-----%
%%select files of different PF-Ru CG MD trajectories%%
TrajRu = transpose({xlsread('CGPF-Ru70_3_1.xlsx') xlsread('CGPF-
Ru70_3_2.xlsx') xlsread('CGPF-Ru70_4_1.xlsx') xlsread('CGPF-
Ru70_4_2.xlsx') xlsread('CGPF-Ru70_5_1.xlsx') xlsread('CGPF-
Ru70_5_2.xlsx') xlsread('CGPF-Ru70_6_1.xlsx') xlsread('CGPF-
Ru70_6_2.xlsx') xlsread('CGPF-Ru70_7_1.xlsx') xlsread('CGPF-
Ru70_7_2.xlsx') xlsread('CGPF-Ru70_10_1.xlsx') xlsread('CGPF-
Ru70_10_2.xlsx')});
trajRu = {cat(1,TrajRu{1:2}) cat(1,TrajRu{3:4}) cat(1,TrajRu{5:6})
cat(1,TrajRu{7:8}) cat(1,TrajRu{9:10}) cat(1,TrajRu{11:12})}';
PolyCount = 80;
RuCount = 70;
FramesPerTraj = 17502;
TrajTime = 700;
timestep = round((TrajTime/FramesPerTraj),2,'significant');
times = [0:timestep:TrajTime]';
%skip timeframes to get 0.2 ns spacing between frames
timeframes = 5;
```

```

timeintervals = timeframes*timestep;
FramesPerSection= 350/timeintervals;
endFPS = timeframes*FramesPerSection;
TimeSection = times(1:timeframes:endFPS);
%starting frame,T0, @~0:25:300 ns
T0 = [5 630 1250 1875 2500 3125 3750 4375 5000 5625 6250 6875 7500]';
% %----END USER DEFINED PARAMETERS-----%
r = length(trajRu);
DIST3D_NN_t0 = cell(r,1);
DIST3D_NNN_t0 = cell(r,1);
%%Follow original NN and NNN pairs to see how their distances evolve
over time
[~,DIST3D_NN,DIST3D_NNN] =
FollowOrigNNDIST(trajRu,timeframes,endFPS,T0,r,FramesPerTraj,FramesPer
Section,PolyCount,RuCount);
%loop over each trajectory
for j = 1:r
    DIST3D_NN_t0{j} = horzcat(DIST3D_NN{j}{:});
    DIST3D_NNN_t0{j} = horzcat(DIST3D_NNN{j}{:});
end
% horz cat framecount x (RuCount*r) of all traj
DIST3D_NN_ = horzcat(DIST3D_NN_t0{:})';
DIST3D_NNN_ = horzcat(DIST3D_NNN_t0{:})';

DIST3D_NN_TIME = horzcat(DIST3D_NN_(:,1), DIST3D_NN_(:,2),
DIST3D_NN_(:,3), DIST3D_NN_(:,4), DIST3D_NN_(:,5), DIST3D_NN_(:,6),
DIST3D_NN_(:,10), DIST3D_NN_(:,50),
DIST3D_NN_(:,500),DIST3D_NN_(:,1750));
DIST3D_NNN_TIME = horzcat(DIST3D_NNN_(:,1), DIST3D_NNN_(:,2),
DIST3D_NNN_(:,3), DIST3D_NNN_(:,4),DIST3D_NNN_(:,5), DIST3D_NNN_(:,6),
DIST3D_NNN_(:,10), DIST3D_NNN_(:,50), DIST3D_NNN_(:,500),
DIST3D_NNN_(:,1750));

save('FollowOrigNNAnalysis_6T_350ns.mat','timeframes','TimeSection','D
IST3D_NN_','DIST3D_NNN_','DIST3D_NN_TIME','DIST3D_NNN_TIME');

toc;
timerVal = tic;

```

### *FollowOrigNNDIST.m*

```

function [DIST3D,DIST3D_NN,DIST3D_NNN] =
FollowOrigNNDIST(trajRu,timeframes,endFPS,T0,r,FramesPerTraj,FramesPer
Section,PolyCount,RuCount)
RuRu = zeros(RuCount,RuCount);
NN = zeros(1,RuCount);
NNN = zeros(1,RuCount);
Dist3D_NN = zeros(FramesPerSection,RuCount);
Dist3D_NNN = zeros(FramesPerSection,RuCount);
Dist3D_NN_t0 = cell(length(T0),1);
Dist3D_NNN_t0 = cell(length(T0),1);

```

```

Dist3D = cell(FramesPerTraj,1);
DIST3D_NN = cell(r,1);
DIST3D_NNN = cell(r,1);
DIST3D_ = cell(r,1);
%%loop over all trajectories%%
for j = 1:r
    Ru = trajRu{j};
    Coord2DRu = mat2cell(Ru,PolyCount*ones(FramesPerTraj,1),3);
    %loop over all frames
    for m = 1:FramesPerTraj
        Coord2D_Ru_f = Coord2DRu{m};
        %choose coordinates to create true 70mer CG MD model
        Coord2D_Ru_frame = Coord2D_Ru_f(5:74,:);
        %loop over all complex pairs
        for k = 1:RuCount
            for i = 1:k
                Xk = Coord2D_Ru_frame(k,1);
                Xi = Coord2D_Ru_frame(i,1);
                Yk = Coord2D_Ru_frame(k,2);
                Yi = Coord2D_Ru_frame(i,2);
                Zk = Coord2D_Ru_frame(k,3);
                Zi = Coord2D_Ru_frame(i,3);
                distance = sqrt((Xk-Xi)^2+(Yk-Yi)^2+(Zk-Zi)^2);
                if Xk == Xi&&Yk == Yi&&Zk == Zi
                    distance = inf;
                end
                %calculate center-center distances between complexes
                RuRu(k,i)= distance;
                RuRu(i,k)= distance;
            end
        end
        Dist3D{m} = RuRu;
    end
    %loop over all starting times, T0
    for t = 1:length(T0)
        M = [T0(t):timeframes:(endFPS+T0(t))-timeframes]';
        %%find orig NN and NNN pairs at each t0%%
        Dist3D_t0 = Dist3D{M(1)};
        [~,NNInd] = sort(Dist3D_t0);
        NN(1,:) = NNInd(1,:);
        NNN(1,:) = NNInd(2,:);
        %save original NN and NNN pairs
        NN_FIXED = horzcat([1:RuCount]',NN(1,:)');
        NNN_FIXED = horzcat([1:RuCount]',NNN(1,:)');
        %%track distances between those orig NN and NNN pairs over time%%
        for k = 1:length(M)
            for n = 1:RuCount
                Dist3D_ = Dist3D{M(k)};
                Dist3D_NN(k,n) = Dist3D_(NN_FIXED(n,1),NN_FIXED(n,2));
                Dist3D_NNN(k,n) = Dist3D_(NNN_FIXED(n,1),NNN_FIXED(n,2));
            end
        end
    end
end
end

```

```

        Dist3D_NN_t0{t} = Dist3D_NN;
        Dist3D_NNN_t0{t} = Dist3D_NNN;
    end
%save original NN and NNN pair distances over time for all T0 and traj
DIST3D_NN{j} = Dist3D_NN_t0;
DIST3D_NNN{j} = Dist3D_NNN_t0;
DIST3D{j} = Dist3D;
end
end

```

### C.1.2. Excited State Diffusion.

Along with the other files listed in Table C.3, our *RU\_DIFFUSION.m* function initially excites a central Ru bead and tracks ES population on Ru beads out to  $\pm 30$  away over 350 ns (Section 4.3.1).

**Table C.3.** Files required to run diffusion calculations on 70mer CG MD models.

Filename	Description
CGPFRu80_N_M.xlsx (n = 1 – 7)	Extracted XYZ Ru coordinates from Part M of the 80mer CG MD Trajectory N
<i>RU_DIFFUSION.m</i>	Initially excites central Ru bead and tracks ES population on Ru beads out to $\pm 30$ away over 350 ns
<i>DISTANCE_MATRIX_DIFF.m</i>	Uses distance function to calculate time-dependent distances between all Ru-Ru and Ru-Os complex pairs
<i>KINETIC_MATRIX_DIFF.m</i>	Calculates time-dependent kinetic matrices from the distance matrices and diagonalizes each kinetic matrix
<i>POPULATION_DIFF.m</i>	Calculates time-dependent populations of Ru* $\pm 30$ away from the initially excited Ru in flexible systems
<i>STATIC_POPULATION_DIFF.m</i>	Calculates time-dependent populations of Ru* $\pm 30$ away from the initially excited Ru in static systems

Matlab files can be found on the following path: *PapanikolasGroup/Workspace/Bowers/PF-RuOs\_CGMDSimulations/Code\_LMRBMac/Matlab/Diffusion.*

Scripts and embedded functions from Table C.3 are described below

*RU\_DIFFUSION.m*



```

clear;
tic;
%-----USER DEFINED PARAMETERS-----%
trajRu = transpose({xlsread('CGPF-Ru70_3_1.xlsx') xlsread('CGPF-
Ru70_3_2.xlsx') xlsread('CGPF-Ru70_4_1.xlsx') xlsread('CGPF-
Ru70_4_2.xlsx') xlsread('CGPF-Ru70_5_1.xlsx') xlsread('CGPF-
Ru70_5_2.xlsx') xlsread('CGPF-Ru70_6_1.xlsx') xlsread('CGPF-
Ru70_6_2.xlsx') xlsread('CGPF-Ru70_7_1.xlsx') xlsread('CGPF-
Ru70_7_2.xlsx') xlsread('CGPF-Ru70_10_1.xlsx') xlsread('CGPF-
Ru70_10_2.xlsx')});
%number of Ru coordinates actually extracted (4 unit cells, each
containing one 20mer)
PolyCount = 80;
%Ru coordinates in 70mer
RuCount = 70;
%each trajectory in trajRu has 8751 frames
FramesPerTraj = 8751;
%each trajectory in trajRu is 350 ns
TrajTime = 350;
%initially excite complexes 31:1:40
excite = [31:1:40]';
%rates of closest approach and decay (ns^-1) and attenuation parameter
(A^-1)
ConstantArray = [k0Ru k0Os kemRu kemOs B]';
RuRadius = (2^(1/6)) * 6.53325;
%Freezing frame for static case
T0 = 1;
%Time allowed for diffusion (ns)
LFIn = 50;
% %----END USER DEFINED PARAMETERS-----%
r = length(trajRu);
timestep = round((TrajTime/FramesPerTraj),2,'significant');
times = [0:timestep:TrajTime]';
LastFrame = LFIn/timestep;
%%Variable setup%%
IR = cell(length(excite),r);
IR_static = cell(length(excite),r);
IR_end = cell(length(excite),r);
IR_static_end = cell(length(excite),r);
KHOP_TRAJ = cell(FramesPerTraj,1);
KHOPstat_TRAJ = cell(FramesPerTraj,1);
KHOP_TRAJ_TIME = cell(r,1);
%loop over trajectories
for j = 1:r
    Ru = trajRu{j};
    Coord2DRu = mat2cell(Ru,PolyCount*ones(FramesPerTraj,1),3);
    %%Calculate time-dependent distance matrices%%
    [Coord2D_Ru_R,Dist3D] =
DISTANCE(Coord2DRu,RuRadius,FramesPerTraj,RuCount);
    %%calculate time-dependent, diagonalized kinetic matrices from
time-dependent distance matrices

```

```

[khop3D_d,Q,KHOP] =
KINETIC(Dist3D,FramesPerTraj,RuCount,timestep,ConstantArray);
%loop over initial excitations
for s = 1:length(excite)
    EXCITE = excite(s);
    InitialArray = zeros(RuCount,1);
    InitialArrayStatic = zeros(RuCount,1);
    %create matrix of initial ES populations with single
excitation
    InitialArray(EXCITE) = 1;
    InitialArrayStatic(EXCITE) = 1;
    DiffRu = [(EXCITE-30):1:(EXCITE+30)]';
    %observe time dependent ES Diffusion to complexes +- 1:30
complexes from inital excitation in flexible and static polymers
    [Iru,Iru_end,Iru_static,Iru_static_end] =
POPULATION(FramesPerTraj,T0,LastFrame,InitialArray,InitialArrayStatic,
DiffRu,khop3D_d,Q);
    IR{s,j} = Iru;
    IR_end {s,j} = Iru_end;
    IR_static{s,j} = Iru_static;
    IR_static_end{s,j} = Iru_static_end;
end
    KHOP_TRAJ(:,j) = KHOP;
    KHOP_TRAJ_TIME{j} = horzcat(KHOP_TRAJ{:,j});
end
IR_end_ = cell(r,1);
IR_static_end_ = cell(r,1);
%Organize populations at end frame by initial excitations
for j = 1:r
    IR_end_{j} = horzcat(IR_end{:,j});
    IR_static_end_{j} = horzcat(IR_static_end{:,j});
end
save('RU_DIFFUSION_B15_12T_Time.mat','ConstantArray','T0','r','IR_end'
,'IR_end_','IR_static_end','IR_static_end_','KHOP_TRAJ','KHOP_TRAJ_TIM
E');

toc;
timerVal = tic;

```

### *DISTANCE.m*

```

function [Coord2D_Ru_R,Dist3D] =
DISTANCE(Coord2DRu,RuRadius,FramesPerTraj,RuCount)
RuRu = zeros(RuCount,RuCount);
Coord2D_Ru_R = cell(FramesPerTraj,1);
Dist3D = cell(FramesPerTraj,1);
%loop over all frames
for m = 1:FramesPerTraj
    Coord2D_Ru_f = Coord2DRu{m};
    %choose coordinates to create true 70mer CG MD model
    Coord2D_Ru_frame = Coord2D_Ru_f(5:74,:);

```

```

%loop over all complex pairs
for k = 1:RuCount
    for i = 1:k
        Xk = Coord2D_Ru_frame(k,1);
        Xi = Coord2D_Ru_frame(i,1);
        Yk = Coord2D_Ru_frame(k,2);
        Yi = Coord2D_Ru_frame(i,2);
        Zk = Coord2D_Ru_frame(k,3);
        Zi = Coord2D_Ru_frame(i,3);
        distance = sqrt((Xk-Xi)^2+(Yk-Yi)^2+(Zk-Zi)^2);
        if Xk == Xi&&Yk == Yi&&Zk == Zi
            distance = inf;
        end
        %calculate periphery-periphery distances between complexes
        RuRu(k,i)= distance-(2*RuRadius);
        RuRu(i,k)= distance-(2*RuRadius);
    end
end
%save 70mer coordinates and distances
Coord2D_Ru_R{m} = Coord2D_Ru_frame;
Dist3D{m} = RuRu;
end
end

```

### ***KINETIC.m***

```

function [khop3D_d,Q_,KHOP] =
KINETIC(Dist3D,FramesPerTraj,RuCount,timestep,ConstantArray)
KHOP = cell(FramesPerTraj,1);
khop3D_d = cell(FramesPerTraj,1);
Q_ = cell(FramesPerTraj,1);
%%loop over distance matrices at each time frame
for i = 1:FramesPerTraj
    Dist = Dist3D{i};
    % Dexter EnT equation for Ru to Ru hopping...input values for
k0Ru and B
    khop = ConstantArray(1)*exp(-ConstantArray(5)*(Dist));
    %loop over diagonal elements - each Ru* can decay by multiple
pathways
        for x = 1:RuCount
            khop(x,x) = 0;
            khop(x,x) = -(sum(khop(x,:)));
            KHOP_ = diag(khop(:, :));
        end
    KHOP{i} = KHOP_;
    %Q returns the eigenvector matrix, D is the eigenvalue matrix.
    [Q,D] = eig(khop);
    %Calculate matrix exponential of diagonalized kinetic matrix *
timestep (40 ps)

```

```

        khop3D_d{i} = expm(D*timestep);
        %Save eigenvector matrix
        Q_{i} = Q;
end
end

```

### *POPULATION.m*

```

function [Iru,Iru_end,Iru_static,Iru_static_end] =
POPULATION(FramesPerTraj,T0,LastFrame,InitialArray,InitialArrayStatic,
DiffRu,khop3D_d,Q_)
    Iru = zeros(length(DiffRu),FramesPerTraj);
    Iru_static = zeros(length(DiffRu),FramesPerTraj);
    solnstatic = (Q_{T0}*khop3D_d{T0}/Q_{T0});
    %loop over all frames
    for j = 1:FramesPerTraj
        %khop3D_d is diagonalized,D,70x70 matrix, time-dependent Q
called at
        %each frame
        soln = (Q_{j}*khop3D_d{j}/Q_{j});
        %kinetics * initial population give final population
        popflex = soln*InitialArray;
        %Get Ru populations +- 1:30 pendants away from initial
excitation
        Iru(:,j) = popflex(DiffRu(:));
        %Propagate ES population - let final population become new
initial population
        InitialArray = popflex;
        popstatic = solnstatic*InitialArrayStatic;
        Iru_static(:,j) = popstatic(DiffRu(:));
        InitialArrayStatic = popstatic;
    end
    Iru_end = Iru(:,LastFrame);
    Iru_static_end = Iru_static(:,LastFrame);
End

```

## REFERENCES

1. Nantalaksakul, A.; Reddy, D. R.; Bardeen, C. J.; Thayumanavan, S., Light Harvesting Dendrimers. *Photosyn. Res.* **2006**, *87* (1), 133-150.
2. Balzani, V.; Campagna, S.; Denti, G.; Juris, A.; Serroni, S.; Venturi, M., Designing Dendrimers Based on Transition-Metal Complexes. Light-Harvesting Properties and Predetermined Redox Patterns. *Accts. Chem. Res.* **1998**, *31* (1), 26-34.
3. Devadoss, C.; Bharathi, P.; Moore, J. S., Energy Transfer in Dendritic Macromolecules: Molecular Size Effects and the Role of an Energy Gradient. *J. Am. Chem. Soc.* **1996**, *118* (40), 9635-9644.
4. Choi, M.-S., A Large Dendritic Multiporphyrin Array as a Mimic of the Bacterial Light-Harvesting Antenna Complex: Molecular Design of an Efficient Energy Funnel for Visible Photons - Choi - 2001 - *Angewandte Chemie International Edition - Wiley Online Library*. *Angew. Chem. Int. Ed.* **2001**, *40*, 3194-3198.
5. Bar-Haim, A.; Klafter, J.; Kopelman, R., Dendrimers as Controlled Artificial Energy Antennae. *J. Am. Chem. Soc.* **1997**, *119* (26), 6197-6198.
6. Bar-Haim, A.; Klafer, J., Dendrimers as light harvesting antennae. *J. Lum.* **1998**, *76*, 197-200.
7. Bar-Haim, A.; Klafter, J., Geometric versus Energetic Competition in Light Harvesting by Dendrimers. *J. Phys. Chem. B* **1998**, *102* (10), 1662-1664.
8. Bradley, P. G.; Kress, N.; Hornberger, B. A.; Dallinger, R. F.; Woodruff, W. H., Vibrational spectroscopy of the electronically excited state. 5. Time-resolved resonance Raman study of tris(bipyridine)ruthenium(II) and related complexes. Definitive evidence for the "localized" MLCT state. *J. Am. Chem. Soc.* **2002**, *103* (25), 7441-7446.
9. Cooley, L. F.; Bergquist, P.; Kelley, D. F., Determination of exciton hopping rates in ruthenium(II)(bipyridine)<sub>3</sub> complexes by picosecond polarized absorption spectroscopy. *J. Am. Chem. Soc.* **2002**, *112* (7), 2612-2617.
10. Damrauer, N. H.; Cerullo, G.; Yeh, A.; Boussie, T. R.; Shank, C. V.; McCusker, J. K., Femtosecond Dynamics of Excited-State Evolution in [Ru(bpy)<sub>3</sub>]<sup>2+</sup>. *Science* **1997**, *275* (5296), 54-57.
11. Collini, E.; Scholes, G. D., Coherent Intrachain Energy Migration in a Conjugated Polymer at Room Temperature. *Science* **2009**, *323* (5912), 369-373.
12. Fleming, C. N.; Brennaman, M. K.; Papanikolas, J. M.; Meyer, T. J., Efficient, long-range energy migration in Ru-II polypyridyl derivatized polystyrenes in rigid media. Antennae for artificial photosynthesis. *Dalton T* **2009**, (20), 3903-3910.

13. Fleming, C. N.; Dupray, L. M.; Papanikolas, J. M.; Meyer, T. J., Energy transfer between Ru(II) and Os(II) polypyridyl complexes linked to polystyrene. *J. Phys. Chem. A* **2002**, *106* (10), 2328-2334.
14. Fleming, C. N.; Jang, P.; Meyer, T. J.; Papanikolas, J. M., Energy Migration Dynamics in a Ru(II)- and Os(II)-Based Antenna Polymer Embedded in a Disordered, Rigid Medium. *J. Phys. Chem. B* **2004**.
15. Fleming, C. N.; Maxwell, K. A.; DeSimone, J. M.; Meyer, T. J.; Papanikolas, J. M., Ultrafast excited-state energy migration dynamics in an efficient light-harvesting antenna polymer based on Ru(II) and Os(II) polypyridyl complexes. *J. Am. Chem. Soc.* **2001**, *123* (42), 10336-10347.
16. Ghiggino, K. P.; Yeow, E. K. L.; Haines, D. J.; Scholes, G. D.; Smith, T. A., Mechanisms of excitation energy transport in macromolecules. *J Photoch Photobio A* **1996**, *102* (1), 81-86.
17. Harmandaris, V. A., Quantitative study of equilibrium and non-equilibrium polymer dynamics through systematic hierarchical coarse-graining simulations. *KOREA-AUST RHEOL J* **2014**, *26* (1), 15-28.
18. Hwang, I.; Beaupre, S.; Leclerc, M.; Scholes, G. D., Ultrafast relaxation of charge-transfer excitons in low-bandgap conjugated copolymers. *Chem Sci* **2012**, *3* (7), 2270-2277.
19. Kurt, K.; S., G. G., Dynamics of entangled linear polymer melts: A molecular-dynamics simulation. *J. Chem. Phys.* **1990**, *92*.
20. Leem, G.; Keinan, S.; Jiang, J. L.; Chen, Z.; Pho, T. A.; Morseth, Z. A.; Hu, Z. Y.; Puodziukynaite, E.; Fang, Z.; Papanikolas, J. M.; Reynolds, J. R.; Schanze, K. S., Ru(bpy)<sub>3</sub>(2+) derivatized polystyrenes constructed by nitroxide-mediated radical polymerization. Relationship between polymer chain length, structure and photophysical properties. *Polym Chem-Uk* **2015**, *6* (47), 8184-8193.
21. Leem, G.; Morseth, Z. A.; Puodziukynaite, E.; Jiang, J. L.; Fang, Z.; Gilligan, A. T.; Reynolds, J. R.; Papanikolas, J. M.; Schanze, K. S., Light Harvesting and Charge Separation in a pi-Conjugated Antenna Polymer Bound to TiO<sub>2</sub>. *J. Phys. Chem. C* **2014**, *118* (49), 28535-28541.
22. Leem, G.; Morseth, Z. A.; Wee, K. R.; Jiang, J. L.; Brennaman, M. K.; Papanikolas, J. M.; Schanze, K. S., Polymer-Based Ruthenium(II) Polypyridyl Chromophores on TiO<sub>2</sub> for Solar Energy Conversion. *Chem-Asian J* **2016**, *11* (8), 1257-1267.
23. Morseth, Z. A.; Pho, T. V.; Gilligan, A. T.; Dillon, R. J.; Schanze, K. S.; Reynolds, J. R.; Papanikolas, J. M., Role of Macromolecular Structure in the Ultrafast Energy and Electron

Transfer Dynamics of a Light-Harvesting Polymer. *J. Phys. Chem. B.* **2016**, *120* (32), 7937-7948.

24. Morseth, Z. A.; Pho, T. V.; Sheridan, M. V.; Meyer, T. J.; Schanze, K. S.; Reynolds, J. R.; Papanikolas, J. M., Interfacial Dynamics within an Organic Chromophore-Based Water Oxidation Molecular Assembly. *Acs Appl Mater Inter* **2017**, *9* (19), 16651-16659.

25. Morseth, Z. A.; Wang, L.; Puodziukynaite, E.; Leem, G.; Gilligan, A. T.; Meyer, T. J.; Schanze, K. S.; Reynolds, J. R.; Papanikolas, J. M., Ultrafast Dynamics in Multifunctional Ru(II)-Loaded Polymers for Solar Energy Conversion. *Accounts Chem Res* **2015**, *48* (3), 818-827.

26. Puodziukynaite, E.; Walczak, R. M.; Wang, L.; Sun, Y. L.; House, R. L.; Schanze, K. S.; Papanikolas, J. M.; Reynolds, J. R., TiO<sub>2</sub> surface anchoring polyfluorene-based ionic transition metal complex polymers as light-harvesting assemblies. *Abstr Pap Am Chem S* **2011**, *242*.

27. Puodziukynaite, E.; Wang, L.; Schanze, K. S.; Papanikolas, J. M.; Reynolds, J. R., Poly(fluorene-co-thiophene)-based ionic transition-metal complex polymers for solar energy harvesting and storage applications. *Polym Chem-Uk* **2014**, *5* (7), 2363-2369.

28. Shaw, G. B.; Papanikolas, J. M., Triplet-triplet annihilation of excited states of polypyridyl Ru(II) complexes bound to polystyrene. *J. Phys. Chem. B.* **2002**, *106* (24), 6156-6162.

29. Wang, L.; Puodziukynaite, E.; Grumstrup, E. M.; Brown, A. C.; Keinan, S.; Schanze, K. S.; Reynolds, J. R.; Papanikolas, J. M., Ultrafast Formation of a Long-Lived Charge-Separated State in a Ru-Loaded Poly(3-hexylthiophene) Light-Harvesting Polymer. *J. Phys. Chem. Lett.* **2013**, *4* (14), 2269-2273.

30. Wang, L.; Puodziukynaite, E.; Vary, R. P.; Grumstrup, E. M.; Walczak, R. M.; Zolotarskaya, O. Y.; Schanze, K. S.; Reynolds, J. R.; Papanikolas, J. M., Competition between Ultrafast Energy Flow and Electron Transfer in a Ru(II)-Loaded Polyfluorene Light-Harvesting Polymer. *J. Phys. Chem. Lett.* **2012**, *3* (17), 2453-2457.

31. Zhang, L. H.; Pei, K.; Yu, M. D.; Huang, Y. L.; Zhao, H. B.; Zeng, M.; Wang, Y.; Gao, J. W., Theoretical Investigations on Donor-Acceptor Conjugated Copolymers Based on Naphtho[1,2-c:5,6-c]bis[1,2,5]thiadiazole for Organic Solar Cell Applications. *J. Phys. Chem. C.* **2012**, *116* (50), 26154-26161.

32. Davis, W. B.; Ratner, M. A.; Wasielewski, M. R., Conformational Gating of Long Distance Electron Transfer through Wire-like Bridges in Donor-Bridge-Acceptor Molecules. *J. Am. Chem. Soc.* **2001**, *123* (32), 7877-7886.

33. Crosby, G. A.; Demas, J. N., Quantum efficiencies on transition metal complexes. II. Charge-transfer luminescence. *J. Am. Chem. Soc.* **2002**, *93* (12), 2841-2847.

34. Muller-Plathe, F., Coarse-Graining in Polymer Simulation: From the Atomistic to the Mesoscopic Scale and Back - Müller-Plathe - 2002 - ChemPhysChem - Wiley Online Library. *Chem. Phys. Chem* **2002**, *3*, 754-769.
35. Abrams, C. F.; Kremer, K., Combined Coarse-Grained and Atomistic Simulation of Liquid Bisphenol A–Polycarbonate: Liquid Packing and Intramolecular Structure. *Macromolecules* **2002**, *36* (1), 260-267.
36. Padding, J. T.; Briels, W. J., Time and length scales of polymer melts studied by coarse-grained molecular dynamics simulations. *J. Chem. Phys.* **2002**, *117* (2).
37. Marrink Siewert J.; de Vries Alex H., a.; E., M. A., Coarse Grained Model for Semiquantitative Lipid Simulations. *J. Phys. Chem. B* **2003**, *108* (2), 750-760.
38. Marrink, S. J.; Jelger, R., H.; Yefimov, S., The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *J. Phys. Chem. B* **2007**, *111* (27), 7812-7824.
39. Bond, P. J., Coarse-grained molecular dynamics simulations of membrane proteins and peptides. *J. Struct. Bio.* **2006**, *157*, 593-605.
40. Levine, Z. A., Life Cycle of an Electropore: Field-Dependent and Field-Independent Steps in Pore Creation and Annihilation | SpringerLink. *J. Memb. Bio.* **2010**, *236* (1), 27-36.
41. Moore, E. B.; Molinero, V., Structural transformation in supercooled water controls the crystallization rate of ice. *Nature* **2011**, *479* (7374), 506.
42. Hickey, K.; Waghorne, W. E., Viscosities and Volumes of Dilute Solutions of Formamide in Water + Acetonitrile and for Formamide and N,N-Dimethylformamide in Methanol + Acetonitrile Mixed Solvents: Viscosity B-Coefficients, Activation Free Energies for Viscous Flow, and Partial Molar Volumes. *J.Chem.Eng. Datta* **2001**, *46* (4), 851-857.
43. A., H. V.; P., A. N.; van der Vegt N. F. A., a.; Kremer, K., Hierarchical Modeling of Polystyrene: From Atomistic to Coarse-Grained Simulations. *Macromolecules* **2006**, *39* (19), 6708-6719.
44. Reith, D.; Putz, M.; Muller-Plathe, F., Deriving effective mesoscale potentials from atomistic simulations - Reith - 2003 - Journal of Computational Chemistry - Wiley Online Library. *J. Comput. Chem.* **2003**, *24*, 1624-1636.
45. Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C., The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128*.
46. Gautieri, A.; Vesentini, S.; Redaelli, A., How to predict diffusion of medium-sized molecules in polymer matrices. From atomistic to coarse grain simulations. *Journal of Molecular Modeling* **2010**, *16* (12), 1845-1851.



47. Bondi, A., van der Waals Volumes and Radii. *J. Phys. Chem.* **1964**, *68* (441).
48. Al-Attar, H.; Griffiths, G.; Moore, T., Highly Efficient, Solution-Processed, Single-Layer, Electrophosphorescent Diodes and the Effect of Molecular Dipole Moment - Al-Attar - 2011 - Advanced Functional Materials - Wiley Online Library. *Adv. Func. Mat.* **2011**, *21* (12), 2376-2382.
49. Berberan-Santos, M. N., The integration of kinetic rate equations by matrix methods. *J. Chem. Ed.* **1990**, *67* (5), 375.