Disruption of Model Membranes by PAMAM Dendrimers
and
Quantatitive Nanoscale Analysis of Type I Collagen

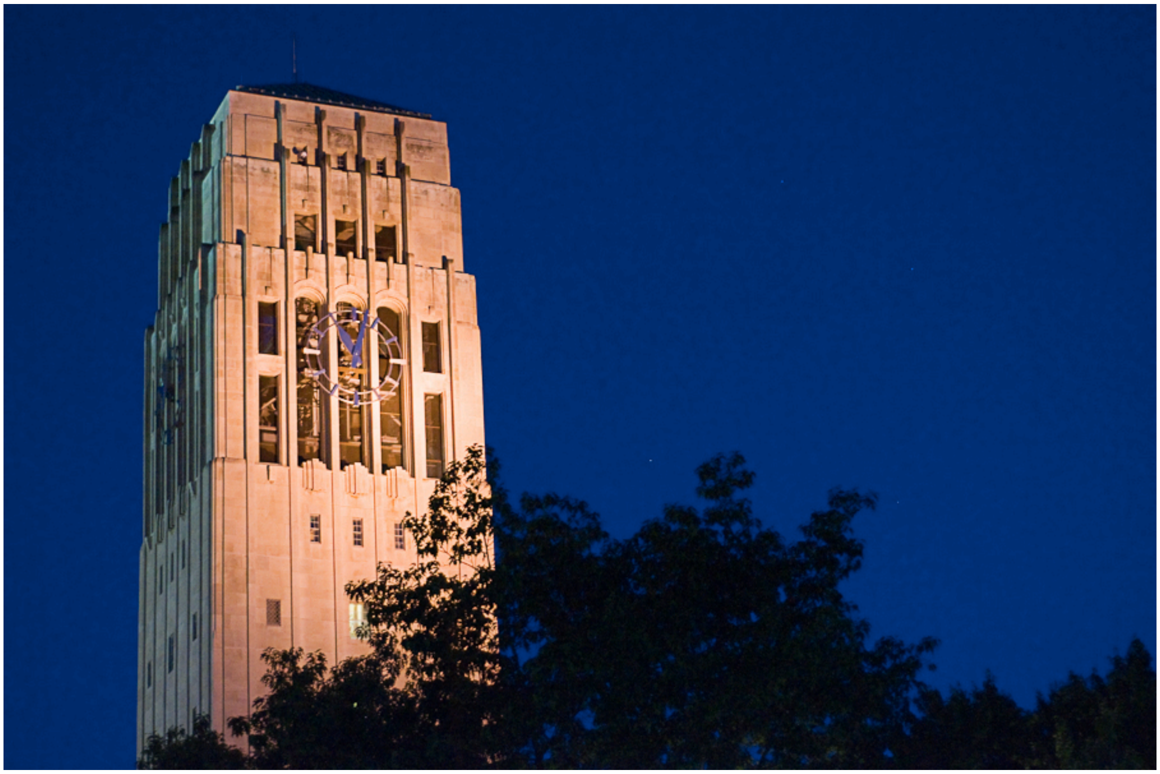
by

Blake William Erickson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biophysics)
in The University of Michigan
2010

Doctoral Committee:

Professor Mark M. Banaszak Holl, Chair
Professor James R. Baker, Jr
Professor Zhan Chen
Professor Bradford G. Orr
Professor Ayyalusamy Ramamoorthy



Sometimes all you have to do is look up.

Dedication

This dissertation is dedicated to the following people who have been with me and guided me throughout my life:

Family: Bill Erickson, Bev Erickson, Blair Erickson, Glenn Erickson, and Marion Erickson, Samuel Edwin Barnette, Irma Barnette, Juanita Rodriguez

Teachers: Mr. George Santich, Ms. Martina Lawlor, and Mr. Phil Miller

Friends: Yolanda Guzman, Jessica Guzman, and Kristine Mangalindian

Acknowledgements

Many thanks to everyone who has helped me along the way through out this project. Without the wealth of support everyone has given me, I would not be here today. I would like to collectively thank each member, past and present, of the Banaszak Holl/Orr laboratory. Each of you have been incredibly supportive and helped me grow as a person and scientist. I would especially like to thank the following people for always being there for me, listening when I needed to say something, and for all the advice over the years: Dr. Joseph Wallace, Dr. Pascale Leroueil, Dr. Chris Kelly, and Dr. Lisa Prevette.

Table of Contents

Dedication ii

Acknowledgements iii

List of Figures viii

List of Tables x

List of Equations xi

List of Appendices xii

Glossary xiii

Abstract xiv

Chapter 1 Introduction 1

Microscopy and the Biophysicist 1

Survanta as a Platform for Studying Nanotoxcicity 4

Enhancing Quantatitive Assessments of Bone 7

Chapter 2 Survanta 9

Introduction 9

Materials and Method 12

Results & Discussion 13

Observed Domain Structures of Survanta Supported Lipid Bilayers 13

Role of fatty acids and proteins 15

Faster AFM scanning 16

Domain Dependence of PAMAM dendrimer interaction...*17*

Disruption Models *21*

Conclusions...34

Chapter 3 Bone Methods 35

Introduction..35

Materials and Methods..36

Animals *36*

Atomic Force Microscopy (AFM) Imaging and Analysis.......................................*37*

Computation..*39*

Statistical Analyses *39*

Results and Discussion 40

Measurement of the D-periodic Gap/Overlap Spacing in Type I Collagen Fibrils .40

Analysis and Associated Uncertainties in D-Periodic Spacing Measurement.........44

Observed Distributions in D-Periodic Spacing Measurements...............................51

Statistical Comparison Between D-Periodic Spacing Distributions 63

Conclusions...65

Chapter 4 Conclusions 66

Conclusions from Survanta: Translating Hypothesizes into Observables.................66

Conclusions from Bone: Improving Nanoscale Quantative Metrics to Address Biological Questions...68

Future Directions and Outlook...70

Appendix A: Deflection Sensitivity...74

Abstract 74

Introduction 74

Methodology and Results 76

Conclusion 81

Appendix B: Bone Code 82

Rational 82

System Requirements and Comments 82

Users' Guide Collagen_fft.m 82

Image Formats 82

Default Parameters 83

A Practical Guide to Drawing Regions of Interest 84

Collagen_fft.m 89

Users' Guide Population_analysis.m 98

Population_analysis.m 99

Example Population Data 116

Licensing Collagen_FFT.m and Population_Analysis.m 117

Appendix C: Flow Cytometry Compensation Code 118

Rational 118

Users' Guide Compensation.m 118

Compensation.m 120

Users' Guide Gating.m 130

Gating.m 131

Users' Guide Half_Life.m 143

Half_Life.m 144

Licensing Compensation.m, Gating.m, and Half_Life.m..147

Bibliography 148

List of Figures

Figure 2.1 PAMAM Dendrimer....11

Figure 2.2 Survanta Bilayer....14

Figure 2.3 Lipid Structure....15

Figure 2.4 Removal by G5 PAMAM....18

Figure 2.5 Removal Highlighted....19

Figure 2.6 Removal by G7 PAMAM....21

Figure 2.7 Clean Removal of Lipid....24

Figure 2.8 Removal by G7 PAMAM: 3D Rendering #1....25

Figure 2.9 Removal by G7 PAMAM: 3D Rendering #2....26

Figure 2.10 Phase Imaging....28

Figure 2.11 3D Rendering of Accumulation #1....29

Figure 2.12 3D Rendering of Accumulation #2....30

Figure 2.13 3D Rendering of Accumulation #3....31

Figure 2.14 Dendrimer Accumulation Step Height....32

Figure 3.1 Schematic Representation of the 2D-FFT Process....39

Figure 3.2 Effects of Angle on 1D FFT Measurements of D-Periodic Spacing....43

Figure 3.3 Model Collagen Fibril and Associated Fast Fourier Transform....46

Figure 3.4 D-Period Uncertainty Calculation....50

Figure 3.5 Histogram of D-Periodic Spacings....51

Figure 3.6 Historical SEM image of Type I Collagen Fibrils..54

Figure 3.7 SAXS scattering data..55

Figure 3.8 SAXS D-Period Spacing 57

Figure 3.9 Example BWA Transform..59

Figure 3.10 Fit of ln(An) vs. l2 61

Figure 3.11 SAXS Calculated D-Period Distribution..62

Figure 3.12 Cumulative Density Function of D-Periodic Spacings 64

Appendix Figure A.1 Deflection Sensitivity Distributions...77

Appendix Figure A.2 Sensitivity vs. Spring Constant..79

Appendix Figure A.3 Measurement Certainty..80

List of Tables

Table 2.1 Summary of Accumulation Step Heights 33

Table 3.1 Table of D-Periodic Spacings 47

Appendix Table B.1 Example Population Data 117

List of Equations

Equation 2.1 Minimum Resolvable Features 16

Equation 3.1 Discrete FFT 48

Equation 3.2 SAXS Baseline 54

Equation 3.3 Determination of D-Period spacing Distribution 60

Appendix Equation A.1 Hook's Law 77

List of Appendices

Appendix A: Deflection Sensitivy 74

Appendix B: Bone Code 82

Appendic C: Flow Cytometry Compensation Code 118

Glossary

AFM	Atomic Force Microscope
PAMAM	Poly(amidoamine)
FFT	Fast Fourier Transform
BWA	Bertaut-Warren-Averbach

Abstract

The work presented in this thesis uses the Atomic Force Microscope (AFM) to study the properties of nano-scaled biological systems. This work is divided into two major sections. The first section uses Survanta as a model system for studying the interaction of nanoparticles with the lung lining. The second section looks at the nanoscale morphology of collagen fibrils.

Survanta forms a phase-separated bilayer when deposited onto mica. At room temperature one domain is comprised of mostly the anionic POPG in the fluid phase, while the other domain contains the zwiterionic DPPC in the gel phase. Generation 5 and 7 Poly(amidoamine) dendrimers preferentially disrupt the fluid layer. For the first time, dendrimer can be observed collecting on top of the lipid preceding removal. The location of the dendrimer is verified by phase imaging. Molecular Dynamics simulations also support that dendrimers are on top of the bilayer. These simulations show that the dendrimer deforms more when interacting with a fluid lipid bilayer. This increase in deformation is observed in AFM images through an analysis of the edge profiles of the dendrimer adhered to the surface. Finally, following removal, no dendrimer is adhered to the mica substrate. This observation, coupled with observations of the dendrimer on top of the lipid, supports that disruption is due to lipid encapsulating dendrimer and not a disruption of the interaction of lipid and substrate.

Turning to the second part of the thesis, collagen fibrils exhibit a characteristic banding pattern roughly 67 nm in spacing. A 2D Fast Fourier Transform technique has been developed to measure both the spacing and orientation of fibrils within AFM images. Computer simulations and experimental data show at least 9 D-Period repeats must be measured to minimize the potential for error. The resolution limit of the 2D-FFT technique is 0.8 nm. Large populations of fibrils from Ovine samples, exhibit a distribution of fibril spacing—significantly larger than the resolution limit—around a central mean. A distribution of spacings also exists in electron microscope images. X-Ray scattering data also shows similar sized distributions in spacing. Finally, statistical comparisons show a significant change in the distribution caused by estrogen depletion in Ovine samples.

Chapter 1

Introduction

Microscopy and the Biophysicist

When broken down into constituent parts, biology is essentially a grand sum of many, many interactions on the nanoscale. For example, proteins range from the relatively small insulin, at 51 amino acids and 5808 Da[1], to the largest human protein (and third most common), Titin at 34,350 amino acids, or roughly 3 million Da[2]. Collagen, the most common protein in mamals[3], forms a quaternary structure with 3 collagen molecules 1.5 nm in diameter by 300 nm long[4]. These in turn, form larger molecules that become the structural basis for everything from bone, to tendon, to skin[5]. Imaging this fundamental single-nanometer length scale is a goal of the biophysicist because it allows interesting biological questions to be probed with the constructionist viewpoint of a physicist by starting at the smallest fundamental element and building in complexity. This provides a useful counterpoint to the deconstructionist view of the traditional biologist who removes layers of complexity from a system.

Starting with Hooke using an early compound microscopes in 1665 to observe cork cell walls for the first time, and Leeuwenhoek observing bacteria and protozoa in water in 1678, to the first images of collagen fibrils with an electron microscope by Hall and Doty in 1958[6], to modern multi-color confocal microscopy highlighting the separate acting and micro-tubule networks that form the cytoskeleton of cells in exquisite detail[7], advances in microscopy have allowed for an ever growing number of interesting biological problems to be investigated.

Certainly, the modern optical microscope is one of the most powerful investigative tools for biology. Optical microscopes are nearly ubiquitous, they are easy to use, perpetually expandable, applicable to an infinite number of different samples, and, most importantly, biologically compatible. Histological samples can be sectioned, stained, and imaged for signs of pathology. Immuno-labeling allows specific proteins within a cell to be imaged directly[8, 9], nucleic acids can be stained and followed through the cell cycle[10-12]. Fluorescence Resonance Energy Transfer and other single molecule techniques can even be used to track conformational changes in nucleic acids[13-15]. Live cells are nearly as easy to work with as fixed cells. Finally, the most significant feature of optical techniques is the incredible range of size and time scales that can be observed. In size, large organisms like humans can be imaged on meter length scales (10^0 m), insects like ants on millimeter length scales (10^{-2} m), single celled organisms on micron length scales (10^{-6} m), before finally being limited in resolution to roughly half the wavelength of light being used for imaging at hundreds of nanometers (10^{-7} m). This is seven orders of magnitude! In time, the difference is even more astounding, ranging from milliseconds on a short end (10^{-3} s) to the doubling time of cells at tens of hours (10^4 s), finally to changes over years, albeit in a series of snapshots spread out over time, in the cases of large animals like humans (10^7 s). This is an even more astounding ten orders of magnitude! Unfortunately, because the wavelength of light (10^{-7} m) sets the fundamental resolution limit in optical techniques, we fall two orders of magnitude shy of the single nanometer length scale (10^{-9} m) that is fundamental to biology.

Getting to the single nanometer length scale is not an impossible task. Electron microscopes have long been able to image at the nanometer scale, recall the first images of collagen in the 1950's. Any number of sub-cellular organelles can be imaged[16, 17]. Unfortunately, this increase in resolution comes at a significant cost. Electron microscopy requires high vacuum in order to operate, which is incompatible with living tissue. This can lead to damage of delicate samples. Furthermore, samples often require exogenous contrast agents to be imaged, which may alter the system[17-21]. Only fixed snapshots of a system can be observed. The wealth of dynamic information available with optical techniques is lost.

The Atomic Force Microscope is an important tool for the biophysicist for a number of reasons. First and foremost, the AFM is capable of nanometer lateral resolution and angstrom vertical resolution[22-24]. Secondly, unlike the electron microscope, the AFM does not require high vacuum or exogenous contrast agents. Samples can be imaged in air or fluid[22, 25], cooled or heated, and with some effort, AFMs can even be integrated to allow simultaneous optical and topographical images of samples[26]. In addition, this biological compatibility offers some limited ability to observe dynamic systems, albeit slowly progressing ones as image capture times rarely fall below multiple minutes in time (10^2 s)[26,27, 28]. One final advantage the AFM has over optical or electron microscopy is the ability to learn some limited materials information about the samples. Unlike optical microscopy, which uses photons to image the sample, or electron microscopes, which use electrons, the AFM uses the physical interaction of an ultra-sharp tip (Tip radius varies on application typically between 2 and 20 nm.) with the sample. As the sample properties change, the interaction with the tip does as well. This, in turn, provides complementary data reflecting these changes in materials properties[29-31]. These properties of the AFM have allowed it to be used on multiple systems in biology, ranging from investigations of model membrane systems[32-35], to amazing images and mechanical testing of collagen and bone performed by Hansma at UCSB[36-39] and Habelietz at UCSF[40-43], to recent work by Fantner at MIT imaging bacteria death caused by anti-microbial peptides in real time.[44]

Survanta as a Platform for Studying Nanotoxcicity

In 2005, Oberdoerster, Oberdoerster, and Oberdoerster wrote a review on the toxicological effects of nanoparticles in which they stated[45]:

> Exposures to airborne nano-sized particles (NSPs; < 100 nm) have been experienced by humans throughout their evolutionary stages, but it is only with the advent of the industrial revolution that such exposures have increased dramatically because of anthropogenic sources such as internal combustion engines, power plants, and many other sources of thermo-degradation. The rapidly developing field of nanotechnology is likely to become yet another source for human exposures to NSPs—engineered nanoparticles (NPs)—by different routes: inhalation (respiratory tract), ingestion [gastrointestinal (GI) tract], dermal (skin), and injection (blood circulation).

And further stated,

> The extraordinarily high number concentrations of NSPs per given mass will likely be of toxicologic significance when these particles interact with

> cells and subcellular components. Likewise, their increased surface area per unit mass can be toxicologically important if other characteristics such as surface chemistry and bulk chemistry are the same. Although the mass of UFPs in ambient air is very low, approaching only 0.5–2 μg/m^3 at background levels (Hughes et al. 1998), it can increase several-fold during high pollution episodes or on highways (Brand et al. 1991; Shi et al. 2001; Zhu et al. 2002).

From these two quotes alone, it becomes clear that understanding the mechanisms that regulate how nanoparticles interact with cell membranes is a critical direction of research. These are a necessary compliment to general toxicity studies because these mechanistic studies have the potential to offer insight on how to mitigate the issues caused.

Existing studies have very clearly shown that charged nanoparticles can disrupt lipid membranes[32-35, 46], and without this charge disruption does not occur[46]. These studies form an excellent foundation for new work, especially since they show a difference in removal between different lipid phases[34]. This is important, because the lipid composition of cell membranes is carefully controlled in order to regulate membrane physical characteristics, including receptor clustering and response to external stressing factors. [47] In particular, it has been hypothesized that cells regulate lipid and protein composition to form domains with different size and lipid phases by adjusting lipids, sphingolipids, sterols, and proteins contents [48, 49]. Optical techniques, used by Veach[48, 50-55] and Keller[56-60], have investigated micro domain formation relating them to phase diagrams of mixtures of lipids and sterols. Of significant interest is the existence of critical points in the phase diagram where multiple states can coexist[56, 57, 59-61]. In these systems, where the lipid and sterol composition is carefully regulated and the temperature varied, significant changes in domain size and behavior can be observed with small changes in temperature[53, 54, 56, 57]. In similar work, Groves has shown that differential thermotropic phase behavior is sufficient to maintain and regulate domain formation and behavior[62-75].

Survanta is an excellent system to study for a number of reasons. First, it's a clinically used lung surfactant for the treatment of respiratory distress syndrome. That is, Survanta is used as a replacement lung lining in cases where this lining is damaged or incomplete. This is of particular interest given the above quotes from Oberdoerster stating that inhalation is a key mechanism by which nano-particles enter the body. Second, Survanta has two distinctly different lipid components with different properties as well as fatty acids and proteins[76, 77]. The beauty of this is that Survanta separates into different domains observable by AFM. This separation provides a perfect test bed to compare how disruption can be influenced by the different properties of these domains, i.e. fluid vs. gel, charge effects, presence vs. absence of proteins.

The final goal of studying the interaction of nanoparticles with Survanta is aimed squarely at learning something new about the mechanistic details of the disruption of the membrane by the polymer. For example, multiple mechanisms have been suggested for the mechanism of disruption by the polymers, from lipid encapsulation, to Gewirth's suggestion of a disruption of the interaction between the lipid and the substrate[78]. These different mechanisms yield different observables—in the encapsulation model, the polymer is taken away, in Gewirth's model it is left behind. This is a feature that can be observed by AFM because of the technique's high lateral resolution. Second, Molecular Dynamic models show that the polymer remains exposed to solvent while interacting with the lipid and that the polymer deforms differently depending on the phase of the lipid it interacts with[79]. These are both testable conditions that can potentially be probed with AFM

Enhancing Quantitative Assessments of Bone

From a materials standpoint, bone is an amazing composite. It is adaptive, self-healing, lightweight, and shows an excellent balance between strength, rigidity, and compliance[80-83]. These properties stem from the hierarchical nature of bone[84-88]; which, at it's most basic level, is composed of Type I collagen fibrils and hydroxyapatite mineral crystals[89]. Because of this hierarchal structure, small changes in the collagen structure can lead to a number of different bone diseases which compromise bone properties[90, 91]. A prime example is Osteogenisis Imperfecta, in which a single point mutation in a collagen molecule leads to collagen fibrils that cannot align properly[92-94]. This, in turn, leads to improper mineralization of the fibrils[95-97], and ultimately bones with increased brittleness compared to normal tissue[98-100]. Before questions about the mechanics of how changes in the collagen or mineral alter bone properties can be addressed, clear methods to characterize the nanoscale characteristics of the tissue ultrastructure must be established. Ideally, the characterization of the ultrastructure would occur on samples with minimally disruptive surface preparation, maintaining the possibility of other characterization methods on the same sample. The most accessible target for characterization of collagen is the D-Periodic axial gap/overlap spacing, which is observed in all Type I collagen fibrils. The 67 nm D-Periodic spacing has been well established over many decades of research, and can be explained by the orientation of the individual collagen molecules within a collagen fibril[101-104].

The goal of this work is to significantly improve the ability to accurately measure this characteristic feature of collagen in a quantitative fashion. Doing so allows for fundamentally new questions to be asked about collagen, a protein that has been studied for more than 50 years. The first question that was answered as a result of this technique is, does collagen exists with a distribution of spacings, unlike the fixed D-Period spacing hypothesized by Hodge and Petruska in 1963? The second question answered was, can pathological conditions lead to observable changes in the D-Period spacing? This study starts by implementing a Two Dimensional Fast Fourier Transform to accurately extract the D-period spacing in a fashion that is unbiased by orientation. Next, theoretical and practical resolution limits are explored to determine prerequisite conditions to ensure accurate results. These resolution limits are small enough that a distribution of D-Period spacings does, in fact, exist in real samples. This distribution is further shown to have been present in literature through separate analysis of historical Scanning Electron Microscopy images and Small Angle X-Ray Data. Finally, the study outlines how to statistically compare the observed distribution of D-Period spacings to identify differences between sham operated and ovariectomized sheep, a model for the early stages of osteopenia[105-109].

Chapter 2

Survanta

Introduction

Understanding the interaction between nanoparticles and the lung lining is an ever-increasing concern as the amount of airborne particulate matter introduced into the environment by human activity grows.[45] Nanoparticles have been implicated in a number of acute and chronic medical conditions, particularly those associated with inflammation and cardiovascular disease.[110-112] A better understanding of the mechanisms by which nanoparticles penetrate the lung lining and enter the lungs would allow for better treatment and prevention of diseases.

Survanta™ is an excellent model system for investigating the mechanisms of nanoparticle transport and toxicity in the lungs because it mimics the composition and function of the alveolar lining while being a relatively well-defined experimental system. Additionally, Survanta is a widely used clinical surfactant for the treatment of Respiratory Distress Syndrome (RDS) making its choice particularly relevant.[113, 114] Survanta's ability to form stable monolayers at the alveoli-air interface that spread with alveoli expansion and contraction to maintain a low surface tension are critical for proper lung function, which is compromised in RDS due to an incomplete surfactant layer. Both mechanistic and *in vivo* studies have been performed addressing these issues, making Survanta one of the best-characterized lung surfactants available.[77, 115, 116]

The two major lipid components in Survanta, 1,2-dipalmitoyl-sn-glycero-3-phosphocholine (DPPC) and 1-palmitoyl-2-oleoyl-sn-glycero-3-[phospho-rac-(1-glycerol)] (POPG), have melting temperatures significantly above and below room temperature, 41 °C and -2 °C respectively.[117] When deposited onto mica, this difference in transition temperature leads to distinct fluid and gel domains that exhibit little structural or phase changes over time at room temperature. These distinct domains allow for direct comparison between nanoparticle interaction with fluid and gel phases. Second, the presence of fatty acids (palmitic) and proteins, observable with atomic force microscopy (AFM),[118, 119] brings us closer to the level of complexity seen in actual cell membranes.

Nanoscale disruption of lipid membranes by charged polymer-based nanoparticles is well documented in the literature.[32-35, 46, 78, 120] These studies demonstrate that charged nanoparticles disrupt biological membranes; however, they leave many open questions regarding the mechanism of disruption, including the role of head group charge,[32, 46] the influence of lipid phase,[32, 34] and the effects of cholesterol, proteins and other membrane components. In 2005, Mecke *et. al.* observed that only lipid in the fluid phase was removed by amine terminated poly(amidoamine) (PAMAM) dendrimers.[34] A number of mechanisms for membrane disruption have been suggested. In one such model, the lipids encapsulate the polymer in a lipid vesicle.[46] More recently, Gewirth and colleagues suggest the possibility that disruption of the electrostatic interactions between the charged mica substrate and the lipid bilayer lead to lipid removal.[78, 120]

PAMAM dendrimers were chosen as the model nanoparticles for interaction with Survanta because of the well-controlled size, excellent polydispersity (1.01), and well-defined surface chemistry.[121-123] Under AFM imaging conditions, unbuffered and neutral pH, all primary amine groups of the PAMAM dendrimers are expected to be protonated.[124, 125] Space filling models of G5 and G7 dendrimers are shown in Figure 2.1.

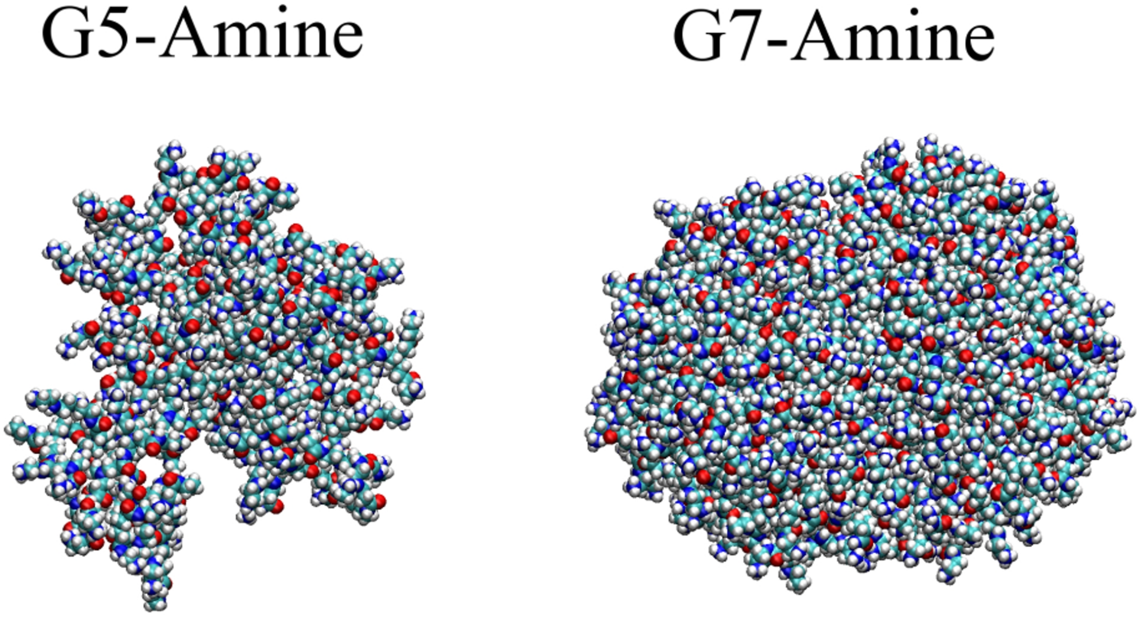


Figure 2.1 PAMAM Dendrimer
Space filling models of equilibrated generation five (G5) and generation seven (G7) poly(amidoamine) PAMAM dendrimers. The G5 dendrimer has one hundred twenty eight (128) surface amines and an approximate diameter of 5 nm. The G7 dendrimer has five hundred twelve (512) surface amines and an approximate diameter of 7 nm.

There are three key results reported in this paper. First, the fluid domain is removed more than an order of magnitude faster than the gel domain. Second, dendrimer accumulation on lipid edges and terraces preceding lipid removal has been directly imaged for both fluid and gel domains and all tested dendrimer generations for the first time. Third, immediately following lipid removal the mica surface is clean. This indicates that lipid defects are not induced by dendrimers binding to mica and displacing the lipid.

Materials and Method

Survanta was generously provided by Ross Laboratories (Columbus, OH) and stored at 4 °C. Survanta, consists of 25 mg/mL of phospholipids: 44-62% 1,2-dipalmitoyl-sn-glycero-3-phosphocholine (DPPC), 13-45% 1-palmitoyl-2-oleoyl-sn-glycero-3-[Phospho-rac-(1-glycerol)] (POPG), 2-7% triglycerides, 5-14% fatty acids most of which is Palmitic acid, and less than 4% surfactant proteins B and C all suspended in a 0.9 M NaCl solution. For these experiments, Survanta was diluted by a factor of ten with PBS buffer (Sigma-Aldrich, St. Louis MO) to a phospholipid concentration of 2.5 mg/mL. 80 mL of the dilute Survanta was warmed to 37 °C and deposited onto freshly cleaved mica surfaces roughly 1 cm^2 in size forming bilayers via the vesicle fusion method. Surfaces were allowed to incubate for at least a half hour in a humid environment at 37 °C, which is above the observed transition temperature for both domains, ensuring layers formed with minimal defects. This observed transition temperature is below that of pure DPPC because both the fatty acids and POPG present in the DPPC enriched domain lower and broaden the observed transition temperature.

Samples were washed with ultra pure water (NERL Diagnostics, East Providence, RI) to remove any excess lipid and transferred to either a Nanoscope IIIa Multimode scanning probe microscope from Digital Instruments equipped with an “E” scanner (Veeco Metrology Group, Santa Barbara, CA) or a PicoPlus 5500 AFM equipped with a multipurpose small scanner from Molecular Imaging (Agilent, Chandler, AZ). Bilayers were kept in ultra pure water during imaging. Images were taken in tapping mode using silicon nitride cantilevers (DI model NPS, spring constant 0.32 N/m, length 100 μm) with a $(3\ \mu m)^2$ scan size, 512 scans per frame, between 2-4 Hz (typically 3 Hz). All imaging was done at room temperature, nominally 25 °C, which led to one domain, mostly DPPC, in the gel phase and the other, mostly POPG, in the fluid phase.

Generation 5 and 7 PAMAM dendrimer were purchased from Dendritech Inc. To remove lower molecular weight impurities and trailing generations, the dendrimer was dialyzed with a 10,000 molecular weight cut-off (MWCO) membrane against de-ionized (DI) water for two days, exchanging washes every 5 hours. The purified dendrimer was lyophilized for three days resulting in a white solid. The number average molecular weight (25,280 g/mol) and PDI (1.015 ± 0.010) was determined by GPC. Potentiometric titration was conducted to determine the average number of primary amines (108) as previously described.[125] Generation 7 PAMAM dendrimer was purchased from Dendritech Inc. To remove lower molecular weight impurities and trailing generations the dendrimer was dialyzed with a 50,000 MWCO membrane against DI water for two days, exchanging four washes. The purified dendrimer was lyophilized for three days resulting in a white solid. The number average molecular weight (105,600 g/mol) and PDI (1.053 ± 0.012) was determined by GPC. Dendrimer solutions, made with ultra pure water, were injected into the fluid cells of either instrument while imaging until the concentration within the cell was between 80 nM and 150 nM. In either instrument, the injected volume was much less than the volume of the cell, preventing significant changes to the resonance frequency of the tip.

Results & Discussion

Observed Domain Structures of Survanta Supported Lipid Bilayers

A typical AFM image of a Survanta bilayer on mica is illustrated in Figure 2.2. Domain A, shown on the left and right edges in light grey is predominantly composed of zwitterionic DPPC in the gel phase.[126] Domain B, the darker grey in the middle of Figure 2.2 is predominantly the anionic POPG in the fluid phase.[119]

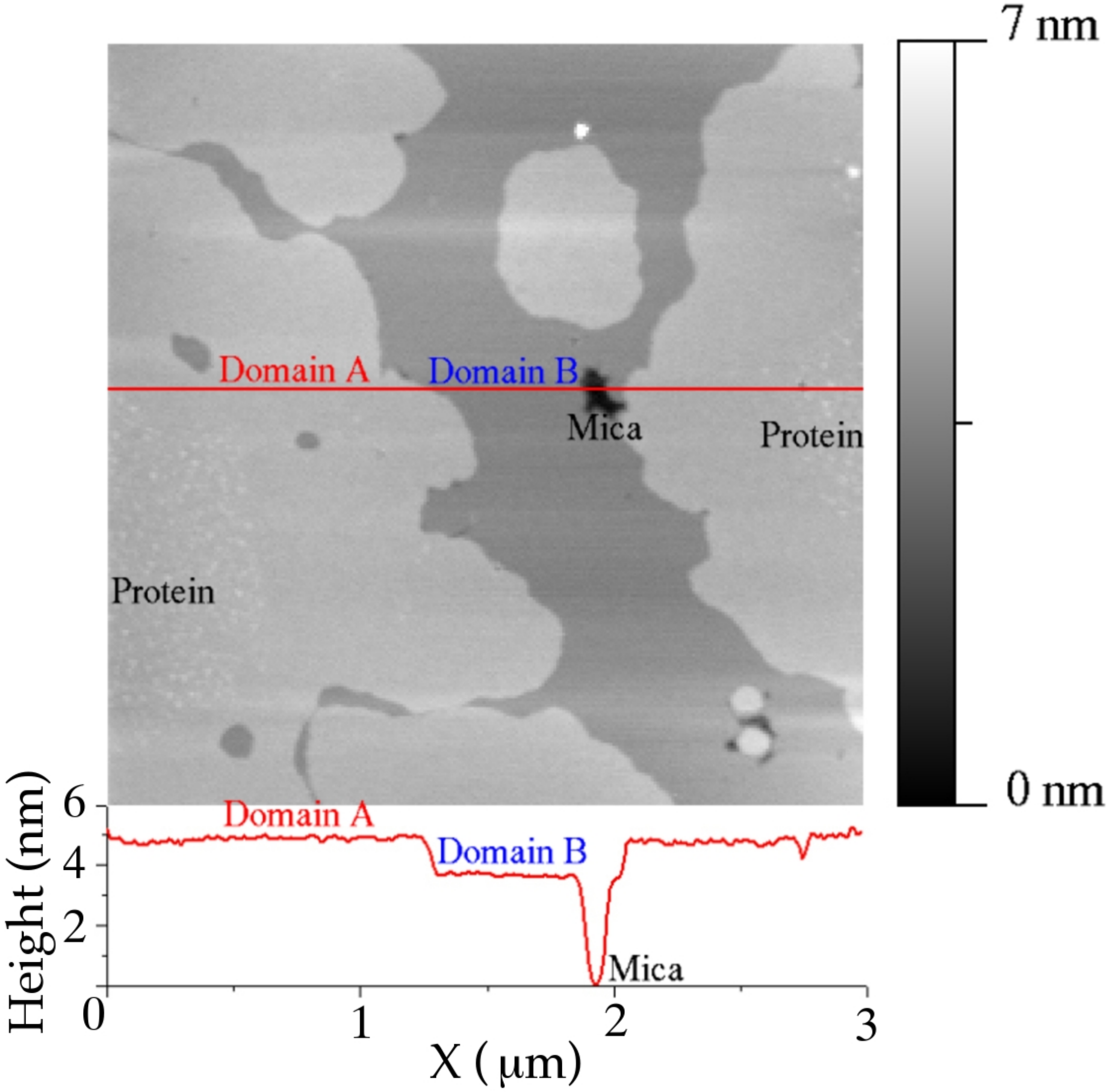


Figure 2.2 Survanta Bilayer
This is an example of a well-formed Survanta bilayer. The image is (3 μm)^2 scan taken at 2 Hz. A line scan, in red, shows the thickness of the primary lipid components. The gel phase (predominantly DPPC) is nominally 5.2 nm, the fluid phase (predominantly POPG) 3.6 nm, and the surfactant proteins and palmitic acid features are 5.6 nm. All heights are relative to the underlying mica substrate

These assignments were made by comparing observed heights to published AFM values for DPPC[126] and a POPG/POPE[118] mixture. The chemical structures of both lipids are shown in Figure 2.3. A defect present in the middle of Domain B reveals the underlying mica substrate shown in black. Finally, the speckled patterns seen in the interior of Domain A are due to the presence of palmitic acid and surfactant proteins as noted by Ding et al.[76, 77] These speckled features are only seen in the interior of large Domains A, while excluded from the edges of Domains A and entirely from Domains B. The observed thickness of Domain A is 5.2 nm, Domain B is 3.6 nm, and the speckled features are 5.6 nm in height.

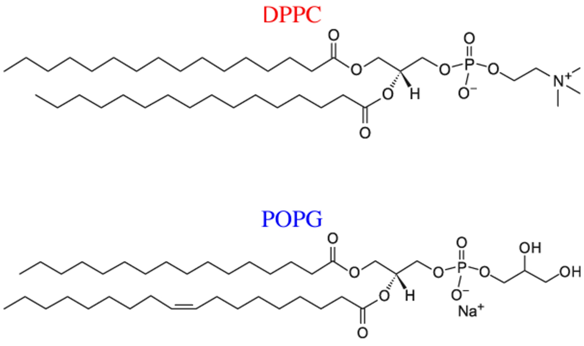


Figure 2.3 Lipid Structure
The structures of the two primary lipid components in Survanta are shown above. The transition temperature of the DPPC is approximately 41 ºC, putting it in the gel phase under experimental conditions. The choline terminal group creates a zwitterionic lipid. The unsaturated acyl chain in POPG leads to a significantly lower transition temperature, 2 ºC, as well as a roughly 1.5 nm difference in bilayer height. The glycerol termination results in an anionic lipid.

Role of fatty acids and proteins

An initial question addressed by these experiments was whether or not fatty acids and proteins present in Survanta could cause surface features that act as nucleation sites for lipid disruption. As shown in Figure 2.2, these features were observed in the Survanta bilayer as speckled regions; however, no activity in or near these speckled regions was detected. This suggests that at these concentrations, interactions between the dendrimer and the fatty acids and proteins do not lead to disruption of the membrane.

Faster AFM scanning

A key improvement in this paper is a significant increase in the imaging speed of the AFM during experiments. Prior efforts by our group have produced images with (1 μm)^2 scan sizes, 256 lines per scan, at 1 line per second.[32-35, 46] In the current work, the limits of these commercial AFM systems have been pushed to increase the tip velocity significantly, allowing for (3 μm)^2 scan sizes, 512 lines per scan, at 3 lines per second. With these imaging conditions, 66% of the lateral resolution of previous works is maintained; however, as the pixel size, 5.9 nm, is significantly less than the radius of curvature of the probes, ~ 20 nm, the same fundamental resolution limit as before exists. For a spherical tip, the smallest feature height or depression that could be resolved by a single pixel is:

$$Feature_{\min} = 2\sqrt{R_c^{\,2} - \left(\tfrac{Pixel}{2}\right)^2}$$

Equation 2.1 Minimum Resolvable Features

For a 20 nm tip with a pixel size of 5.9 nm (these experiments), the feature height resolvable is 0.2 nm. For a 20 nm tip with a pixel size of 3.9 nm (previous experiments), the feature height resolvable is 0.1 nm. As this difference is within the experimental noise of the system, no resolution is lost. The improvement comes from the fact that 9 times the total area is scanned 1.5 times as fast as what was previously done. This greatly enhances the number of events that can be followed within a given field of view, significantly increasing the probability that an intermediate state, such as dendrimer accumulating on lipid, can be observed.

Domain Dependence of PAMAM dendrimer interaction

For the Suvanta bilayer, removal of lipid by the positively-charged PAMAM dendrimers was observed for both Domain A and Domain B; however, Domain B, primarily consisting of a fluid-phase POPG lipid bilayer with a negatively charged phosphatidyl glycerol headgroup, was preferentially disrupted and removed by both G5 and G7 PAMAM dendrimer.

In Figure 2.4, almost exclusive removal of Domain B by G5 PAMAM dendrimer is observed. Most of the lipid is removed from the edges of existing defects, shown with blue arrows; however, there are two instances of lipid removal (Figure 2.4D), shown with purple arrows, which are not extensions of existing observable defects.

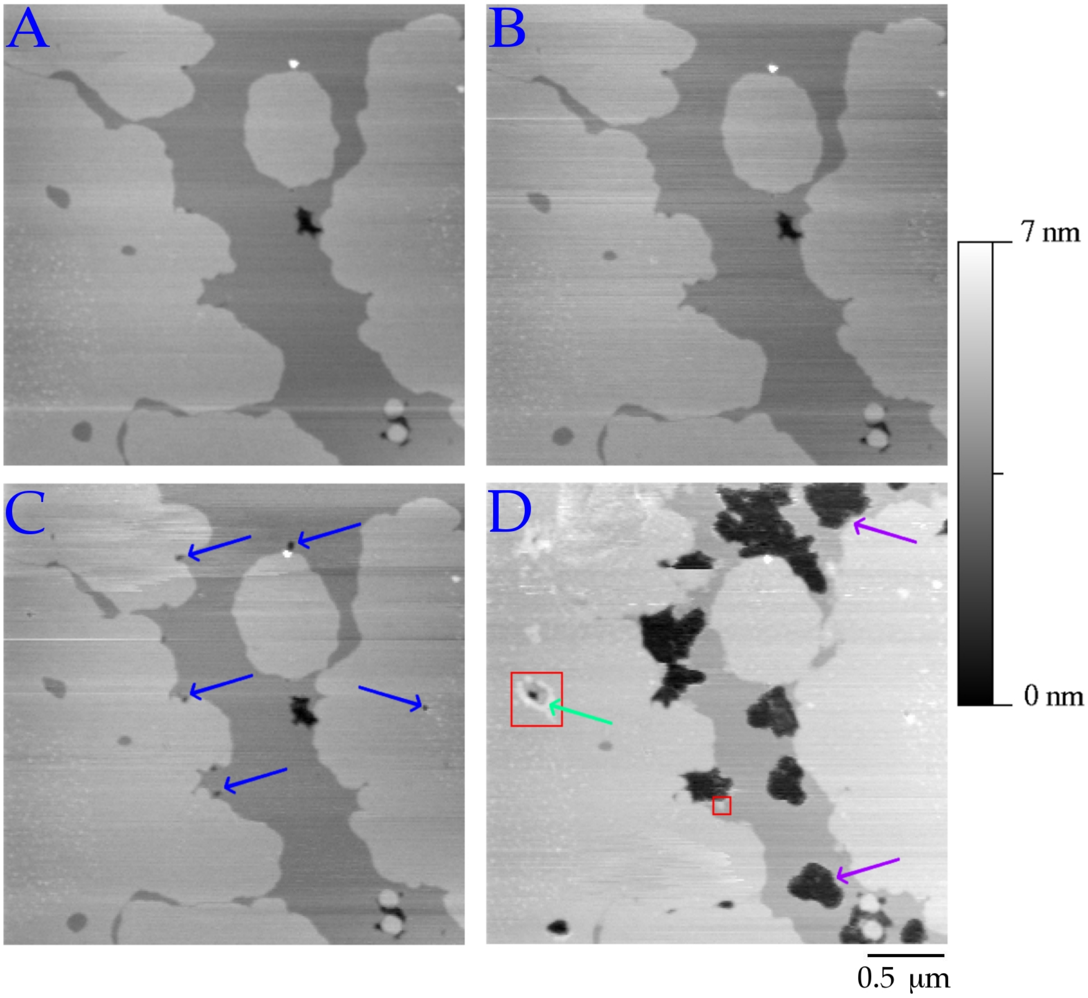


Figure 2.4 Removal by G5 PAMAM
This figure shows the time progression of defects caused by 90 nM G5-PAMAM dendrimer. Image A is approximately 2 minutes before the addition of dendrimer, Images B, C, and D are 2, 4, and 6 minutes following the addition of dendrimer. The blue arrows in Image C show expansion of existing defects by the dendrimer, predominantly in Domain B. Purple arrows in Image D show new defect formation. The teal arrow in Image D shows the accumulation of material around a defect. Red boxes highlight linescan regions used for Figure 2.11.

Figure 2.5 further highlights removal of Domain A by G5 PAMAM dendrimer.

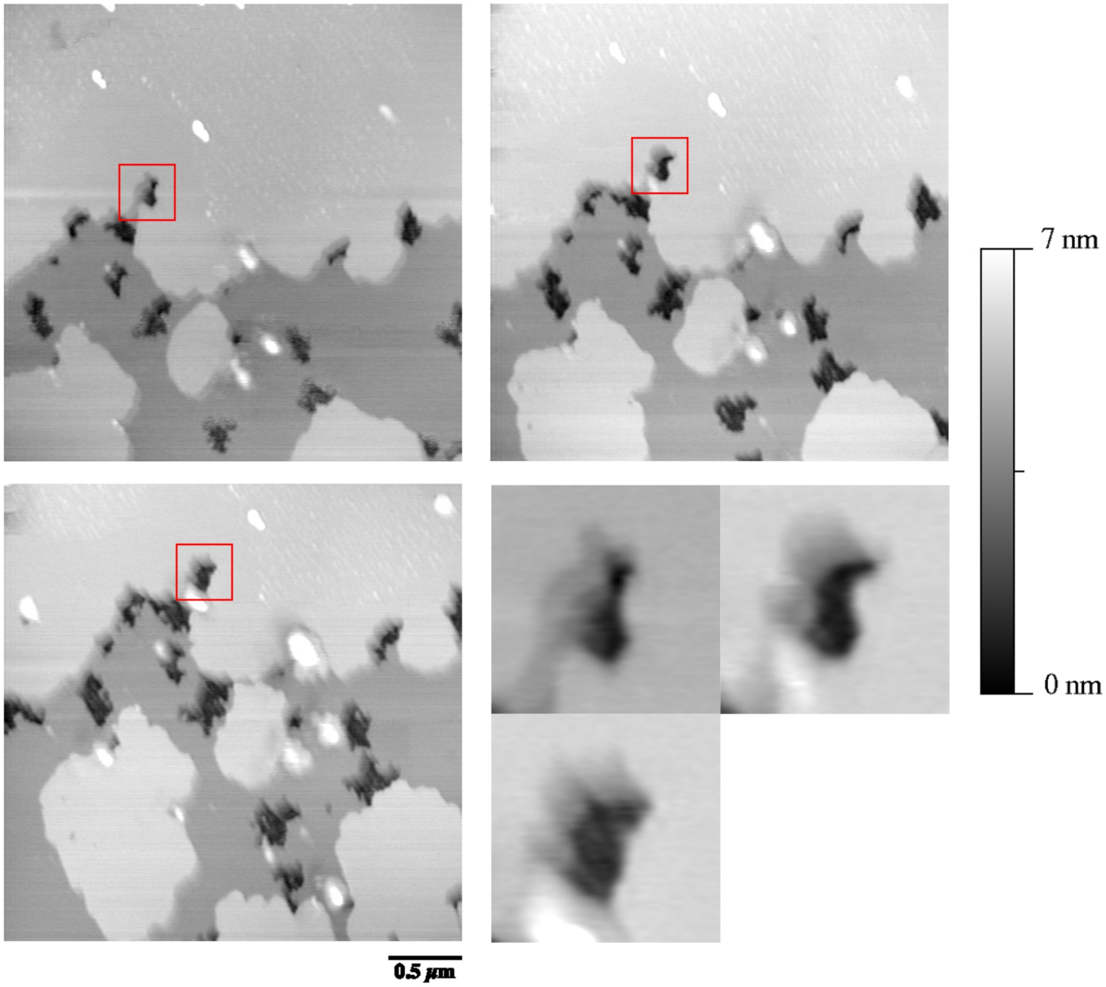


Figure 2.5 Removal Highlighted
This figure shows disruption of a Survanta bilayer by G5 PAMAM dendrimers. The three large images are spaced approximately 9 minutes apart. The scan size is (3 µm)^2 with a 7 nm Z scale. Red boxes highlight regions where Domain A is disrupted by the dendrimer. These regions, displayed in the lower right panel of the figure, are 375 nm wide. The Z values in the magnified images are unchanged from their sources.

Figure 2.6 shows that Domain B is preferentially removed by G7 PAMAM dendrimer. Comparing the amount of lipid removed in a fixed amount of time from Domain A, indicated by blue arrows in Figure 2.6D to Figure 2.6E, to the amount removed from Domain B shows at least an order of magnitude increase.

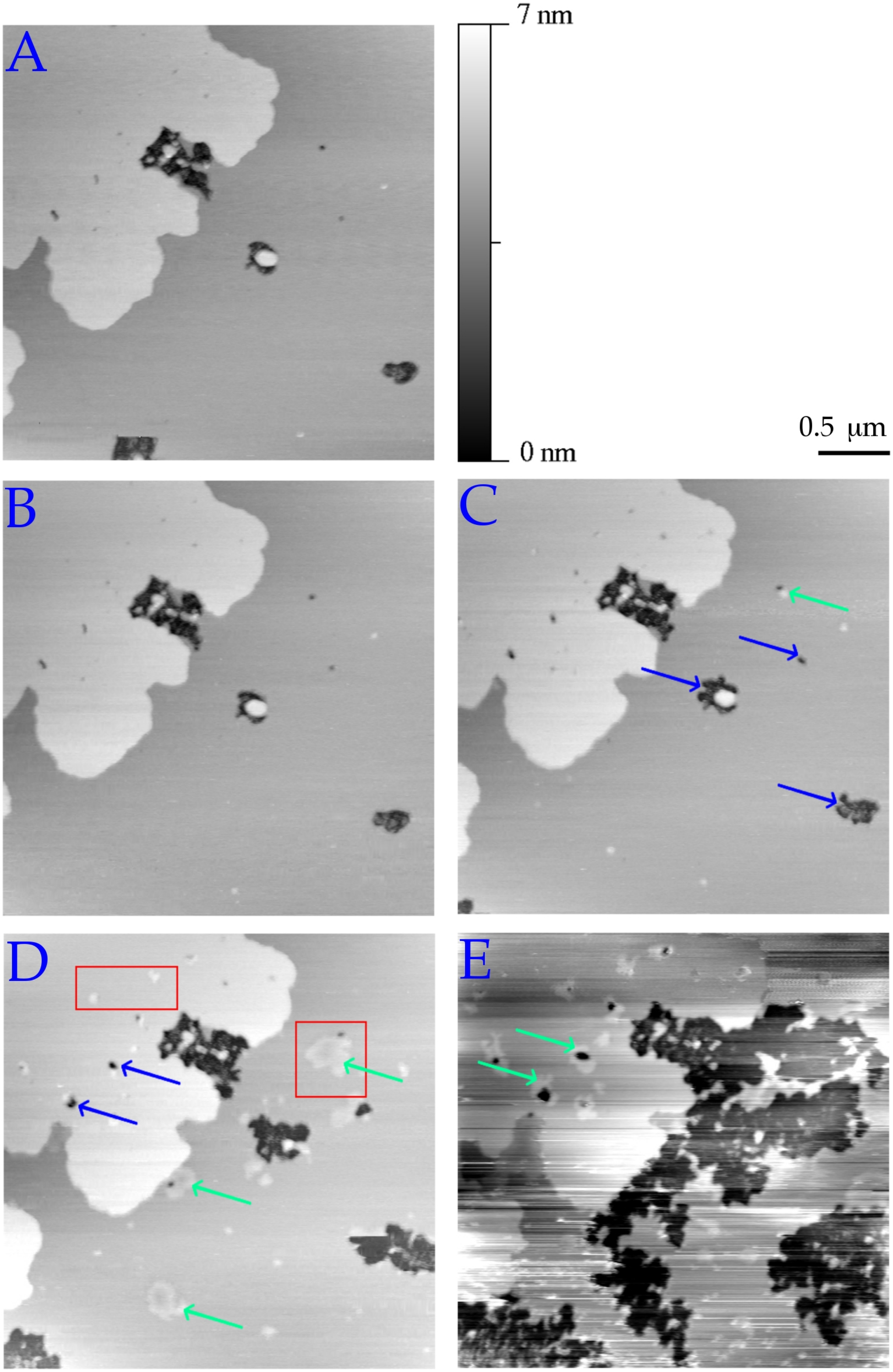

A
7 nm
0 nm
0.5 μm
B
C
D
E

Figure 2.6 Removal by G7 PAMAM
This figure shows the time progression of defects caused by 110 nM G7-PAMAM dendrimer. Image A is approximately 4 minutes before the addition of dendrimer, Images B, C, D, and E are 8, 12, 16, and 20 minutes following the addition of dendrimer. The blue arrows in Images C and D again show expansion of existing defects. Teal arrows in Images C and D indicate an accumulation of material that precedes lipid removal. Red boxes show regions used for line scans for Figure 2.14.

Experimental limitations prevent examination of the steady-state equilibrium of the system, thus precluding conclusions regarding the overall thermodynamics of the interaction. However, it is clear that the energy barrier to remove lipid, which sets the kinetic rate, is much higher for Domain A than Domain B. This could be due to a shielding effect from the positively charged choline termination reducing interaction with the phosphate group, the increased ordering of the gel phase DPPC or, most likely, a combination of the two.

We believe that a combination of electrostatic attraction and hydrophobic effects drive the disruption of the membrane. In previous reports, exclusive removal of fluid domains, similar to Domain B, was observed.[32, 34] In this experiment, significant removal of gel phase material has been observed because of the presence of two large, distinct domains which have transition temperatures significantly above and below experimental temperatures. In prior reports, imaging was done near the transition temperature of DMPC which led to a smaller fraction of material in the gel phase, which reduces the probability of observing an interaction with the gel domain.[127, 128]

Disruption Models

Lipid removal by charged polymers generally falls into one of two categories, adherence and disruption. In adherence mechanisms the charged polymer, in this case

PAMAM dendrimers, attaches to the lipid head groups through electrostatic interactions and the hydrophobic effect maximizes exposure between the hydrophobic tails of the bilayer and the hydrophobic core of the dendrimer. These forces disorder the membrane and allow for lipid removal. Disruptive methods will either disrupt the head groups, tails, or, as suggested by Gewirth and colleagues, the electrostatic interactions between the bilayer and mica.[78, 120] Regardless of the mechanism of disruption, these methods do not always result in encapsulation of the polymer. In Figure 2.7, lipid removal from the fluid phase by G7 PAMAM dendrimer (nominally 512 surface amines) is shown in two (3 μm)^2 scans approximately two minutes apart.

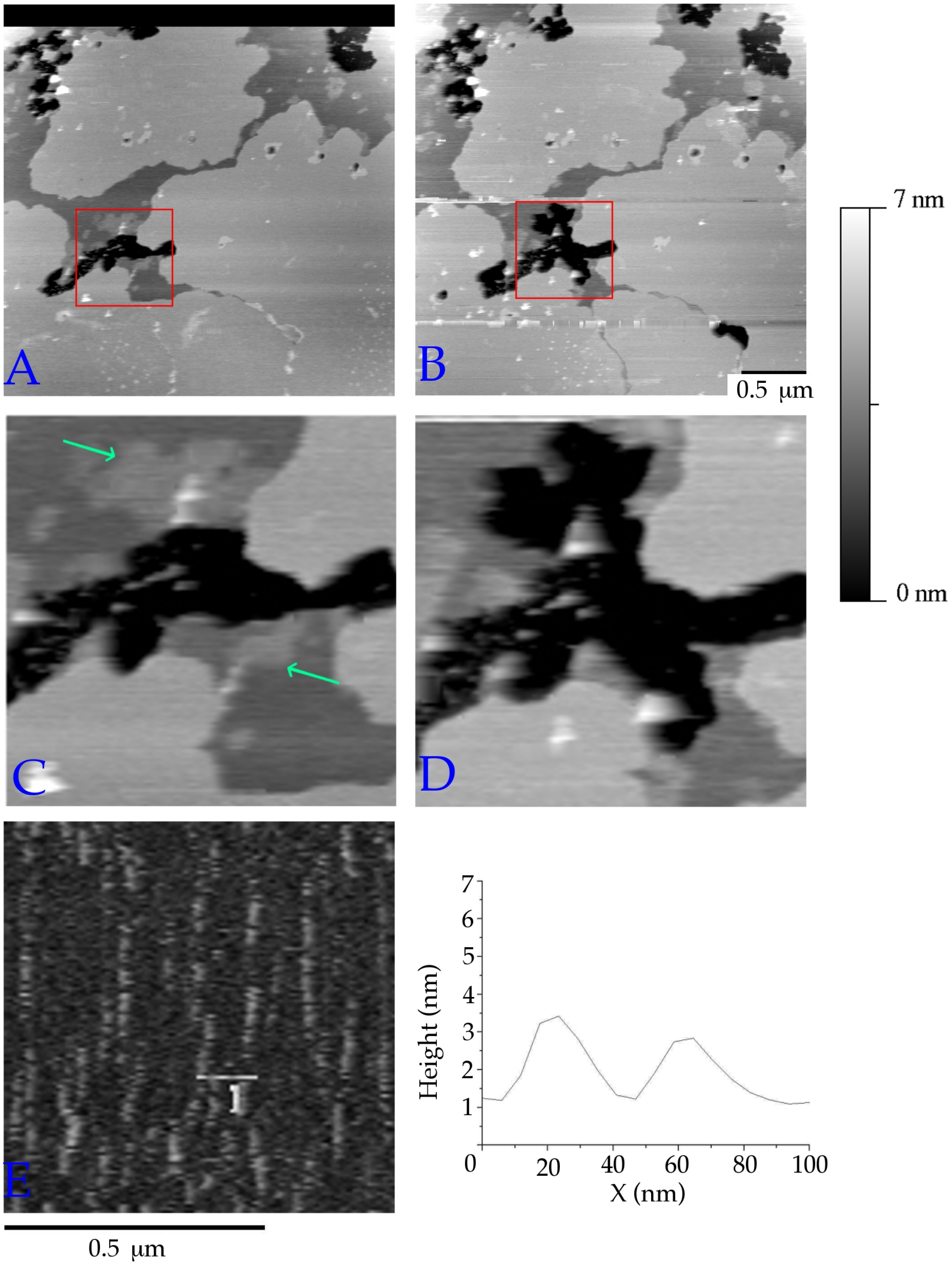

A
B
0.5 μm
7 nm
0 nm
C
D
E
0.5 μm
Height (nm)
X (nm)
0
1
2
3
4
5
6
7
0
20
40
60
80
100

Figure 2.7 Clean Removal of Lipid
In this figure, lipid removal by 110 nM G7-Amine is illustrated. All images, including the line scan, share a 7 nm height scale. Images A and B are approximately two minutes apart. Red boxes highlight the regions for Images C and D, which have been magnified by a factor of four. Teal arrows indicate regions of accumulated dendrimer that precede lipid removal. No dendrimers are observed on mica surface immediately following lipid removal. E: image of G5-Amine dendrimer on mica at the same imaging speeds and scan size as before. 40 mL of 1.1 mM G5-Amine was spin coated onto freshly cleaved mica preceding tapping mode imaging in water. The observed streaking is consistent with work done by Betley,[22] which found low pH ~1 was needed to abolish streaking and fully adhere the dendrimers to the mica. The image has been cropped to the same scale as images C and D. The grey line labeled "1" indicates where a linescan was taken to determine dendrimer size. Bottom left: line scan across the dendrimers. They are between 2 and 3 nm in height and 15 nm in with. The width of the peaks shown in the linescan is ~ 36 nm. This is due to the radius of curvature of the tip (20 nm).

Looking at the lipid removed between images Figure 2.7C and Figure 2.7D at the upper edge of the defect, highlighted with a teal arrow, shows that the lipid is removed cleanly. That is, only the smooth mica that underlies the bilayer is observed. Previous studies have shown G5 dendrimer adhering to mica substrate are approximately 1 to 2 nm tall and 10 to 15 nm in diameter.[129] If lipid were removed because the dendrimer bound to the mica and displaced the lipid, the dendrimer should remain attached to the mica. At the size scale in Figure 2.7, if the dendrimer was still attached to the mica a rough texture would be observed. The lack of this textured surface leads to the conclusion that lipid removal is not due to displacement by dendrimer. As a control, G5 dendrimer, was spin coated onto freshly cleaved mica. The sample was then imaged via tapping mode in water at the same (3 μm)^2 scan size and rate as before. Features between 2 and 3 nm tall and 15 nm in diameter are present. This agrees with previous reports. G5 was used as a control for this experiment specifically because the smaller size, relative to G7, would make it more difficult to observe. Because of this, G7 should have been observable in

Figure 2.7D. This provides confidence that the lipid removal is not due to the dendrimer interacting strongly with the mica substrate and displacing the lipid. Additionally, regions of lipid removal by G7 dendrimer from Figures 5 and 6 have been rendered in 3D in Figure 2.8 and Figure 2.9 to further elucidate this point.

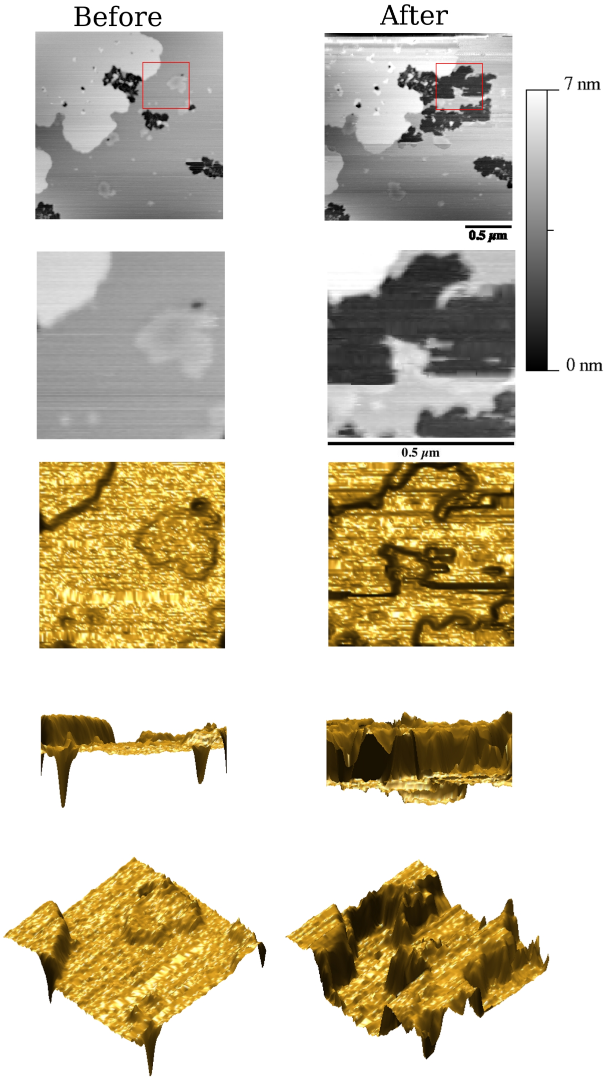


Figure 2.8 Removal by G7 PAMAM: 3D Rendering #1
This figure shows disruption of a Survanta bilayer by G7-Amine terminated dendrimer. The “Before” image (upper left) is the same as Figure 6D. The “After” image is the next

image acquired. Red boxes highlight where dendrimer accumulates on the lipid prior to removal. These areas are shown below in both a 2D representation (Row 2) and three 3D views. The angles of rotation are the following: (0,0), (0,-π/2), and(π/4,-π/4). These views will be used all subsequent 3D renderings.

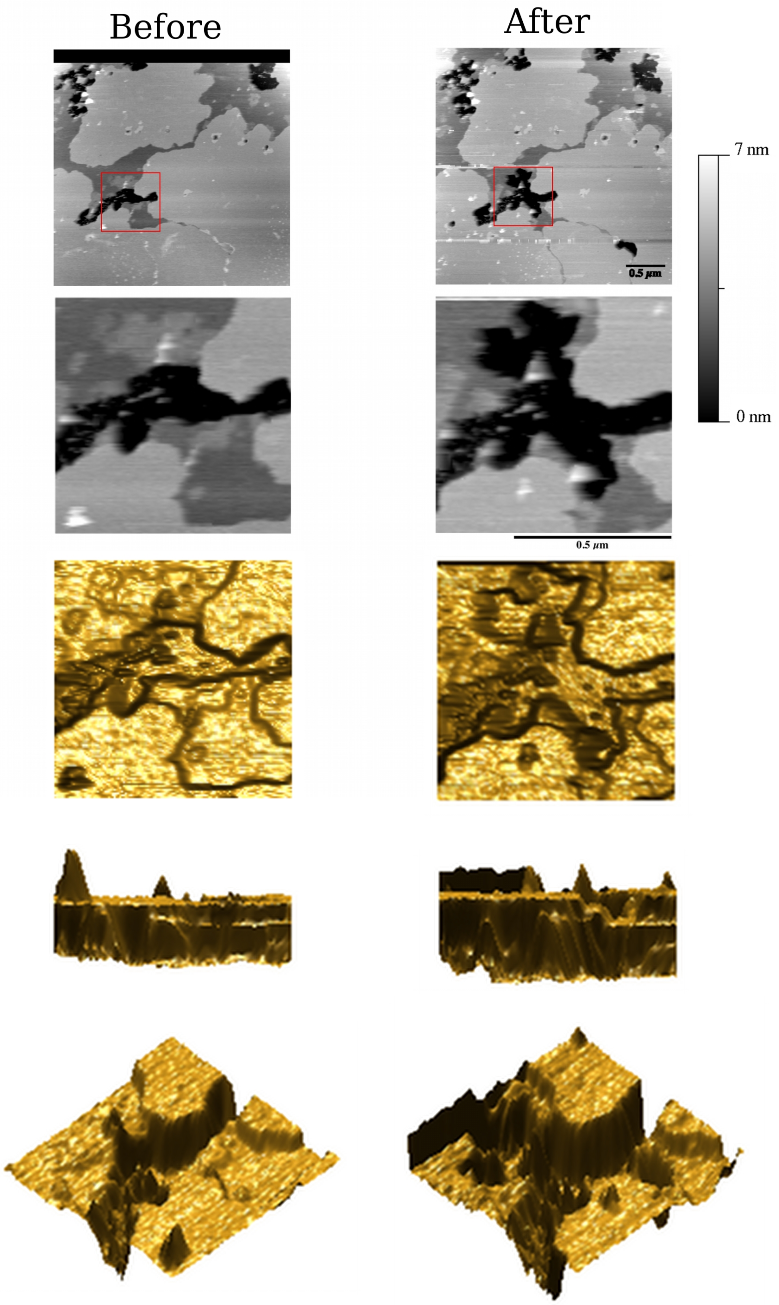


Figure 2.9 Removal by G7 PAMAM: 3D Rendering #2
This figure shows disruption of a Survanta bilayer by G7-Amine terminated dendrimer. The 2D representations are the same as in Figure 2.7. The same region of interest as in Figure 7 has been represented in 3D to aid the reader.

In previous studies, G3 and G5 PAMAM dendrimers have been observed accumulating near the edges of preexisting defects in supported DMPC bilayers, but not with G7.[46] In Figure 2.4 and Figure 2.6, dendrimer accumulation preceding hole expansion is highlighted with teal arrows and red boxes. Since very little material accumulates prior to removal, faster imaging speeds and larger image acquisition allows us to capture the intermediate state—where material has accumulated but not yet been removed—more often. Figure 2.10 shows the phase deflection image of Figure 2.6D and a line scan through Domain A, Domain B, and material that has accumulated on top of both domains.

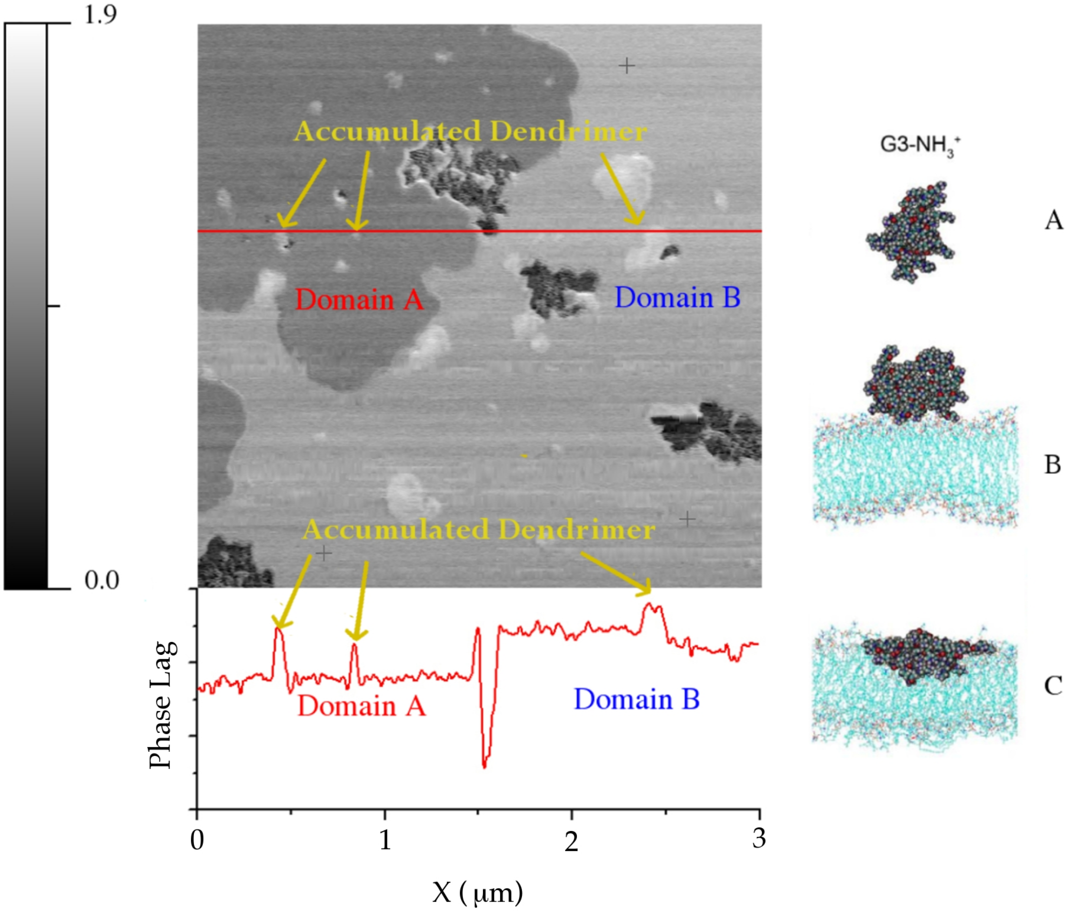


Figure 2.10 Phase Imaging
This figure shows the phase image during Figure 2.6D. The phase difference measures the phase lag between the oscillation driving the cantilever and the readout at the split photo diode. Hard, perfectly elastic materials, like mica, appear black, while softer and stickier surfaces will have a larger phase lag and will thus be lighter. An advantage of phase imaging is that it can be used to differentiate materials. The phase lag in Domain A is significantly lower than in Domain B, presumably due to the gel nature of Domain A. More importantly, the regions corresponding to accumulated dendrimer show an additional phase lag from the surrounding lipid in both Domain A and Domain B. This suggests that the dendrimer is either sitting on top of the lipid, or partially intercalated into the hydrophilic region yet still somewhat exposed. On the right, are images from a molecular dynamics simulation showing the interaction of a G3 PAMAM dendrimer with a DMPC bilayer. In Image A the dendrimer is effectively not interacting with the bilayer. In Image B, the dendrimer is interacting with a gel DMPC bilayer (similar to domain A). In Image C the dendrimer is interacting with a fluid DMPC bilayer.

Phase imaging can be used to distinguish between different materials because “stickier” and “softer” (less elastic) materials will cause a larger phase lag between the driving signal and the deflection signal. In this color scheme, brighter values indicate a stickier or softer material. In this image, regions of dendrimer accumulation, originally identified by topography, are, in fact, regions of different exposed material. Moreover, when compared to the results of molecular dynamics simulations of G3 PAMAM dendrimer interacting with a bilayer made from 1,2-dimyristoyl-sn-glycero-3-phosphocholine (DMPC) additional evidence for why the dendrimer should be interacting with the upper leaflet of the bilayer is presented. The simulation images, (right side of Figure 2.10) show how both the dendrimer and the bilayer deform to maximize their interaction. The increased disorder of the fluid layer explains why the observed phase lag between Domain B and the G7 PAMAM dendrimer is smaller than for the dendrimer and Domain A.[79]

In order to understand what is going on in these regions where the dendrimers accumulate, boxed regions in Figure 2.4D and Figure 2.6D have been selected for closer examination. 3D renderings of these regions are provided in Figure 2.11, Figure 2.12, and Figure 2.13. These regions were selected because they show accumulation of material that precedes lipid removal and these regions are elevated above the underlying lipid with a distinct edge.

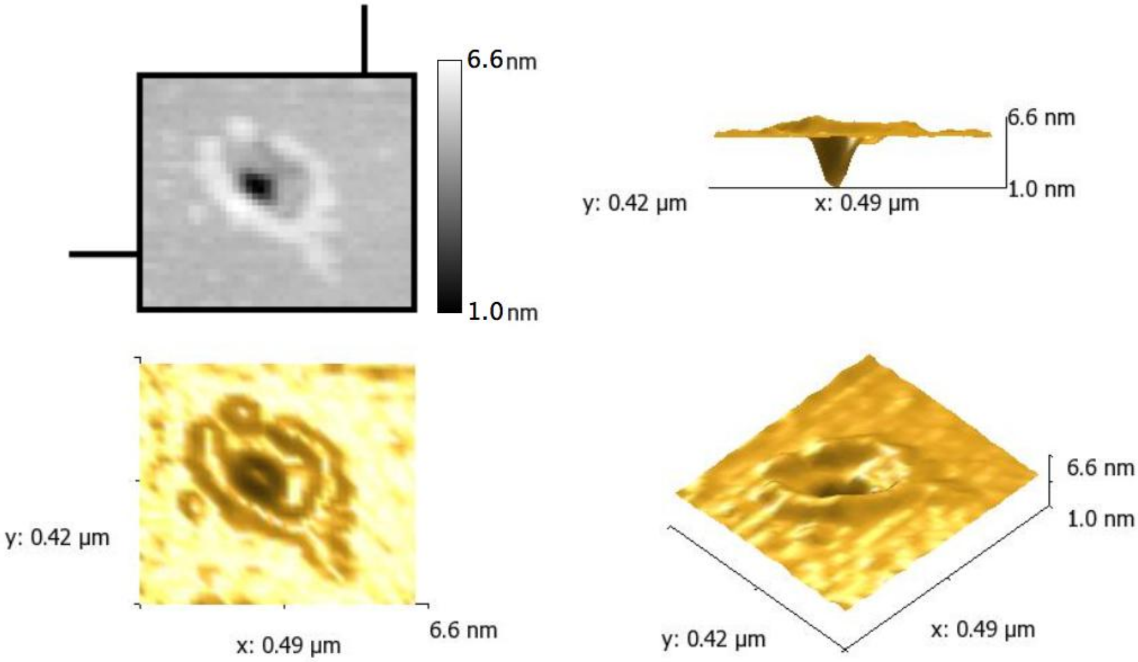


Figure 2.11 3D Rendering of Accumulation #1
This figure is a three dimensional rendering of the defect boxed in red on the left in Figure 2.4. Significant accumulation of material on Domain A is shown in this rendering.

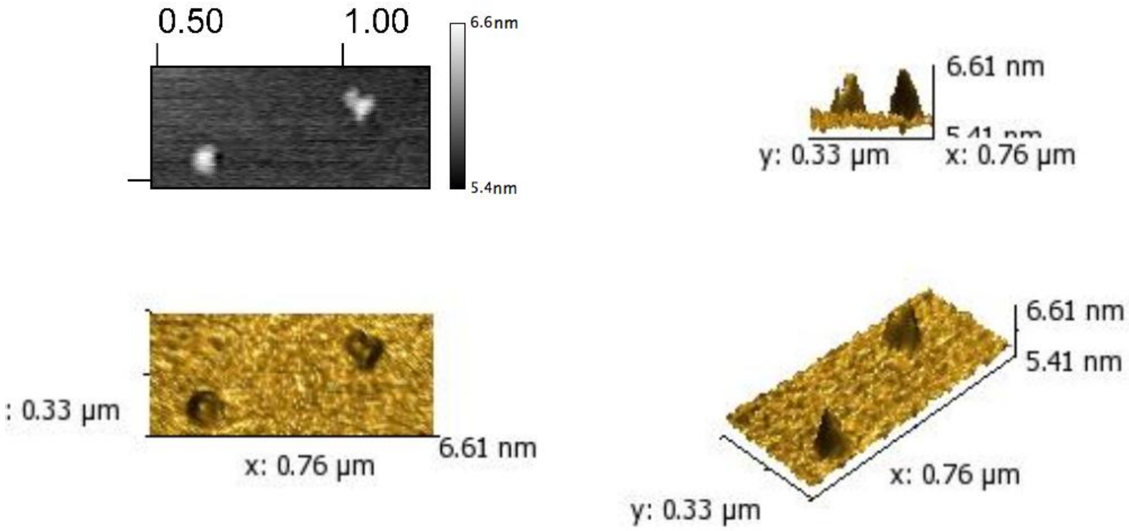


Figure 2.12 3D Rendering of Accumulation #2
This figure is a three dimensional rendering of the defect boxed in red on the left of Figure 2.6D. These raised areas are of similar height to what is seen in Domain A with a G5 dendrimer in Figure 2.4. As in Figure 2.4E, accumulation occurs around existing defects and precedes lipid removal. The rate of this removal is much slower than what is observed in Domain B.

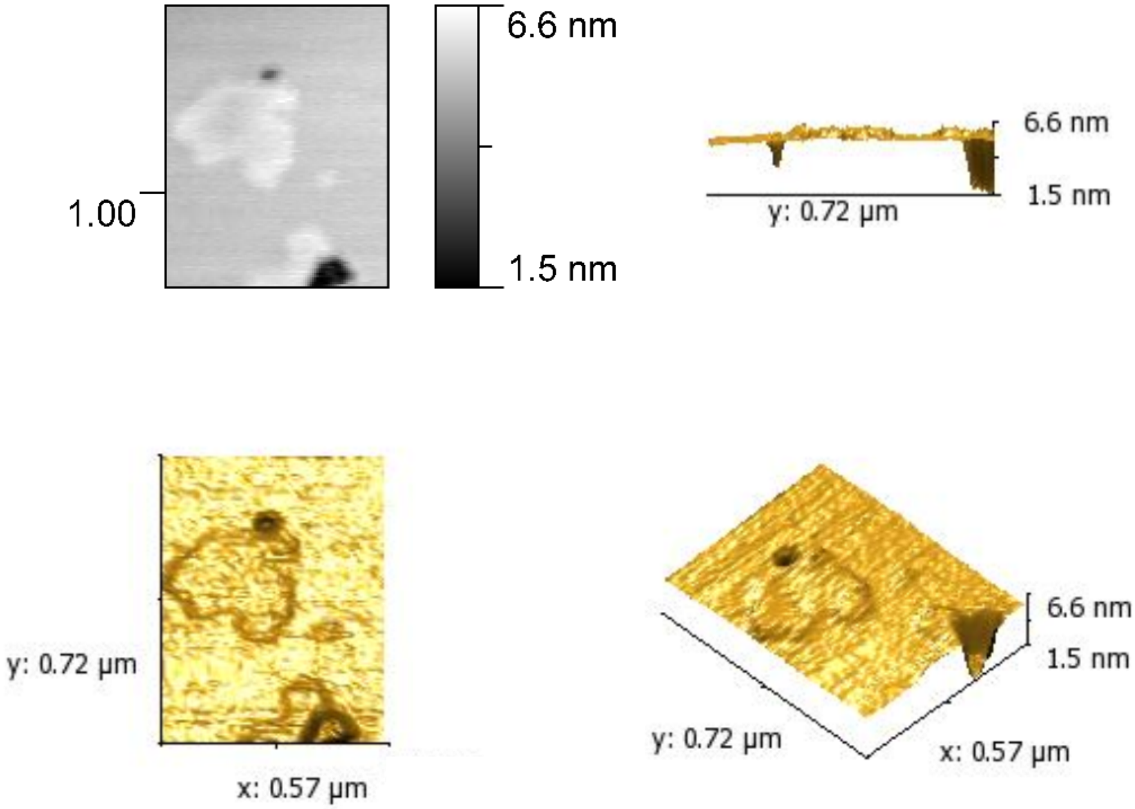


Figure 2.13 3D Rendering of Accumulation #3
This is a three dimensional rendering of the defect boxed in red on the right of Figure 2.6D. The area of accumulation is much greater than what is seen with any other combination of dendrimer and domain. These regions are removed completely in Figure 2.5E, which suggests that the dendrimers have to work in a concerted fashion to remove lipid.

Multiple line-scans have been taken across these edges in the boxed regions in Figure 2.4D and Figure 2.6D so that the line scans start the same distance from the edge. The line scans were averaged to generate representative profiles. This allows for examination of the "step height" going across this edge from one lipid domain to the accumulated material on top of it, in terms of both generation and underlying domain.

The step height is plotted in Figure 2.14 with the height of the reference domain set to zero.

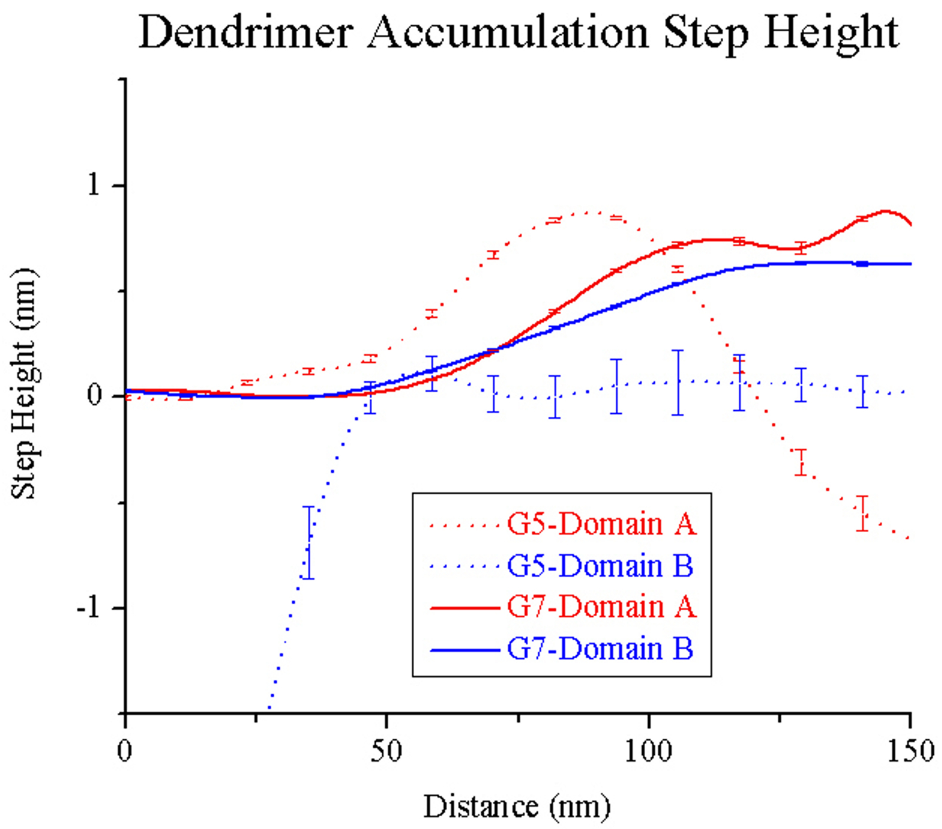


Figure 2.14 Dendrimer Accumulation Step Height
Highlighted regions in Figure 2.4 and Figure 2.6 were selected because they showed raised edges preceding lipid removal. Between 5 and 18 line scans were taken across each region and averaged to generate the figure. Each line scan has been positioned so the step from lipid to the accumulated material occurs at the same position. In each case, the height of the reference lipid has been set to 0. For G5-Domain A, G7-Domain A, and G7-Domain B the reference lipid is on the right and the accumulated material on the left. For G5-Domain B the scan goes from a hole at the edge of the accumulated materiel across the step and down to the reference lipid. The lines scans were taken parallel to the fast scan axis of each AFM image. Error bars are the standard error at each location (pixel). For G5 interacting with Domain A, and G7 interacting with both Domain A and Domain B a significant accumulation of material greater then 0.5 nm indicates this is not a phase transition.

What is interesting about this plot is the difference in slope between Domain A and Domain B is independent of dendrimer generation. Factoring in the higher step heights—summarized in Table 2.1—for Domain A and Domain B, the dendrimer, then, should be able to spread out more and disturb more of the lipid bilayer possibly explaining why lipid is removed faster from the Domain B than Domain A.

	Domain A	*Domain B*
G5	0.85 ± 0.01 nm	0.11 ± 0.08 nm
G7	0.84 ± 0.01 nm	0.63 ± 0.01 nm

Table 2.1 Summary of Accumulation Step Heights
This table shows the accumulation height at the leading edge of each materiel. Statistically identical heights in the gel phase for G5 and G7, yet much smaller values for Domain B than Domain A indicates that the ability of the dendrimer to reach out and interact with the lipid and or penetrate the bilayer is mediated primarily by the fluidity of the layer and not electrostatic interactions. The stated error for G5-Domain B is from the standard error of the data set. For the rest the calculated error was below significance for both instruments so the error was rounded to the nearest significant figure.

These conclusions match the molecular dynamics results shown in Figure 2.10 Phase Imaging where the dendrimer on fluid DMPC is more extended and deformed than on the gel layer.[79]

Considering the information from the phase images, the inaccessibility of the hydrophobic tails to the dendrimer, molecular dynamics simulations showing the dendrimer is on top of the bilayer, our evidence indicating that the dendrimers do not accumulate between the bilayer and mica substrate, and the morphology of the accumulated regions, the conclusion is that the dendrimers are sitting on top of the lipid bilayer or partially intercalated into the hydrophilic head region.

Conclusions

Clinical lung surfactants, Survanta being one example, provide opportunities to study aspects not traditionally captured in single component lipid bilayer experiments. Questions regarding lipid phase, composition, roles of fatty acids and proteins can all be examined in a relatively controlled environment. In the case of polycationic nanoparticle induced disruptions, lipid-polycation interactions still appear dominant. The data presented supports lipid removal caused by dendrimers adsorbing onto the upper leaflet and interacting strongly with the hydrophillic head groups, disrupting the lipid. For the first time, the adsorption of material has been observed preceding lipid removal in all cases, the rate of which is governed by the underlying domain phase.

An oft asked question is how can nanoscale results be related to global behavior? In this case, how would the domain segregation and preferential disruption of Domain B affect the behavior of the surfactant layer in a living animal? To answer this question, the authors direct the reader to an excellent book, Lung Surfactant Function and Disorder, edited by Kaushik Nag.[130] This book features a very thorough review of the composition, function and structure of surfactants from the nanoscale through the macroscale. The book relates how physical properties at all scales support proper lung function, and how disruptions can lead to diseased states. The reader will find chapters six, ten, and twelve particularly relevant.

Chapter 3

Bone Methods

Introduction

The collagen family are the most abundant structural proteins in animals.[3] These proteins are based on trimeric polypeptide chains, each of which includes a repeating Gly-X-Y triplet region (X and Y are often proline and hydroxyproline, respectively). A major class of collagen is the fibrillar-forming type which has an approximately 300 nm long, uninterrupted triple helix.[4] The most common form of collagen, Type I, is found throughout the body as fibrils in teeth, bones, tendons, skin, arterial walls and the cornea[5], and is synthesized in response to injury.

At the nanoscale, the most readily quantified morphological characteristic of Type I collagen fibrils is the D-periodic axial gap/overlap spacing. This feature is thought to arise from the staggered parallel alignment of collagen molecules within the collagen fibril, resulting in "gap" and "overlap" regions.[131] Known as the Hodge-Petruska model, this repeating pattern of gaps and overlaps results in an oscillating surface topography with a 67 nm periodicity based on theoretical models of collagen fibrils in isolation.[131] Since this original assertion, X-ray diffraction[132] and electron microscopy[133, 134] studies have supported a mean spacing of 67 nm; however, limited analysis did not reveal the underlying distribution. Despite this commonly held view of a singular spacing, the hierarchical complexity of the collagen fibril itself, the variety of tissues in which these fibrils are incorporated, and the potential for morphology variation with damage and disease, led us to examine the distribution of D-Period spacings more closely.

Developing techniques to characterize and quantify morphological features in Type I collagen-based tissues is imperative to understanding normal tissue architecture,[135] a necessary step in deciphering how damage and disease may alter these nanoscale morphological features. These methods may also serve as surrogate techniques for disease diagnosis in collagen-based tissues.[136] The current study details a systematic method developed to accurately measure the D-periodic spacing in Type I collagen, and addresses the uncertainties associated with such measures. The distributions of morphologies that are present in surface-based techniques have been further compared to X-ray data. Statistical methods have been developed and employed to compare distributions between normal and diseased tissues, demonstrating the utility of such methods for disease diagnosis.

Materials and Methods

Animals

The Les laboratory at the Henry Ford Hospital (Detroit, MI) conducted all the animal studies. Samples were graciously provided to support this work.

Five year-old Columbia-Ramboulliet cross sheep were anesthetized and ovariectomized (OVX, n=3) or subjected to a sham surgery (n=3) [Colorado State University, ACUC # 03-010A-02]. After 2 years, the ewes were sacrificed with an intravenous overdose of a barbiturate, and 2 beams approximately 1.75 mm x 1.75 mm x 9 mm were removed from the mid-diaphysis of the left radius as previously described.[83]

Atomic Force Microscopy (AFM) Imaging and Analysis

Dr. Joseph Wallace performed AFM sample preparation, imaging, and biological experimental design within our laboratory. Image analysis design and implementation was done in collaboration with Dr. Wallace.

Bones were processed and imaged as previously described.[136] Briefly, bones were mounted to a steel disk using a thin layer of cyanoacrylate glue. A flat polished surface was created using a 3 μm polycrystalline water-based diamond suspension and a 0.05 mm deagglomerated alumina suspension (Buehler LTD; Lake Bluff, Il), then sonicated for 15 seconds to remove polishing residue and debris. To remove extrafibrillar surface mineral and expose underlying collagen fibrils, the surface of each bone was demineralized using 0.5M EDTA at a pH of 8 for up to 45 minutes, then vigorously rinsed with ultrapure water and sonicated to remove any surface bound mineral.

Samples were imaged in air using a PicoPlus 5500 Atomic Force Microscope (AFM, Agilent) in tapping mode using a silicon cantilever with tip radius < 10 nm (VistaProbes T300R, force constant 40 N/m, resonance frequency 300 kHz) at line scan rates of 2 Hz or lower at 512 lines per frame. Images were acquired from at least 3 axial locations in each sample. At each location, 3.5 μm x 3.5 μm amplitude (error) images were analyzed without leveling to investigate the D-periodic axial gap/overlap spacing (SPIP v4.8.2, Image Metrology; Hørsholm, Denmark). Following image capture, a rectangular region of interest (ROI) was chosen along straight segments of individual fibrils (Figure 3.1). The selected regions spanned consistent topographical features (i.e. from gap region to gap region and though the middle of a given fibril). For each evaluated fibril, a two dimensional Fast Fourier Transform (2D FFT) was performed and the 2D power spectrum was analyzed to determine the value of the D-periodic gap/overlap spacing. One dimensional power spectra through the primary fibril orientation were also acquired for future analysis.

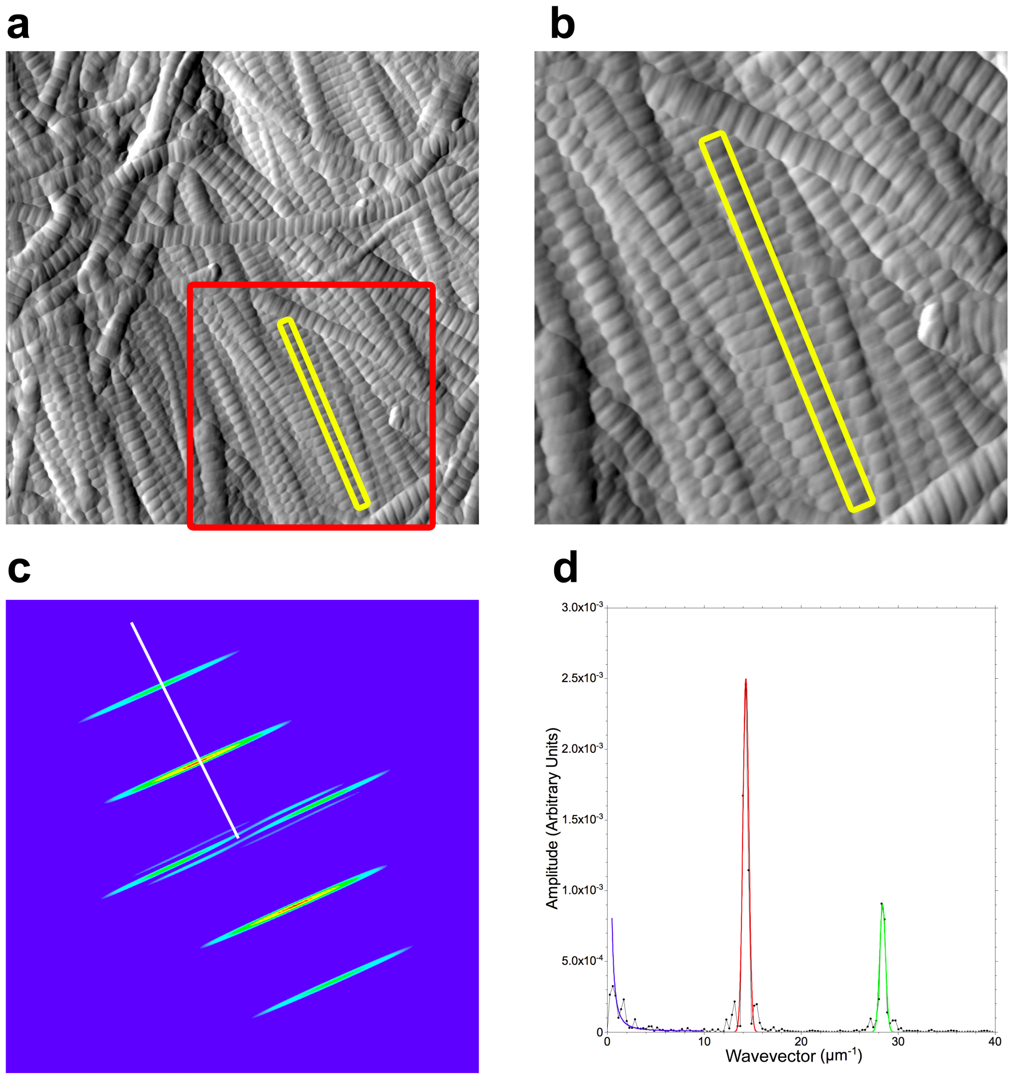


Figure 3.1 Schematic Representation of the 2D-FFT Process
Panel A shows an AFM amplitude image of a typical collagen sample (bone in this case). The D-periodicity is visible as a striped patterned perpendicular to the fibril axis. A fibril of interest is denoted with the yellow box. The region cropped for a 4x enlargement in Panel B is shown in red. The yellow box in Panel B highlights the same fibril marked in panel A. Panel C shows the 2D FFT of the selected fibril. The white line runs through the maximum value of the first peak, which corresponds to the D-Periodic spacing. Panel D shows the 1D-FFT along this line, normal to the D-Periodic spacing and through the maxima in the 2D power spectrum. The first two harmonics are visible and can be fitted numerically to determine the D-Periodic spacing and the associated uncertainty. The fit to the D-Periodic spacing is shown in red, the second harmonic in green, and the low frequency noise in blue. Dr. Joseph Wallace acquired AFM images in Panels A and B

Computation

Model collagen fibrils were constructed in Matlab (The Mathworks: Nattic, MA) as a summation of repeating Gaussians. The power spectrum of these model fibrils was computed in Matlab and analyzed to determine the D-Periodic spacing (the peak of the power spectrum) and D-Periodic uncertainty (the standard deviation of the distribution of D-Periodic spacing observed in the power spectrum). The results were further analyzed and compared to results obtained from SPIP on real fibrils experimentally imaged using AFM.

Statistical Analyses

Following the determination of the D-Periodic spacing of each fibril, various statistical analyses were performed using Matlab. To investigate differences in fibril morphology due to estrogen deficiency, D-periodic spacing values measured from an individual sheep sample were pooled, yielding an average value for that sample, and then statistically compared using One Way ANOVA. Histograms were computed using a 1.0

nm bin size. To examine differences in the distribution of fibril morphologies between sham and OVX groups, the Cumulative Distribution Function (CDF) of each group was computed. The CDF shows what fraction of a given sample is contained up to a particular value, easily demonstrating differences between distributions in both mean and standard deviation. To test for statistical significance between distributions, two-sample Kolmogorov-Smirnov (K-S) tests were then applied to the data sets. A custom Matlab script was written to run and automate each step of the analysis. This script is included as Appendix B.

Results and Discussion

Measurement of the D-periodic Gap/Overlap Spacing in Type I Collagen Fibrils

The D-periodic axial gap/overlap spacing was chosen as the key metric of collagen fibril morphology in this study. This measure captures aspects of fibril structure, which may be related to the state of the individual molecular triple helices, post-translational modifications and/or cross-linking. Although the molecular origin of these morphological features has not been elucidated, Figure 3.1 illustrates that the D-period is well resolved with techniques such as AFM.

Although this study is not the first to analyze the D-periodic spacing in Type I collagen, previous studies have relied primarily on line scans and 1D FFTs and did not perform quantitative analyses of larger data sets.[36, 42, 137, 138] Both line scans and 1D FFTs suffer from a number of limitations. The first limitation is a question of resolution. Lateral resolution is limited by a combination of factors including the pixel size of the image (~ 7 nm in images acquired for analysis) and the radius of curvature of the probe

(< 10 nm). The periodic nature of the fibril can be exploited with a 1D-FFT to resolve the average spacing below the single pixel limit. The resolution limit of the 1D-FFT is the ability to determine the center of the peak in reciprocal space. The smallest wavevector that can be resolved is:

$$F_{\min} = \frac{F_{sampling}}{2N}$$

Equation 3.1 FFT Frequency Limit

The true reciprocal spacing must lie within ± ½ of a the smallest resolvable wavevector (any more or less would shift the peak into the next bin), so this sets an absolute upper bound for determination of the frequency component. For example, assume we have the following conditions, which are based off of experimental data: a 3500 nm image, with 512x512 pixels, a collagen fibril with a true spacing of 67.0 nm and 9 repeats to measure. To start with, we have a pixel pitch of 6.8 nm, which is sampling at $1.50*10^{-1}$ nm^{-1}. Our fibril is 603 nm long, or 88 pixels long. Our minimum wavevector is then, $8.3*10^{-4}$ nm^{-1}. The reciprocal spacing of our 67.0 nm D-Period spaced fibril is $1.50*10^{-2}$ nm^{-1}. Dividing the true reciprocal spacing by the wavevector spacing says what bin the spacing will fall into, in this case the 18th bin. Looking at experimental data, we could say that the true peak in this case lies somewhere between 17.5 and 18.5 bins. 17.5 bins correspond to a wavevector of $1.45*10^{-2}$ nm^{-1} or 68.8 nm. 18.5 bins correspond to a wavevector of $1.54*10^{-2}$ nm^{-1} or 65.0 nm. Since the spacing falls within the 18th bin, we can say with certainty then that the spacing is 66.8 nm ± 1.9 nm. (This is 28% of a single pixel.)

In practice, resolution is significantly higher than this absolute limit given the signal to noise in the observed peaks. (The peaks in frequency are narrow in comparison to the frequency sampling). In Figure 3.1, the observed harmonics have Full Widths at Half Maximum (FWHM) on the order of 2 frequency units. For a Gaussian distribution, the FWHM is defined as $2(2\ln 2)^{1/2}\sigma$ (FWHM is ~2.35σ), where σ is the standard deviation of the distribution. In Figure 3.1, σ is on the order of 0.85 bins. The ability to determine the center of a well-resolved Gaussian is easily within 1/4 of σ, or ~ 0.2 bins in the case of Figure 3.1. Returning to our example above, the peak center would lie between 17.8 and 18.2 bins. This corresponds to 67.6 nm and 66.1 nm respectively. We can reasonably say then, that our peak is 66.8 nm ± 0.8 nm. (This is 12% of a single pixel) Further refinements could be made to this error with peak fitting; but accuracy to a tenth of a nanometer by simple peak selection is sufficient.

The second, and more significant limitation of line scans and 1D FFTs is that they both rely on the user drawing a line along the length of a fibril, exactly normal to the D-spacing.[136] From 1D FFT measurements on fibrils in the current study, as little as a 5° deviation in this line away from normal can alter the value measured for the spacing by as much as 8% (Figure 3.2).

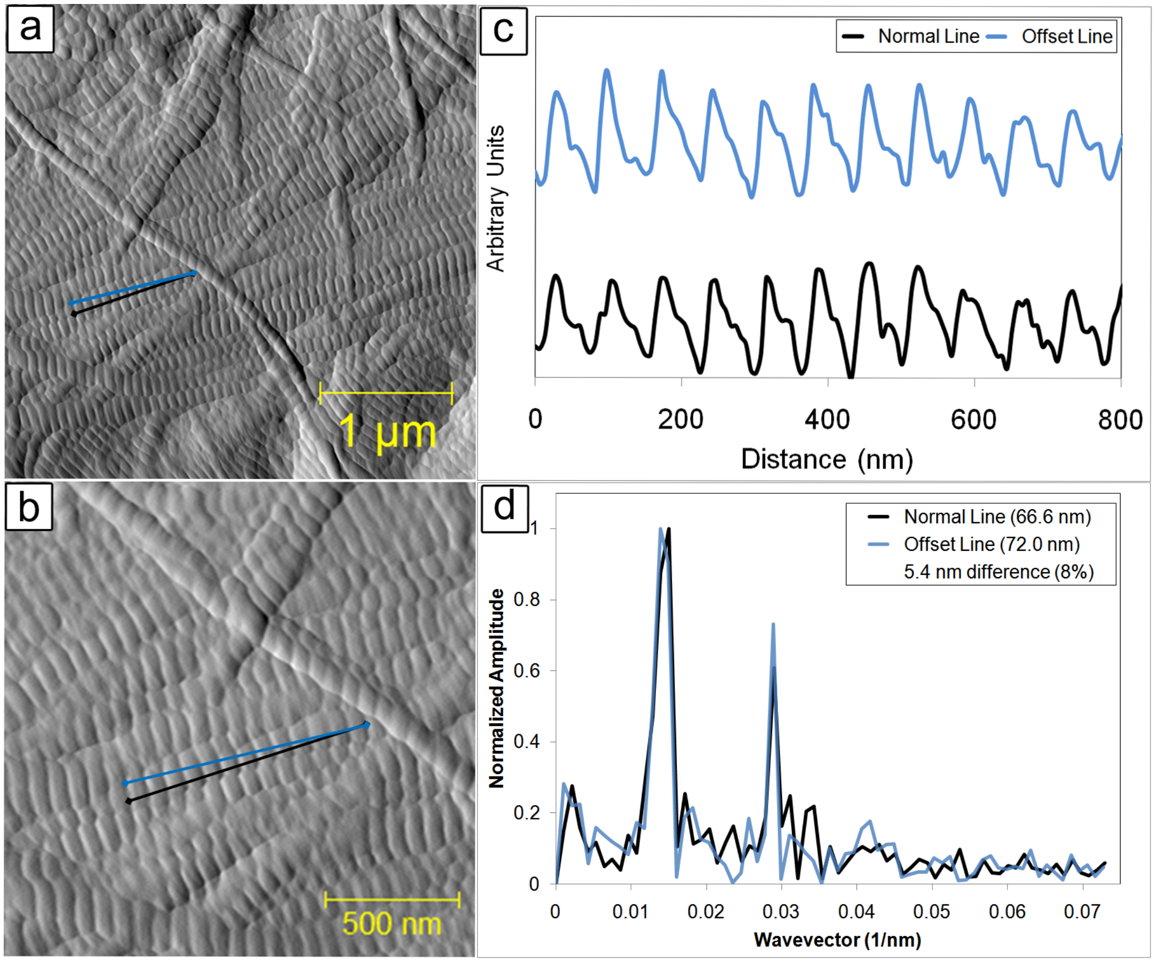


Figure 3.2 Effects of Angle on 1D FFT Measurements of D-Periodic Spacing
This figure demonstrates the effect of changing the angle of the user-drawn line on the D-periodic spacing measurement derived using a 1D FFT. Panel a is a 3.5 µm x 3.5 µm amplitude image showing a fibril chosen for analysis. Panel b is an enlargement of the analyzed area. The same fibril was measured using a line drawn normal to the D-periodic spacing (black) and one with a 5° tilt from normal (blue line). Panel c shows the corresponding line scans from each line in panel b and panel d shows the 1D FFT derived from each line. In this example, there was a 5.4 nm difference in the measured D-periodic spacing (an 8% difference). This Figure was prepared by Dr. Joseph Wallace.

Errors of this magnitude are larger than the population standard deviations noted below, preventing accurate conclusions from being drawn from the distributions. In addition, this error may have resulted in the presence of the distribution not being previously discovered. The main factors that contribute to this discrepancy are the number of repeat units measured (i.e. the length of the fibril) and pixilation caused by finite sampling. Employing 2D FFT measurement methodology decouples determination of the D-Periodic spacing from user bias in angle selection. An example image of a collagen fibril imaged by AFM and measured using a 2D FFT approach is shown in Figure 3.1. Panel A shows an amplitude image from a 3.5 x 3.5 μm scan of a bone surface. Multiple collagen fibrils running parallel to each other can be observed, with the D-Periodic spacing visible as a striped pattern perpendicular to the fibril axis. Panel B shows a 4x enlargement of the boxed region in Panel A, with the rectangular region analyzed by the 2D FFT technique outlined in yellow. In order to minimize edge effects that can degrade resolution, the rectangular box extends from the edge of one gap zone to the edge of another gap zone, and stays within the width of the fibril. Panel C shows the 2D FFT power spectrum from the selected region. The white line is normal to the gap/overlap axis and passes through the maximum value in the fundamental peak, corresponding to the D-Periodic spacing. A 1D FFT along this normal was extracted and is shown in Panel D. Both the D-Periodic spacing and the second harmonic are visible and well resolved.

Analysis and Associated Uncertainties in D-Periodic Spacing Measurement

In order to provide a foundation for the parameters that can be measured by this method, and to investigate the method's resolution limits, model collagen fibrils were generated in Matlab (Figure 3.3). In these simulations, the D-Periodic spacing was modeled with a repeating set of Gaussian drops (the gap regions), the amplitude and width of which were set by comparison to AFM images of real Type I collagen fibrils. The spacing between each gap was allowed to vary by a user-defined amount, and the sampling of the "fibril" was set to match experimental imaging conditions. The D-periodic spacing and its associated uncertainty were determined by fitting the FFT as shown in Figure 3.3.

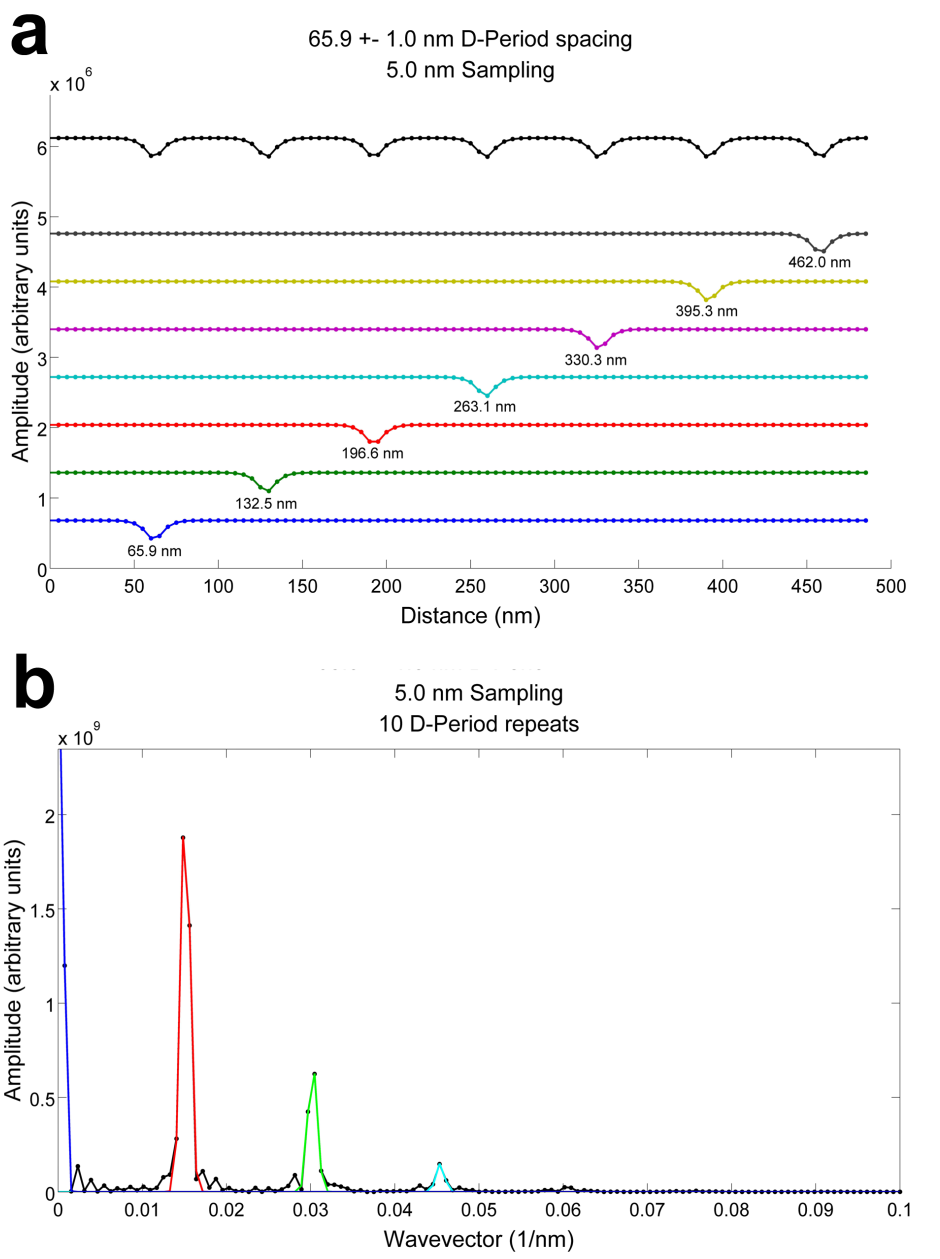


Figure 3.3 Model Collagen Fibril and Associated Fast Fourier Transform
The upper figure shows the construction of the first five repeat units of a model collagen fibril. The fibril is the summation of multiple Gaussian curves with 65.9 ± 1.0 nm spacing, sampled every 5 nm. The width of each Gaussian is 15 nm. A constant DC offset has been applied to the amplitude of the fibril. All parameters in the model fibril were experimentally determined from representative AFM scans of Type I collagen fibrils. The 1D FFT shows expected peaks corresponding to the D-periodic spacing, along with low wavelength components.

A fit of the data to a function of the form $\gamma e^{\left(-\frac{(f-f_c)^2}{2\sigma^2}\right)}$, is shown in red. In this equation, γ is an amplitude scaling factor, f is the frequency, f_c is the D-periodic spacing and σ is the associated uncertainty of the D-periodic spacing. The fit to the zero f component is shown in blue. The D-periodic spacing and the associated uncertainty were recorded for future use.

To put an upper bound on the uncertainty associated with measurements from real experimental samples, this 2D FFT and fitting approach was applied to collagen fibrils imaged within the cortex of sham-operated sheep bones that were used as a control for a previously published experiment.[136] The D-Periodic spacing for each fibril was computed for multiple fibrils from multiple images using a minimum of 9 and a maximum of 21 D-period repeat units. The mean and standard deviation of the D-Periodic spacing for each fibril is shown in Table 3.1. The mean of the standard deviations was used to determine the upper bound on the D-Period uncertainty, which was found to be 0.8 nm. This result verifies that to avoid biasing measurements from low numbers of D-Period repeat units, a minimum of 9 D-Period repeats should be used and further sets the minimum bin size for histograms at 0.8 nm.

	Sham		OVX	
Fibril	Mean (nm)	Standard Deviation (nm)	Mean (nm)	Standard Deviation (nm)
1	71.3	1.2	59.5	0.4
2	69.5	1.1	59.3	0.9
3	67.6	1.2	63.1	0.6
4	68.0	0.3	61.8	0.7
5	66.4	0.6	61.7	1.0
6	66.1	0.6	65.3	0.9
7	70.2	0.2	65.5	0.5
8	72.2	1.0	65.9	1.0
9	72.2	0.7	61.4	0.5
10	68.9	1.0	61.9	0.7
Mean	69.3	0.8	62.6	0.7
Standard Deviation	2.2	0.4	2.5	0.2

Table 3.1 Table of D-Periodic Spacings
This table shows the results of 2D-FFT measurements from multiple Type I collagen fibrils in Sham operated and OVX bone. The D-Periodic spacing for each fibril was computed using a minimum of 9 and a maximum of 21 D-Periods. The mean and standard deviation of the D-Period Spacing of each fibril is shown. The mean of the standard deviations was computed to determine the upper and lower bounds on the D-Period uncertainty. The calculated uncertainty ranges from 0.7 nm for OVX to 0.8 nm for Sham fibrils.

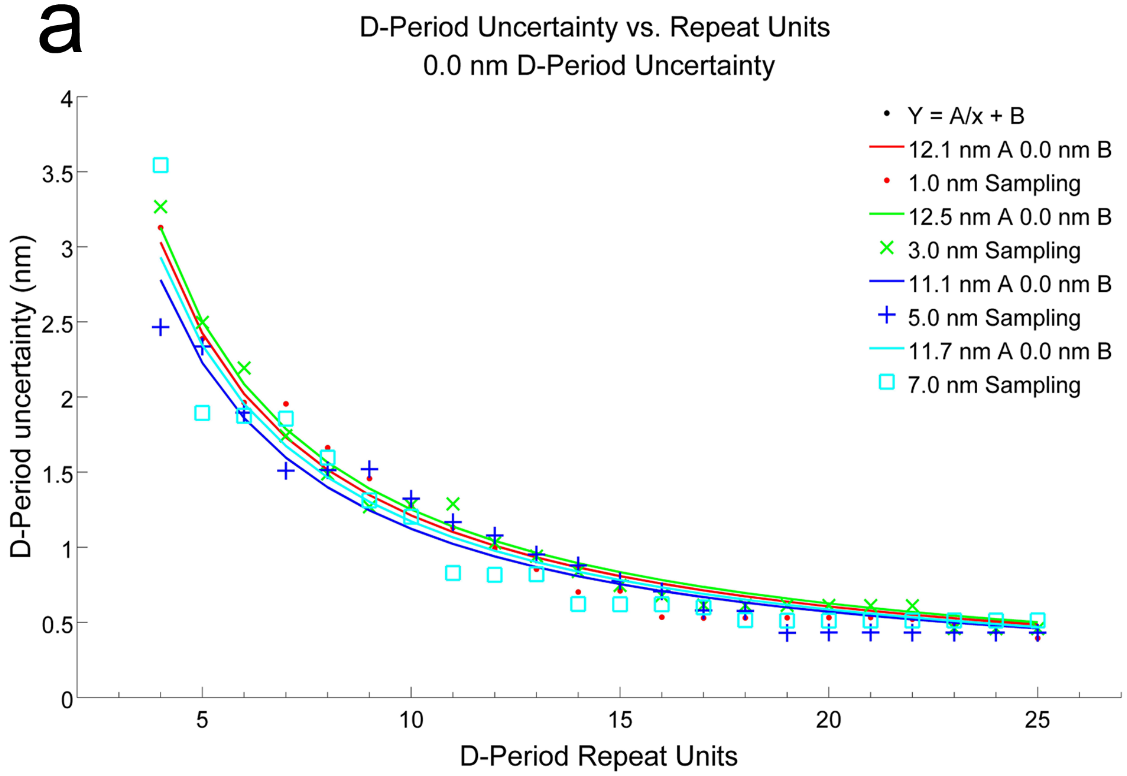
Sampling theory dictates that the uncertainty associated with the D-Periodic spacing measurement should scale as the inverse of the number of repeat units until the fundamental fibril variability is reached. The measured D-Periodic spacing is also invariant of the sampling length scale, provided the feature is well resolved. Figure 3.4 shows the results of varying both the sampling length along the model fibril and the D-Periodic spacing. In both panels, each line is a different pixel density (from fine to much longer given experimental condition). Panel a shows a fibril with perfect D-period spacing (i.e. no variability in spacing) and Panel b allows for ± 1.0 nm of variability which is evenly distributed across the interval. The figure demonstrates that the 2D FFT method is insensitive to the sampling length scales, given the clear 1/x dependence in both panels. A Discrete FFT takes the form:

$$F_n \equiv \sum_{k=0}^{N-1} f_k e^{-\frac{2\pi i n k}{N}}$$

Equation 3.2 Discrete FFT

Where *N* is the total number of data points, *k* is the index of the data point in the series, f_k is the k^{th} data point in the initial series and *n* represents the wavevector. Because of this discrete sampling, there is a spread in wavevectors that scales with 1/*N.*[139] Since *N* is proportional to the number of fibril repeat unit ($N = xl$, where *l* is the sample length and x is the number of repeat units), this spread is also proportional to 1/*x*. Since the number of fibril repeat units is a more natural length scale for these purposes, the spread in wavevectors was fit against a 1/*x* relation.

a


D-Period Uncertainty vs. Repeat Units
0.0 nm D-Period Uncertainty
D-Period uncertainty (nm)
D-Period Repeat Units
0
0.5
1
1.5
2
2.5
3
3.5
4
5
10
15
20
25
Y = A/x + B
12.1 nm A 0.0 nm B
1.0 nm Sampling
12.5 nm A 0.0 nm B
3.0 nm Sampling
11.1 nm A 0.0 nm B
5.0 nm Sampling
11.7 nm A 0.0 nm B
7.0 nm Sampling

b

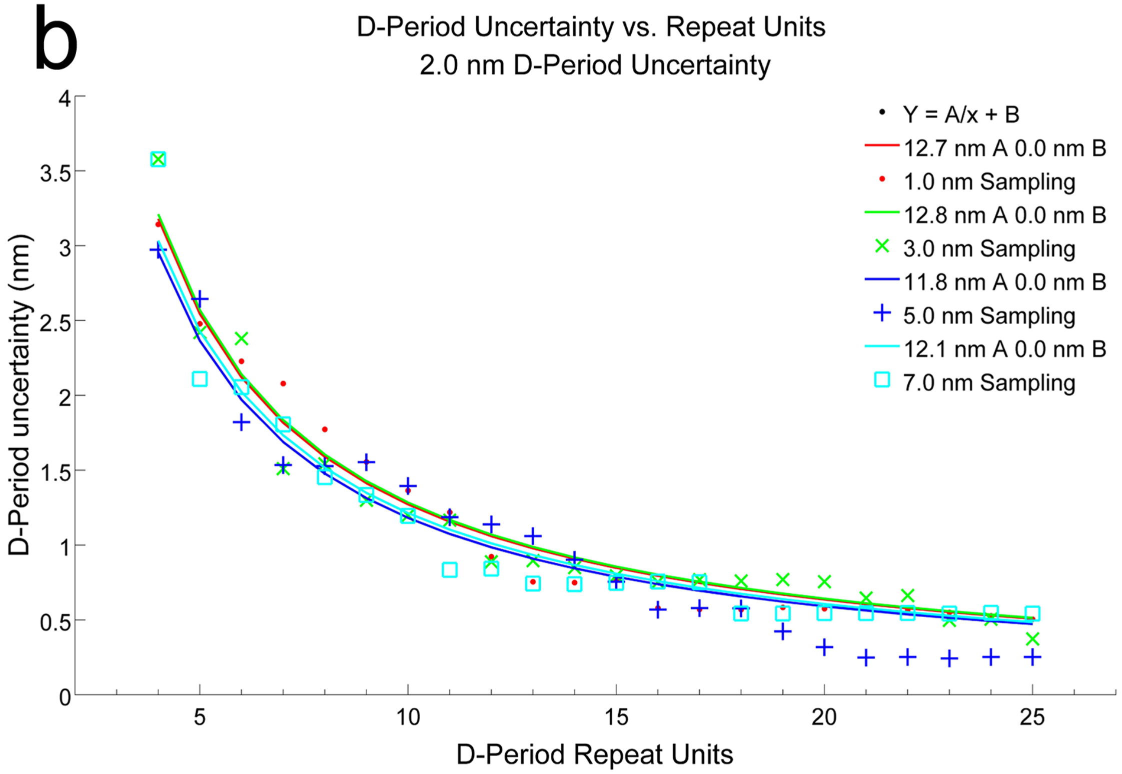

D-Period Uncertainty vs. Repeat Units
2.0 nm D-Period Uncertainty
D-Period uncertainty (nm)
D-Period Repeat Units
0
0.5
1
1.5
2
2.5
3
3.5
4
5
10
15
20
25
Y = A/x + B
12.7 nm A 0.0 nm B
1.0 nm Sampling
12.8 nm A 0.0 nm B
3.0 nm Sampling
11.8 nm A 0.0 nm B
5.0 nm Sampling
12.1 nm A 0.0 nm B
7.0 nm Sampling

Figure 3.4 D-Period Uncertainty Calculation
This figure shows the D-period uncertainty as a function of the number of D-Period repeat units or the sample length, and the variability within the spacing used to create a given model fibril. The uncertainty is expected to scale with the number of repeats as *1/x* because of finite sampling.[139] An additional component is expected due to the contributions from the inherent variability within a particular fibril. Each data set has therefore been fit to a function of the form $\frac{A}{x}+B$. Both panels show fibrils computed with a range of sampling lengths. A sampling length of 1 nm represents an ideal case. Sampling lengths of 3, 5 and 7 nm are more experimentally realistic values. The upper panel shows the results from a fibril with no variability in the repeat spacing. The B term in all fits approaches 0 nm. The lower panel shows the results from a fibril with 2.0 nm variability in the repeat spacing. In this case, the *B* term hovers around 0.2 nm because of the inherent variability in the fibril. The most important feature of both panels is that the D-Period uncertainty does not depend on the sampling length, meaning that as long as features can be resolved experimentally, the D-Period length can be determined without bias from the pixel size of the image or the orientation of the fibril within the image.

It is important to point out that making absolute x-y distance measurements with AFM has multiple limitations, which make it difficult to compare the absolute D-periodic spacing value measured in different studies. Prior to sampling, calibration of the AFM system was performed according to manufacturer guidelines. A calibration grating with a 10 mm period was imaged with the scan size set just under the range of the piezo (80 μm scan size) at 512 pixels by 512 pixels. This results in a pixel size of roughly 160 nm. To overcome limitations imposed by pixel size, multiple consecutive periods were measured. The measured period was defined as the total measured length divided by the number of periods. The calibration was adjusted until the measured period was within 50 nm of the actual 10 μm period. This calibration method results in a maximum error of 0.5%, which is less than the 1.0% tolerance specified by the manufacturer. For the imaging of collagen fibrils, scan sizes were reduced to 3.5 μm at 512 x 512 pixels.

Because of the effective linearity of the scans, error scales with the reduction of scan size reducing the 50 nm tolerance to 2.2 nm.

Observed Distributions in D-Periodic Spacing Measurements

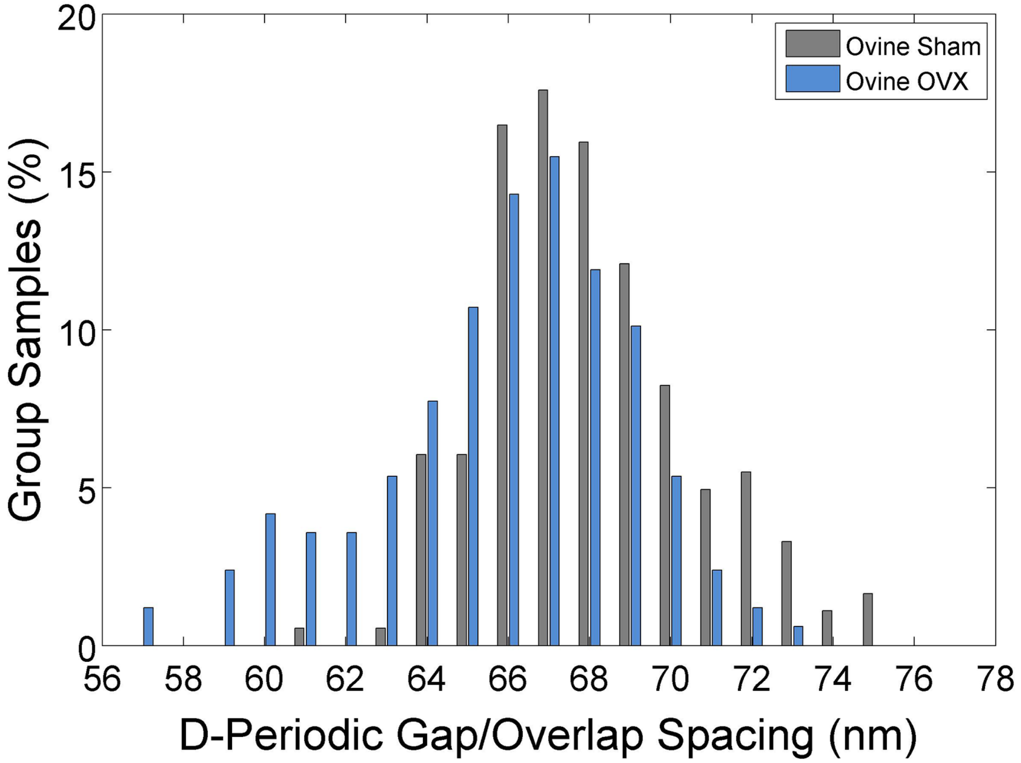


Figure 3.5 Histogram of D-Periodic Spacings
This figure shows the distribution of observed Type I Collagen D-Periodic spacing values measured by AFM from same-operated (grey) and estrogen-depleted (OVX, blue) bone. The data in this figure was derived from previously published work.[136] The mean D-Periodic spacing of the Sham population was 68.0 ± 2.6 nm. The mean D-Period Spacing of the OVX population was 65.9 ± 3.1 nm. In the case of the OVX data, there was an observable shift in the population towards lower D-Periodic spacings, especially evident between 57 and 63 nm. The data in this figure was imaged and analyzed by Dr. Joseph Wallace.

This 2D FFT approach was next applied to Type I collagen fibrils imaged within the cortex of sham-operated sheep bones using AFM (used as a control for a previously published experiment[136]). The mean D-Periodic spacing of the 182 measured fibrils was

68.0 nm, with a population standard deviation of 2.6 nm. Figure 3.5 shows these data plotted as a histogram with a bin size of 1 nm, which is greater than the maximum observed variation along a single fibril of 0.8 nm shown above. This histogram demonstrates that in these normal sham-operated bone samples, there was a distribution of observed D-Periodic spacing values. The existence of a distribution of D-Periodic spacings may be a fundamental characteristic of Type I collagen, but is overlooked in most studies. As an example, this distribution was not discussed in a thorough recent book that presents a detailed review of collagen structure and mechanics.[140] In order to verify that this distribution is real and not caused by sample preparation or AFM imaging artifacts, the 2D FFT analysis was run on historical SEM images from human Achilles tendons (Figure 3.6, reproduced with permission[134]). Qualitatively, the fibrils in Panel a look similar to images of collagen fibrils observed by AFM. Panel B shows the histogram results of the 2D FFT analysis from 13 observed fibrils within the image (the included scale bar was used to set the length scale). The mean D-Periodic spacing was 66.9 nm with a population standard deviation of 1.6 nm. This mean value is well within experimental error of the AFM results. More importantly, a distribution of D-Periodic spacings was observed in the SEM data. This observed distribution supports the contention that the distribution of Type I collagen morphologies is a real feature that does not arise from sample treatment or other imaging artifacts of AFM.

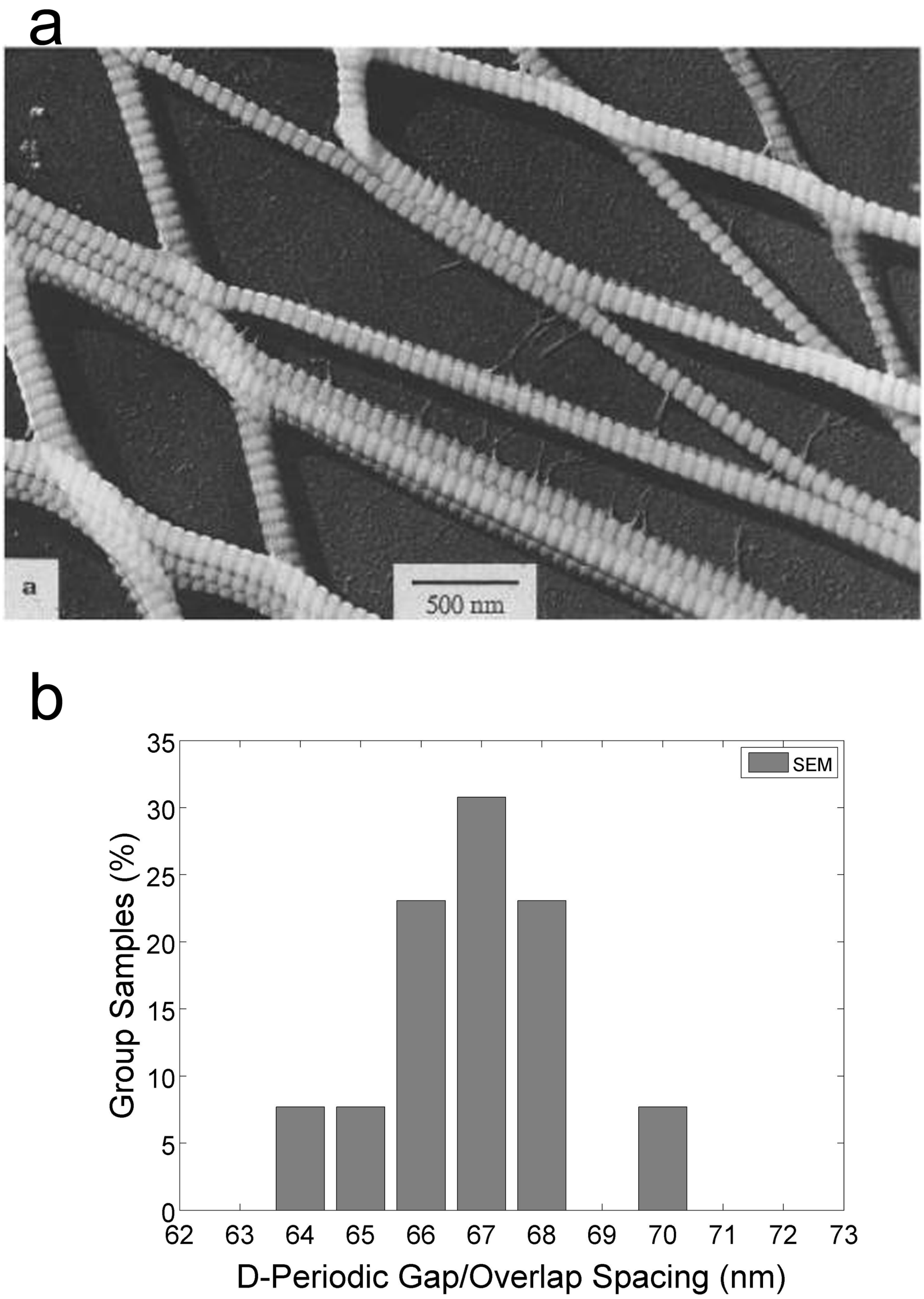

a
a
500 nm
b
SEM
Group Samples (%)
0
5
10
15
20
25
30
35
62
63
64
65
66
67
68
69
70
71
72
73
D-Periodic Gap/Overlap Spacing (nm)

Figure 3.6 Historical SEM image of Type I Collagen Fibrils
Panel A reproduces SEM images of isolated Type I collagen fibrils from Chapman et al.[134] The D-Periodic spacing of these fibrils was determined using the 2D FFT technique outlined above. The internal reference bar set the length scale. Panel B shows the distribution of D-Periodic spacings within the image. The distribution had a mean of 66.9 nm and a standard deviation of 1.6 nm. The presence of a distribution of spacings corroborates that the existence of a distribution is a real feature and not an imaging artifact.

To investigate other methods that have been used to analyze Type I Collagen morphology, historical Small Angle X-Ray Scatter (SAXS) data from fresh, non-mineralized, turkey tendon samples were analyzed.[141] Multiple D-period harmonics are present in this curve. This image was analyzed to determine if inhomogeneity in D-period spacing is sufficient to explain the observed peak widths. A modified Bertaut-Warren-Averbach technique, as outlined by Drits[48], has been used to determine what portion of the peak widths can be ascribed by this inhomogeneity in D-Period spacing. This analysis will demonstrate if the distribution of D-Period spacings measured by AFM is consistent with the observed widths of X-Ray peaks. Initially, the data was digitized and scaled from the published image to provide a working data set. The resulting data was linearly interpolated to provide even scaling along the x-axis. Next, fitting each data point that was not clearly a harmonic peak to a power law of the form identified the background scattering intensity:

$$Y = Ax^{-B} + C$$

Equation 3.3 SAXS Baseline

where $A = 1.1$, $B = -3.2$, $C = 10.9$. This fitted form has no physical meaning; it is simply an approximation to the background scattering intensity from the sample. For future studies, a blank spectrum is a more appropriate background to use. The results are shown in Figure 3.7.

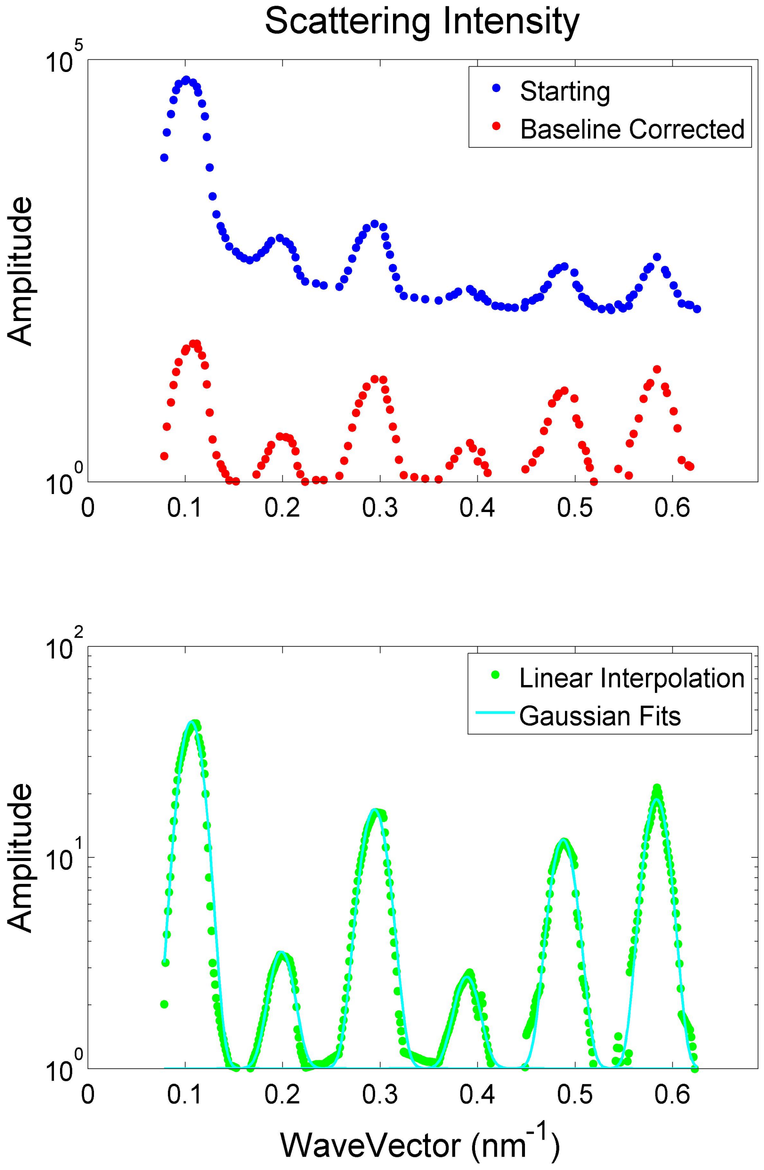


Figure 3.7 SAXS scattering data
SAXS data from Fratzl *et. al.* is reproduced above.[141] The extracted intensities are shown as blue dots. The red dots show baseline corrected data fit to the following form: $Y = Ax^{-B} + C$, where A = 1.1, B = -3.2, C = 10.9. The green data is a linear interpolation of the red data so that all data points are equally spaced. Finally the teal lines are Gaussian fits to each scattering harmonic.

Drits initially describes the intensity from a sample as the product of the Lorentz and polarization factors, the structure factors, and interference function (Drits, Equation 1). The interference can then be determined by dividing the scattering by these background components (Drits, Equation 2). To accomplish this, the scattering intensity was divided by the fitted background. Each peak was fit to a Gaussian in order to identify the peak center of each scattering harmonic in reciprocal space. The reciprocal values of the spacing reported on the x-axis were computed and multiplied by the appropriate harmonic to determine the real values of the D-Periodic spacing. Baseline could not be adequately separated from peak intensity at low x values for the first harmonic. Because of this, the Gaussian fit to the first harmonic may not be accurate and appears to be biased to higher value in reciprocal space (shorter D-Period spacing). Because of this apparent bias, the first harmonic is excluded from the average calculation. The average D-Periodic spacing (harmonics 2 through 6) was determined to be 64.2 ± 0.6 nm. The extracted D-Period spacings are shown in Figure 3.8

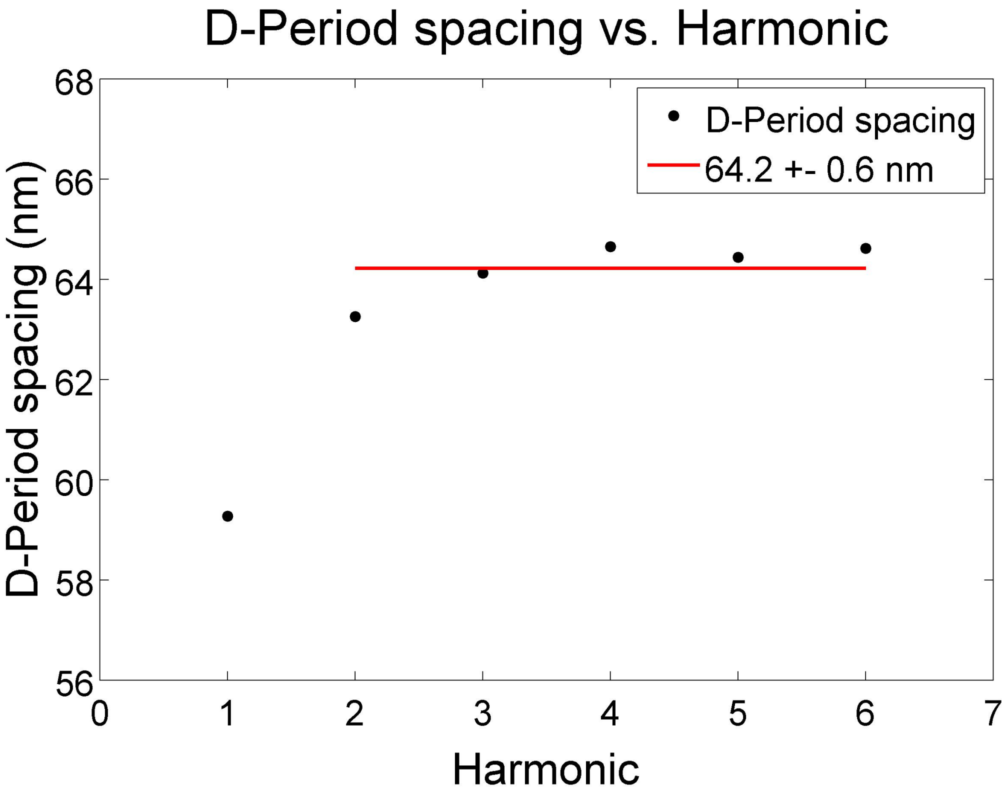


Figure 3.8 SAXS D-Period Spacing
The calculated D-Period spacing from each harmonic in Figure 3.8 is shown above. The first harmonic is not included in the average because a lack of baseline on both sides of the peak leads to a bias in the Gaussian Fit. The average D-Period spacing is 64.2 ± 0.6 nm.

The breadth of each peak was not previously discussed. In order to determine the uncertainty in the D-Period spacing, a variant of the Bertaut-Warren-Averbach technique, described by Drits *et. al.*[142] was applied to extract the underlying distribution of D-Periodic spacings from the scattering peaks. In short, the scattering intensity for a sample with a known D-Period spacing can be described as a Fourier series (Drits, Equation 13), the coefficients of which can be determined by an inverse Fourier transform (Drits, Equation 14). Since these coefficients can be calculated analytically, they provide a useful way to investigate the scattering curves. In the case that the D-Period spacing is not uniform, and there is a distribution of spacings around a given mean, the scattering intensity can still be fully described by a Fourier series (Drits, Equation 20). This new series can also be inversely transformed to generate a different set of Fourier coefficients (Drits, Equation 21), which can be simplified, since the scattering peaks are symmetric, to a sum of cosine terms (Drits, Equation 23). Assuming a Gaussian distribution of D-Period spacings, which approximates distributions observed by AFM, the reduced Fourier coefficients can be written as the product of the coefficients from a perfect sample and a different Gaussian modifying term, the parameters of which describe the distribution of spacings (Drits, Equation 26), which is both separable and linear (Drits, Equation 27). This means that one can take a set of observed scattering peaks, Fourier transform them and fully separate the pure scattering components from broadening because of variation in D-Period spacings.

This method was applied to the baseline corrected turkey tendon data in the following way: each peak was separated onto an interval spanning the average peak separation centered on the peak. The peak center used is the center from the Gaussian fits used to determine the average D-Period spacing. This interval was scaled by the average peak separation so that the final interval was symmetric ranging from -½ to ½ with a peak centered at 0. The Fourier transform of the scattering peak over this symmetric interval was then computed in order to extract the components. Since the coefficients are required to be strictly positive and the sine terms sum to 0, the amplitude of the overall transform was used for the next calculations. Transforming the raw data or fitted Gaussian yield identical results. The transform of Harmonic Three is shown as an example in Figure 3.9.

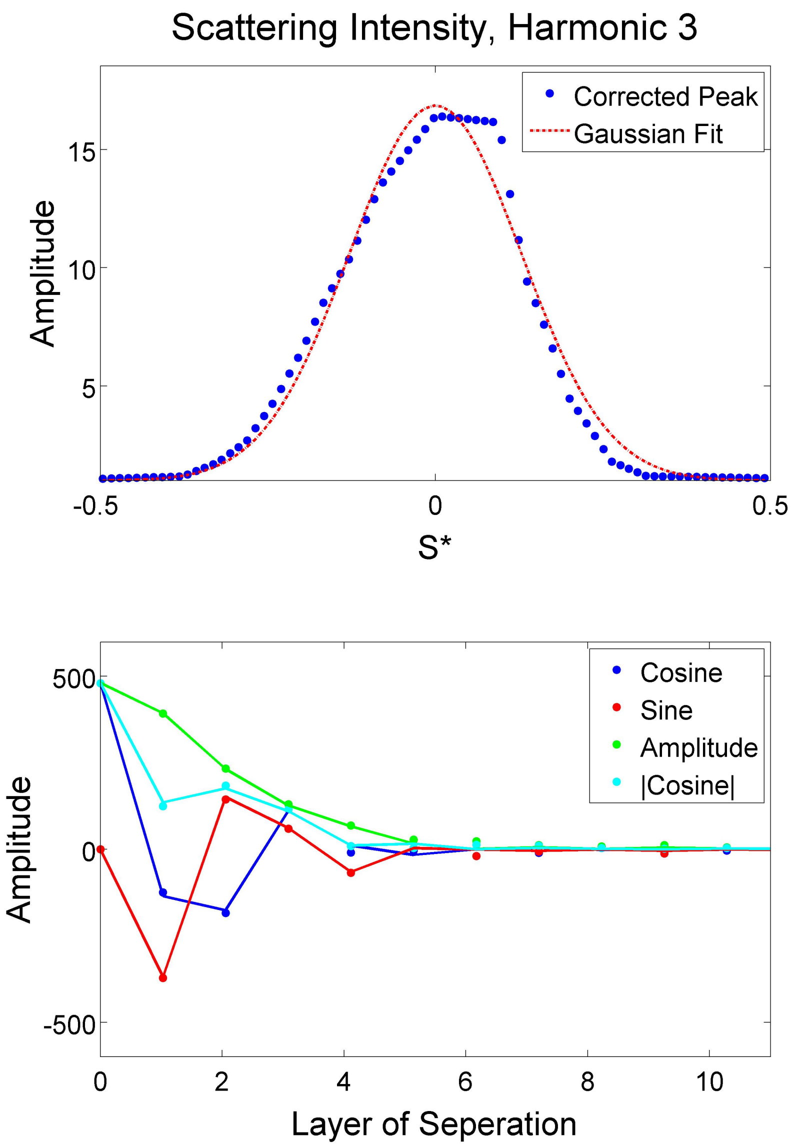


Figure 3.9 Example BWA Transform

This figure shows an example of the transforms used to calculate the distribution of D-Period spacings. In the upper panel the blue dots are the baseline corrected scattering intensity. The red line is a Gaussian fit to this peak. The peak has been centered over a scaled interval spanning the average reciprocal D-Period Spacing. Below is the Fourier transform of the above data. The dots are transforms of the raw data, and the solid lines are the transforms of the Gaussian fit. In blue are the cosine components, red are the average components, green is the amplitude of the transform, and teal the absolute value of just the cosine component. The amplitude of the transform is used in the next steps because the problem requires strictly positive values, and the sine terms sum to zero.

Following Drits' method, a plot of the Fourier coefficients (A_n) vs. the square of the harmonic (l^2) was constructed, where the slope of the line contains the information about the distribution of D-Period spacings:

$$m = \frac{-2\pi^2 n \overline{\sigma_1^2}}{d^2(001)}$$

$$\overline{\sigma_1} = \sqrt{\frac{md^2(001)}{-2\pi^2 n}}$$

Equation 3.4 Determination of D-Period spacing Distribution

where m is the slope of the line, n is the number of the coefficient, $\overline{\sigma_1}$ is the standard deviation of the distribution of D-Period spacings around $d(001)$, the reciprocal of the D-Period spacing.

For this particular data set, the first, third, fifth and sixth harmonics were used to calculate the distribution of D-Period spacings. Harmonic one can be included, in this case, because each peak is required to be centered at 0. Harmonics two and four are excluded because Fratzl notes significant dampening in amplitude, which they suggest arises from partial hydration of the fibril leading to reduced contrast.[141] The fits are shown in Figure 3.10

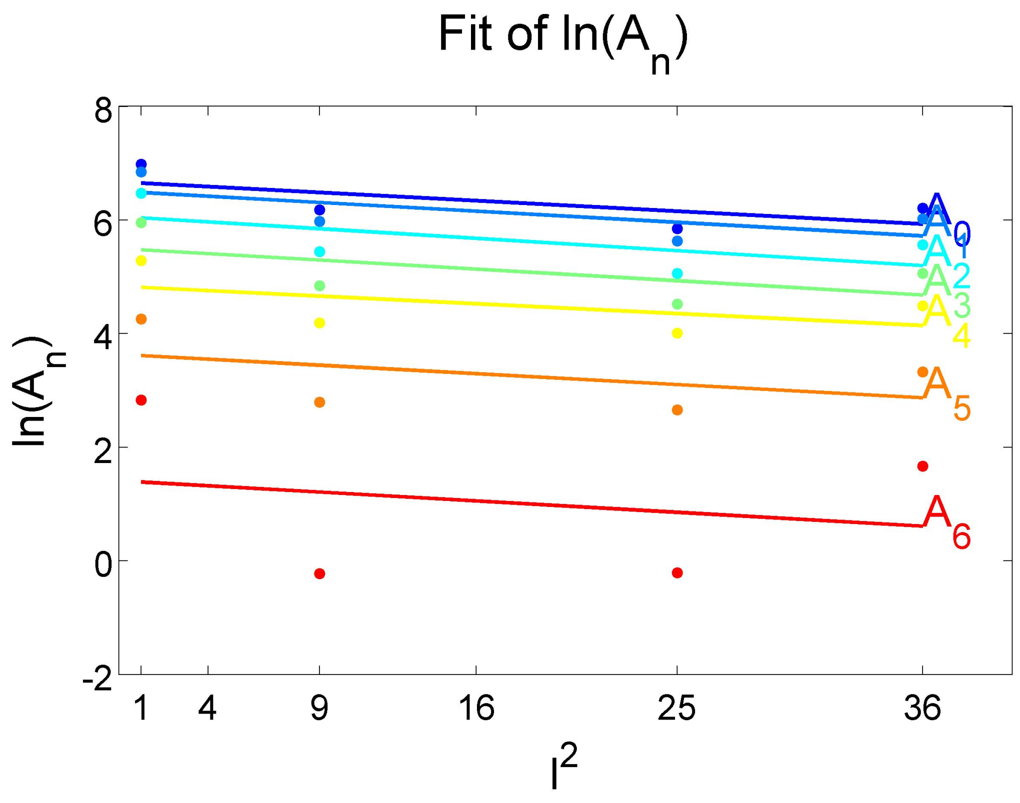


Figure 3.10 Fit of ln(An) vs. l2
This figure plots the natural log of the Fourier coefficients (separated by n layers) of each peak against the square of the harmonic. The slope of each line is calculated in order to determine the distribution of D-Period spacings, as described in Equation 3.4

Since this entire method is dependant on the relative amplitudes of the different scattering harmonics, these two harmonics must be excluded. The calculated standard deviation of the D-Period spacing is 2.1 nm. The extracted D-Period spacing is shown below in Figure 3.11

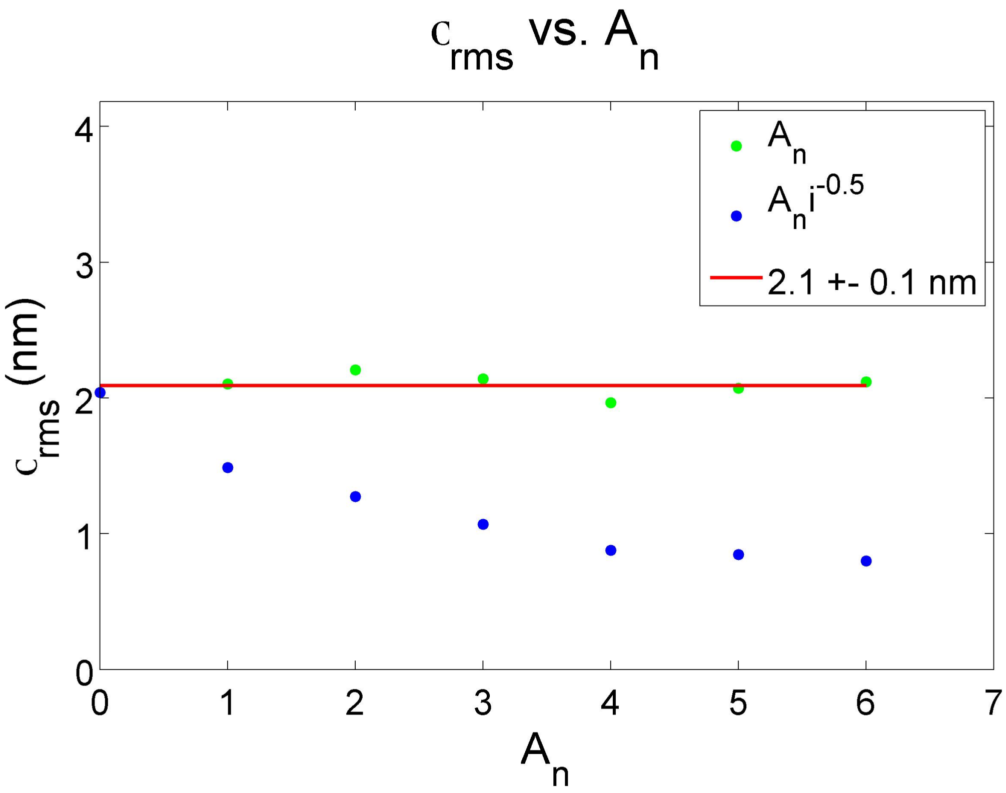


Figure 3.11 SAXS Calculated D-Period Distribution
This Figure shows the calculated D-Period standard deviations calculated from Figure 3.10. The average D-Period standard deviation is 2.16 nm

This shows two things: first, assuming inhomogeneities in the D-Period spacing is sufficient to explain the observed peak width in X-Ray data; and, second, the distribution width determined from this X-Ray analysis is comparable to what is observed by the 2D-FFT technique. The 2D-FFT technique has one significant advantage: the fundamental resolution limit is sufficiently small to directly observe the shape of the sample distribution. The modified BWA technique cannot do this by definition because only one distribution is defined in the algebra, and the ability to fit the transformed data is insufficient to discriminate between different functional forms for an underlying distribution.

Statistical Comparison Between D-Periodic Spacing Distributions

The existence of a distribution of D-Periodic spacings is an important observation which stands in contrast to the fixed 67 nm value put forth by the Hodge-Petruska model.[131] Finding this distribution in current AFM data along with historical SEM and SAXS data supports the assertion that this is a fundamental characteristic of Type I collagen. Given the ubiquitous nature of Type I collagen in animal tissues,[5] and the number of disease that can affect these tissues, it is likely that alterations in the Type I collagen ultrastructure will occur with disease. Using a model of estrogen-depletion in sheep (ovariectomy, OVX), which may by a model system for age-related estrogen deficiency and osteoporosis in humans, it was hypothesized that it would be possible to distinguish between normal bone (sham-operated) and bone from animals with a known disease state through changes in the distribution of collagen fibril morphologies. The mean D-Periodic spacing of the Sham population was 68.0 ± 2.6 nm, significantly different that the 65.9 ± 3.1 nm value in OVX samples (p=0.048 using One Way ANOVA). When viewed as histograms (Figure 3.5), there was an observable shift in the OVX population towards lower D-Periodic spacing values. This difference in populations is further highlighted when viewing the data as Cumulative Density Functions (CDF, Figure 3.12). The CDF plot shows both the shift in the mean of the OVX population as well as the change in the shape of the distribution.

Demonstrating that qualitative population differences exist between normal and diseased states is an important finding, but proving statistical significance between populations is an important next step. The distribution of D-Periodic spacings in the Sham population was compared to that of the OVX population using a Kolmogorov-

Smirnov test (K-S test). The K-S test was chosen because it is a non-parametric comparison between distributions, and is sensitive to changes in means as well as in the width of distributions. Further, the test does not require normally distributed data. Applying a two-sample K-S test demonstrated that there was a significant difference in the population distributions between Sham and OVX samples (p<0.001). The fact that distributions are present in both normal and diseased fibril populations, and that there are statistical changes in these distributions with disease, is further proof that the distribution itself is a true characteristic of Type I collagen fibrils.

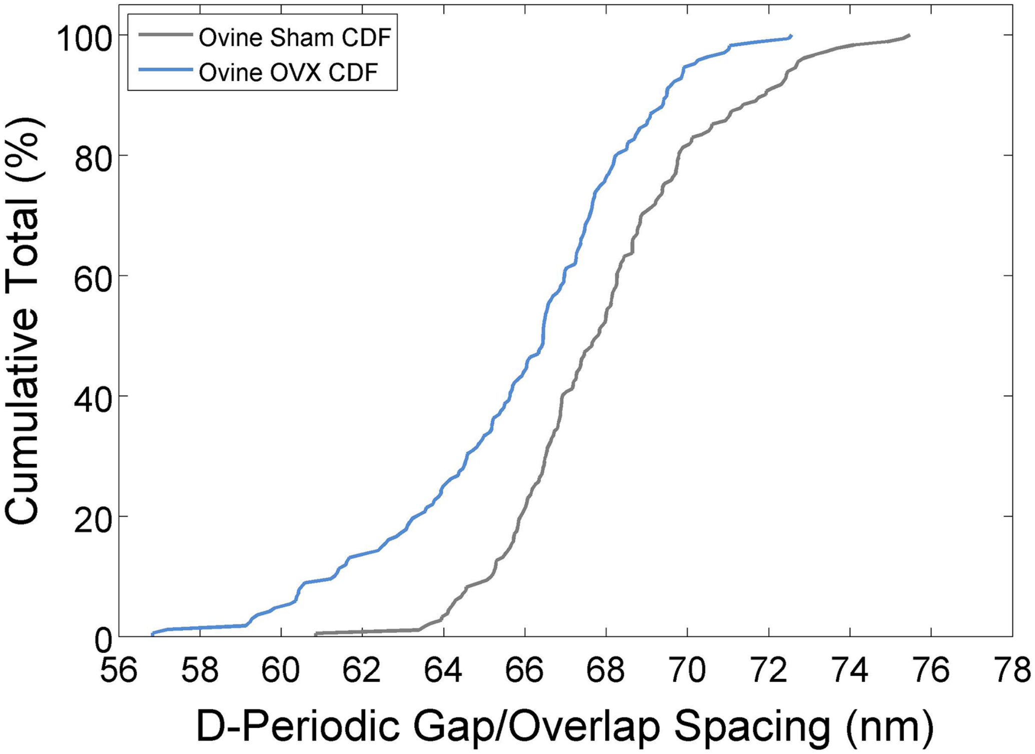


Figure 3.12 Cumulative Density Function of D-Periodic Spacings
This figure shows the cumulative density function (CDF) of Type I collagen fibril morphologies measured from sham-operated (Sham, grey) and ovariectomized (OVX, blue) sheep bone. The data in this figure was derived from previously published work [9]. Plotting the CDFs highlights the overall difference in the distribution of D-Period spacing between the two data sets. A two-sample Kolmogorov-Smirnov (K-S) test was applied to the data from each set. The K-S test is a nonparametric test that is sensitive to changes in

both the mean and standard deviation of a distribution, and does not require normally distributed data sets. There was a significant difference in distributions between the 2 groups ($p<0.001$), indicating that estrogen-deficiency leads to a significant alteration in Type I Collagen morphology. This figure is generated from the data collected and analyzed by Dr. Joseph Wallace in Figure 3.5.

Conclusions

In conclusion, this paper details a systematic method to analyze and compare Type I collagen fibril morphology distributions. The measurement of the D-periodic spacing of Type I collagen was performed using a 2D FFT approach. The ability to accurately measure the D-periodic spacing led to the observation of a distribution of Type I collagen morphologies in AFM images, as well as historical SEM images. The uncertainty associated with such measurements was analyzed and compared with historical SAXS data to verify the presence of the distribution. Finally, statistically significant changes in these population distributions were noted in samples from a model of human estrogen depletion. These results suggest that this method can be used to analyze images of Type I collagen fibril using any surface-based microscopy technique, and that monitoring the distribution may represent a new diagnostic technique for disease in a number of collagen based-tissues.

Chapter 4

Conclusions

Conclusions from Survanta: Translating Hypothesizes into Observables

The heart of the Survanta study was translating hypothesizes about how nanoparticles interact with biological systems into new observables. In particular, the study investigated how polycationic PAMAM dendrimers interacted with a clinical lung surfactant, Survanta, as a controllable model for inhalation toxicity since the lung lining is one of the primary barriers nanoparticles have to cross. This study allowed for the testing of three major concepts: first, do the dendrimers interact differently with the kinds of substructures that exist in biology; second, can any of the proposed mechanisms for membrane be supported or contradicted; and third, can the results of Molecular Dynamics simulations be translated into real, physical, observables?

In respect to the first question, a number of different conclusions can be made. Significantly, membrane proteins do not appear to nucleate membrane disruption. A fluid, negatively charged, lipid domain is preferentially removed with respect to a gel, zwitterionic, lipid domain. Existing defects appear to be significantly easier to enlarge than generating new defects is. This observation is corroborated by the pronounced difference in apparent rate of lipid removal.

With respect to the second question, a wealth of mechanistic data has been revealed. Dendrimers have been directly observed adhering to the membrane prior to lipid removal. This strongly suggests that the dendrimers have to work in a concentrated fashion as aggregated material to induce disruption. Second, the mechanism proposed by Gewirth, nanoparticles disrupting the interaction between the lipid bilayer and substrate can be eliminated. This is a substantive results because we have taken a mechanistic hypothesis: that polymer tunnels through the layer and sits between the layer and substrate, shielding electrostatics, and allowing for removal. This was translated into an observable: the polymer that disrupted this electrostatic interaction between lipid and substrate will remain attached to the substrate following lipid removal. More importantly, this observable could be directly tested with our experimental setup. Polymer was not observed on the substrate following disruption, disproving Gewirth's hypothesis.

In respect to the third question, this study has allowed for a direct link of Molecular Dynamics calculations to physically observable properties on a nanometer length scale. This is a significant observation for both the fields of computational biology and microscopy. Molecular Dynamics simulations showed that PAMAM dendrimers interacted more strongly with and deformed to a greater extent during interaction with, fluid lipid bilayers than gel lipid bilayers. Additionally, the dendrimer remains partially exposed to solvent during this interaction. These simulation results can be corroborated through AFM investigation. Dendrimer accumulated on top of the bilayer is supported through a combination of both topographical and phase images. The topographical images show a distinct elevation change, confirming the presence of polymer at the membrane. Phase contrast images show that this material is at least partially exposed to solvent because of the significant different in phase lag between the region with aggregated material and the underlying lipid domain, reflecting substantive changes in the interaction of the tip with the material beneath it.

Conclusions from Bone: Improving Nanoscale Quantative Metrics to Address Biological Questions

The heart of the bone study was extracting quantifiable metrics from existing observable nanoscale parameters to test fundamental biological hypothesis. Collagen is the most prevalent protein in the human body, and forms the basis for multiple tissues ranging from bone, to tendon, to corneas, to arterial walls. Collagen is also interesting from the perspective that it is one of the longest studied proteins. In 1963, Hodge and Petruska describe a theoretical model for the structure of collagen fibrils, namely that the fibrils exhibit a characteristic 67 nm gap-overlap spacing.

This study questioned whether or not, collagen is truly a perfectly regular material. Is there, really, only one characteristic D-Period spacing of collagen; or, can biological changes manifest alterations to the structure of collagen? The short answer to this question is absolutely yes! This study demonstrates, through careful analysis, that a distribution of D-Period spacings exists in biological samples; furthermore, the study demonstrates that pathological models induce statistically significant differences in this distribution.

This study shows how a 2D-FFT technique can be used to analyze the D-Period spacing of collagen fibrils at accuracy below both the radius of curvature of the AFM probe used for imaging, and the pixel size of the captured image. The shows uses numerical simulations to elucidate fundamental resolution limits of the analytical technique, as well as confirm the independence from pixel size. The resolution limits determined from the simulated collagen fibrils were confirmed through analysis of actual collagen fibrils found in bone. The combination of both the theoretical studies and experimental studies set the requirements for accurate measurement of the D-Period spacing, and the uncertainty associated with a single measurement. Including 9 D-Period repeats or greater ensures single measurement accuracy below 1 nm.

The culmination of this image analysis method and its associated resolution certainty allows for the first question to be addressed: is the D-Period spacing of collagen a singular value? The remarkable answer is no! The D-Period spacing exists as a distribution of values in biology! This single result flies in the face of dogma established since the Hodge-Petruska model. The reality of this distribution, and proof that it was not simply an artifact of the AFM imaging process demanded proof. This fact was something I, the author of the study itself, was not expecting to find and reluctant to believe in the first place!

This startling result was not something that could be left unaddressed. The beauty of collagen as a scientific system is that there is a wealth of research dating back many, many decades begging for additional careful analyses. The first, and most obvious target, are historical SEM images of isolated collagen fibrils. This type of data is idea because it is visually virtually indistinguishable from AFM results and the 2D-FFT technique can be directly applied to this kind of data. To this extent, data from an excellent review of collagen imaging by Chapmann was analyzed with the 2D-FFT technique. Amazingly, this distribution was observed again with a width similar to that of the AFM derived data!

Determined to prove this distribution truly a real feature, SAXS data was analyzed utilizing a technique pulled from the literature on clay. This technique effectively breaks scattering data into separable components arising from scattering from samples with a perfect mean D-Period spacing, and a secondary component arising from any variations in D-Period spacing around this mean. The truly remarkable aspect of this analysis is that the calculated D-Period spacing distribution was again similar to what was observed by both AFM and SEM. A direct comparison of AFM and SEM data cannot be performed because of both an absolute calibration difference between the two techniques and the low number fibrils available for measurement in the SEM image limits the statistical significance of any comparison.

The observation of a distribution of D-Period spacings by a separate, fundamentally independent technique, proves that the distribution of D-Period spacings is a real, observable, feature of the biological system

The final aspect of the study addressed the second major question: can this observable be used to test a relevant biological hypothesis. To this extent, a non parametric, Komologorov-Smirnov, test was implemented to show conclusive differences between control sheep and estrogen depleted sheep, which are utilized as a model for early onset Osteopenia.

Future Directions and Outlook

The Atomic Force Microscope is an important tool in biology as both this study and a wealth of literature shows. Academic and commercial development of AFMs continues to enhance the capabilities of the microscopes, which, in turn, allows for new experiments to be performed. There are three key areas of development right now that will make significant strides towards improving the utility of the AFM for biology: developing faster scanning instruments, improving integration with optical microscopes, and developing new feedback modes.

Making faster AFMs is, the easiest improvement that can be made to these instruments. Imaging right now simply takes a long time. Things that occur on fast timescales cannot be observed. Careful, detailed studies, like the investigation of collagen, require an incredible amount of time and effort to do well when each of hundreds of images takes many minuets to acquire. Building faster microscopes well requires three major components: smaller cantilevers and detection systems to use them, stiffer scanners with higher resonance modes, and faster intelligent control electronics to drive it all. Small cantilevers have been the major target of development since the late 1990s[143, 144] The cantilevers resonate at much higher frequencies and suffer from significantly less viscous drag, both of which allow for orders of magnitude higher line rates. Small cantilever systems have been used to directly image biomolecules[145, 146], and significantly enhance signal to noise in force spectroscopy experiments[37, 147, 148]. Almost a decade later, Asylum Research's (Santa Barbara, CA) Cypher AFM is the first and only commercially available small cantilever commercial microscope in the United States. It is truly an impressive feat of engineering with high line rate, closed loop functionality and built in temperature regulation. Prototype fluid cells are still in development, so the ultimate utility for biological applications remains to be seen. Ultra small cantilevers are no longer the exclusive domain of laboratories with built in fabrication facilities as SCL Sensortech (Vienna, Austria) has emerged as the leading supplier of commercial small cantilevers. Further improvements to scanning rates can be made through improving the physical scanners themselves by minimizing parasitic resonances either by scanning at resonance modes,[149, 150] or by designing much stiffer comb-scanners,[151-153] which have become standard in both Veeco and Asylum's high end AFMs. Finally, even further improvements are being made by enhancements to the data acquisition and control electronics needed to operate at these high-speeds.[154-158]

The utility of these fast AFM's can be shown by at least two metrics, reduction in time required for imaging (ultimately resulting in happier researchers)[159] or the opening of new experiments which have not been performed before. These kinds of experiment include translation[160] of kinesin along microtubules and movement of myosin, to measurements of ATP dependant conformation changes in proteins[161], to the wealth of work done at the Hansma lab on bone[36, 37, 39, 138].

The other two avenues for improvement, optical integration and new feedback modes, afford many opportunities for development. Despite the obvious advantages in spatial resolution and closing gap in temporal resolution, the AFM will never supersede the power of fluorescent optical imaging. Individual proteins will never be able to be tagged color labeled, and localized in three-dimensional space by AFM. Thankfully, there is no fundamental reason for AFM and optical techniques to continue to remain virtually isolated techniques. Veeco's Bioscope Catalyst is the first commercially available instrument in which optical integration on both a hardware and software side was a design requirement. The original Bioscope, Asylum's MFP3D, Agilent's 5500ILM have all had serious flaws, many of which thrashed optical resolution and software integration is critically absent. The new Catalyst allows for easy, real-time alignment of optical and atomic force images. Regions of interest can be identified optically, and then imaged with the AFM, which has been difficult, at best, until now. The last piece of the puzzle is continued development of alternate feedback methods. The most promising mode at the moment is Veeco's new Peak Force Tapping mode. In essence, rather than looking at changes in the amplitude of the oscillations of the cantilever, pixel by pixel force curves are used for feedback. Samples can be imaged with minimal force; an effective Young's modulus can be measured at each and every location. At each and every point in an image, real, quantifiable, materials properties can be correlated with topological features. In the context of the work of this thesis, this would allow for lipid domains of different phases but similar heights to be observed, applied forces could be minimized, potentially highlighting other, masked, features of the sample. In context of the bone study, any correlation between differences in D-Period spacing and materials properties could be investigated on an individual fibril basis, which is simply not possible with any other technique!

Appendix A

Deflection Sensitivity

Abstract

We present a method for improving the accuracy of force spectroscopy measurements by atomic force microscopy methods. This involves measurement of the deflection sensitivity of cantilevers while maintaining contact forces below the damage threshold to biological samples. By using different cantilevers on three different substrates, a simple method for determining the minimum number of deflection sensitivity measurements required, in all cases, for improving the accuracy of force measurements by 10-20% can be achieved. This procedure is applicable to all commercial AFMs, and significantly decreases potential damage to both AFM tip and sample.

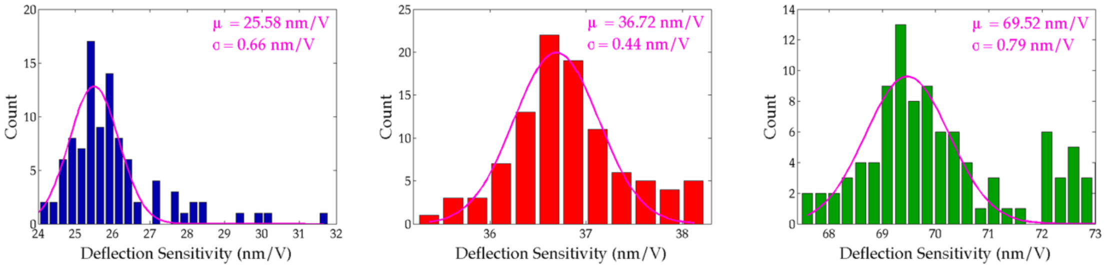
Introduction

Single molecule force spectroscopy has become a powerful tool for gaining insights into biological interactions that are not possible with ensemble measurements. An atomic force microscope (AFM) is the most widely employed instrument for single molecule force spectroscopy.[162] A typical AFM based single molecule study requires functionalizing both a surface with a molecule of interest, and the AFM cantilever with a suitable recognition molecule complementary to that of the surface molecule. Several such interactions have been studied, including: DNA interactions[163-165], ligand-receptor interactions[166-169], and ligand-cell based interactions[168, 170]. In a typical AFM force pulling experiment, the complementary molecules functionalized on both AFM probe and surface are brought together momentarily and pulled apart. The resulting force-distance curve provides a quantitative means for measuring the applied force at bond rupture. Repeating this measurement hundreds, or thousands, of times enables one to examine the distribution of forces at rupture, obtaining the most probable rupture force for a single molecule interaction. This measurement requires accurate determination of two separate parameters. One is the deflection sensitivity, which is the ratio of the deflection of the cantilever in the z-direction and the measured voltage at the split photodiode. The second is the spring constant of the cantilever.[171] The thermal noise method, used by all commercial instruments, to determine the spring constant of a cantilever relies on the precise measurement of the deflection sensitivity.[172-175] This means that accurate determination of the deflection sensitivity is critical for single molecule force measurements. In most commercial AFM instruments, the cantilever is pushed against a hard surface to obtain a deflection and the slope of the contact region, selected by the user, in the force-distance curve is taken as a single input for determining the deflection sensitivity. The region selected, quite readily, becomes a potential source of error. By shifting the range used for a linear fit, the deflection sensitivity is altered.[176] The user can also measure multiple slopes from multiple curves and select a single value from the set. Combining these error sources can drastically change value of the measured deflection sensitivity.

When the cantilever probe and surface are functionalized with biomolecules, obtaining a large number of deflection sensitivity measurements by pushing against the surface is not advisable due to potential damage to the molecular mechanical system. For typical single molecule force pulling experiments on biological systems, knowing the spring constant of the cantilever and the deflection sensitivity prior to taking measurements is a pre-requisite for applying a constant loading force. In these experiments, the loading force, or the contact force between the AFM tip and surface, is kept at or below 200 pN in order to limit damage to the surface and tip bound molecules, and to observe predominantly single molecule rupture events[165, 167, 169]. The scale of forces required for accurate deflection sensitivity calibration, however, can easily extend into the 10's of nN. Limiting the contact force requires knowing both the spring constant and deflection sensitivity of the cantilever *a priori*. In this study, we have examined three different sets of cantilevers to determine the minimum number of deflection sensitivity measurements required to accurately measure the deflection sensitivity of each of the cantilevers during the calibration procedure. This minimizes any damage to the functionalized molecules on both the AFM probe and surface while increases the accuracy in single molecule force measurements

Methodology and Results

Appendix Figure A.1 shows the distribution of measured deflection sensitivities for three cantilevers with measured spring constants ranging from extremely soft (Olympus Biolever 0.030 N/m, Veeco DNP 0.291 N/m) to extremely stiff (Veeco FESP 1.672 N/m) on a gold substrate, along with their relevant statistical parameters.



Appendix Figure A.1 Deflection Sensitivity Distributions
Deflection sensitivity distributions for different cantilevers, on gold, are shown above. Each bin is 0.25 nm/V in width. The solid line displays the Gaussian fit to each determining the mean deflection sensitivity (μ) and standard deviation (s). The OBL cantilever, blue-left, had a spring constant of 0.030 N/m. The DNP cantilever, red-middle, had a spring constant of 0.291 N/m. The FESP cantilever, green-right, had a spring constant of 1.672 N/m.

Each of the distributions shows a significant number of outliers from a normal distribution. Their presence in considerable numbers indicates that a single deflection sensitivity value cannot guarantee the accuracy of any measurement. The measured force, *F*, is the product of the spring constant of the cantilever (*k*), the deflection sensitivity (*d*) and the measured voltage difference at the photo detector *(ΔV).*

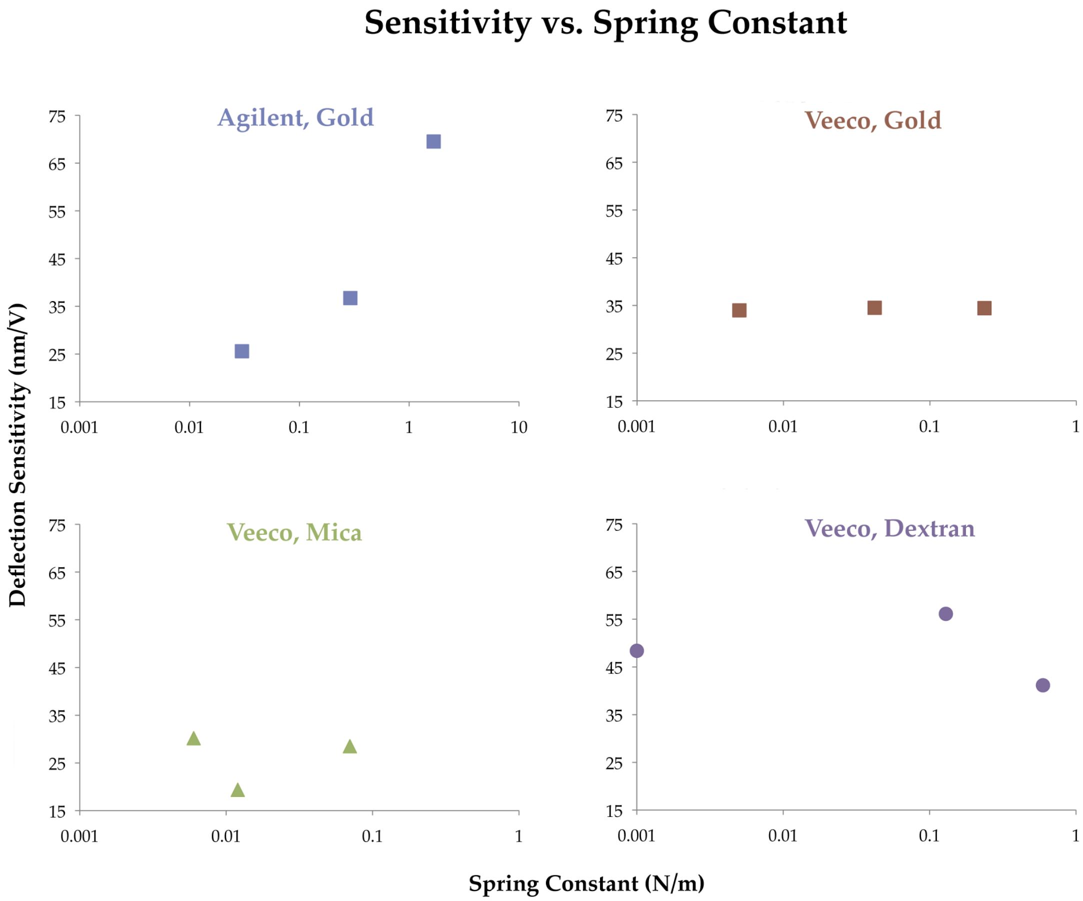
$$F = kd\Delta V$$

Appendix Equation A.1 Hook's Law

In many biological applications forces on the order of 10 pN are relevant. Using the OBL tip (k = 0.030 N/m, d = 25.58 ± 0.66 nm/V) as a reference, the voltage change needed to observe 10 pN of force is 12.61 mV.

If a deflection sensitivity value of 30.00 nm/V (an outlier) were used, the measured force would be 11.35 pN , which is a 13.5% difference. It becomes clear that reducing the uncertainty in measuring the deflection sensitivity is a straightforward way to increase the accuracy of spectroscopic measurements. Unfortunately, when working with biological samples and functionalized tips, this type of statistical analysis is not always possible before starting an experiment, as both the sample and the tip can be damaged quite easily.

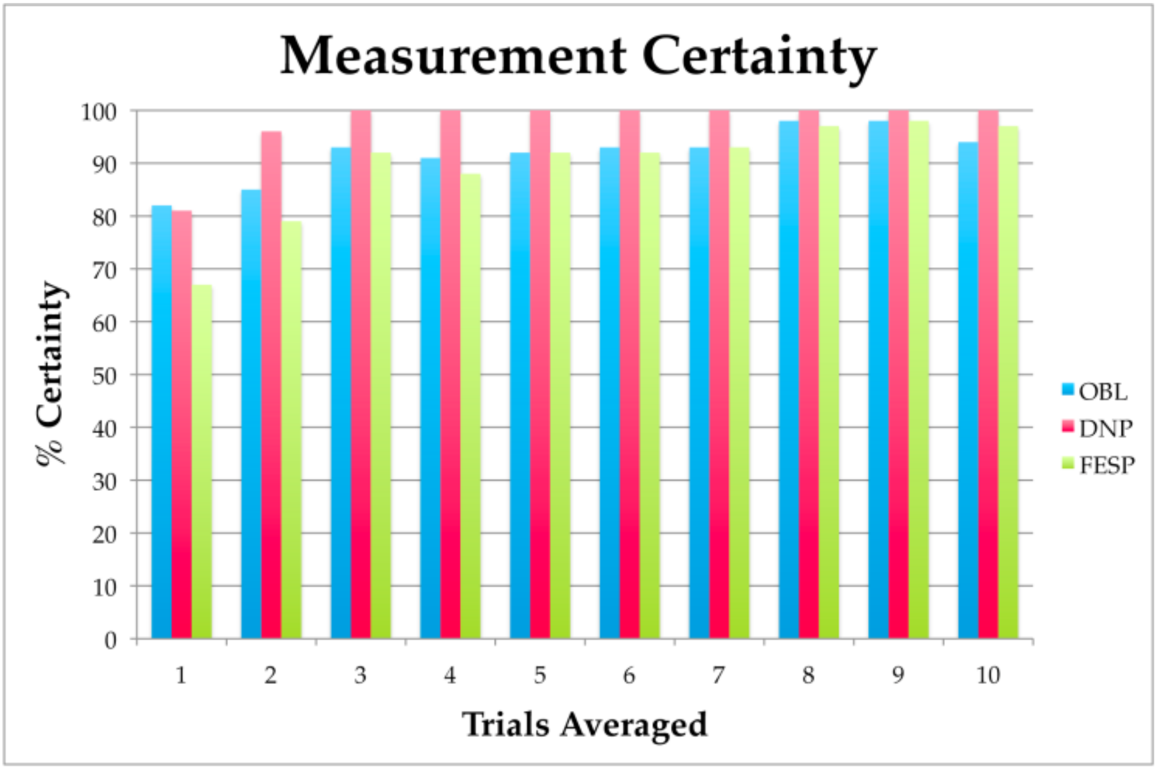
The first critical step to minimize damage to the sample and tip is to ensure that the force set point on approach is appropriate. If the set point is too low, the instrument constantly false engages. If the set point is too high, the tip crashes into the surface, destroying both the surface and the tip. Since the applied force depends on both the spring constant, *k*, and deflection sensitivity, *d*, both must be determined beforehand. While, *k* can be estimated without contacting the surface based on manufacturer specifications; however, the deflection sensitivity cannot be determined without contacting the surface. The solution then, is to estimate *d* for a given instrument and cantilever before approaching the sample. This can be done with a reference sample as shown in Appendix Figure A.2.



Appendix Figure A.2 Sensitivity vs. Spring Constant
Mean deflection sensitivity vs. measured spring constant for the different tips used. Error bars are not included because they are smaller than the symbol size. This plot allows for an initial estimate of the deflection sensitivity of any given cantilever.

By measuring *d* on spare samples with multiple tips with the same nominal spring constant, reasonable values for *k* and *d* can be determined. By measuring *k* first, *d* can be estimated for a given surface, prior to approaching the sample. This minimizes the potential for damage to the tip or sample. This type of deflection calibration should be done for all substrates of interest on each instrument separately.

Following tip approach, the final step in the process is to measure the actual deflection sensitivity of the tip. As each cycle has the possibility of damaging the tip or sample (since they have to be run over the full extent of the linear response range for a given cantilever), minimizing the number of deflection curves needed reduces the potential for damage. In order to determine the minimum number of values for *d* needed to ensure accuracy, one hundred random samples were taken from the data, representing the average of between 1 and 10 different measurements. The number of times the average value fell within ±2σ of μ was recorded to produce Appendix Figure A.3.



Appendix Figure A.3 Measurement Certainty
The number of times a random sample (one hundred in total) of all deflection sensitivity values for a given tip lies within ±2σ of μ. The horizontal axis represents the number of values averaged to generate a particular sample. For example a value of 5 means that five deflection sensitivities were averaged to generate a particular sample. Random sampling (Trials = 1) has a relatively low certainty for all tips. Averaging two samples increases certainty for all tips. Averaging between three and ten samples all provide increased, yet statistically identical, levels of certainty.

A single measurement, for either the OBL or DNP tips, results in an accurate measurement about 80% of the time. Taking the average of just two measurements significantly increases the accuracy of measuring *d*. Averaging three or more measurements returns a deflection sensitivity value within in ±2σ of μ more than 90% of the time. Repeating this analysis on data from the PicoForce instrument yields the same conclusion.

Conclusion

In summary, a procedure has been developed for reducing the risk of sample and tip damage, while increasing the accuracy of force spectroscopy measurements. First generate a calibration plot of deflection sensitivity vs. spring constant for each instrument on a reference substrate. Next, measure the spring constant of a given tip, and use it to estimate the deflection sensitivity before engaging the sample. Third, average at least three deflection sensitivities before beginning the experiment to ensure accurate force measurements. Finally, withdraw and measure the spring constant of the tip again in order to determine an accurate value.

Appendix B

Bone Code

Rational

The fundamental goal of this appendix is to outline how the 2D-FFT technique can be implemented is a fast and accurate way to minimize time needed to analyze data. The first script eliminates the need for multiple commercial applications and significantly reduced the time need process individual images. The second script automates the data analysis and figure generation to improve both speed of implementation and consistency of results.

System Requirements and Comments

Matlab------------------------- >= R2008b

Image processing toolkit ---- >= R2008b

Statistics toolkit-------------- >= R2008b

All code below is strictly for reference only. Electronic copies are in the possession of Professor Mark M. Banaszak Holl, myself, and available online at:

https://sourceforge.net/projects/collagenfourier/

Users' Guide Collagen_fft.m

Image Formats

The script "collagen_fft" is used to identify, measure, and annotate the D-Period spacing of measured collagen fibrils from image files.

Tiff, jpeg, gif, and bitmap files are all valid inputs

Input images must be of square sizes, ideally with 2^n number of data points. n should be between 8 and 12 as required by the image size to resolve fibrils. That is, images between 256x256 and 4096x4096 pixels work well.

Non-square or other irregular images must be cropped or embedded to a square format with the correct sizes. This is generally not a problem with AFM images. The final file should be a RGB format. This is generally not a problem for most software as this is the default setting.

Default Parameters

The parameters on line 6 "Default_path" sets the initial image directory to search in. This should be changed to the source image directory. Matlab's help describes how to define these paths. The following are suggested defaults

Windows:

'C:\Documents and Settings\"user name"\directory'

Mac OS X:

'~/Documents/"directory"/'

Line 7 sets the default search parameters used to automate the peak fitting. They take the following form:

{'image size in nm','expected D-period spacing in nm','search window size in nm'} The first parameter sets the lateral scaling of the image, and should be adjusted to the default sizes.

Upon running the script a dialog box will open asking you to pick a source image file and whether or not you would like to start new measurements or resume previous measurements. Please pick appropriately. New measurements on a previously analyzed image will overwrite values.

Another dialog box will open with inputs for the search parameters to verify that they are correct for this particular image.

A Practical Guide to Drawing Regions of Interest

Once an image is loaded you will have to draw boxes around fibrils of interest. The minimum requirements are outlined in detail in the body of the methods paper. Unfortunately, absolute best practices are much harder to define, as defining a region of interest is inherently a subjective set of conditions. There are a number of things that can help make things a bit easier in the end. They are as follows:

Calibrate the System Correctly

> Small amounts of additional effort, which start even before image acquisition can far outweigh any number of heroic attempts later on during analysis. Taking the time to do a few, relatively basic, things right the first time is entirely worth it. Most of the following comments are directed at AFM users; however, these general principles apply to other imaging techniques as well
>
> Calibrate, align, recalibrate, maintain, and understand your imaging system well. More than anything else, this is absolutely the most critical step in pre-

acquisition. AFM calibration is a bit of a black art to say the least, and commercial guides are meant for basic qualitative imaging, not quantitative measurements! The first step is certainly to follow the manufacturer's procedure as an initial starting point. This will help establish a reasonable baseline for everything from which you can make small adjustments and corrections. The first thing that has to be established is an orthogonal basis with no rotation relative to the sample plane. For SEMs and TEMs this is done by the manufacturer (Lucky EM folks). For SPM users, this simply is not the case. If your system has sample plane adjustments, (Agilent PicoPlus, Veeco Explorer, Caliber) take the day necessary to set the sample plane perpendicular to the Z-axis of the scanner. For the PicoPlus this involves imaging a sample holder, with all flattening off, and adjusting each alignment screw until the sample's flat. Use cheap tips and have patience. You will break many in this process. Make small changes, see if things improve and adjust accordingly. This process will take a day to do well. It's just how it goes. In the end, the less you waste dynamic range of the AFM on sample tilt, the better your images are going to be.

Now that the Z-Axis axis is perpendicular to the sample plane, the X and Y-Axes need to be orthogonalized. A basic calibration should get things pretty close. The angle between known perpendicular samples will need to be measured (calibration grid) on a sample w/o sample drift. If there is drift you will need to set the fast scan axis of the scanner parallel to the direction of the drift. After that is done, calibrating the scanner is another iterative process. Make small changes at any given time getting things closer. Calculating the true "correction factor" is of limited use as this is implicitly dependant on the correct calibration, which hasn't been set yet. Once the scan is "square," rotate the scan 90°. If the scan is still square then all is good. If the scan is no longer square then the scanner is probably not parallel to the sample drift. Check this in multiple regions, not just the center of the scanner's range! Non-linearity in the piezo can lead to distorted regions. Avoid any distorted regions for actual measurements.

After the scanner's basis is orthoganalized, the X-Y-Z voltage sensitivities have to be set carefully so that the distance scales are correct and equal. Again this is an iterative process. Adjust each parameter until the error is minimized. For quantitative work, errors less than 0.25% can be obtained with a reasonable amount of effort. Images will need to be collected with a high enough pixel density to resolve the grid very well.

Check the calibration through a range of reasonable imaging sizes and line rates. If the results are not consistent, the scanner absolutely must be calibrated again at the experimental range of interest.

Acquire the Best Image Possible

This sounds like a simple enough of a statement; but, the truth is, getting high quality images is a difficult task that takes as much luck as it does perseverance and skill. The first recommendation is if something's known to be bad, stop using it. If there is a bad tip, get a new one. If the sample is not prepared right, get another one. If the scanner is not calibrated correctly, calibrate it... Too many other things can prove challenging, it is not worth fighting something you know is wrong.

Imaging parameters must be optimized for each sample. What works well at one image scale does not always work well at another. Lower applied forces/dissipated energy allows for smaller feature to be observed at the cost of larger features. Higher integral and proportional gains respond faster to changes at the expense of increased "ringing." A general method that works well for most samples is to engage at relatively low forces, lower the setpoint until the tip just stops interacting with the surface then raise the setpoint until the scanner barely tracks the surface. From there increase the gains until the scanner almost "rings." Using this as a baseline the setpoint and gains can be adjusted to get the best image. This strikes a good balance of low imaging forces and fast response time.

The next big step is to avoid known distortions of a particular scanner. Many scanners are prone to oscillations when the scanner changes directions at the end of each raster line. Slight overscans can often put these oscillations outside of the acquired image. Most scanners exhibit some degree of piezo creep when moved over large distances, when adjusting scan centers, or scan size, a few scan restarts in rapid succession can quickly dampen this creep. If sample drift occurs to a noticeable extent, rotate the scanner axis so the drift is parallel to the fast scan direction. This will minimize any cumulative errors along the slow scan axis. Most scanners are extremely temperature sensitive. Minimizing disturbance and temperature fluctuation is an important consideration.

Once these steps have been performed, an actual digitized image has to be acquired. A reasonable pixel density must be used for each sample. While it may be tempting to use lower pixel densities, especially since the 2D-FFT technique handles this relatively gracefully, to acquire images more rapidly, this should be avoided whenever possible as this makes it harder on the user to identify the edges of features.

Preparing Images For Analysis

Any imaging chain is only as good as the weakest link; this is especially true for quantitative work. Mistakes at this step negate any effort done prior, and are often the easiest mistakes to miss. Even though an amplitude or deflection image is virtually flat since they are error signals, a simple 3rd order polynomial flattening goes a long way to remove general artifacts due to slightly faster scanning than ideal. The last step before analysis is to ensure that the image is properly formatted, exported, and utilizes the full dynamic range of the final format. To do this, the peak of the image histogram should be centered within the dynamic range. This will be the case for a reasonably flat sample following background correction; otherwise it should be adjusted manually. Next the color range should be reduced as much as possible without clipping highlights and shadows while remaining symmetric around the center. (Basically just move the sliders until you reach the edge of the Gaussian). Finally the image should be exported as an uncompressed, color, TIFF file to avoid any image compression artifacts, which may bias the results.

After each of these steps has been completed it is possible to properly analyze individual fibrils. The first point of the box should start at the bottom left most corner of the fibril. The second point should run parallel to the fibril to the bottom right most corner. The third point should be placed at the upper right corner. The final should be at the upper left most corner. Double clicking will close the box and save the region allowing you to continue. (Right clicks also make those options available)

Following this, the FFT will be presented for review. We suggest a minimum of 3 acceptable repetitions per fibril. Clicking on repeat fibril will allow you to measure the same fibril, clicking new fibril will allow you to measure a new fibril, and clicking on end will end the session and save all data.

All data is saved after each run to facilitate recovery at intermediate steps

The following is recorded per measurement:

D-Period Spacing (pixel max), D-Period orientation (pixel max), D-Period Spacing (centroid weighted), D-Period orientation (centroid weighted). Centroid weighted values are only recorded when they are within a two pixel radius of the max value. These serve mostly as consistency checks.

Collagen_fft.m

```
%Disable Obnoxious warnings...
warning('off','Images:initSize:adjustingMag');
warning('off','MATLAB:MKDIR:DirectoryExists');


Default_path = '~/Desktop/';
Default_parameters = {'3500','67','20'};



%Full code starts here.  Please do not modify.
```

```
recover_button = questdlg('Recover from a Previous Crash or Start a New
Set of Measurements','Recover Previous?','Recover','New','New');
if strcmp(recover_button, 'New')
[filename, pathname]    = uigetfile( ...
                                    {'*.jpg;*.tif;*.gif;*.bmp','Image
Files (*.jpg,*.tif,*.gif,*.bmp)';
                                    '*.jpg',  'JPEGs (*.jpg)'; ...
                                    '*.tif','TIFFs (*.tif)'; ...
                                    '*.gif','GIFs (*.gif)'; ...
                                    '*.bmp','Bitmaps (*.bmp)'}, ...
                                    'Pick a file',...
                                    Default_path);
Input_param             = inputdlg( {'Image Size in nm','Default
Spacing nm','Search Range'},...
                                    'Image Search Settings',...
                                    1,...
                                    Default_parameters);
Image_size              = str2num(Input_param{1});
ant_space               = str2num(Input_param{2});
search_radius           = str2num(Input_param{3});

I = rgb2gray(imread([pathname,filename]));
x_test = 0;
y_test = 0;
radius = 0;
radius_prime = 0;
new_fibril = 1;
repeat_fibril = 1;
end_program = 0;
fibril = 1;
button = 'Repeat Fibril';
mkdir([pathname,'Matlab/']);

else
[filename, pathname]    = uigetfile( ...
                                    {'*.jpg;*.tif;*.gif;*.bmp','Image
Files (*.jpg,*.tif,*.gif,*.bmp)';
                                    '*.jpg',  'JPEGs (*.jpg)'; ...
                                    '*.tif','TIFFs (*.tif)'; ...
                                    '*.gif','GIFs (*.gif)'; ...
                                    '*.bmp','Bitmaps (*.bmp)'}, ...
                                    'Pick a file',...
                                    Default_path);
Input_param             = inputdlg( {'Image Size in nm','Default
Spacing nm','Search Range'},...
                                    'Image Search Settings',...
                                    1,...
                                    Default_parameters);
Image_size              = str2num(Input_param{1});
ant_space               = str2num(Input_param{2});
search_radius           = str2num(Input_param{3});
min_repeat_num          = 9;

I = rgb2gray(imread([pathname,filename]));
```

```
box_import =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'_box.txt'];
bounding_box = importdata(box_import);
x_test(1:5,1:size(bounding_box,1)) =
bounding_box(1:size(bounding_box,1),1:5)';
y_test(1:5,1:size(bounding_box,1)) =
bounding_box(1:size(bounding_box,1),6:10)';

radius_import =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'.txt'];
radius = importdata(radius_import);
fibril = size(bounding_box,1)+1;
radius_prime = 0;
new_fibril = 1;
repeat_fibril = 1;
end_program = 0;
button = 'Repeat Fibril';
mkdir([pathname,'Matlab/']);
end



while end_program == 0;

reps = 1;
 while repeat_fibril == 1;
 try

figure('Name',  'Image', 'NumberTitle', 'off' );
imshow(I);
hold on;
if reps == 1
    for i = 1:fibril-1
        if sqrt((x_test(2,i)-x_test(1,i))^2 + (y_test(2,i)-
y_test(1,i))^2)*Image_size/size(I,1) >= min_repeat_num*ant_space

text(x_test(1,i),y_test(1,i),num2str(i),'Color','y','FontSize',16);
          patch (x_test(:,i),
y_test(:,i),'y','FaceColor','none','EdgeColor','y','LineWidth',1);

        else

text(x_test(1,i),y_test(1,i),num2str(i),'Color','r','FontSize',16);
          patch (x_test(:,i),
y_test(:,i),'r','FaceColor','none','EdgeColor','r','LineWidth',1);
        end
 end

elseif fibril > 1

    for i = 1:fibril-1
        if sqrt((x_test(2,i)-x_test(1,i))^2 + (y_test(2,i)-
y_test(1,i))^2)*Image_size/size(I,1) >= min_repeat_num*ant_space

text(x_test(1,i),y_test(1,i),num2str(i),'Color','y','FontSize',16);
```

```
          patch (x_test(:,i),
y_test(:,i),'y','FaceColor','none','EdgeColor','y','LineWidth',1);

        else

text(x_test(1,i),y_test(1,i),num2str(i),'Color','r','FontSize',16);
          patch (x_test(:,i),
y_test(:,i),'r','FaceColor','none','EdgeColor','r','LineWidth',1);
        end
 end
end

[BW x_i y_i] = roipoly;
hold off;
close gcf;
x_test(1:length(x_i),fibril) = x_i;
y_test(1:length(x_i),fibril) = y_i;


%Save the box locations so they can be reloaded later
boxes = '';
for i = 1:fibril
 boxes = [boxes,[num2str(x_test(1,i), '%0.2f'),'\t',...
                 num2str(x_test(2,i), '%0.2f'),'\t',...
                 num2str(x_test(3,i), '%0.2f'),'\t',...
                 num2str(x_test(4,i), '%0.2f'),'\t',...
                 num2str(x_test(5,i), '%0.2f'),'\t',...
                 num2str(y_test(1,i), '%0.2f'),'\t',...
                 num2str(y_test(2,i), '%0.2f'),'\t',...
                 num2str(y_test(3,i), '%0.2f'),'\t',...
                 num2str(y_test(4,i), '%0.2f'),'\t',...
                 num2str(y_test(5,i), '%0.2f'),'\n']];
end
Output_Name_results_box =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'_box.txt'];
fid = fopen(Output_Name_results_box,'wt');
fprintf(fid,boxes,'char');
fclose(fid);

%Saving is over now



Masked_area = immultiply(I,BW);
fft_array = fftshift(abs(fft2(Masked_area)));
fft_array = fft_array.*fft_array;
fft_array = fft_array/max(max(fft_array));

figure('Name',  'FFT', 'NumberTitle', 'off' );
imshow(fft_array);
hold on;
scale_map = colormap(jet);

search_angle = -pi()+atan((y_i(2)-y_i(1))/(x_i(2)-x_i(1)));
points_space = linspace(search_angle-
pi()/6,search_angle+pi()/6,size(I,1)/2);
inner_x = Image_size/(ant_space +
search_radius/2)*cos(points_space)+size(I,1)/2+1;
inner_y = Image_size/(ant_space +
search_radius/2)*sin(points_space)+size(I,2)/2+1;
outer_x = Image_size/(ant_space -
search_radius/2)*cos(points_space)+size(I,1)/2+1;
outer_y = Image_size/(ant_space -
search_radius/2)*sin(points_space)+size(I,2)/2+2;
plot(inner_x,inner_y,'--r','LineWidth',2);
plot(outer_x,outer_y,'--g','LineWidth',2);

scale_mask = (-1*poly2mask([size(I,1)/2,inner_x],
[size(I,2)/2,inner_y], size(I,1),
size(I,2))+1).*(poly2mask([size(I,1)/2,outer_x], [size(I,2)/2,outer_y],
size(I,1), size(I,2))) ;
scale_mask = (bwlabel(scale_mask) == 1);
my_clim = [0 ,1*max(max(immultiply(scale_mask,fft_array)))];
caxis(my_clim);
colorbar;



fft_crop = immultiply(scale_mask,fft_array);
final_fft_mask = bwlabel(fft_crop > 1/2*max(max(fft_crop)));

% Centroid weighted average location.
centroid =
regionprops(final_fft_mask,immultiply(final_fft_mask,fft_crop),'Weighte
dCentroid');
index_a_centroid = centroid.WeightedCentroid(2);
index_b_centroid = centroid.WeightedCentroid(1);

% There is a known condition in matlab where 0 frequency component is
not
% at the center of the axis after a shift; but rather at the (center_x
+
% 1,center_y + 1)  I don't know why this is.  It's corrected for below.

[val_a index_a_peak] = max(fft_crop);
[val_b index_b_peak] = max(val_a);
index_a_peak = index_a_peak(index_b_peak);
center_x    =  (size(I,1)/2 + 1);

if sqrt((index_a_centroid-index_a_peak)^2+(index_b_centroid-
index_b_peak)^2) < 4
    radius(fibril,4*reps-3) = Image_size/sqrt((center_x -
index_a_peak)^2     + (center_x - index_b_peak)^2);
    radius(fibril,4*reps-1) = Image_size/sqrt((center_x -
index_a_centroid)^2 + (center_x - index_b_centroid)^2);
    radius(fibril,4*reps-2) = 180-atan((center_x - index_a_peak)/(
center_x - index_b_peak))*180/pi();
    radius(fibril,4*reps-0) = 180-atan((center_x - index_a_centroid)/(
center_x - index_b_centroid))*180/pi();
else
```

```
    radius(fibril,4*reps-3) = Image_size/sqrt((center_x -
index_a_peak)^2 + (center_x - index_b_peak)^2);
    radius(fibril,4*reps-2) = 180-atan((center_x -
index_a_peak)/(center_x - index_b_peak))*180/pi();
    radius(fibril,4*reps-1) = NaN;
    radius(fibril,4*reps-0) = NaN;
end
disp({[  'Fibril ' ,num2str(fibril),' Repeat ',num2str(reps)];...
      [  'Radius pixel '      ,num2str(radius(fibril,4*reps-3)),'
nm'];...
      [  'Angle pixel '       ,num2str(radius(fibril,4*reps-2)),'
deg'];...
      [  'Radius centroid '   ,num2str(radius(fibril,4*reps-1)),'
nm'];...
      [  'Angle centroid '    ,num2str(radius(fibril,4*reps-0)),'
deg']});

plot(index_b_peak,index_a_peak,'+black','MarkerSize',20);
plot(index_b_centroid,index_a_centroid,'xmagenta','MarkerSize',20);

axis([round(size(I,1)*3/8),round(size(I,1)*5/8),round(size(I,2)*3/8),ro
und(size(I,2)*5/8)]);
legend( [num2str(ant_space + search_radius/2), ' nm'],...
        [num2str(ant_space - search_radius/2), ' nm'],...
        [num2str(radius(fibril,4*reps-3), '%0.2f'),' nm @ ',...
         num2str(radius(fibril,4*reps-2), '%0.2f'),'^\o'],...
        [num2str(radius(fibril,4*reps-1), '%0.2f'),' nm @ ',...
         num2str(radius(fibril,4*reps-0), '%0.2f'),'^\o']);
hold off;

if  sqrt((x_test(2,fibril)-x_test(1,fibril))^2 + (y_test(2,fibril)-
y_test(1,fibril))^2)*Image_size/size(I,1) <= min_repeat_num*ant_space
    button = questdlg('This fibril is too short, how do you want to
proceed?',':)','Repeat Fibril','New Fibril','End','Repeat Fibril');
    if strcmp(button, 'Repeat Fibril')
        repeat_fibril = 1;
        reps = 0;
    elseif strcmp(button, 'New Fibril')
        repeat_fibril = 1;
        fibril = fibril+1;
        reps = 0;
    elseif strcmp(button, 'End')
        end_program = 1;
        repeat_fibril = 0;
    end
else
button = questdlg('How would you like to proceed?',':)','Repeat
Fibril','New Fibril','End','New Fibril');
    if strcmp(button, 'Repeat Fibril')
        repeat_fibril = 1;
    elseif strcmp(button, 'New Fibril')
        repeat_fibril = 1;
        fibril = fibril+1;
        reps = 0;
    elseif strcmp(button, 'End')
```

```
        end_program = 1;
        repeat_fibril = 0;
    end
end



 close gcf;

catch ME
reps = 0;
x_test = x_test(1:length(x_test),1:fibril-1);
y_test = y_test(1:length(y_test),1:fibril-1);
close gcf;
 end
  reps = reps +1;


%This is the portion of the code that saves the radius values


Output_Name_results =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'.txt'];
fid = fopen(Output_Name_results,'wt');
for i = 1:size(radius,1)
        for j = 1:size(radius,2)
            if j == size(radius,2)
                fprintf(fid,[num2str(radius(i,j),
'%0.2f'),'\n'],'char');
            else
                fprintf(fid,[num2str(radius(i,j),
'%0.2f'),'\t'],'char');
            end
        end
end
fclose(fid);
% Saving ends here

 end

%This is the portion of the code that saves the radius values


Output_Name_results =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'.txt'];
fid = fopen(Output_Name_results,'wt');
for i = 1:size(radius,1)
        for j = 1:size(radius,2)
            if j == size(radius,2)
                fprintf(fid,[num2str(radius(i,j),
'%0.2f'),'\n'],'char');
            else
                fprintf(fid,[num2str(radius(i,j),
'%0.2f'),'\t'],'char');
```

```
            end
        end
end
fclose(fid);
% Saving ends here
end

%This is the portion of the code that saves the radius values


Output_Name_results =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'.txt'];
fid = fopen(Output_Name_results,'wt');
for i = 1:size(radius,1)
        for j = 1:size(radius,2)
            if j == size(radius,2)
                fprintf(fid,[num2str(radius(i,j),
'%0.2f'),'\n'],'char');
            else
                fprintf(fid,[num2str(radius(i,j),
'%0.2f'),'\t'],'char');
            end
        end
end
fclose(fid);
% Saving ends here


%This is the portion of the code that saves the final radius values
for i = 1:size(radius,1)
    for j = 1:size(radius,2)/4;
       if radius(i,size(radius,2)-3) == 0;
           radius(i,5:size(radius,2)) = radius(i,1:size(radius,2)-4);
           radius(i,1:4) = NaN;
       end
    end
end

radius_prime = zeros(size(radius));
for k = 1:4
if k == 1
radius_prime(:,size(radius,2)/4*(k-1)+1:size(radius,2)/4*k) =
radius(:,1:4:size(radius,2));
elseif k == 2
radius_prime(:,size(radius,2)/4*(k-1)+1:size(radius,2)/4*k) =
radius(:,3:4:size(radius,2));
elseif k == 3
radius_prime(:,size(radius,2)/4*(k-1)+1:size(radius,2)/4*k) =
radius(:,2:4:size(radius,2));
elseif k == 4
radius_prime(:,size(radius,2)/4*(k-1)+1:size(radius,2)/4*k) =
radius(:,4:4:size(radius,2));
end
end
radius_prime;
```

```
Output_Name_results_sorted =
[pathname,'Matlab/',regexprep(filename,'\.','_'),'_sorted.txt'];
fid = fopen(Output_Name_results_sorted,'wt');
for i = 1:size(radius_prime,1)+1
    if i ==1
        for j = 1:5
            if j ==1
                fprintf(fid,['Fibril','\t'],'char');
            elseif j == 2
                label = '';
                for k = 1:size(radius_prime,2)/4
                label = [label,'Radius (nm) (Pixel Max)','\t'];
                end
                fprintf(fid,label,'char');

            elseif j == 3
                label = '';
                for k = 1:size(radius_prime,2)/4
                label = [label,'Radius (nm) (Centroid)','\t'];
                end
                fprintf(fid,label,'char');

            elseif j == 4
                label = '';
                for k = 1:size(radius_prime,2)/4
                label = [label,'Angle (deg) (Pixel Max)','\t'];
                end
                fprintf(fid,label,'char');
            elseif j ==5
                label = '';
                for k = 1:size(radius_prime,2)/4-1
                label = [label,'Angle (deg) (Centroid)','\t'];
                end
                label = [label,'Angle (deg) (Centroid)','\n'];
                fprintf(fid,label,'char');

            end
        end
    else
        for j = 1:size(radius_prime,2)+1
            if j ==1
                fprintf(fid,[num2str(i-1),'\t'],'char');
            elseif j == size(radius_prime,2)+1
                fprintf(fid,[num2str(radius_prime(i-1,j-1),
'%0.2f'),'\n'],'char');
            else
                fprintf(fid,[num2str(radius_prime(i-1,j-1),
'%0.2f'),'\t'],'char');
            end
        end
    end
end
fclose(fid);
% Saving ends here
```

```
figure('Name',  'Final Image', 'NumberTitle', 'off' );
imshow(I);
hold on;

 for i = 1:size(x_test,2)
        if sqrt((x_test(2,i)-x_test(1,i))^2 + (y_test(2,i)-
y_test(1,i))^2)*Image_size/size(I,1) >= min_repeat_num*ant_space

text(x_test(1,i),y_test(1,i),num2str(i),'Color','y','FontSize',16);
          patch (x_test(:,i),
y_test(:,i),'y','FaceColor','none','EdgeColor','y','LineWidth',1);

        else

text(x_test(1,i),y_test(1,i),num2str(i),'Color','r','FontSize',16);
          patch (x_test(:,i),
y_test(:,i),'r','FaceColor','none','EdgeColor','r','LineWidth',1);
        end
 end

axis('tight','square');
print('-djpeg' ,[pathname,'Matlab/',filename]);
```

Users' Guide Population_analysis.m

The population_analysis.m script analyzes the distribution of observed D-Period spacings in order to look for statistically different populations. Users must reformat the output of the collagen_fft.m script as appropriate for their particular application. The files must be tab delimited text files with descriptive column headers. Any number of populations greater than or equal to 2 can be handled. The columns should be ordered as the following: Column 2*n-1 = sample identifier (numeric values only), Column 2*n = D-Period spacing (nm). The column headers for the sample identifier and D-Period spacing should be the same and must not contain non-alphanumeric symbols (in particular: "\", "/", or "." as these will cause major errors). An example file is included with this document as example.txt

On line 12, the default directory can be set. This follows the same convention as above.

The rest of the parameters can be easily set and interpreted based on the included comments. Of particular interest are line 38 which sets the output colors in RGB space, and lines 44-47 which control the axis formatting.

Lines 52 through 66 contain output formatting specific options, which can be adjusted to suit the user's taste and application.

Population_analysis.m

```
%Clears All previous variables to clean memory
clear all;


%Image Filename and path
warning('off','Images:initSize:adjustingMag');
warning('off','MATLAB:MKDIR:DirectoryExists');

[Filename, ImageDir]    = uigetfile({'*.txt','Text Files (*.txt)';
                                    '*.txt','Tab Delimited Text Files
(*.txt)'}, ...
                                    'Pick a file',...
                                    '~/Desktop/',...
                                    'MultiSelect','off');
[tok, file_type] = strtok(Filename,'.');
tok = strfind(Filename,'.');
Filename_edit = Filename(1:tok(length(tok))-1);

number_of_peaks     = 1;
% Number of populations to look for
min_spacing         = 54;
% Minimum D-Spacing.  Good for getting rid of low value peaks in the
kernel density
asymptotic_bins     = 0;
% Number of asymptotic bins to include at the end
seperation_factor   = 2;
%
bin_size_spacing    = 1;
% D-Period bin size
min_spacing_stdev   = .5;
% Minimum Fit width in nm
max_spacing_stdev   = 4;
% Maximum Fit width in nm
number_bins_move    = 1;
% Number of bins the center of the fitted peaks can move
amplitude_fract     = .2;
% Fraction the peak height can change
spacing_kernel      = .5;
% Rupture Kernel size.  Bigger = smoother, smaller = more populations
spec_spacing_center = 'Yes';
spacing_center      = [67];
```

```
bins_to_label       = 2;
kernel_restriction  = 'unbounded';
% Restricts the kernel density.  'positive' for only positive values
otherwise 'unbounded'

% changing this requires resetting the kernel sizes


% For the colors:   [.332 .555 .832]    = joey's blue
%                   [.5 .5 .5]          = joey's grey
%                   [.570 .813 .313]    = joey's green
bar_color           = {[.5 .5 .5] [.332 .555 .832] [.570 .813 .313]};
% Histogram colors.  First is for forces, second is rates
disp_gaussian       = 'Ye';
% Turns the Histogram fitting display on or off "Yes" = on, everything
else = off
disp_kernel         = 'Yes';
% Turns the Kernel density display on or off "Yes" = on, everything
else = off
disp_kernel_fit     = 'Ye';
% Turns the Kernel fitting display on or off "Yes" = on, everything
else = off
disp_legend         = 'Yes';
% Turns the Legends on or off "Yes" = on, everything else = off

auto_axis           = 'o';
% 'on' Automatically sets axis heights anything else overides with user
specified  defaults
max_prob            = 35;
% User defined max Percentage
min_spacing         = 54;
% User defined minimun spacing to display
max_spacing         = 78;
% User defined maximum spacing to display

max_iteration       = 5e4;
% Max variable iteration for fits
max_eval            = 5e4;
% Max fit evaluation

x_axis_label        = 'D-Periodic Gap/Overlap Spacing (nm)';
y_axis_label_hist   = 'Group Samples (%)';
y_axis_label_cdf    = 'Cumulative Total (%)';
hist_title          = '';
cdf_title           = '';

Axis_FontSize       = 20;
% Font Size for images
Label_FontSize      = 25;
Title_FontSize      = 28;
Legend_FontSize     = 14;
fig_width           = 10;
fig_height          = 7;
MarkerSize          = 20;
LineWidth           = 2;
```

```
FontName            = 'Arial';
% Font for images



% Make the Base Directories to save the images
mkdir([ImageDir,'Matlab/']);
mkdir([ImageDir,'Matlab/Figure Jpegs/']);
mkdir([ImageDir,'Matlab/Figure Figs/']);
mkdir([ImageDir,'Matlab/Text Files/']);







newData1_columns = importdata([ImageDir,Filename]);
newData1.data =
zeros(size(newData1_columns.data,1),size(newData1_columns.data,2)/2);
newData1.colheaders = '';
newData1.data(:,1:size(newData1_columns.data,2)/2) =
newData1_columns.data(:,2:2:size(newData1_columns.data,2));
newData1.colheaders =
newData1_columns.colheaders(2:2:size(newData1_columns.data,2));


% Create new variables in the base workspace from those fields.
vars = fieldnames(newData1);
for i = 1:length(vars)
    assignin('base', vars{i}, newData1.(vars{i}));
end


min_data_points = zeros(1,length(newData1.data(1,:)));
for i = length(newData1.data(1,:))
   min_data_points(i) = find(newData1.data(:,i)<Inf,1,'last');
end
bin_cdf_calc = seperation_factor*max(mean(diff(sort(newData1.data(1:
min(min_data_points(min_data_points > 0)),:)))));





for column = 1:length(newData1.colheaders)

rupture_sorted  = data(:,column);
data_label      = newData1.colheaders{column};
bin_start       = floor(min(min(data)))-
asymptotic_bins*bin_size_spacing;
bin_end         =
ceil(max(max(data)))+asymptotic_bins*bin_size_spacing;

% Here's the fun part.
% Cycle through the potential populations in the total data set.
```

```
Caluclate a
% histogram of the data set, Fit componenents and make pretty graphs

if strcmp(auto_axis,'on')
max_spacing = ceil(max(max(newData1.data))+bin_size_spacing);
min_spacing = floor(min(min(newData1.data))-bin_size_spacing);
else
end



% Caluclates the kernel density of the spacings.  Requires all the
% values to be positive.  Saves the kernel density of a file for future
use
if column == 1
kdens_rupture = zeros(512,length(newData1.colheaders));
x_val_rupture = zeros(512,length(newData1.colheaders));
else
end

[kdens_rupture(:,column),x_val_rupture(:,column)] =
ksdensity(rupture_sorted,bin_start:(bin_end-bin_start)/511:bin_end,
'support',kernel_restriction,'npoints',512,'width',spacing_kernel);
kdens_rupture(:,column) = kdens_rupture(:,column)*100*bin_size_spacing;
Output_Name_rupture = [ImageDir,'Matlab/Text
Files/',Filename_edit,'_kernel__',data_label,'.txt'];
kernel_rupture = [x_val_rupture(:,column) ,
kdens_rupture(:,column)*100];
fid = fopen(Output_Name_rupture, 'w');
fprintf(fid, ['D-Period Spacing (nm)','\t', 'Kernel ',data_label,'\n'],
'char');
fclose(fid);
save(Output_Name_rupture, 'kernel_rupture' ,'-ASCII', '-tabs','-
append');

% Does the histogram with the speficied bin width
if column == 1
count_rupture = zeros((bin_end-
bin_start)/bin_size_spacing+1,length(newData1.colheaders));
x_rupture = zeros((bin_end-
bin_start)/bin_size_spacing+1,length(newData1.colheaders));
else
end

[count_test x_test] =
hist(rupture_sorted,bin_start:bin_size_spacing:bin_end);
count_test = transpose(count_test);
x_test = transpose(x_test);

% Converts histogram to probablility values
count_test_total = sum(count_test);
count_test = count_test/sum(count_test)*100;

x_rupture(:,column) = x_test;
count_rupture(:,column) = count_test;
```

```
% Finds the first value above the minimum force for a rupture event
min_index_hist = find(x_rupture > min_spacing,1);
min_index_kern = find(x_val_rupture > min_spacing,1);


% Finds peaks in the rupture data.  Upto "number_ruputres" peaks can be
% found.  Peaks have to be greater than the minimum spacing and are
found in
% order of increasing spacing.
if strcmp(spec_spacing_center, 'Yes')
locs_rupture_hist = find(x_rupture(:,column) >=
spacing_center,1,'first');
peaks_rupture_hist = count_rupture(locs_rupture_hist,column);
locs_rupture_kern = zeros(1,length(locs_rupture_hist));
peaks_rupture_kern = zeros(1,length(locs_rupture_hist));
for i = 1:length(locs_rupture_hist)
locs_rupture_kern(i) = find(x_val_rupture(:,column) >
spacing_center(i),1,'First');
end
peaks_rupture_kern = kdens_rupture(locs_rupture_kern);

else
[peaks_rupture_hist,locs_rupture_hist] =
findpeaks(count_rupture(min_index_hist:length(count_rupture),column),'M
INPEAKDISTANCE',1,'NPEAKS',number_of_peaks);
[peaks_rupture_kern,locs_rupture_kern] =
findpeaks(kdens_rupture(min_index_kern:length(kdens_rupture),column),'M
INPEAKDISTANCE',1,'NPEAKS',number_of_peaks);
locs_rupture_hist = locs_rupture_hist + min_index_hist -1;
locs_rupture_hist = find(x_rupture(:,column) ==
locs_rupture_hist,1,'first');
locs_rupture_kern = locs_rupture_kern + min_index_kern -1;
end

min_spacing_count = find(count_rupture >0,1,'first') -1;



if column == 1
cdf_hist = zeros(length(newData1.data),2*length(newData1.colheaders));
else
end

[cdf_overwrite cdf_x_overwrite] =ecdf(newData1.data(:,column));
cdf_overwrite = 100*cdf_overwrite;
cdf_hist(1:length(cdf_overwrite),2*column-1:2*column) =
[cdf_x_overwrite cdf_overwrite];

[cdf_zero_row cdf_zero_column] = find(cdf_hist(2:length(cdf_hist),:) ==
0);
cdf_zero_row = cdf_zero_row + 1;
A = [cdf_zero_row cdf_zero_column];
for I = 1:length(A)
  cdf_hist(A(I,1),A(I,2))=NaN;
end
```

```
if column == 1
cdf_kern =
zeros(length(kernel_rupture(:,2)),length(newData1.colheaders));
else
end
for index = 1:length(kernel_rupture(:,2))
    cdf_kern(index,column) =
sum(kernel_rupture(1:index,2))/sum(kernel_rupture(1:length(kernel_ruptu
re(:,2)),2))*100;
end

    % Identifies the peak of the histogram and uses that to set the
fitting
    % paramters for the gaussian fit.

    % Initializes the varibles
    func_form = [];
    t_hist_lower_a = zeros(1,length(locs_rupture_hist));
    t_hist_lower_s = zeros(1,length(locs_rupture_hist));
    t_hist_lower_x = zeros(1,length(locs_rupture_hist));

    t_hist_upper_a = zeros(1,length(locs_rupture_hist));
    t_hist_upper_s = zeros(1,length(locs_rupture_hist));
    t_hist_upper_x = zeros(1,length(locs_rupture_hist));

    t_hist_start_a = zeros(1,length(locs_rupture_hist));
    t_hist_start_s = zeros(1,length(locs_rupture_hist));
    t_hist_start_x = zeros(1,length(locs_rupture_hist));

    % Cycles through the different peaks found in the histogram to
generate the starting
    % values and bounds  Peaks are constrained to within 5% of the peak
    % value, widths are defined by the user above and, and the centers
of
    % the gausians can only move upto 1 bind width.  (The center stays
    % within the same bin)
    % The last step creates the fitting function which is a sum of
    % gaussians equal to the number of peaks

    for i = 1:length(locs_rupture_hist)
    t_hist_lower_a(i) = (1-amplitude_fract)*peaks_rupture_hist(i);
    t_hist_lower_s(i) = min_spacing_stdev;
    t_hist_lower_x(i) = x_rupture(locs_rupture_hist(i))-
number_bins_move/2*bin_size_spacing;

    t_hist_upper_a(i) = (1+amplitude_fract)*peaks_rupture_hist(i);
    t_hist_upper_s(i) = max_spacing_stdev;
    t_hist_upper_x(i) =
x_rupture(locs_rupture_hist(i))+number_bins_move/2*bin_size_spacing;

    t_hist_start_a(i) = 1.00*peaks_rupture_hist(i);
    t_hist_start_s(i) = (min_spacing_stdev+max_spacing_stdev)/2;
    t_hist_start_x(i) = x_rupture(locs_rupture_hist(i));
    func_form = [func_form,'a',num2str(i),'*exp(-(x-
x',num2str(i),')^2/(2*(s',num2str(i),')^2))'];
    if i < length(peaks_rupture_hist)
        func_form = [func_form,'+'];
    end
    end

    % Combines all the bouds into a form for the fits
    t_hist_lower = [t_hist_lower_a,t_hist_lower_s,t_hist_lower_x];
    t_hist_upper = [t_hist_upper_a,t_hist_upper_s,t_hist_upper_x];
    t_hist_start = [t_hist_start_a,t_hist_upper_s,t_hist_upper_x];

    % Fits the data with the speficied conditions
    t_hist = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',      t_hist_lower,...
               'Upper',      t_hist_upper,...
               'Startpoint', t_hist_start,...
               'MaxIter',    max_iteration,...
               'MaxFunEvals',max_eval);
    f_Number_hist = fittype(func_form,  'options' ,t_hist);
    [fit_rupture_gaussian,gof_rupture_gaussian] =
fit(x_rupture(2:length(x_rupture),column),
count_rupture(2:length(x_rupture),column), f_Number_hist);

    % Creates an array with the fitting paramters, amplitude, center
and
    % width.  Saves the array to a text file
    rupture_hist_values = zeros(length(peaks_rupture_hist),4);
    for i = 1:length(peaks_rupture_hist)
    rupture_hist_values(i,1) = i;
    rupture_hist_values(i,2) =
eval(['fit_rupture_gaussian.a',num2str(i)]);
    rupture_hist_values(i,3) =
eval(['fit_rupture_gaussian.x',num2str(i)]);
    rupture_hist_values(i,4) =
eval(['fit_rupture_gaussian.s',num2str(i),'/sqrt(count_test_total)']);
    end




    Output_Name_hist_force = [ImageDir,'Matlab/Text
Files/',Filename_edit,'_hist_',data_label,'.txt'];
    fid = fopen(Output_Name_hist_force, 'w');
    fprintf(fid, ['Peak','\t', 'Amplitude','\t',
'Area','\t','Sigma','\n'], 'char');
    fclose(fid);
    save(Output_Name_hist_force, 'rupture_hist_values' ,'-ASCII', '-
tabs','-append');


    hist_cdf_out = cdf_hist(:,2*column-1:2*column);
    Output_Name_hist_cdf = [ImageDir,'Matlab/Text
Files/',Filename_edit,'_hist_cdf_',data_label,'.txt'];
    fid = fopen(Output_Name_hist_cdf, 'w');
    fprintf(fid, ['Bin (nm)','\t', 'CDF Hist ',data_label,'\n'],
'char');
    fclose(fid);
    save(Output_Name_hist_cdf, 'hist_cdf_out' ,'-ASCII', '-tabs','-
append');

    kern_cdf_out = [kernel_rupture(:,1),cdf_kern(:,column)];
    Output_Name_kernel_cdf = [ImageDir,'Matlab/Text
Files/',Filename_edit,'_kernel_cdf_',data_label,'.txt'];
    fid = fopen(Output_Name_kernel_cdf, 'w');
    fprintf(fid, ['Bin (nm)','\t', 'CDF Kern ',data_label,'\n'],
'char');
    fclose(fid);
    save(Output_Name_kernel_cdf, 'kern_cdf_out' ,'-ASCII', '-tabs','-
append');

    % Cycles through the different peaks found in the kernel density to
generate the starting
    % values and bounds  Peaks are constrained to within 5% of the peak
    % value, widths are defined by the user above and, and the centers
of
    % the gausians can only move upto 20% of one bin width.  (The
center stays
    % within the same bin)
    % The last step creates the fitting function which is a sum of
    % gaussians equal to the number of peaks

    func_form = [];
    t_kern_lower_a = zeros(1,length(peaks_rupture_kern));
    t_kern_lower_s = zeros(1,length(peaks_rupture_kern));
    t_kern_lower_x = zeros(1,length(peaks_rupture_kern));

    t_kern_upper_a = zeros(1,length(peaks_rupture_kern));
    t_kern_upper_s = zeros(1,length(peaks_rupture_kern));
    t_kern_upper_x = zeros(1,length(peaks_rupture_kern));

    t_kern_start_a = zeros(1,length(peaks_rupture_kern));
    t_kern_start_s = zeros(1,length(peaks_rupture_kern));
    t_kern_start_x = zeros(1,length(peaks_rupture_kern));

    for i = 1:length(peaks_rupture_kern)
    t_kern_lower_a(i) = (1-amplitude_fract)*peaks_rupture_kern(i);
    t_kern_lower_s(i) = min_spacing_stdev;
    t_kern_lower_x(i) = x_val_rupture(locs_rupture_kern(i))-
number_bins_move/2*bin_size_spacing;

    t_kern_upper_a(i) = (1+amplitude_fract)*peaks_rupture_kern(i);
    t_kern_upper_s(i) = max_spacing_stdev;
    t_kern_upper_x(i) =
x_val_rupture(locs_rupture_kern(i))+number_bins_move/2*bin_size_spacing
;

    t_kern_start_a(i) = 1.00*peaks_rupture_kern(i);
    t_kern_start_s(i) = (min_spacing_stdev+max_spacing_stdev)/2;
    t_kern_start_x(i) = x_val_rupture(locs_rupture_kern(i));
    func_form = [func_form,'a',num2str(i),'*exp(-(x-
x',num2str(i),')^2/(2*(s',num2str(i),')^2))'];
    if i < length(peaks_rupture_kern)
        func_form = [func_form,'+'];
    end
    end

    % Combines the bounds into a useful form
    t_kern_lower = [t_kern_lower_a,t_kern_lower_s,t_kern_lower_x];
    t_kern_upper = [t_kern_upper_a,t_kern_upper_s,t_kern_upper_x];
    t_kern_start = [t_kern_start_a,t_kern_upper_s,t_kern_upper_x];

    % Fits the kernel density
    t_kern = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',      t_kern_lower,...
               'Upper',      t_kern_upper,...
               'Startpoint', t_kern_start,...
               'MaxIter', max_iteration,...
               'MaxFunEvals', max_eval);
    f_Number_kern = fittype(func_form,  'options' ,t_kern);
    [fit_rupture_kern,gof_rupture_kern] =
fit(x_val_rupture(:,column),kdens_rupture(:,column),f_Number_kern);


    % Saves the fitting results to the kernel density to a text file
    rupture_kern_values = zeros(length(peaks_rupture_kern),4);
    for i = 1:length(peaks_rupture_kern)
    rupture_kern_values(i,1) = i;
    rupture_kern_values(i,2) = eval(['fit_rupture_kern.a',num2str(i)]);
    rupture_kern_values(i,3) = eval(['fit_rupture_kern.x',num2str(i)]);
    rupture_kern_values(i,4) =
eval(['fit_rupture_kern.s',num2str(i),'/sqrt(count_test_total)']);
    end
    Output_Name_kern_force = [ImageDir,'Matlab/Text
Files/',Filename_edit,'_kern_forces_',data_label,'.txt'];
    fid = fopen(Output_Name_kern_force, 'w');
    fprintf(fid, ['Peak','\t', 'Amplitude','\t',
'Area','\t','Sigma','\n'], 'char');
    fclose(fid);
    save(Output_Name_kern_force, 'rupture_kern_values' ,'-ASCII', '-
tabs','-append');

    % Calculates the values of the gaussian components
    % Initialize the array
    fit_Value_rupture =
zeros(length(x_val_rupture),length(peaks_rupture_hist));
    fit_Value_rupture_kern =
zeros(length(x_val_rupture),length(peaks_rupture_kern));

    % Cycles through the number of peaks in the histogram.  Calculates
the
    % values of that component
    for i = 1:length(peaks_rupture_hist)
    for x=1:length(x_val_rupture)
         fit_Value_rupture(x,i) =
eval(['fit_rupture_gaussian.a',num2str(i),'*exp(-
(x_val_rupture(',num2str(x),') -
fit_rupture_gaussian.x',num2str(i),')^2/(2*(fit_rupture_gaussian.s',num
2str(i),')^2))']);

    end
    end

    % Cycles through the number of peaks in the kernel density.
Calculates the
    % values of that component
    for i = 1:length(peaks_rupture_kern)
    for x=1:length(x_val_rupture)
        fit_Value_rupture_kern(x,i) =
eval(['fit_rupture_kern.a',num2str(i),'*exp(-
(x_val_rupture(',num2str(x),') -
fit_rupture_kern.x',num2str(i),')^2/(2*(fit_rupture_kern.s',num2str(i),
')^2))']);
    end
    end


    % Sets up the order of things in the plot legend
    if strcmp(disp_gaussian,'Yes') && strcmp(disp_kernel,'Yes') &&
strcmp(disp_kernel_fit,'Yes')
    leg_loc = length(peaks_rupture_hist)+2;
    force = cell(1,length(peaks_rupture_hist) +
length(peaks_rupture_kern)+2);

    elseif strcmp(disp_gaussian,'Yes') && strcmp(disp_kernel,'Yes')
    leg_loc = length(peaks_rupture_hist)+2;
    force = cell(1,length(peaks_rupture_hist) +2);

    elseif strcmp(disp_gaussian,'Yes') && strcmp(disp_kernel_fit,'Yes')
    leg_loc = length(peaks_rupture_hist)+1;
    force   = cell(1,length(peaks_rupture_hist) +
length(peaks_rupture_kern));

    elseif strcmp(disp_kernel,'Yes') && strcmp(disp_kernel_fit,'Yes')
    leg_loc = 2;
    force = cell(1,length(peaks_rupture_kern)+2);

    elseif strcmp(disp_gaussian,'Yes')
    force = cell(1,length(peaks_rupture_hist) + 1);

    elseif strcmp(disp_kernel_fit,'Yes')
    leg_loc = 2;
    force = cell(1,length(peaks_rupture_kern));

    elseif strcmp(disp_kernel,'Yes')
    leg_loc = 2;
    force = cell(1,length(peaks_rupture_kern));

    else
    force = cell(1,1);
    end
    % Plots the histogram and the fitted gaussian.  Makes it look
pretty,
    % saves the figure as a jpeg and an editible matlab figure.
    figure('Name',  'D-Period ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]) ,
bar(x_rupture(:,column),count_rupture(:,column),'FaceColor',bar_color{c
olumn},'EdgeColor',bar_color{column});
    hold on ;

    % Plot the fits to the histogram if that option is on.  Add a
legend
    % entry with the peak force and standard error
    if strcmp(disp_gaussian,'Yes');

        for i = 1:length(peaks_rupture_hist)
    plot(x_val_rupture,fit_Value_rupture(:,i),'--
','color',bar_color{column}, 'LineWidth', LineWidth);
    force{1+i} = ['Gaussian Fit ','Mean =
',num2str(eval(['fit_rupture_gaussian.x',num2str(i)]),'%3.2f'),' +- ',
num2str(eval(['fit_rupture_gaussian.s',num2str(i),'/sqrt(count_test_tot
al)']),'%3.2f'),' pN'];
        end
    end

    % Plot the kernel density and the fits to the kernel density if
that option is on.  Add a legend
    % entry with the peak force and standard error
    if strcmp(disp_kernel,'Yes')  && strcmp(disp_kernel_fit,'Yes');

plot(x_val_rupture(:,column),kdens_rupture(:,column),'color',bar_color{
column}, 'LineWidth', LineWidth);
    force{leg_loc} = 'Kernel Density';

    for i = 1:length(peaks_rupture_kern)
    plot(x_val_rupture(:,column),fit_Value_rupture_kern(:,i),'-
.','color',bar_color{column}, 'LineWidth', LineWidth);
    force{leg_loc+i} = ['Kernel Density ','Mean =
',num2str(eval(['fit_rupture_kern.x',num2str(i)]),'%3.2f'),' +- ',
num2str(eval(['fit_rupture_kern.s',num2str(i),'/sqrt(count_test_total)'
]),'%3.2f'),' pN'];
    end

    elseif strcmp(disp_kernel_fit,'Yes')
    for i = 1:length(peaks_rupture_kern)
    plot(x_val_rupture(:,column),fit_Value_rupture_kern(:,i),'-
.','color',bar_color{column}, 'LineWidth', LineWidth);
    force{leg_loc+i-1} = ['Kernel Density ','Mean =
',num2str(eval(['fit_rupture_kern.x',num2str(i)]),'%3.2f'),' +- ',
num2str(eval(['fit_rupture_kern.s',num2str(i),'/sqrt(count_test_total)'
]),'%3.2f'),' pN'];
    end

    elseif strcmp(disp_kernel,'Yes')
```

```
plot(x_val_rupture(:,column),kdens_rupture(:,column),'color',bar_color{
column}, 'LineWidth', LineWidth);
    force{leg_loc} = 'Kernel Density';
    end

    % Does the figure final formatting and saves the output
    force{1} =['D-Period Spacing ',data_label];
    if strcmp(disp_legend,'Yes')
    force_leg = legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    end
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label_hist,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(hist_title,'FontSize', Title_FontSize, 'FontName', FontName
);
    if strcmp(auto_axis,'on')
        max_prob = 1.25*max(max(count_rupture));
    end
    axis([min_spacing max_spacing 0 max_prob]);
    box(gca,'on');

set(gca,'XTick',min_spacing:bins_to_label*bin_size_spacing:max_spacing)
;
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename_edit,'_d-period_',data_label,'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure Figs/',Filename_edit,'_d-
period','.fig'],'fig');


    %Plot the CDF

    force = cell(1,1);
    figure('Name',  'D-Period CDF ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height])
    hold on
    stairs(cdf_hist(:,2*column-
1),cdf_hist(:,2*column),'.','color',bar_color{column},'Markersize',Mark
erSize);

stairs(kernel_rupture(:,1),cdf_kern(:,column),'color',bar_color{column}
,'LineWidth',LineWidth);


    % Does the figure final formatting and saves the output
    force{1} =['CDF ',data_label];
    force{2} =['Kernel CDF ',data_label];
    if strcmp(disp_legend,'Yes')
    force_leg = legend(force,'Location','northwest','FontName',
FontName,'FontSize',Legend_FontSize);
    end
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label_cdf,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(cdf_title,'FontSize', Title_FontSize, 'FontName', FontName );
    if strcmp(auto_axis,'on')
        max_prob = 1.25*max(count_rupture);
    end
    axis([min_spacing max_spacing 0 105]);
    box(gca,'on');

set(gca,'XTick',min_spacing:bins_to_label*bin_size_spacing:max_spacing)
;
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename_edit,'_d-period_cdf_',data_label,'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure Figs/',Filename_edit,'_d-
period_cdf','.fig'],'fig');

end

%Plot All Histograms
    figure('Name',  'All D-Period ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
    force = cell(length(newData1.colheaders),1);
    hold on;
    bar(x_rupture(:,1),count_rupture(:,:),'group');
    map = zeros(length(newData1.colheaders),3);
    for column = 1:length(newData1.colheaders)
    force{column} = newData1.colheaders{column};
    map(column,:) = bar_color{column};
    end
    colormap(map);
    if strcmp(disp_legend,'Yes')
    force_leg = legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    end
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label_hist,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(hist_title,'FontSize', Title_FontSize, 'FontName', FontName
);
    if strcmp(auto_axis,'on')
        max_prob = 1.25*max(max(count_rupture));
    end
    axis([min_spacing max_spacing 0 max_prob]);
    box(gca,'on');
    xtick = (min_spacing:bins_to_label*bin_size_spacing:max_spacing);
    set(gca,'XTick',xtick);
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename_edit,'_d-period_all.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure Figs/',Filename_edit,'_d-
period_all','.fig'],'fig');


%Plot All Kernel Densities
    figure('Name',  'All Kernel Density ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
    force = cell(2*length(newData1.colheaders),1);
    hold on;
    for column = 1:length(newData1.colheaders)
    force{2*column-1} = [newData1.colheaders{column}, ' Kernel' ];
    force{2*column} = [newData1.colheaders{column},' Histogram' ];

plot(x_val_rupture(:,column),kdens_rupture(:,column),'color',bar_color{
column}, 'LineWidth', LineWidth);

plot(x_rupture(:,column),count_rupture(:,column),'.','color',bar_color{
column},'MarkerSize', MarkerSize);
    end
    colormap(map);
    if strcmp(disp_legend,'Yes')
    force_leg = legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    end
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label_hist,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(hist_title,'FontSize', Title_FontSize, 'FontName', FontName
);
    if strcmp(auto_axis,'on')
        max_prob = 1.25*max(max(count_rupture));
    end
    axis([min_spacing max_spacing 0 max_prob]);
    box(gca,'on');

set(gca,'XTick',min_spacing:bins_to_label*bin_size_spacing:max_spacing)
;
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename_edit,'_kernel_all.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename_edit,'_kernel_all','.fig'],'fig');

%Plot All CDF's
figure('Name',  'All CDF ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
    %force = cell(2*length(newData1.colheaders),1);
    force = cell(length(newData1.colheaders),1);
    hold on;
    for column = 1:length(newData1.colheaders)
%     force{2*column -1} = [newData1.colheaders{column} ' Kernel'];
%     force{2*column} = [newData1.colheaders{column} ' Histogram'];
    force{column} = [newData1.colheaders{column} ' CDF '];

%plot(x_val_rupture(:,column),cdf_kern(:,column),'color',bar_color{colu
mn}, 'LineWidth', LineWidth);
    stairs(cdf_hist(:,2*column-
1),cdf_hist(:,2*column),'color',bar_color{column}, 'LineWidth',
LineWidth);

%plot(x_rupture(:,column)+bin_size_spacing/2,cdf_hist(:,column),'.','co
lor',bar_color{column}, 'MarkerSize', MarkerSize);
    end
    colormap(map);
    if strcmp(disp_legend,'Yes')
    force_leg = legend(force,'Location','northwest','FontName',
FontName,'FontSize',Legend_FontSize);
    end
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label_cdf,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(cdf_title,'FontSize', Title_FontSize, 'FontName', FontName );
    if strcmp(auto_axis,'on')
        max_prob = 1.25*max(max(count_rupture));
    end
    axis([min_spacing max_spacing 0 105]);
    box(gca,'on');

set(gca,'XTick',min_spacing:bins_to_label*bin_size_spacing:max_spacing)
;
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename_edit,'_cdf_all.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename_edit,'_cdf_all','.fig'],'fig');


    %Plot All Boxplots
    figure('Name',  'BoxPlots ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
    hold on;
    h =
boxplot(data,newData1.colheaders,'boxstyle','outline','medianstyle','li
ne','colors',cell2mat(transpose(bar_color)),'Whisker',Inf);
    hcc = get(get(gca,'Children'),'Children');
    hl = findobj(hcc,'type','text');
    set(h(1:length(h(:,1))-1,:),'LineWidth',LineWidth);
    set(hl,'FontSize', Axis_FontSize, 'FontName',
FontName,'VerticalAlignment','Top');
    ylabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(hist_title,'FontSize', Title_FontSize, 'FontName', FontName
);
    means = zeros(1,length(newData1.colheaders));
    for a= 1:length(newData1.colheaders)
        means(a) = mean(data(1:find(data(:,a) <=
max(data(:,a)),1,'last'),a));
    plot(a,means(a),'.','MarkerSize',MarkerSize,'Color',bar_color{a});
    end
    axis([0 length(newData1.colheaders)+1 min_spacing
max_spacing+5*bin_size_spacing]);
    box(gca,'on');

set(gca,'YTick',min_spacing:bins_to_label*bin_size_spacing:(max_spacing
+5*bin_size_spacing));
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename_edit,'_box_plot.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename_edit,'_box_plot','.fig'],'fig');


results_mean =
cell(length(newData1.colheaders)+1,length(newData1.colheaders)+1);
for i = 1:length(newData1.colheaders)
newData1_columns.data(:,2*i-1:2*i) =
sortrows(newData1_columns.data(:,2*i-1:2*i));
end

average = [];
groups = [];
for i = 1:length(newData1.colheaders)
groups = [groups ; newData1_columns.data(:,2*i-1)];
end
groups = sort(groups);
[peaks vals] = findpeaks(diff(groups));
if length(peaks) >= 1
    groups = groups([vals, vals(length(vals))+1]);
else
    groups = groups(1);
end
for i = 1:length(newData1.colheaders)
```

```
   for j = 1:length(groups)
    if length(find(newData1_columns.data(:,2*i-1) == groups(j))) > 0
        average(j,i) = 
mean(newData1_columns.data(find(newData1_columns.data(:,2*i-1) == 
groups(j)),2*i));
    else
        average(j,i) = NaN;
    end
   end
end

if length(groups) >=2
[anova_p,anova_m,anova_t] = anova1(average,newData1.colheaders,'off');
[anova_c,compare_m] = 
multcompare(anova_t,'ctype','bonferroni','display','off');

    for a = 1:length(newData1.colheaders)
        for b = a+1:length(newData1.colheaders)
            results_mean{1,b+1} = [newData1.colheaders{b},' 
',num2str(compare_m(b,1),'%10.2f'),' +-
',num2str(compare_m(b,2)*sqrt(anova_t.n(b)-1),'%10.2f'), ' nm'];
            results_mean{a+1,1} = [newData1.colheaders{a},' 
',num2str(compare_m(a,1),'%10.2f'),' +-
',num2str(compare_m(a,2)*sqrt(anova_t.n(a)-1),'%10.2f'), ' nm'];
            if anova_c(b-a,3) < 0  && 0 < anova_c(b-a,5)
                results_mean{a+1,b+1} = 'Not Significant';
            else
                results_mean{a+1,b+1} = 'Significant';
            end
         end
    end
    disp('Mean ANOVA results');
    disp(results_mean);
    Output_Name_results_anova = [ImageDir,'Matlab/Text 
Files/',Filename_edit,'_mean_anova_test_results.txt'];
        fid = fopen(Output_Name_results_anova,'wt');
        for i = 1:length(results_mean)
        for j = 1:length(results_mean)
            if j == length(results_mean)
            fprintf(fid,[results_mean{i,j},'\n'],'char');
            else
            fprintf(fid,[results_mean{i,j},'\t'],'char');
            end

        end
        end
        fclose(fid);
end

    %Calculate K-S Test results
    results_hist = 
cell(length(newData1.colheaders)+1,length(newData1.colheaders)+1);
    for a = 1:length(newData1.colheaders)
        for b= a+1:length(newData1.colheaders)
            [h,p] = kstest2(newData1.data(:,a),newData1.data(:,b));
            results_hist{1,b+1} = newData1.colheaders{b};
            results_hist{a+1,1} = newData1.colheaders{a};
            results_hist{a+1,b+1} = ['P Value = ',
num2str(p,'%10.4f')];
        end
    end
    disp('K-S results');
    disp(results_hist);
    Output_Name_results = [ImageDir,'Matlab/Text
Files/',Filename_edit,'_k-s_test_results.txt'];
        fid = fopen(Output_Name_results,'wt');
        for i = 1:length(results_hist)
        for j = 1:length(results_hist)
            if j == length(results_hist)
            fprintf(fid,[results_hist{i,j},'\n'],'char');
            else
            fprintf(fid,[results_hist{i,j},'\t'],'char');
            end

        end
        end
        fclose(fid);
```

Example Population Data

Set A	Set A	Set B	Set B	Set C	Set C
1	66.03	1	66.92	1	67.42
1	66.20	1	66.50	1	66.22
1	66.81	1	67.57	1	67.80
1	67.34	1	67.19	1	66.13
1	67.65	1	67.29	1	66.74
2	66.42	2	66.42	2	67.00
2	66.14	2	67.23	2	66.65
2	67.13	2	67.97	2	66.39
2	66.27	2	66.71	2	66.53
2	67.58	2	67.82	2	66.88
3	66.53	3	67.69	3	66.02
3	67.11	3	67.15	3	67.00
3	67.24	3	67.81	3	66.91
3	67.69	3	66.91	3	66.49
3	67.08	3	67.50	3	66.27
4	66.84	4	66.49	4	66.39
4	67.53	4	67.49	4	67.77
4	67.31	4	66.31	4	67.44
4	66.03	4	66.31	4	67.47
4	67.63	4	67.95	4	67.37
5	67.61	5	66.39	5	67.37

5	67.21	5	67.48	5	66.64
5	67.44	5	66.55	5	66.93
5	66.15	5	66.80	5	66.91
5	67.44	5	66.47	5	67.22

Appendix Table B.1 Example Population Data
Example data with column headers and group labels is shown above. The data is centerd at 67nm with a random difference on the interval [-1,1] added to it. Data formatted like this should be saved at tab-delimted text files.

Licensing Collagen_FFT.m and Population_Analysis.m

Appendix C

Flow Cytometry Compensation Code

Rational

Flow cytometery is a great tool for the biologist. It allows huge numbers of cells to be analyzed at a given time and with careful staining any number of different properties and cellular processes can be measured. Qualitative results from Flow are fast and incredibly useful if used correctly. The trouble is that the fluorescent dyes used to label the cells have significant overlap. Without proper compensation—removal of this spectral overlap—accurate quantitative measurements are impossible to make. Typical compensation methods are based on simple population matching by the user. This is fine for two color qualitative assessments; but it is nowhere near accurate enough for multi-color quantitative assessments. In order to address this, a set of Matlab scripts were written to correctly compensate, gate, and process flow cytometery data for quantitative analysis

Users' Guide Compensation.m

The Compensation.m file is reasonably straightforward to use. It measures the spill over from three single dye controls, calculates the inverse transform, and applies it to the experimental data at a particular time point. The corrected data is then saved for future use. Each file requires three independent Columns of data, the first for channel 1, the second for channel 2, and the third for channel 3. In the case of data with only two colors of real data, simply add a third column of zeros to each file the length of the dataset, except for the third channel single dye control. The third column in that file should linearly increase the length of the file. Each file should be saved as a tab delimited .txt file. Only one file is used for each single dye control. As for real data, any number of control data sets (>= 1) can be selected, followed by any number of experimental files (>= 1).

On line 5, the “fit_cutoff” parameter set the minimum fraction data to be excluded to avoid bias from auto fluorescence. 0.2 works very well.

On line 6, the “slope_lim” array sets the maximum spillover possible between channels. The maximum value for any element is 1. Changing this array can help remove bias from auto-fluorescence.

The parameters on line 15, "Default_path" sets the initial directory to search in. This should be changed to the source directory for the data files. Matlab's help describes how to define these paths. The following are suggested defaults

Windows:

'C:\Documents and Settings\"user name"\directory'

Mac OS X:

'~/Documents/"directory"/'

Compensation.m

```
clear all;
warning('off','MATLAB:MKDIR:DirectoryExists');
warning('off','MATLAB:MKDIR:DirectoryExists');
warning('off','MATLAB:MKDIR:OSError');

fit_cutoff = 0.2;
slope_lim = [1.000,0.030,0.020;...
             0.275,1.000,0.040;...
             0.030,0.275,1.000];

[Filename_1, ImageDir]    = uigetfile( ...
                                    {'*.txt','Data Files (*.txt)';...
                                    '*.txt','Tab Delimeted Text File
(*.txt)'}, ...
                                    'Channel 1 only control',...
                                    '/Users/berick/Desktop/Becky Flow
Data/',...
                                    'MultiSelect','on');
[Filename_2, ImageDir]    = uigetfile( ...
                                    {'*.txt','Data Files (*.txt)';...
                                    '*.txt','Tab Delimeted Text File
(*.txt)'}, ...
                                    'Channel 2 only control',...
                                    ImageDir,...
                                    'MultiSelect','on');

[Filename_3, ImageDir]    = uigetfile( ...
                                    {'*.txt','Data Files (*.txt)';...
                                    '*.txt','Tab Delimeted Text File
(*.txt)'}, ...
                                    'Channel 3 only control',...
                                    ImageDir,...
                                    'MultiSelect','on');

[Filename_6, ImageDir]    = uigetfile( ...
                                    {'*.txt','Data Files (*.txt)';...
                                    '*.txt','Tab Delimeted Text File
(*.txt)'}, ...
                                    'Blank Plasmid',...
                                    ImageDir,...
                                    'MultiSelect','on');

[Filename_7, ImageDir]    = uigetfile( ...
                                    {'*.txt','Data Files (*.txt)';...
                                    '*.txt','Tab Delimeted Text File
(*.txt)'}, ...
                                    'CFP Plasmid',...
                                    ImageDir,...
                                    'MultiSelect','on');

control_files = Filename_2;
ex1_files = Filename_6;
ex2_files = Filename_7;
```

```
randomize = questdlg('Preform Intra-Channel Randomization','Binning
correction','Yes','No','Yes');




% For the colors:   [.332 .555 .832]    = joey's blue
%                   [.5 .5 .5]          = joey's grey
%                   [.570 .813 .313]    = joey's green
%                   [.570 .297 .297]    = becky's red
%                   [.414 .344 .5]      = becky's purple
%                   [.379 .539 .344]    = becky's dark green

bar_color           = {[.5 .5 .5] [.332 .555 .832] [.570 .813 .313]
[.570 .297 .297] [.414 .344 .5] [.379 .539 .344]};
% Histogram colors.  First is for forces, second is rates
max_iteration       = 5e4;
% Max variable iteration for fits
max_eval            = 5e4;
% Max fit evaluation



Axis_FontSize       = 20;
% Font Size for images
Label_FontSize      = 25;
Title_FontSize      = 28;
Legend_FontSize     = 14;
fig_width           = 10;
fig_height          = 7;
MarkerSize          = 8;
LineWidth           = 3;
FontName            = 'Arial';
% Font for images

% Make the Base Directories to save the images
mkdir([ImageDir,'Matlab/']);
mkdir([ImageDir,'Matlab/Figure Jpegs/']);
mkdir([ImageDir,'Matlab/Figure Figs/']);
mkdir([ImageDir,'Matlab/Compensated/']);


for i = 1:3
    size_data = zeros(size(eval(['Filename_',num2str(i)]),2),3);
    rep_matrix = zeros(size(eval(['Filename_',num2str(i)]),2),3);
    for j = 1:size(eval(['Filename_',num2str(i)]),2)
    data(:,:,i,j) =
importdata([ImageDir,eval(['Filename_',num2str(i),'{',num2str(j),'}'])]
);
    size_data(i,j) = length(data(:,:,i,j));
    rep_matrix(i,j) = size(eval(['Filename_',num2str(i)]),2);
    end
end
```

```
colheaders = data(:,:,1,1).colheaders;

data_start = NaN(max(max(rep_matrix))*max(max(size_data)),3,4);
for i = 1:3
    for j = 1:size(eval(['Filename_',num2str(i)]),2)
    data_start(size(data(:,:,i,j).data,1)*(j-
1)+1:size(data(:,:,i,j).data,1)*j,:,i) = data(:,:,i,j).data;
    end
end

data_start(data_start == 0) = NaN;
data_start(1:size(data_start,1),:,1) =
sortrows(data_start(1:size(data_start,1),:,1),1);
data_start(1:size(data_start,1),:,2) =
sortrows(data_start(1:size(data_start,1),:,2),2);
data_start(1:size(data_start,1),:,3) =
sortrows(data_start(1:size(data_start,1),:,3),3);
data_original = data_start;

% This handles the inter channel randomization.  It "fixes" the display
% artifacts caused by binning and correctly centers the datapoints into
the
% center of a bin as opposed to the left edges
if strcmp(randomize, 'Yes') ==1
for i = 1:3
    for j = 1:3
        data_start(:,:,i) = sortrows(data_start(:,:,i),j);
        [peaks, index] = findpeaks(diff(data_start(:,j,i)));
        for k = 1:length(index)
            if k ==1
                data_start(1:index(k),j,i) = data_start(1:index(k),j,i)
+ peaks(k)*rand(index(k),1);
            else
                data_start(index(k-1)+1:index(k),j,i) =
data_start(index(k-1)+1:index(k),j,i) + peaks(k)*rand(index(k)-index(k-
1),1);
            end
        end
    end
end
else
end
%This ends the inter channel randomization

data_max = 1.1*max(max(max(max(data_start))));
M = zeros(3,3);
force = cell(6,1);
for i = 1:3
  for j = 1:3
      s = fitoptions('Method','NonlinearLeastSquares',...
                'Lower',0,...
                'Upper',slope_lim(j,i),...
                'Startpoint', 0,...
                'MaxIter', 1000,...
                'MaxFunEvals', 1000);
    f_Number = fittype('m*x',  'options' ,s);
    [fit_Number,gof_Number] = fit(data_start(find(data_start(:,i,i) <
fit_cutoff*max(data_original(:,i,i)),1,'first')+1
:find(data_start(:,i,i) == max(data_original(:,i,i)),1,'first')-
2,i,i),data_start(find(data_start(:,i,i) <
fit_cutoff*max(data_original(:,i,i)),1,'first')+1:find(data_start(:,i,i
) == max(data_original(:,i,i)),1,'first')-2,j,i), f_Number);
    M(j,i) = fit_Number.m;
  end
end

for j = 1:3
    for i = 1:3
        data_start(data_start(:,i,j) == max(data_start(:,i,j)),:,j) =
NaN(size(data_start(data_start(:,i,j) == max(data_start(:,i,j)),:,j)));
    end
end


for i = 1:3
figure('Name', ['Channel ',num2str(i),' Control'], 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
hold on;

  for j = 1:3

plot(data_original(:,i,i),data_original(:,j,i),'.','MarkerSize',MarkerS
ize,'color',bar_color{j+1});

plot(data_start(:,i,i),data_start(:,j,i),'.','MarkerSize',MarkerSize,'c
olor',bar_color{j});
    plot(1:data_max,M(j,i)*(1:data_max),'-
','MarkerSize',MarkerSize,'color',bar_color{j});
    force{3*j-2} = colheaders{j};
    force{3*j-1} = [colheaders{j},' Useable'];
    force{3*j} = [colheaders{j},' Fit'];
  end
    force_leg = legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    xlabel(colheaders(i),'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(['Channel ','"N"'],'FontSize', Label_FontSize, 'FontName',
FontName );
    title(['Spillover Determination ',colheaders{i}],'FontSize',
Title_FontSize, 'FontName', FontName );
    axis([0 1e4 0 1e4]);
    box(gca,'on');
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_width], 'PaperPosition', [0 0
fig_width fig_width]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/','Spillover_',colheaders{i},'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/','Spillover_',colheaders{i},'.fig'],'fig');
    close gcf;
end
M_inv = inv(M);

disp(ImageDir);
disp('Spillover Matrix');
disp(M);
disp('Inverse Spillover Matrix');
disp(M_inv);

dlmwrite([ImageDir,'Matlab/Compensated/M.txt'], M,'-append',
'delimiter', '\t','precision', 4);
dlmwrite([ImageDir,'Matlab/Compensated/Minv.txt'], M_inv,'-append',
'delimiter', '\t','precision', 4);


data_comp = zeros(size(data_start));
for i = 1:3
data_comp(:,:,i) = (M_inv*data_start(:,:,i)')';
end
data_min =
1.1*min(min(min(min(min(data_start)))),min(min(min(min(data_comp)))));
data_max =
1.1*min(max(max(max(max(data_start)))),max(max(max(max(data_comp)))));

leg_new = cell(3,1);
for l = 1:3
    for i = 1:3
        for j = i+1:3
            % Linear Scales
            figure('Name',  'Scatter Plot', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
            hold on;

plot(data_original(:,i,l),data_original(:,j,l),'.','MarkerSize',MarkerS
ize,'color',bar_color{j+1});

plot(data_start(:,i,l),data_start(:,j,l),'x','MarkerSize',MarkerSize,'c
olor',bar_color{i});

plot(data_comp(:,i,l),data_comp(:,j,l),'.','MarkerSize',MarkerSize,'col
or',bar_color{j});
            leg_new{1} =  colheaders{j};
            leg_new{2} = [colheaders{j},' Useable'];
            leg_new{3} = [colheaders{j},' Compensated'];
            legend(leg_new,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
            xlabel(colheaders{i},'FontSize', Label_FontSize,
'FontName', FontName );
            ylabel(colheaders{j},'FontSize', Label_FontSize,
'FontName', FontName );
            title('Linear','FontSize', Title_FontSize, 'FontName',
FontName );
            axis([0 1e4 0 1e4]);
            axis('square');
            box(gca,'on');
            set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
            set(gcf, 'PaperUnits', 'inches');
            set(gcf, 'Papersize', [fig_width fig_width],
'PaperPosition', [0 0 fig_width fig_width]);
            hold off;
            print( '-r300', '-loose',  '-djpeg' ,
[ImageDir,'Matlab/Figure
Jpegs/',eval(['Filename_',num2str(l),'{1}']),'_scatter_',num2str(i),'_'
,num2str(j),'.jpg']);
            saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',eval(['Filename_',num2str(l),'{1}']),'scatter_',num2str(i),'_',n
um2str(j),'.fig'],'fig');
            close gcf;

            % Log Scales
            figure('Name',  'Scatter Plot Log', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
            hold on;

plot(data_start(:,i,l),data_start(:,j,l),'.','MarkerSize',MarkerSize,'c
olor',bar_color{j+1});

plot(data_original(:,i,l),data_original(:,j,l),'x','MarkerSize',MarkerS
ize,'color',bar_color{i});

plot(data_comp(:,i,l),data_comp(:,j,l),'.','MarkerSize',MarkerSize,'col
or',bar_color{j});
            leg_new{1} = colheaders{j};
            leg_new{2} = [colheaders{j},' Useable'];
            leg_new{3} = [colheaders{j},' Compensated'];
            legend(leg_new,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
            xlabel(colheaders{i},'FontSize', Label_FontSize,
'FontName', FontName );
            ylabel(colheaders{j},'FontSize', Label_FontSize,
'FontName', FontName );
            title('Log','FontSize', Title_FontSize, 'FontName',
FontName );
            axis([1 1e4 1 1e4]);
            set(gca,'Xscale','log');
            set(gca,'Xtickmode','auto');
            set(gca,'Yscale','log');
            set(gca,'Ytickmode','auto');
            axis('square');
            box(gca,'on');
            set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
            set(gcf, 'PaperUnits', 'inches');
            set(gcf, 'Papersize', [fig_width fig_width],
'PaperPosition', [0 0 fig_width fig_width]);
            hold off;
            print( '-r300', '-loose',  '-djpeg' ,
[ImageDir,'Matlab/Figure
Jpegs/',eval(['Filename_',num2str(l),'{1}']),'_scatter_log_',num2str(i)
,'_',num2str(j),'.jpg']);
            saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',eval(['Filename_',num2str(l),'{1}']),'scatter_log_',num2str(i),'
_',num2str(j),'.fig'],'fig');
            close gcf;
        end
    end
end

leg_new = cell(2,1);
figure('Name',  '3D Plot axis', 'NumberTitle', 'off');
plot3(data_start(:,1,1) ,data_start(:,2,1)  ,data_start(:,3,1),'b.');
hold on;
plot3(data_comp(:,1,1)  ,data_comp(:,2,1)   ,data_comp(:,3,1),'r.');
plot3(data_start(:,1,2) ,data_start(:,2,2)  ,data_start(:,3,2),'b.');
plot3(data_comp(:,1,2)  ,data_comp(:,2,2)   ,data_comp(:,3,2),'r.');
plot3(data_start(:,1,3) ,data_start(:,2,3)  ,data_start(:,3,3),'b.');
plot3(data_comp(:,1,3)  ,data_comp(:,2,3)   ,data_comp(:,3,3),'r.');
axis([-1e3 1e4 -1e3 1e4 -1e3 1e4]);
grid(gca,'on');
box(gca,'on');
xlabel(colheaders{1},'FontSize', Label_FontSize, 'FontName', FontName
);
ylabel(colheaders{2},'FontSize', Label_FontSize, 'FontName', FontName
);
zlabel(colheaders{3},'FontSize', Label_FontSize, 'FontName', FontName
);
leg_new{1} = 'Uncompensated';
leg_new{2} = 'Compensated';
legend(leg_new,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
title('Compensated Axis','FontSize', Title_FontSize, 'FontName',
FontName );
set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
hold off;
print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/3d-axis.jpg']);
saveas(gcf,  [ImageDir,'Matlab/Figure Figs/3d-axis.fig'],'fig');
close gcf,

for i = 1:3
fid =
fopen([ImageDir,'Matlab/Compensated/',eval(['Filename_',num2str(i),'{1}
'])],'wt');
        for j = 1:length(colheaders)
            if j == length(colheaders)
                fprintf(fid,[eval('colheaders{j}'),'\n'],'char');
            else
                fprintf(fid,[eval('colheaders{j}'),'\t'],'char');
            end
        end
fclose(fid);
end

dlmwrite([ImageDir,'Matlab/Compensated/',Filename_1{1}],
data_comp(:,:,1), '-append', 'delimiter', '\t','precision', 4);
dlmwrite([ImageDir,'Matlab/Compensated/',Filename_2{1}],
data_comp(:,:,2), '-append', 'delimiter', '\t','precision', 4);
dlmwrite([ImageDir,'Matlab/Compensated/',Filename_3{1}],
data_comp(:,:,3), '-append', 'delimiter', '\t','precision', 4);

Filename_5 = [Filename_1,Filename_2,Filename_3,Filename_6,Filename_7];
for input= 1:length(Filename_5)
Filename_4 = Filename_5{input};
newData4 = importdata([ImageDir,Filename_4]);
data_start(:,:,4) = NaN(size(data_start(:,:,4)));
data_start(1:size(newData4.data,1),:,4) = newData4.data;

if input <= length([Filename_1,Filename_2,Filename_3,Filename_5])
data_start(data_start(:,1,4) == max(data_start(:,1,4)),1,4) = NaN;
data_start(data_start(:,2,4) == max(data_start(:,2,4)),2,4) = NaN;
data_start(data_start(:,3,4) == max(data_start(:,3,4)),3,4) = NaN;
end

% This handles the inter channel randomization.  It "fixes" the display
% artifacts caused by binning and correctly centers the datapoints into
the
% center of a bin as opposed to the left edges
if strcmp(randomize, 'Yes') ==1
    for j = 1:3
        data_start(:,:,4) = sortrows(data_start(:,:,4),j);
        [peaks, index] = findpeaks(diff(data_start(:,j,4)));
        for k = 1:length(index)
            if k ==1
                data_start(1:index(k),j,4) = data_start(1:index(k),j,4)
+ peaks(k)*rand(index(k),1);
            else
                data_start(index(k-1)+1:index(k),j,4) =
data_start(index(k-1)+1:index(k),j,4) + peaks(k)*rand(index(k)-index(k-
1),1);
            end
        end
    end
else
end
%This ends the inter channel randomization

data_comp(:,:,4) = (M_inv*data_start(:,:,4)')';



figure('Name',  Filename_4, 'NumberTitle', 'off');
plot3(data_start(:,1,4),data_start(:,2,4),data_start(:,3,4),'b.');
hold on;
plot3(data_comp(:,1,4),data_comp(:,2,4),data_comp(:,3,4),'r.');
axis([-1e3 1e4 -1e3 1e4 -1e3 1e4]);
grid(gca,'on');
box(gca,'on');
xlabel(colheaders{1},'FontSize', Label_FontSize, 'FontName', FontName
);
```

```
ylabel(colheaders{2},'FontSize', Label_FontSize, 'FontName', FontName
);
zlabel(colheaders{3},'FontSize', Label_FontSize, 'FontName', FontName
);
leg_new{1} = 'Uncompensated';
leg_new{2} = 'Compensated';
legend(leg_new,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
title(Filename_4,'FontSize', Title_FontSize, 'FontName', FontName );
set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
hold off;
print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/3d-scatter',Filename_4,'.jpg']);
saveas(gcf,  [ImageDir,'Matlab/Figure Figs/3d-
scatter',Filename_4,'.fig'],'fig');
close gcf;

fid =
fopen([ImageDir,'Matlab/Compensated/',eval(['Filename_',num2str(4)])],'
wt');
        for j = 1:length(eval('colheaders'))
            if j == length(eval('colheaders'))
                fprintf(fid,[eval('colheaders{j}'),'\n'],'char');
            else
                fprintf(fid,[eval('colheaders{j}'),'\t'],'char');
            end
        end
fclose(fid);
dlmwrite([ImageDir,'Matlab/Compensated/',Filename_4], data_comp(:,:,4),
'-append', 'delimiter', '\t','precision', 4);

end

ctrl_array  =
NaN(size(data_start,1),3*max(length(Filename_6),length(Filename_7)));
blank_array =
NaN(size(data_start,1),3*max(length(Filename_6),length(Filename_7)));
cfp_array   =
NaN(size(data_start,1),3*max(length(Filename_6),length(Filename_7)));

for i = 1:max(length(Filename_6),length(Filename_7))
    if i <= length(Filename_2)
        testdata =
importdata([ImageDir,'Matlab/Compensated/',Filename_2{i}]);
        ctrl_array(1:length(testdata.data),3*(i-1)+1:3*(i))      =
testdata.data;
    else
        testdata =
importdata([ImageDir,'Matlab/Compensated/',Filename_2{length(Filename_2
)}]);
        ctrl_array(1:length(testdata.data),3*(i-1)+1:3*(i))      =
testdata.data;
    end

    if i <= length(Filename_6)
```

```
        testdata =
importdata([ImageDir,'Matlab/Compensated/',Filename_6{i}]);
        blank_array(1:length(testdata.data),3*(i-1)+1:3*(i))     =
testdata.data;
    else
        testdata =
importdata([ImageDir,'Matlab/Compensated/',Filename_6{length(Filename_6
)}]);
        blank_array(1:length(testdata.data),3*(i-1)+1:3*(i))     =
testdata.data;
    end

    if i <= length(Filename_7)
        testdata =
importdata([ImageDir,'Matlab/Compensated/',Filename_7{i}]);
        cfp_array(1:length(testdata.data),3*(i-1)+1:3*(i))     =
testdata.data;
    else
        testdata =
importdata([ImageDir,'Matlab/Compensated/',Filename_7{length(Filename_7
)}]);
        cfp_array(1:length(testdata.data),3*(i-1)+1:3*(i))     =
testdata.data;
    end
end

fid = fopen([ImageDir,'Matlab/Compensated/blank.txt'],'wt');
        for j = 1:max(length(Filename_6),length(Filename_7))
                fprintf(fid,['Ctrl ',eval('colheaders{1}'),'\t','Ctrl
',eval('colheaders{2}'),'\t','Ctrl
',eval('colheaders{3}'),'\t'],'char');
        end

        for j = 1:max(length(Filename_6),length(Filename_7))
            if j == max(length(Filename_6),length(Filename_7))
                fprintf(fid,['Blank ',eval('colheaders{1}'),'\t','Blank
',eval('colheaders{2}'),'\t','Blank
',eval('colheaders{3}'),'\n'],'char');
            else
                fprintf(fid,['Blank ',eval('colheaders{1}'),'\t','Blank
',eval('colheaders{2}'),'\t','Blank
',eval('colheaders{3}'),'\t'],'char');
            end
        end

fclose(fid);
dlmwrite([ImageDir,'Matlab/Compensated/blank.txt'],[ctrl_array,blank_ar
ray], '-append', 'delimiter', '\t','precision', 4);


fid = fopen([ImageDir,'Matlab/Compensated/cfp.txt'],'wt');
        for j = 1:max(length(Filename_6),length(Filename_7))
                fprintf(fid,['Ctrl ',eval('colheaders{1}'),'\t','Ctrl
',eval('colheaders{2}'),'\t','Ctrl
',eval('colheaders{3}'),'\t'],'char');
```

```
        end

        for j = 1:max(length(Filename_6),length(Filename_7))
            if j == max(length(Filename_6),length(Filename_7))
                fprintf(fid,['CFP ',eval('colheaders{1}'),'\t','CFP
',eval('colheaders{2}'),'\t','CFP
',eval('colheaders{3}'),'\n'],'char');
            else
                fprintf(fid,['CFP ',eval('colheaders{1}'),'\t','CFP
',eval('colheaders{2}'),'\t','CFP
',eval('colheaders{3}'),'\t'],'char');
            end
        end

fclose(fid);
dlmwrite([ImageDir,'Matlab/Compensated/cfp.txt'],[ctrl_array,cfp_array]
, '-append', 'delimiter', '\t','precision', 4);
```

Users' Guide Gating.m

The Gating.m script takes the compensated output from the Compensation.m script and gates the data along the second channel based on an external control. This gate can either be defined directly by the control population or by a user set variable. The final output gives the mean and median intensities of both populations along the first axis, as well as averages and standard deviations across all replicates

On line 30, "num_of_replicates" sets the number of replicates found in each file and should be adjusted accordingly.

On line 31, "std_trans" sets the threshold to gate populations as a multiple above the mean of the population.

On line 32, "min_trans" sets an alternative minimum threshold (if there is linked auto-fluorescence), which overrides the population-based threshold if and only if it is the greater of the two.

On Line 33, “thesh_7aad” sets a threshold for the 3rd channel above which to exclude data points (Non viable cells in the case of this experiment)

The subsequent parameters are reasonably self explanatory, and set formatting parameter for the final output. These should be adjusted according to the experiment

Gating.m

```
%Clears All previous variables to clean memory
clear all;

%Image Filename and path

[Filename, ImageDir]    = uigetfile( ...
                                    {'*.txt','Data Files (*.txt)';...
                                    '*.txt','Tab Delimeted Text File
(*.bmp)'}, ...
                                    'Pick a file',...
                                    '/Users/berick/Desktop/',...
                                    'MultiSelect','on');



Filename_array = cell(length(Filename),1);
timepoints = cell(length(Filename),1);
for i = 1: length(Filename)
[Filename_array{i}, file_type] = strtok(Filename{i},'.');
[timepoints{i}, points] = strtok(Filename{i},' hours');
end
[timepoints index_sort] = sort(str2double(timepoints));
Filename_array = Filename_array(index_sort);



num_of_replicates   = 3;
% Number of replicates
std_trans        = 2;
min_trans        = 60;
thresh_7aad         = 100;
x_axis_label        = 'CFP Intensity';
y_axis_label        = 'PKH26 Intensity';
scatter_title       = 'PKH26 Intensity vs. CFP Intensity';


% For the colors:   [.332 .555 .832]    = joey's blue
%                   [.5 .5 .5]          = joey's grey
%                   [.570 .813 .313]    = joey's green
%                   [.570 .297 .297]    = becky's red
%                   [.414 .344 .5]    = becky's purple
%                   [.379 .539 .344]    = becky's dark green
```

```
bar_color           = {[.5 .5 .5] [.332 .555 .832] [.570 .813 .313]
[.570 .297 .297] [.414 .344 .5] [.379 .539 .344]};
% Histogram colors.  First is for forces, second is rates
disp_legend         = 'Yes';
% Turns the Legends on or off "Yes" = on, everything else = off

auto_axis           = 'on';
% 'on' Automatically sets axis heights anything else overides with user
specified  defaults
max_prob            = 20;
% User defined max rupture probablility
min_x               = 0;
% User defined max rupture force
max_x               = 3000;

max_iteration       = 5e4;
% Max variable iteration for fits
max_eval            = 5e4;
% Max fit evaluation



Axis_FontSize       = 20;
% Font Size for images
Label_FontSize      = 25;
Title_FontSize      = 28;
Legend_FontSize     = 14;
fig_width           = 10;
fig_height          = 7;
MarkerSize          = 20;
LineWidth           = 2;
FontName            = 'Arial';
% Font for images



% Make the Base Directories to save the images
mkdir([ImageDir,'Matlab/']);
mkdir([ImageDir,'Matlab/Figure Jpegs/']);
mkdir([ImageDir,'Matlab/Figure Figs/']);
mkdir([ImageDir,'Matlab/Text Files/']);





results = zeros(3,3*num_of_replicates + 6,length(Filename_array));
for num_files = 1:length(Filename_array)
Filename = Filename_array{num_files};
hist_title          = Filename;
cdf_title           = Filename;

newData1 = importdata([ImageDir,Filename,'.txt']);
for i = 1:num_of_replicates
newData1.data(newData1.data(:,3*i) > thresh_7aad,3*i-2:3*i) =
NaN(size(newData1.data(newData1.data(:,3*i) > thresh_7aad,3*i-2:3*i)));
end


% Create new variables in the base workspace from those fields.
vars = fieldnames(newData1);
for i = 1:length(vars)
    assignin('base', vars{i}, newData1.(vars{i}));
end


% Do the 2D-Scatter Plot Linear Scales
    figure('Name',  'Scatter Plot', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
    hold on;
    force = cell(length(newData1.colheaders)/3,1);
    for column=length(newData1.colheaders)/3:-1:1
    plot(data(:,3*column-2),data(:,3*column-
1),'.','MarkerSize',MarkerSize,'color',bar_color{column});
    end
    for column=1:length(newData1.colheaders)/3
    force(column) = newData1.colheaders(length(newData1.colheaders)-
3*column+2);
    end
    legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(scatter_title,'FontSize', Title_FontSize, 'FontName',
FontName );
    axis([-1e2 1e4 -1e2 1e4]);
    axis('square');
    set(gca,'Xtick',[0 1e3 2e3 3e3 4e3 5e3 6e3 7e3 8e3 9e3 1e4]);
    set(gca,'Ytick',[0 1e3 2e3 3e3 4e3 5e3 6e3 7e3 8e3 9e3 1e4]);
    box(gca,'on');
    grid(gca,'on');
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_width], 'PaperPosition', [0 0
fig_width fig_width]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename,'_scatter_',num2str(column),'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename,'scatter_',num2str(column),'.fig'],'fig');
    close gcf;

 figure('Name',  'Scatter Plot Log', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
    hold on;
    force = cell(length(newData1.colheaders)/3,1);
    for column=length(newData1.colheaders)/3:-1:1
    plot(data(:,3*column-2),data(:,3*column-
1),'.','MarkerSize',MarkerSize,'color',bar_color{column});
    end
    for column=1:length(newData1.colheaders)/3
    force(column) = newData1.colheaders(length(newData1.colheaders)-
3*column+2);
    end
    legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(scatter_title,'FontSize', Title_FontSize, 'FontName',
FontName );
    axis([1e-0 1e4 1e-0 1e4]);
    axis('square');
    grid(gca,'on');
    set(gca,'Xscale','log');
    set(gca,'Xtickmode','auto');
    set(gca,'Yscale','log');
    set(gca,'Ytickmode','auto');
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename,'_scatter_log_',num2str(column),'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename,'scatter_log_',num2str(column),'.fig'],'fig');
    close gcf;


ctrl_mean       = zeros(num_of_replicates,1);
ctrl_std        = zeros(num_of_replicates,1);
for replicates  = 1:num_of_replicates
control_mean    = mean(data(data(:,3*replicates-2)>0,3*replicates-2));
control_std     =  std(data(data(:,3*replicates-2)>0,3*replicates-2));
ctrl_mean(replicates)   = control_mean;
ctrl_std(replicates)    = control_std;
end
control_mean = mean(ctrl_mean);
control_std  = mean(ctrl_std);

for replicates = 1:num_of_replicates
data_ctrl  = zeros(1,1);
data_norm  = zeros(1,1);
data_trans  = zeros(1,1);
ctrl_cfp  = zeros(1,1);
norm_cfp  = zeros(1,1);
trans_cfp  = zeros(1,1);

cutoff_trans = max( [control_mean + std_trans*control_std,min_trans]);
for sample = 1:3
rupture_sorted = zeros(1,1);



if strcmp(auto_axis,'on')
```

```
min_x = max(round(floor(min(min(data)))),0);
max_x = ceil(max(max(data)));
tick_start  = max(round(floor(min(min(data)))),0);
tick_end    = ceil(max(max(data)));
else
tick_start  = min_x;
tick_end    = max_x;
end

if sample ==1

rupture_sorted  = data(data(:,3*replicates-1) > 0,3*replicates-1);
ctrl_cfp        = data(data(:,3*replicates-1) > 0,3*replicates-2);
bin_start       = max(round(floor(min(min(data)))),0);
bin_end         = round(ceil(max(max(data))));
column = 1;
newData1.colheaders{1} = ['Ctrl ',num2str(replicates)];
data_label      = newData1.colheaders{1};
data_ctrl = rupture_sorted;

elseif sample ==2
rupture_sorted  = data(data(:,3*num_of_replicates + 3*replicates-2) <
cutoff_trans,3*num_of_replicates + 3*replicates-1);
norm_cfp        = data(data(:,3*num_of_replicates + 3*replicates-2) <
cutoff_trans,3*num_of_replicates + 3*replicates-2);
bin_start       = max(round(floor(min(min(data)))),0);
bin_end         = round(ceil(max(max(data))));column = 2;
newData1.colheaders{2} = ['NonExpressing ',num2str(replicates)];
data_label      = newData1.colheaders{2};
data_norm = rupture_sorted;

elseif sample ==3
rupture_sorted  = data(data(:,3*num_of_replicates + 3*replicates-2) >
cutoff_trans,3*num_of_replicates + 3*replicates -1);
trans_cfp       = data(data(:,3*num_of_replicates + 3*replicates-2) >
cutoff_trans,3*num_of_replicates + 3*replicates -2);
bin_start       = max(round(floor(min(min(data)))),0);
bin_end         = round(ceil(max(max(data))));column = 3;
newData1.colheaders{3} = ['Expressing ',num2str(replicates)];
data_label      = newData1.colheaders{3};
data_trans = rupture_sorted;

end
newData1.colheaders{4} = ['All ',num2str(replicates)];
end

% Plot Seperate Scatter
    figure('Name',  'Scatter Plot Seperated', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
    hold on;
    force = cell(3,1);


plot(ctrl_cfp,data_ctrl,'.','MarkerSize',MarkerSize,'color',bar_color{1
});
```

```
plot(norm_cfp,data_norm,'.','MarkerSize',MarkerSize,'color',bar_color{2
});

plot(trans_cfp,data_trans,'.','MarkerSize',MarkerSize,'color',bar_color
{3});

    force{1} = ['Ctrl ',num2str(replicates)];
    force{2} = ['NonExpressing ',num2str(replicates)];
    force{3} = ['Expressing ',num2str(replicates)];

    if strcmp(disp_legend,'Yes')
    legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
    end
    xlabel(x_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    ylabel(y_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(scatter_title,'FontSize', Title_FontSize, 'FontName',
FontName );
    axis([-1e2 1e4 -1e2 1e4]);
    axis('square');
    box(gca,'on');
    grid(gca,'on');
    set(gca,'Xtick',[0 1e3 2e3 3e3 4e3 5e3 6e3 7e3 8e3 9e3 1e4]);
    set(gca,'Ytick',[0 1e3 2e3 3e3 4e3 5e3 6e3 7e3 8e3 9e3 1e4]);
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [fig_width fig_width], 'PaperPosition', [0 0
fig_width fig_width]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename,'_scatter_seperate_',num2str(column),'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename,'scatter_seperate_',num2str(column),'.fig'],'fig');

    axis([1e-0 1e4 1e-0 1e4]);
    axis('square');
    set(gca,'Xscale','log');
    set(gca,'Xtickmode','auto');
    set(gca,'Yscale','log');
    set(gca,'Ytickmode','auto');
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename,'_scatter_seperate_log_',num2str(column),'.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename,'scatter_seperate_log_',num2str(column),'.fig'],'fig');
    close gcf;


    %Plot All Boxplots
    data_points =  max([length(data_ctrl) length(data_norm)
length(data_trans) length([data_norm;data_trans])]);
    data_extracted = NaN(data_points,sum([size(data_ctrl,1) >
0,size(data_norm,1) > 0,size(data_trans,1) >0,1]));
```

```
    data_extracted(1:length(data_ctrl),1) = data_ctrl;
    data_extracted(1:length(data_norm),2) = data_norm;
    data_extracted(1:length(data_trans),3) = data_trans;
    data_extracted(1:length(data_trans)+length(data_norm),4) =
[data_norm;data_trans];
    figure('Name',  'BoxPlots ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
    hold on;
    h =
boxplot(data_extracted,newData1.colheaders(1:4),'boxstyle','outline','m
edianstyle','line','colors',cell2mat(transpose(bar_color)),'Whisker',In
f);
    hcc = get(get(gca,'Children'),'Children');
    hl = findobj(hcc,'type','text');
    set(h(1:length(h(:,1))-1,:),'LineWidth',LineWidth);
    set(hl,'FontSize', Axis_FontSize, 'FontName',
FontName,'VerticalAlignment','Top');
    ylabel(y_axis_label,'FontSize', Label_FontSize, 'FontName',
FontName );
    title(hist_title,'FontSize', Title_FontSize, 'FontName', FontName
);
    means = zeros(1,4);
    medians = zeros(1,4);
    fraction_data = zeros(1,4);
    for a= 1:4
        means(a)    =
mean(data_extracted(data_extracted(:,a)<Inf,a));
        medians(a)  =
median(data_extracted(data_extracted(:,a)<Inf,a));

    if a == 1
        fraction_data(a) = 100;
    elseif a == 4
        fraction_data(a) = 100;
    else
        fraction_data(a) =
100*length(data_extracted(data_extracted(:,a)<Inf,a))/(length(data_extr
acted(data_extracted(:,2)<Inf,2))+length(data_extracted(data_extracted(
:,3)<Inf,3)));
    end
    plot(a,means(a),'.','MarkerSize',MarkerSize,'Color',bar_color{a});
    end
    axis([0 5 -1e2 1e4]);
    set(gca,'Ytickmode','auto');
    set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
    set(gcf, 'PaperUnits', 'inches');
    set(gcf, 'Papersize', [2*fig_width fig_height], 'PaperPosition',
[.1*fig_width .1*fig_width 0.9*fig_width 0.9*fig_height]);
    hold off;
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename,'_rep_',num2str(replicates),'_box_plot.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename,'_box_plot','_rep_',num2str(replicates),'.fig'],'fig');
```

```
    axis([0 5 9e0 1.1e4]);
    set(gca,'Yscale','log');
    set(gca,'Ytickmode','auto');
    print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/',Filename,'_rep_log_',num2str(replicates),'_box_plot.jpg']);
    saveas(gcf,  [ImageDir,'Matlab/Figure
Figs/',Filename,'_box_plot','_rep_log_',num2str(replicates),'.fig'],'fi
g');
    close gcf;

%Calculate K-S Test results Kernel CDF
    results_kernel = cell(4,4);
    for a = 1:size(data_extracted,1)
        for b= a:size(data_extracted,2)
            [h,p] = kstest2(data_extracted(:,a),data_extracted(:,b));
            results_kernel{1,b+1} = newData1.colheaders{b};
            results_kernel{a+1,1} = newData1.colheaders{a};
            results_kernel{a+1,b+1} = ['P Value = ',
num2str(p,'%10.4f')];
        end
    end
    disp(['K-S Test Results Replicate # ',num2str(replicates)])
    disp(results_kernel);

    results_average = cell(4,4);
    for a = 1:3
        for b= a:4
            if b <= 3
            results_average{1,b+1} = newData1.colheaders{b};
            else
            results_average{1,b+1} = 'All';
            end
            results_average{2,1} = 'Mean';
            results_average{3,1} = 'Median';
            results_average{4,1} = '% of Total';
            results_average{2,b+1} = num2str(means(a),'%10.2f');
            results_average{3,b+1} = num2str(medians(a),'%10.2f');
            results_average{4,b+1} =
num2str(fraction_data(a),'%10.2f');
        end
    end
    disp(['Averages Replicate # ',num2str(replicates)])
    disp(results_average);

        Output_Name_results = [ImageDir,'Matlab/Text
Files/',Filename,'_rep_',num2str(replicates),'_k-s_test_results_.txt'];
        fid = fopen(Output_Name_results,'wt');
        for i = 1:length(results_kernel)
        for j = 1:length(results_kernel)
            if j == length(results_kernel)
            fprintf(fid,[results_kernel{i,j},'\n'],'char');
            else
            fprintf(fid,[results_kernel{i,j},'\t'],'char');
            end
        end
        end
        fclose(fid);

        Output_Name_results_average = [ImageDir,'Matlab/Text
Files/',Filename,'_rep_',num2str(replicates),'_averages.txt'];
        fid = fopen(Output_Name_results_average,'wt');
        for i = 1:length(results_average(:,1))
        for j = 1:length(results_average(1,:))
            if j == length(results_average)
            fprintf(fid,[results_average{i,j},'\n'],'char');
            else
            fprintf(fid,[results_average{i,j},'\t'],'char');
            end

        end
        end
        fclose(fid);


        results(1,0*num_of_replicates+replicates,num_files) = means(1);
        results(1,1*num_of_replicates+replicates,num_files) = means(2);
        results(1,2*num_of_replicates+replicates,num_files) = means(3);
        results(1,3*num_of_replicates+replicates,num_files) = means(4);

        results(2,0*num_of_replicates+replicates,num_files) =
medians(1);
        results(2,1*num_of_replicates+replicates,num_files) =
medians(2);
        results(2,2*num_of_replicates+replicates,num_files) =
medians(3);
        results(2,3*num_of_replicates+replicates,num_files) =
medians(4);

        results(3,0*num_of_replicates+replicates,num_files) =
fraction_data(1);
        results(3,1*num_of_replicates+replicates,num_files) =
fraction_data(2);
        results(3,2*num_of_replicates+replicates,num_files) =
fraction_data(3);
        results(3,3*num_of_replicates+replicates,num_files) =
fraction_data(4);

end


for j=1:3
results(j,4*num_of_replicates+1,num_files) =
mean(results(j,0*num_of_replicates+1:1*num_of_replicates,num_files));
results(j,4*num_of_replicates+2,num_files) =
std(results(j,0*num_of_replicates+1:1*num_of_replicates,num_files));
results(j,4*num_of_replicates+3,num_files) =
mean(results(j,1*num_of_replicates+1:2*num_of_replicates,num_files));
results(j,4*num_of_replicates+4,num_files) =
std(results(j,1*num_of_replicates+1:2*num_of_replicates,num_files));
```

```
results(j,4*num_of_replicates+5,num_files) =
mean(results(j,2*num_of_replicates+1:3*num_of_replicates,num_files));
results(j,4*num_of_replicates+6,num_files) =
std(results(j,2*num_of_replicates+1:3*num_of_replicates,num_files));
results(j,4*num_of_replicates+7,num_files) =
mean(results(j,3*num_of_replicates+1:4*num_of_replicates,num_files));
results(j,4*num_of_replicates+8,num_files) =
std(results(j,3*num_of_replicates+1:4*num_of_replicates,num_files));
end


results_output = cell(size(results,1)+1,size(results,2));
results_output{2,1} = 'Mean';
results_output{3,1} = 'Median';
results_output{4,1} = 'Expression Eff';
results_output{1,4*num_of_replicates+2} = 'Avg Ctrl';
results_output{1,4*num_of_replicates+3} = 'Std Ctrl';
results_output{1,4*num_of_replicates+4} = 'Avg NonExpressing';
results_output{1,4*num_of_replicates+5} = 'Std NonExpressing';
results_output{1,4*num_of_replicates+6} = 'Avg Expressing';
results_output{1,4*num_of_replicates+7} = 'Std Expressing';
results_output{1,4*num_of_replicates+8} = 'Avg All';
results_output{1,4*num_of_replicates+9} = 'Std All';

for i=1:num_of_replicates
 results_output{1,0*num_of_replicates+i+1} = ['Ctrl',num2str(i)];
 results_output{1,1*num_of_replicates+i+1} =
['NonExpressing',num2str(i)];
 results_output{1,2*num_of_replicates+i+1} = ['Expressing',num2str(i)];
 results_output{1,3*num_of_replicates+i+1} = ['All',num2str(i)];
end


for i = 1:size(results,1)
    for j = 1:size(results,2)
    results_output{i+1,j+1} = num2str(results(i,j,num_files),'%10.2f');
    end
end
disp(results_output);

Output_Name_results_average_compiled = [ImageDir,'Matlab/Text
Files/',Filename,'_all','_averages.txt'];
fid = fopen(Output_Name_results_average_compiled,'wt');
        for i = 1:size(results_output,1)
        for j = 1:size(results_output,2)
            if j == size(results_output,2)
            fprintf(fid,[results_output{i,j},'\n'],'char');
            else
            fprintf(fid,[results_output{i,j},'\t'],'char');
            end

        end
        end
        fclose(fid);
```

```
end


%Plot Final mean Data
figure('Name',  'Mean Data ', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
force = cell(8,1);
hold on;

errorbar(timepoints,results(1,13,:),results(1,14,:),'.-','color',
bar_color{1}, 'LineWidth', LineWidth, 'MarkerSize', MarkerSize);
errorbar(timepoints,results(2,13,:),results(2,14,:),'.--
','color',bar_color{1}, 'LineWidth', LineWidth, 'MarkerSize',
MarkerSize);
errorbar(timepoints,results(1,15,:),results(1,16,:),'.-','color',
bar_color{2}, 'LineWidth', LineWidth, 'MarkerSize', MarkerSize);
errorbar(timepoints,results(2,15,:),results(2,16,:),'.--
','color',bar_color{2}, 'LineWidth', LineWidth, 'MarkerSize',
MarkerSize);
errorbar(timepoints,results(1,17,:),results(1,18,:),'.-','color',
bar_color{3}, 'LineWidth', LineWidth, 'MarkerSize', MarkerSize);
errorbar(timepoints,results(2,17,:),results(2,18,:),'.--
','color',bar_color{3}, 'LineWidth', LineWidth, 'MarkerSize',
MarkerSize);
errorbar(timepoints,results(1,19,:),results(1,20,:),'.-','color',
bar_color{4}, 'LineWidth', LineWidth, 'MarkerSize', MarkerSize);
errorbar(timepoints,results(2,19,:),results(2,20,:),'.--
','color',bar_color{4}, 'LineWidth', LineWidth, 'MarkerSize',
MarkerSize);


force{1} = 'Ctrl Mean';
force{2} = 'Ctrl Median';
force{3} = 'NonExpressing Mean';
force{4} = 'NonExpressing Median';
force{5} = 'Expressing Mean';
force{6} = 'Expressing Median';
force{7} = 'All Mean';
force{8} = 'All Median';

legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
xlabel('Time (Hrs)','FontSize', Label_FontSize, 'FontName', FontName );
ylabel(y_axis_label,'FontSize', Label_FontSize, 'FontName', FontName );
title('PKH26 Intensity vs. Time','FontSize', Title_FontSize,
'FontName', FontName );
max_prob = 1.25*max(max(max(results(:,10:15,:))));
axis([floor(timepoints(1)-0.1*(timepoints(length(timepoints))-
timepoints(1)))-1
ceil(timepoints(length(timepoints))+0.1*(timepoints(length(timepoints))
-timepoints(1)))+1 0 max_prob]);
box(gca,'on');
set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
set(gcf, 'PaperUnits', 'inches');
set(gcf, 'Papersize', [fig_width 2*fig_height], 'PaperPosition',
[.1*fig_width .1*fig_height 0.9*fig_width 0.9*fig_height]);
hold off
print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/means.jpg']);
saveas(gcf,  [ImageDir,'Matlab/Figure Figs/means.fig'],'fig');

%Export saved values
means_export = zeros(length(timepoints),17);
for i = 1:length(timepoints)
   means_export(i,1)     = timepoints(i);
   means_export(i,2)     = results(1,13,i);
   means_export(i,3)     = results(1,14,i);
   means_export(i,4)     = results(1,15,i);
   means_export(i,5)     = results(1,16,i);
   means_export(i,6)     = results(1,17,i);
   means_export(i,7)     = results(1,18,i);
   means_export(i,8)     = results(1,19,i);
   means_export(i,9)     = results(1,20,i);
   means_export(i,10)    = results(2,13,i);
   means_export(i,11)    = results(2,14,i);
   means_export(i,12)    = results(2,15,i);
   means_export(i,13)    = results(2,16,i);
   means_export(i,14)    = results(2,17,i);
   means_export(i,15)    = results(2,18,i);
   means_export(i,16)    = results(2,19,i);
   means_export(i,17)    = results(2,20,i);
end
header = [  'Time','\t',...
            'Ctrl Mean','\t','Ctrl Std','\t',...
            'NonExpressing Mean','\t', 'NonExpressing Std','\t',...
            'Expressing Mean','\t', 'Expressing Std','\t',...
            'Ctrl Median','\t','Ctrl Std','\t',...
            'NonExpressing Median','\t', 'NonExpressing Std','\t',...
            'Expressing Median','\t', 'Expressing Std','\t',...
            'All Median','\t', 'All Std','\n'];
fid = fopen([ImageDir,'Matlab/Text Files/means.txt'],'wt');
fprintf(fid,header,'char');
fclose(fid);
dlmwrite([ImageDir,'Matlab/Text Files/means.txt'], means_export,'-
append', 'delimiter', '\t','precision', 4);


%Plot Final Trans effc
figure('Name',  'Expression Eff', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_height]);
force = cell(1,1);
hold on;
errorbar(timepoints,results(3,17,:),results(3,18,:),'.-','color',
bar_color{1}, 'LineWidth', LineWidth, 'MarkerSize', MarkerSize);
force{1} = 'Expression Effiency';
force_leg = legend(force,'Location','northeast','FontName',
FontName,'FontSize',Legend_FontSize);
xlabel('Time (Hrs)','FontSize', Label_FontSize, 'FontName', FontName );
ylabel('% Cells Expressing CFP','FontSize', Label_FontSize, 'FontName',
FontName );
```

```
title('Protein Expression vs. Time','FontSize', Title_FontSize,
'FontName', FontName );
max_prob = 1.25*max(max(max(results(3,17,:))));
axis([floor(timepoints(1)-0.1*(timepoints(length(timepoints))-
timepoints(1)))-1
ceil(timepoints(length(timepoints))+0.1*(timepoints(length(timepoints))
-timepoints(1)))+1 0 max_prob]);
box(gca,'on');
set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
set(gcf, 'PaperUnits', 'inches');
set(gcf, 'Papersize', [fig_width 2*fig_height], 'PaperPosition',
[.1*fig_width .1*fig_height 0.9*fig_width 0.9*fig_height]);
hold off
print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/express.jpg']);
saveas(gcf,  [ImageDir,'Matlab/Figure Figs/express.fig'],'fig');

% Save Values
means_export = zeros(length(timepoints),3);
for i = 1:length(timepoints)
   means_export(i,1)    = timepoints(i);
   means_export(i,2)    = results(3,17,i);
   means_export(i,3)    = results(3,18,i);
end
header = ['Time','\t','Expression','\t','Expression Std','\n'];
fid = fopen([ImageDir,'Matlab/Text Files/expression.txt'],'wt');
fprintf(fid,header,'char');
fclose(fid);
dlmwrite([ImageDir,'Matlab/Text Files/expression.txt'], means_export,'-
append', 'delimiter', '\t','precision', 4);
```

Users' Guide Half_Life.m

This final script fits a set of values to an exponential decay in order to extract the half-life of a dividing population. The script requires a tab delimited .txt file with columns and column headers for a minimum of three columns, where the first is always time, the second is an average value and the third is the standard deviation around that average. (Subsequent columns follow this paring system)

Each pair of columns is fit to the form Y = A*2^(x/x0) + B

On line 9, "A_down" sets the minimum value for A.

On line 10, "A_up" sets the maximum value for A.

On line 11, “x0_down” sets the minimum rate for x0.

On line 12, “x0_up” sets the maximum rate for x0.

On line 13, “B_max” sets the maximum value for the offset. The minimum is fixed to 0 since negative fluorescence is non-physical.

The remaining parameters are reasonably straightforward, set formatting parameters, and should be adjusted as needed.

Half_Life.m

```
clear all;
warning('off','Images:initSize:adjustingMag');
warning('off','MATLAB:MKDIR:DirectoryExists');
warning('off','MATLAB:MKDIR:OSError');
warning('off','MATLAB:NonIntegerInput');
warning('off','MATLAB:xlsread:Mode');
warning('off','MATLAB:xlswrite:NoCOMServer');


A_down  = 2000;
A_up    = 2000;
x0_down = 20;
x0_up   = 100;
B_max   = 00;

bar_color           = {[.5 .5 .5] [.332 .555 .832] [.570 .813 .313]
[.570 .297 .297] [.414 .344 .5] [.379 .539 .344]};
% Histogram colors.  First is for forces, second is rates
Axis_FontSize       = 20;
% Font Size for images
Label_FontSize      = 25;
Title_FontSize      = 28;
Legend_FontSize     = 14;
fig_width           = 10;
fig_height          = 7;
MarkerSize          = 20;
LineWidth           = 2;
FontName            = 'Arial';
% Font for images


[Filename, ImageDir]    = uigetfile({   '*.txt','Excel Files (*.txt)';
                                        '*.txt','Excel Data Files
(*.txt)'}, ...
                                        'Pick a file',...
                                        '/Users/berick/Desktop/',...
                                        'MultiSelect','off');


% Make the Base Directories to save the images
mkdir([ImageDir,'Matlab/']);
mkdir([ImageDir,'Matlab/Figure Jpegs/']);
mkdir([ImageDir,'Matlab/Figure Figs/']);
mkdir([ImageDir,'Matlab/Text Files/']);




newData1 = importdata([ImageDir,Filename]);
figure('Name',  'Time Decay', 'NumberTitle',
'off','Units','inches','Position', [0 0 fig_width fig_width]);
hold on;
fit_leg = cell(8,1);
 for i = 1:4

errorbar(newData1.data(:,1),newData1.data(:,2*i),newData1.data(:,2*i+1)
,'.','Color',bar_color{i},'MarkerSize',MarkerSize,'LineWidth',LineWidth
);

    if A_up == A_down && B_max == 0
    s = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',     x0_down,...
               'Upper',     x0_up,...
               'Startpoint',(x0_up + x0_down)/2,...
               'MaxIter',   1000,...
               'MaxFunEvals',1000);
    f_Number = fittype([num2str(A_up),'*2^(-x/x0)+',num2str(B_max)],
'options' ,s);
    [fit_Number,gof_Number] = fit(
newData1.data(isfinite(newData1.data(:,2*i)),1),...

newData1.data(isfinite(newData1.data(:,2*i)),2*i), f_Number);
    plot(0:60,A_up*(1/2).^((0:60)/fit_Number.x0),'-
','Color',bar_color{i},'MarkerSize',MarkerSize,'LineWidth',LineWidth)
    fit_leg{2*i-1}   = newData1.colheaders{2*i};
    fit_leg{2*i} = [num2str(A_up,'%10.2f'),'*2^{(-
x/',num2str(fit_Number.x0,'%10.2f'),')}'];

    elseif A_up == A_down && B_max > 0
    s = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',     [0,x0_down],...
               'Upper',     [B_max,x0_up],...
               'Startpoint',[0,(x0_up + x0_down)/2],...
               'MaxIter',   1000,...
               'MaxFunEvals',1000);
    f_Number = fittype([num2str(A_up),'*2^(-x/x0)+B'],  'options' ,s);
    [fit_Number,gof_Number] = fit(
newData1.data(isfinite(newData1.data(:,2*i)),1),...

newData1.data(isfinite(newData1.data(:,2*i)),2*i), f_Number);
    plot(0:60,A_up*(1/2).^((0:60)/fit_Number.x0)+fit_Number.B,'-
','Color',bar_color{i},'MarkerSize',MarkerSize,'LineWidth',LineWidth)
    fit_leg{2*i-1}   = newData1.colheaders{2*i};
    fit_leg{2*i} = [num2str(A_up,'%10.2f'),'*2^{(-
x/',num2str(fit_Number.x0,'%10.2f'),')}+',num2str(fit_Number.B,'%10.2f'
)];

    elseif A_up > A_down && B_max == 0
    s = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',     [A_down,x0_down],...
               'Upper',     [A_up,x0_up],...
               'Startpoint',[(A_up+A_down)/2,(x0_up + x0_down)/2],...
               'MaxIter',   1000,...
               'MaxFunEvals',1000);
    f_Number = fittype('A*2^(-x/x0)',  'options' ,s);
    [fit_Number,gof_Number] = fit(
newData1.data(isfinite(newData1.data(:,2*i)),1),...

newData1.data(isfinite(newData1.data(:,2*i)),2*i), f_Number);
    plot(0:60,fit_Number.A*(1/2).^((0:60)/fit_Number.x0),'-
','Color',bar_color{i},'MarkerSize',MarkerSize,'LineWidth',LineWidth)
    fit_leg{2*i-1}   = newData1.colheaders{2*i};
    fit_leg{2*i} = [num2str(fit_Number.A,'%10.2f'),'*2^{(-
x/',num2str(fit_Number.x0,'%10.2f'),')}'];

    else
    s = fitoptions('Method','NonlinearLeastSquares',...
               'Lower',     [A_down,0,x0_down],...
               'Upper',     [A_up,B_max,x0_up],...
               'Startpoint',[(A_up+A_down)/2,0,(x0_up + x0_down)/2],...
               'MaxIter',   1000,...
               'MaxFunEvals',1000);
    f_Number = fittype('A*2^(-x/x0)+B',  'options' ,s);
    [fit_Number,gof_Number] = fit(
newData1.data(isfinite(newData1.data(:,2*i)),1),...

newData1.data(isfinite(newData1.data(:,2*i)),2*i), f_Number);

plot(0:60,fit_Number.A*(1/2).^((0:60)/fit_Number.x0)+fit_Number.B,'-
','Color',bar_color{i},'MarkerSize',MarkerSize,'LineWidth',LineWidth);
    fit_leg{2*i-1}   = newData1.colheaders{2*i};
    fit_leg{2*i} = [num2str(fit_Number.A,'%10.2f'),'*2^{(-
x/',num2str(fit_Number.x0,'%10.2f'),')}+',num2str(fit_Number.B,'%10.2f'
)];
    end

end
legend(fit_leg);
xlabel('Time (Hrs)','FontSize', Label_FontSize, 'FontName', FontName );
ylabel('PKH26 Intensity','FontSize', Label_FontSize, 'FontName',
FontName );
title('Doubling Time','FontSize', Title_FontSize, 'FontName', FontName
);
axis([0 60 500 2000]);
box(gca,'on');
set(gca,'FontSize', Axis_FontSize, 'FontName', FontName);
set(gcf, 'PaperUnits', 'inches');
set(gcf, 'Papersize', [fig_width fig_width], 'PaperPosition', [0 0
fig_width fig_width]);
hold off;
print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/intensity fitting.jpg']);
saveas(gcf,  [ImageDir,'Matlab/Figure Figs/intensity
fitting.fig'],'fig');
```

```
set(gca,'Yscale','log');
set(gca,'Ytickmode','auto');
print( '-r300', '-loose',  '-djpeg' , [ImageDir,'Matlab/Figure
Jpegs/intensity fitting logy.jpg']);
saveas(gcf,  [ImageDir,'Matlab/Figure Figs/intensity fitting
logy.fig'],'fig');
```

Licensing Compensation.m, Gating.m, and Half_Life.m

Bibliography

1. Chang, X. Q.; Jorgensen, A. M. M.; Bardrum, P.; Led, J. J., Solution structures of the R-6 human insulin hexamer. *Biochemistry* **1997,** 36, (31), 9409-9422.

2. Labeit, S.; Barlow, D. P.; Gautel, M.; Gibson, T.; Holt, J.; Hsieh, C. L.; Francke, U.; Leonard, K.; Wardale, J.; Whiting, A.; Trinick, J., A Regular Pattern of 2 Types of 100-Residue Motif in the Sequence of Titin. *Nature* **1990,** 345, (6272), 273-276.

3. Kadler, K. E.; Holmes, D. F.; Trotter, J. A.; Chapman, J. A., Collagen fibril formation. *Biochemical Journal* **1996,** 316, 1-11.

4. Kadler, K. E.; Baldock, C.; Bella, J.; Boot-Handford, R. P., Collagens at a glance. *Journal of Cell Science* **2007,** 120, (12), 1955-1958.

5. Canty, E. G.; Kadler, K. E., Procollagen trafficking, processing and fibrillogenesis. *Journal of Cell Science* **2005,** 118, (7), 1341-1353.

6. Hall, C. E.; Doty, P., A Comparison between the Dimensions of Some Macromolecules Determined by Electron Microscopy and by Physical Chemical Methods. *Journal of the American Chemical Society* **1958,** 80, (6), 1269-1274.

7. Adams, M. C.; Salmon, W. C.; Gupton, S. L.; Cohan, C. S.; Wittmann, T.; Prigozhina, N.; Waterman-Storer, C. M., A high-speed multispectral spinning-disk confocal microscope system for fluorescent speckle microscopy of living cells. *Methods* **2003,** 29, (1), 29-41.

8. Cregger, M.; Berger, A. J.; Rimm, D. L., Immunohistochemistry and quantitative analysis of protein expression. *Journal Information* **2006,** 130, (7).

9. Cristino, L.; De Petrocellis, L.; Pryce, G.; Baker, D.; Guglielmotti, V.; Di Marzo, V., Immunohistochemical localization of cannabinoid type 1 and vanilloid transient receptor potential vanilloid type 1 receptors in the mouse brain. *Neuroscience* **2006,** 139, (4), 1405-1415.

10. Gratzner, H. G., Monoclonal antibody to 5-bromo-and 5-iododeoxyuridine: a new reagent for detection of DNA replication. *Science* **1982,** 218, (4571), 474.

11. Krishan, A., Rapid flow cytofluorometric analysis of mammalian cell cycle by propidium iodide staining. *Journal of Cell Biology* **1975,** 66, (1), 188.

12. Porter, K. G.; Feig, Y. S., The use of DAPI for identifying and counting aquatic microflora. *Limnology and Oceanography* **1980,** 25, (5), 943-948.

13. Walter, N. G., Structural dynamics of catalytic RNA highlighted by fluorescence resonance energy transfer. *Methods* **2001,** 25, (1), 19-30.

14. Bokinsky, G.; Rueda, D.; Misra, V. K.; Rhodes, M. M.; Gordus, A.; Babcock, H. P.; Walter, N. G.; Zhuang, X., Single-molecule transition-state analysis of RNA folding. *Proceedings of the National Academy of Sciences* **2003,** 100, (16), 9302.

15. Zhuang, X.; Kim, H.; Pereira, M. J. B.; Babcock, H. P.; Walter, N. G.; Chu, S., Correlating structural dynamics and function in single ribozyme molecules. *Science* **2002,** 296, (5572), 1473.

16. Rambourg, A.; Leblond, C. P., Electron microscope observations on the carbohydrate-rich cell coat present at the surface of cells in the rat. *Journal of Cell Biology* **1967,** 32, (1), 27.

17. Reynolds, E. S., The use of lead citrate at high pH as an electron-opaque stain in electron microscopy. *Journal of Cell Biology* **1963,** 17, (1), 208.

18. Venable, J. H.; Coggeshall, R., A simplified lead citrate stain for use in electron microscopy. *The Journal of Cell Biology* **1965**, 407-408.

19. Watson, M. L., Staining of tissue sections for electron microscopy with heavy metals: II. Application of solutions containing lead and barium. *Journal of Cell Biology* **1958,** 4, (6), 727.

20. Sperling, S., Early morphological changes in organ cultured human corneal endothelium. *Acta Ophthalmologica* **2009,** 56, (5), 785-792.

21. Taylor, D. L., Chloroplasts as symbiotic organelles in the digestive gland of Elysia viridis [Gastropoda: opisthobranchia]. *Journal of the Marine Biological Association of the UK* **2009,** 48, (01), 1-15.

22. Drake, B.; Prater, C. B.; Weisenhorn, A. L.; Gould, S. A.; Albrecht, T. R.; Quate, C. F.; Cannell, D. S.; Hansma, H. G.; Hansma, P. K., Imaging crystals, polymers, and processes in water with the atomic force microscope. *Science* **1989,** 243, (4898), 1586.

23. Binnig, G.; Quate, C. F.; Gerber, C., Atomic force microscope. *Physical review letters* **1986,** 56, (9), 930-933.

24. Alexander, S.; Hellemans, L.; Marti, O.; Schneir, J.; Elings, V.; Hansma, P. K.; Longmire, M.; Gurley, J., An atomic resolution atomic force microscope implemented using an optical lever. *Journal of Applied Physics* **1989,** 65, 164.

25. Hansma, H. G.; Vesenka, J.; Siegerist, C.; Kelderman, G.; Morrett, H.; Sinsheimer, R. L.; Elings, V.; Bustamante, C.; Hansma, P. K., Reproducible imaging and dissection of plasmid DNA under liquid with the atomic force microscope. *Science* **1992,** 256, 1180-1184.

26. Matzke, R.; Jacobson, K.; Radmacher, M., Direct, high-resolution measurement of furrow stiffening during division of adherent cells. *Nature cell biology* **2001,** 3, (6), 607-610.

27. Oberleithner, H.; Schneider, S.; Bustamante, J. O., Atomic force microscopy visualizes ATP-dependent dissociation of multimeric TATA-binding protein before translocation into the cell nucleus. *Pfl¸gers Archiv European Journal of Physiology* **1996,** 432, (5), 839-844.

28. Rotsch, C.; Jacobson, K.; Radmacher, M., Dimensional and mechanical dynamics of active and stable edges in motile fibroblasts investigated by using atomic force microscopy. *Proceedings of the National Academy of Sciences* **1999,** 96, (3), 921.

29. Cleveland, J. P.; Anczykowski, B.; Schmid, A. E.; Elings, V. B., Energy dissipation in tapping-mode atomic force microscopy. *Applied Physics Letters* **1998,** 72, 2613.

30. Bar, G.; Thomann, Y.; Brandsch, R.; Cantow, H. J.; Whangbo, M. H., Factors affecting the height and phase images in tapping mode atomic force microscopy. Study of phase-separated polymer blends of poly (ethene-co-styrene) and poly (2, 6-dimethyl-1, 4-phenylene oxide). *Langmuir* **1997,** 13, (14), 3807-3812.

31. Ishida, N.; Inoue, T.; Miyahara, M.; Higashitani, K., Nano bubbles on a hydrophobic surface in water observed by tapping-mode atomic force microscopy. *Langmuir* **2000,** 16, (16), 6377-6380.

32. Hong, S. P.; Bielinska, A. U.; Mecke, A.; Keszler, B.; Beals, J. L.; Shi, X. Y.; Balogh, L.; Orr, B. G.; Baker, J. R.; Holl, M. M. B., Interaction of poly(amidoamine) dendrimers with supported lipid bilayers and cells: Hole formation and the relation to transport. *Bioconjugate Chemistry* **2004,** 15, (4), 774-782.

33. Hong, S. P.; Leroueil, P. R.; Janus, E. K.; Peters, J. L.; Kober, M. M.; Islam, M. T.; Orr, B. G.; Baker, J. R.; Holl, M. M. B., Interaction of polycationic polymers

with supported lipid bilayers and cells: Nanoscale hole formation and enhanced membrane permeability. *Bioconjugate Chemistry* **2006,** 17, (3), 728-734.

34. Mecke, A.; Lee, D. K.; Ramamoorthy, A.; Orr, B. G.; Holl, M. M. B., Synthetic and natural polycationic polymer nanoparticles interact selectively with fluid-phase domains of DMPC lipid bilayers. *Langmuir* **2005,** 21, (19), 8588-8590.

35. Mecke, A.; Uppuluri, S.; Sassanella, T. M.; Lee, D. K.; Ramamoorthy, A.; Baker, J. R.; Orr, B. G.; Holl, M. M. B., Direct observation of lipid bilayer disruption by poly(amidoamine) dendrimers. *Chemistry and Physics of Lipids* **2004,** 132, (1), 3-14.

36. Hassenkam, T.; Fantner, G. E.; Cutroni, J. A.; Weaver, J. C.; Morse, D. E.; Hansma, P. K., High-resolution AFM imaging of intact and fractured trabecular bone. *Bone* **2004,** 35, (1), 4-10.

37. Thompson, J. B.; Kindt, J. H.; Drake, B.; Hansma, H. G.; Morse, D. E.; Hansma, P. K., Bone indentation recovery time correlates with bond reforming time. *Nature* **2001,** 414, (6865), 773-776.

38. Fantner, G. E.; Hassenkam, T.; Kindt, J. H.; Weaver, J. C.; Birkedal, H.; Pechenik, L.; Cutroni, J. A.; Cidade, G. A. G.; Stucky, G. D.; Morse, D. E., Sacrificial bonds and hidden length dissipate energy as mineralized fibrils separate during bone fracture. *Nature materials* **2005,** 4, (8), 612-616.

39. Gutsmann, T.; Fantner, G. E.; Venturoni, M.; Ekani-Nkodo, A.; Thompson, J. B.; Kindt, J. H.; Morse, D. E.; Fygenson, D. K.; Hansma, P. K., Evidence that collagen fibrils in tendons are inhomogeneously structured in a tubelike manner. *Biophysical Journal* **2003,** 84, (4), 2593-2598.

40. Habelitz, S.; Marshall, S. J.; Marshall, G. W., The functional width of the dentino-enamel junction determined by AFM-based nanoscratching. *Journal of Structural Biology* **2001,** 135, (3), 294-301.

41. Habelitz, S.; Marshall Jr, G. W.; Balooch, M.; Marshall, S. J., Nanoindentation and storage of teeth. *Journal of biomechanics* **2002,** 35, (7), 995-998.

42. Habelitz, S.; Balooch, M.; Marshall, S. J.; Balooch, G.; Marshall, G. W., In situ atomic force microscopy of partially demineralized human dentin collagen fibrils. *Journal of Structural Biology* **2002,** 138, (3), 227-236.

43. Habelitz, S.; Marshall, S. J.; MarshallJr, G. W.; Balooch, M., Mechanical properties of human dental enamel on the nanometre scale. *Archives of Oral Biology* **2001,** 46, (2), 173-183.

44. Fantner, G. E.; Barbero, R. J.; Gray, D. S.; Belcher, A. M., Kinetics of antimicrobial peptide activity measured on individual bacterial cells using high-speed atomic force microscopy. *Nature Nanotechnology*.

45. Oberdorster, G.; Oberdorster, E.; Oberdorster, J., Nanotoxicology: An emerging discipline evolving from studies of ultrafine particles. *Environmental Health Perspectives* **2005,** 113, (7), 823-839.

46. Mecke, A.; Majoros, I. J.; Patri, A. K.; Baker, J. R.; Holl, M. M. B.; Orr, B. G., Lipid bilayer disruption by polycationic polymers: The roles of size and chemical functional group. *Langmuir* **2005,** 21, (23), 10348-10354.

47. Veatch, S. L.; Cicuta, P.; Sengupta, P.; Honerkamp-Smith, A.; Holowka, D.; Baird, B., Critical Fluctuations in Plasma Membrane Vesicles. *ACS Chem. Biol.* **2008,** 3, (5), 287-293.

48. Veatch, S. L.; Keller, S. L., A closer look at the canonical "raft mixture" in model membrane studies. *Biophysical Journal* **2003,** 84, (1), 725-726.

49. Liu, J.; Qi, S.; Groves, J. T.; Chakraborty, A. K., Phase Segregation on Different Length Scales in a Model Cell Membrane System. *Journal of Physical Chemistry B* **2005,** 109, (42), 19960-19969.

50. Veatch, S. L.; Gawrisch, K.; Keller, S. L., Closed-loop miscibility gap and quantitative tie-lines in ternary membranes containing diphytanoyl PC. *Biophysical Journal* **2006,** 90, (12), 4428-4436.

51. Veatch, S. L.; Keller, S. L., Organization in Lipid Membranes Containing Cholesterol. *Physical Review Letters* **2002,** 89, (26), 268101/1-268101/4.

52. Veatch, S. L.; Keller, S. L., Separation of liquid phases in giant vesicles of ternary mixtures of phospholipids and cholesterol. *Biophysical Journal* **2003,** 85, (5), 3074-3083.

53. Veatch, S. L.; Keller, S. L., Seeing spots: Complex phase behavior in simple membranes. *Biochim. Biophys. Acta, Mol. Cell Res.* **2005,** 1746, (3), 172-185.

54. Veatch, S. L.; Keller, S. L., Miscibility Phase Diagrams of Giant Vesicles Containing Sphingomyelin. *Physical Review Letters* **2005,** 94, (14), 148101/1-148101/4.

55. Veatch, S. L.; Polozov, I. V.; Gawrisch, K.; Keller, S. L., Liquid domains in vesicles investigated by NMR and fluorescence microscopy. *Biophysical Journal* **2004,** 86, (5), 2910-2922.

56. Honerkamp-Smith, A. R.; Cicuta, P.; Collins, M. D.; Veatch, S. L.; den Nijs, M.; Schick, M.; Keller, S. L., Line tensions, correlation lengths, and critical exponents in lipid membranes near critical points. *Biophysical Journal* **2008,** 95, (1), 236-246.

57. Honerkamp-Smith, A. R.; Veatch, S. L.; Keller, S. L., An introduction to critical points for biophysicists; observations of compositional heterogeneity in lipid membranes. *Biochim. Biophys. Acta, Biomembr.* **2009,** 1788, (1), 53-63.

58. Keller, S. L., Miscibility Transitions and Lateral Compressibility in Liquid Phases of Lipid Monolayers. *Langmuir* **2003,** 19, (5), 1451-1456.

59. Keller, S. L.; Anderson, T. G.; McConnell, H. M., Miscibility critical pressures in monolayers of ternary lipid mixtures. *Biophysical Journal* **2000,** 79, (4), 2033-2042.

60. Keller, S. L.; McConnell, H. M., Stripe Phases in Lipid Monolayers near a Miscibility Critical Point. *Physical Review Letters* **1999,** 82, (7), 1602-1605.

61. Keller, S. L., Coexisting liquid phases in lipid monolayers and bilayers. *Journal of Physics: Condensed Matter* **2002,** 14, (19), 4763-4766.

62. Groves, J. T.; Boxer, S. G., Micropattern formation in supported lipid membranes. *Accounts of Chemical Research* **2002,** 35, (3), 149-157.

63. Sapuri, A. R.; Baksh, M. M.; Groves, J. T., Electrostatically targeted intermembrane lipid exchange with micropatterned supported membranes. *Langmuir* **2003,** 19, (5), 1606-1610.

64. Parthasarathy, R.; Groves, J. T., Protein patterns at lipid bilayer junctions. *Proc. Natl. Acad. Sci. U. S. A.* **2004,** 101, (35), 12798-12803.

65. Rozovsky, S.; Kaizuka, Y.; Groves, J. T., Formation and Spatio-Temporal Evolution of Periodic Structures in Lipid Bilayers. *Journal of the American Chemical Society* **2005,** 127, (1), 36-37.

66. Yamazaki, V.; Sirenko, O.; Schafer, R. J.; Groves, J. T., Lipid Mobility and Molecular Binding in Fluid Lipid Membranes. *Journal of the American Chemical Society* **2005,** 127, (9), 2826-2827.

67. Forstner, M. B.; Yee, C. K.; Parikh, A. N.; Groves, J. T., Lipid Lateral Mobility and Membrane Phase Structure Modulation by Protein Binding. *Journal of the American Chemical Society* **2006,** 128, (47), 15221-15227.

68. Groves, J. T., Unveiling the Membrane Domains. *Science (Washington, DC, U. S.)* **2006,** 313, (5795), 1901-1902.

69. Groves, J. T., Supported lipid bilayers as mimics for cell surfaces and as tools for biotechnology. *BioMEMS Biomed. Nanotechnol.* **2006,** 3, 305-323.

70. Liu, J.; Groves, J. T.; Chakraborty, A. K., Kinetic Pathways of Phase Ordering in Lipid Raft Model Systems. *Journal of Physical Chemistry B* **2006,** 110, (16), 8416-8421.

71. Sapuri-Butti, A. R.; Li, Q.; Groves, J. T.; Parikh, A. N., Nonequilibrium Patterns of Cholesterol-Rich Chemical Heterogeneities within Single Fluid Supported Phospholipid Bilayer Membranes. *Langmuir* **2006,** 22, (12), 5374-5384.

72. Parthasarathy, R.; Yu, C.-h.; Groves, J. T., Curvature-Modulated Phase Separation in Lipid Bilayer Membranes. *Langmuir* **2006,** 22, (11), 5095-5099.

73. Groves, J. T.; Parthasarathy, R.; Forstner, M. B., Fluorescence imaging of membrane dynamics. *Annu. Rev. Biomed. Eng.* **2008,** 10, 311-338.

74. Groves, J. T., Bending mechanics and molecular organization in biological membranes. *Annual Review of Physical Chemistry* **2007,** 58, 697-717.

75. Goksu, E. I.; Nellis, B. A.; Lin, W.-C.; Satcher, J. H.; Groves, J. T.; Risbud, S. H.; Longo, M. L., Effect of Support Corrugation on Silica Xerogel-Supported Phase-Separated Lipid Bilayers. *Langmuir* **2009,** 25, (6), 3713-3717.

76. Ding, J. Q.; Doudevski, I.; Warriner, H. E.; Alig, T.; Zasadzinski, J. A., Nanostructure changes in lung surfactant monolayers induced by interactions between palmitoyloleoylphosphatidylglycerol and surfactant protein B. *Langmuir* **2003,** 19, (5), 1539-1550.

77. Ding, J. Q.; Takamoto, D. Y.; von Nahmen, A.; Lipp, M. M.; Lee, K. Y. C.; Waring, A. J.; Zasadzinski, J. A., Effects of lung surfactant proteins, SP-B and SP-C, and palmitic acid on monolayer stability. *Biophysical Journal* **2001,** 80, (5), 2262-2272.

78. Spurlin, T. A.; Gewirth, A. A., Poly-L-lysine-induced morphology changes in mixed anionic/zwitterionic and neat zwitterionic-supported phospholipid bilayers. *Biophysical Journal* **2006,** 91, (8), 2919-2927.

79. Kelly, C. V.; Leroueil, P. R.; Orr, B. G.; Holl, M. M. B.; Andricioaei, I., Poly(amidoamine) dendrimers on lipid bilayers II: Effects of bilayer phase and dendrimer termination. *Journal of Physical Chemistry B* **2008,** 112, (31), 9346-9353.

80. Burr, D. B., Bone material properties and mineral matrix contributions to fracture risk or age in women and men. *J. Musculoskeletal Neuronal Interact.* **2002,** 2, (3), 201-204.

81. Gupta, H. S.; Wagermaier, W.; Zickler, G. A.; Raz-Ben Aroush, D.; Funari, S. S.; Roschger, P.; Wagner, H. D.; Fratzl, P., Nanoscale deformation mechanisms in bone. *Nano letters* **2005,** 5, (10), 2108.

82. Thompson, J. B.; Kindt, J. H.; Drake, B.; Hansma, H. G.; Morse, D. E.; Hansma, P. K., Bone indentation recovery time correlates with bond reforming time. *Nature* **2001,** 414, (6865), 773.

83. Les, C. M.; Spence, C. A.; Vance, J. L.; Christopherson, G. T.; Patel, B.; Turner, A. S.; Divine, G. W.; Fyhrie, D. P., Determinants of ovine compact bone viscoelastic properties: effects of architecture, mineralization, and remodeling. *Bone* **2004,** 35, (3), 729-738.

84. Amanata, N.; He Li, H.; Swain Michael, V.; Little David, G., The effect of zoledronic acid on the intrinsic material properties of healing bone: an indentation study. *Med Eng Phys* **2008,** 30, (7), 843-7.

85. Haiat, G.; Padilla, F.; Peyrin, F.; Laugier, P., Variation of ultrasonic parameters with microstructure and material properties of trabecular bone: a 3D model simulation. *J Bone Miner Res* **2007,** 22, (5), 665-74.

86. Weiner, S.; Traub, W., Bone structure: from angstroms to microns. *The FASEB journal : official publication of the Federation of American Societies for Experimental Biology; The FASEB journal : official publication of the Federation of American Societies for Experimental Biology* **1992,** 6, (3), 879.

87. Weiner, S.; Traub, W.; Wagner, H. D., Lamellar bone: structure-function relations. *Journal of structural biology; Journal of structural biology* **1999,** 126, (3), 241.

88. Weiner, S.; Wagner, H. D., The Material Bone: Structure-Mechanical Function Relations. *Annual Review of Materials Science* **1998,** 28, (1), 271.

89. Fratzl, P.; Gupta, H. S.; Paschalis, E. P.; Roschger, P., Structure and mechanical quality of the collagen,Àìmineral nano-composite in bone. *Journal of Materials Chemistry* **2004,** 14, (14), 2115.

90. Lane, N. E.; Yao, W.; Balooch, M.; Nalla, R. K.; Balooch, G.; Habelitz, S.; Kinney, J. H.; Bonewald, L. F., Glucocorticoid-treated mice have localized changes in trabecular bone material properties and osteocyte lacunar size that are

not observed in placebo-treated or estrogen-deficient mice. *J. Bone Miner. Res.* **2006,** 21, (3), 466-476.

91. Axelrod, D. W., Pathogenesis of the osteoporoses. *J Clin Rheumatol* **1997,** 3, (2 Suppl), 17-21.

92. Bonadio, J.; Holbrook, K. A.; Gelinas, R. E.; Jacob, J.; Byers, P. H., Altered triple helical structure of type I procollagen in lethal perinatal osteogenesis imperfecta. *Journal of Biological Chemistry* **1985,** 260, (3), 1734.

93. Byers, P. H.; Wallis, G. A.; Willing, M. C., Osteogenesis imperfecta: translation of mutation to phenotype. *Journal of medical genetics* **1991,** 28, (7), 433.

94. Cassella, J. P.; Ali, S. Y., Abnormal collagen and mineral formation in osteogenesis imperfecta. *Bone and mineral* **1992,** 17, (2), 123.

95. Camacho, N. P.; Hou, L.; Toledano, T. R.; Ilg, W. A.; Brayton, C. F.; Raggio, C. L.; Root, L.; Boskey, A. L., The material basis for reduced mechanical properties in oim mice bones. *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research* **1999,** 14, (2), 264.

96. Camacho, N. P.; Landis, W. J.; Boskey, A. L., Mineral changes in a mouse model of osteogenesis imperfecta detected by Fourier transform infrared microscopy. *Connective tissue research* **1996,** 35, (1-4), 259.

97. Glorieux, F. H.; Ward, L. M.; Rauch, F.; Lalic, L.; Roughley, P. J.; Travers, R., Osteogenesis imperfecta type VI: a form of brittle bone disease with a mineralization defect. *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research* **2002,** 17, (1), 30.

98. Blair-Levy, J. M.; Watts, C. E.; Fiorentino, N. M.; Dimitriadis, E. K.; Marini, J. C.; Lipsky, P. E., A type I collagen defect leads to rapidly progressive osteoarthritis in a mouse model. *Arthritis and Rheumatism* **2008,** 58, (4), 1096.

99. Chen, T.; Kozloff, K. M.; Goldstein, S. A.; Morris, M. D., Bone tissue ultrastructural defects in a mouse model for osteogenesis imperfecta: a Raman spectroscopy study. *Proceedings of SPIE* **2004,** 5321, 85.

100. Misof, K.; Landis, W. J.; Klaushofer, K.; Fratzl, P., Collagen from the osteogenesis imperfecta mouse model (oim) shows reduced resistance against tensile stress. *The Journal of clinical investigation* **1997,** 100, (1), 40.

101. Katz, E. P.; Wachtel, E.; Yamauchi, M.; Mechanic, G. L., The structure of mineralized collagen fibrils. *Connective tissue research* **1989,** 21, (1-4), 149.

102. Chernoff, E. A. G.; Chernoff, D. A., Atomic force microscope images of collagen fibers. *Journal of Vacuum Science & Technology A Vacuum Surfaces and Films* **1992,** 10, (4), 596.

103. Lees, S.; Hukins, D. W. L., X-ray diffraction by collagen in the fully mineralized cortical bone of cow tibia. *Bone and mineral* **1992,** 17, (1), 59.

104. Cell Atomic force microscopy study of the collagen fibre structure.

105. Peng, Z. Q.; Vaananen, H. K.; Tuukkanen, J., Ovariectomy-induced bone loss can be affected by different intensities of treadmill running exercise in rats. *Calcified tissue international* **1997,** 60, (5), 441.

106. Miller, S. C.; Bowman, B. M., Comparison of bone loss during normal lactation with estrogen deficiency osteopenia and immobilization osteopenia in the rat. *The Anatomical Record* **1998,** 251, (2), 265.

107. Wu, J.; Wang, X. X.; Takasaki, M.; Ohta, A.; Higuchi, M.; Ishimi, Y., Cooperative effects of exercise training and genistein administration on bone mass in ovariectomized mice. *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research* **2001,** 16, (10), 1829.

108. Nielsen, K. L.; Allen, M. R.; Bloomfield, S. A.; Andersen, T. L.; Chen, X. D.; Poulsen, H. S.; Young, M. F.; Heegaard, A. M., Biglycan deficiency interferes with ovariectomy-induced bone loss. *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research* **2003,** 18, (12), 2152.

109. Li, C. Y.; Schaffler, M. B.; Wolde-Semait, H. T.; Hernandez, C. J.; Jepsen, K. J., Genetic background influences cortical bone response to ovariectomy. *Journal of bone and mineral research : the official journal of the American Society for Bone and Mineral Research* **2005,** 20, (12), 2150.

110. Tetley, T. D., Health effects of nanomaterials. *Biochemical Society Transactions* **2007,** 35, 527-531.

111. Nemmar, A.; Hoet, P. H. M.; Thomeer, M.; Nemery, B.; Vanquickenborne, B.; Vanbilloen, H.; Mortelmans, L.; Hoylaerts, M. F.; Verbruggen, A.; Dinsdale, A., Passage of inhaled particles into the blood circulation in humans - Response. *Circulation* **2002,** 106, (20), E141-E142.

112. Nemmar, A.; Hoylaerts, M. F.; Hoet, P. H. M.; Nemery, B., Possible mechanisms of the cardiovascular effects of inhaled particles: systemic translocation and prothrombotic effects. *Toxicology Letters* **2004,** 149, (1-3), 243-253.

113. Halliday, H. L., Recent clinical trials of surfactant treatment for neonates. *Biology of the Neonate* **2006,** 89, (4), 323-329.

114. Escobedo, M. B.; Gunkel, J. H.; Kennedy, K. A.; Shattuck, K. E.; Sanchez, P. J.; Seidner, S.; Hensley, G.; Cochran, C. K.; Moya, F.; Morris, B.; Denson, S.; Stribley, R.; Naqvi, M.; Lasky, R. E.; Grp, T. N. R., Early surfactant for neonates with mild to moderate respiratory distress syndrome: A multicenter, randomized trial. *Journal of Pediatrics* **2004,** 144, (6), 804-808.

115. Ivanova, T.; Minkov, I.; Panaiotov, I.; Saulnier, P.; Proust, J. E., Dilational properties and morphology of surface films spread from clinically used lung surfactants. *Colloid and Polymer Science* **2004,** 282, (11), 1258-1267.

116. Zasadzinski, J. A.; Alonso, C.; Alig, T. J.; Warriner, H. E., Modifying the Shear Viscosity of Monolayers - what fatty acids, hexadecanol and surfactant proteins have in common. *Biophysical Journal* **2003,** 84, (2), 309A-309A.

117. Silvius, J. R., *Lipid-Protein Interactions*. John Wiley & Sons: New York, 1982.

118. Domenech, O.; Merino-Montero, S.; Montero, M. T.; Hernandez-Borrell, J., Surface planar bilayers of phospholipids used in protein membrane reconstitution: An atomic force microscopy study. *Colloids and Surfaces B-Biointerfaces* **2006,** 47, (1), 102-106.

119. Domenech, O.; Morros, A.; Cabanas, M. E.; Montero, M. T.; Hernandez-Borrell, J., Supported planar bilayers from hexagonal phases. *Biochimica Et Biophysica Acta-Biomembranes* **2007,** 1768, (1), 100-106.

120. Spurlin, T. A.; Gewirth, A. A., Effect of C-60 on solid supported lipid bilayers. *Nano Letters* **2007,** 7, (2), 531-535.

121. Zeng, F. W.; Zimmerman, S. C., Dendrimers in supramolecular chemistry: From molecular recognition to self-assembly. *Chemical Reviews* **1997,** 97, (5), 1681-1712.

122. Tomalia, D. A., Dendrimer Molecules. *Scientific American* **1995,** 272, (5), 62-66.

123. Betley, T. A.; Holl, M. M. B.; Orr, B. G.; Swanson, D. R.; Tomalia, D. A.; Baker, J. R., Tapping mode atomic force microscopy investigation of poly(amidoamine) dendrimers: Effects of substrate and pH on dendrimer deformation. *Langmuir* **2001,** 17, (9), 2768-2773.

124. Majoros, I. J.; Thomas, T. P.; Mehta, C. B.; Baker, J. R., Poly(amidoamine) dendrimer-based multifunctional engineered nanodevice for cancer therapy. *Journal of Medicinal Chemistry* **2005,** 48, (19), 5892-5899.

125. Majoros, I. J.; Myc, A.; Thomas, T.; Mehta, C. B.; Baker, J. R., PAMAM dendrimer-based multifunctional conjugate for cancer therapy: Synthesis, characterization, and functionality. *Biomacromolecules* **2006,** 7, (2), 572-579.

126. Pedersen, T. B.; Kaasgaard, T.; Jensen, M. O.; Frokjaer, S.; Mouritsen, O. G.; Jorgensen, K., Phase behavior and nanoscale structure of phospholipid membranes incorporated with acylated C-14-peptides. *Biophysical Journal* **2005,** 89, (4), 2494-2503.

127. Enders, O.; Ngezahayo, A.; Wiechmann, M.; Leisten, F.; Kolb, H. A., Structural calorimetry of main transition of supported DMPC billayers by temperature-controlled AFM. *Biophysical Journal* **2004,** 87, (4), 2522-2531.

128. Tokumasu, F.; Jin, A. J.; Dvorak, J. A., Lipid membrane phase behaviour elucidated in real time by controlled environment atomic force microscopy. *Journal of Electron Microscopy* **2002,** 51, (1), 1-9.

129. Mecke, A.; Lee, I.; Baker, J. R.; Holl, M. M. B.; Orr, B. G., Deformability of poly(amidoamine) dendrimers. *European Physical Journal E* **2004,** 14, (1), 7-16.

130. Nag, K., *Lung surfactant function and disorder*. Taylor & Francis: Boca Raton, 2005; p xix, 493 p.

131. Hodge, A. J.; Petruska, J. A., *Recent Studes with the Electron Microscope on Ordered Aggregates of the Tropocollagen Molecule*. Academic Press: New York, 1963.

132. Bigi, A.; Koch, M. H. J.; Panzavolta, S.; Roveri, N.; Rubini, K., Structural aspects of the calcification process of lower vertebrate collagen. *Connective Tissue Research* **2000,** 41, (1), 37-43.

133. Lin, A. C.; Goh, M. C., Investigating the ultrastructure of fibrous long spacing collagen by parallel atomic force and transmission electron microscopy. *Proteins-Structure Function and Genetics* **2002,** 49, (3), 378-384.

134. Chapman, J. A.; Tzaphlidou, M.; Meek, K. M.; Kadler, K. E., The Collagen Fibril - a Model System for Studying the Staining and Fixation of a Protein. *Electron Microscopy Reviews* **1990,** 3, (1), 143-182.

135. Wallace, J. M.; Chen, Q.; Fang, M.; Erickson, B.; Orr, B. G.; Banaszak Holl, M. M., Type I Collagen Exists as a Distribution of Nanoscale Morphologies in Teeth, Bones, and Tendons. *Langmuir* **2010,** In Press.

136. Wallace, J. M.; Erickson, B.; Les, C. M.; Orr, B. G.; Holl, M. M. B., Distribution of Type I Collagen Morphologies in Bone: Relation to Estrogen Depletion *Bone* **2009,** In Press.

137. Baranauskas, V.; Garavello-Freitas, I.; Jingguo, Z.; Cruz-Hofling, M. A., Observation of the bone matrix structure of intact and regenerative zones of tibias by atomic force microscopy. *Journal of Vacuum Science & Technology A* **2001,** 19, (4), 1042-1045.

138. Kindt, J. H.; Thurner, P. J.; Lauer, M. E.; Bosma, B. L.; Schitter, G.; Fantner, G. E.; Izumi, M.; Weaver, J. C.; Morse, D. E.; Hansma, P. K., In situ observation of fluoride-ion-induced hydroxyapatite-collagen detachment on bone fracture surfaces by atomic force microscopy. *Nanotechnology* **2007,** 18, (13), -.

139. Arfken, G. B.; Weber, H. J., *Mathematical Methods for Physicists*. Elsevier Academic Press: Burlington, MA, 2005.

140. Fratzl, P., *Collagen: Structure and Mechanics*. Springer: 2008.

141. Fratzl, P.; Fratzlzelman, N.; Klaushofer, K., Collagen Packing and Mineralization - an X-Ray-Scattering Investigation of Turkey Leg Tendon. *Biophysical Journal* **1993,** 64, (1), 260-266.

142. Drits, V. A.; Eberl, D. D.; Srodon, J., XRD measurement of mean thickness, thickness distribution and strain for illite and illite-smectite crystallites by the Bertaut-Warren-Averbach technique. *Clays and Clay Minerals* **1998,** 46, (1), 38-50.

143. Schaffer, T. E.; Cleveland, J. P.; Ohnesorge, F.; Walters, D. A.; Hansma, P. K., Studies of vibrating atomic force microscope cantilevers in liquid. *Journal of Applied Physics* **1996,** 80, (7), 3622-3627.

144. Schaeffer, T. E.; Viani, M.; Walters, D. A.; Drake, B.; Runge, E. K.; Cleveland, J. P.; Wendman, M. A.; Hansma, P. K. In *Atomic force microscope for small cantilevers*, 1997; 1997; p 48.

145. Hansma, H. G., Surface Biology of DNA by Atomic Force Microscopy. *Annual Review of Physical Chemistry* **2001,** 52, (1), 71-92.

146. Viani, M. B.; Sch‰ffer, T. E.; Paloczi, G. T.; Pietrasanta, L. I.; Smith, B. L.; Thompson, J. B.; Richter, M.; Rief, M.; Gaub, H. E.; Plaxco, K. W., Fast imaging

and fast force spectroscopy of single biopolymers with a new atomic force microscope designed for small cantilevers. *Review of Scientific Instruments* **1999,** 70, 4300.

147. Viani, M. B.; Sch‰ffer, T. E.; Chand, A.; Rief, M.; Gaub, H. E.; Hansma, P. K., Small cantilevers for force spectroscopy of single molecules. *Journal of Applied Physics* **1999,** 86, 2258.

148. Viani, M. B.; Pietrasanta, L. I.; Thompson, J. B.; Chand, A.; Gebeshuber, I. C.; Kindt, J. H.; Richter, M.; Hansma, H. G.; Hansma, P. K., Probing proteinñprotein interactions in real time. *Nature Structural & Molecular Biology* **2000,** 7, (8), 644-647.

149. Kodera, N.; Yamashita, H.; Ando, T., Active damping of the scanner for high-speed atomic force microscopy. *Review of Scientific Instruments* **2005,** 76, 053708.

150. Ando, T.; Kodera, N.; Takai, E.; Maruyama, D.; Saito, K.; Toda, A., A high-speed atomic force microscope for studying biological macromolecules. *Proceedings of the National Academy of Sciences* **2001,** 98, (22), 12468.

151. Fantner, G. E.; Schitter, G.; Kindt, J. H.; Ivanov, T.; Ivanova, K.; Patel, R.; Holten-Andersen, N.; Adams, J.; Thurner, P. J.; Rangelow, I. W., Components for high speed atomic force microscopy. *Ultramicroscopy* **2006,** 106, (8-9), 881-887.

152. Kindt, J. H.; Fantner, G. E.; Cutroni, J. A.; Hansma, P. K., Rigid design of fast scanning probe microscopes using finite element analysis. *Ultramicroscopy* **2004,** 100, (3-4), 259-265.

153. Picco, L. M.; Bozec, L.; Ulcinas, A.; Engledew, D. J.; Antognozzi, M.; Horton, M. A.; Miles, M. J., Breaking the speed limit with atomic force microscopy. *Nanotechnology* **2007,** 18, (4), 44030.

154. Sulchek, T.; Hsieh, R.; Adams, J. D.; Yaralioglu, G. G.; Minne, S. C.; Quate, C. F.; Cleveland, J. P.; Atalar, A.; Adderton, D. M., High-speed tapping mode imaging with active Q control for atomic force microscopy. *Applied Physics Letters* **2000,** 76, 1473.

155. Yamashita, H.; Kodera, N.; Miyagi, A.; Uchihashi, T.; Yamamoto, D.; Ando, T., Tip-sample distance control using photothermal actuation of a small cantilever for high-speed atomic force microscopy. *Review of Scientific Instruments* **2007,** 78, 083702.

156. Schitter, G.; Stemmer, A., Model-based signal conditioning for high-speed atomic force and friction force microscopy. *Microelectronic Engineering* **2003,** 67, 938-944.

157. Fantner, G. E.; Hegarty, P.; Kindt, J. H.; Schitter, G.; Cidade, G. A. G.; Hansma, P. K., Data acquisition system for high speed atomic force microscopy. *Review of Scientific Instruments* **2005,** 76, 026118.

158. Schitter, G.; Allgower, F.; Stemmer, A., A new control strategy for high-speed atomic force microscopy. *Nanotechnology* **2004,** 15, (1), 108-114.

159. Hansma, P. K.; Schitter, G.; Fantner, G. E.; Prater, C., High-speed atomic force microscopy. *SCIENCE-NEW YORK THEN WASHINGTON-* **2006,** 314, (5799), 601.

160. Ando, T.; Uchihashi, T.; Kodera, N.; Yamamoto, D.; Miyagi, A.; Taniguchi, M.; Yamashita, H., High-speed AFM and nano-visualization of biomolecular processes. *Pfl¸gers Archiv European Journal of Physiology* **2008,** 456, (1), 211-225.

161. Yokokawa, M.; Wada, C.; Ando, T.; Sakai, N.; Yagi, A.; Yoshimura, S. H.; Takeyasu, K., Fast-scanning atomic force microscopy reveals the ATP/ADP-dependent conformational changes of GroEL. *The EMBO journal* **2006,** 25, (19), 4567.

162. Neuman, K. C.; Nagy, A., Single-molecule force spectroscopy: optical tweezers, magnetic tweezers and atomic force microscopy. *Nature Methods* **2008,** 5, (6), 491-505.

163. Pope, L. H.; Davies, M. C.; Laughton, C. A.; Roberts, C. J.; Tendler, S. J. B.; Williams, P. M., Force-induced melting of a short DNA double helix. *European Biophysics Journal with Biophysics Letters* **2001,** 30, (1), 53-62.

164. Strunz, T.; Oroszlan, K.; Schafer, R.; Guntherodt, H. J., Dynamic force spectroscopy of single DNA molecules. *Proceedings of the National Academy of Sciences of the United States of America* **1999,** 96, (20), 11277-11282.

165. Wal, M. V.; Kamper, S.; Headley, J.; Sinniah, K., Effects of contact force and salt concentration on the unbinding of a DNA duplex by force spectroscopy. *Langmuir* **2006,** 22, (3), 882-886.

166. Dupres, V.; Menozzi, F. D.; Locht, C.; Clare, B. H.; Abbott, N. L.; Cuenot, S.; Bompard, C.; Raze, D.; Dufrene, Y. F., Nanoscale mapping and functional analysis of individual adhesins on living bacteria. *Nature Methods* **2005,** 2, (7), 515-520.

167. Kamper, S. G.; Porter-Peden, L.; Blankespoor, R.; Sinniah, K.; Zhou, D. J.; Abell, C.; Rayment, T., Investigating the specific interactions between carbonic anhydrase and a sulfonamide inhibitor by single-molecule force spectroscopy. *Langmuir* **2007,** 23, (25), 12561-12565.

168. Muller, D. J.; Dufrene, Y. F., Atomic force microscopy as a multifunctional molecular toolbox in nanobiotechnology. *Nature Nanotechnology* **2008,** 3, (5), 261-269.

169. Porter-Peden, L.; Kamper, S. G.; Wal, M. V.; Blankespoor, R.; Sinniah, K., Estimating Kinetic and Thermodynamic Parameters from Single Molecule Enzyme-Inhibitor Interactions. *Langmuir* **2008,** 24, (20), 11556-11561.

170. Lee, C. K.; Wang, Y. M.; Huang, L. S.; Lin, S. M., Atomic force microscopy: Determination of unbinding force, off rate and energy barrier for protein-ligand interaction. *Micron* **2007,** 38, (5), 446-461.

171. Ohler, B., In Veeco Appication Note: 2007; Vol. AN94.

172. Ohler, Practical Advice on the Determination of Cantilever Spring Constants. *Veeco Application Note* **2007,** AN94.

173. Schaffer, T. E., Calculation of thermal noise in an atomic force microscope with a finite optical spot size. *Nanotechnology* **2005,** 16, (6), 664-670.

174. Schaffer, T. E.; Hansma, P. K., Characterization and optimization of the detection sensitivity of an atomic force microscope for small cantilevers. *Journal of Applied Physics* **1998,** 84, (9), 4661-4666.

175. Schaffer, T. E., Force spectroscopy with a large dynamic range using small cantilevers and an array detector. *Journal of Applied Physics* **2002,** 91, (7), 4739-4746.

176. Naeem, S.; Liu, Y.; Nie, H. Y.; Lau, W. M.; Yang, J., Revisiting atomic force microscopy force spectroscopy sensitivity for single molecule studies. *Journal of Applied Physics* **2008,** 104, (11), -.