12-2022

Investigating Spatial Heterogeneity In Myocardial Wound Environments To Improve Therapy

Michael Potter
mjpotte@clemson.edu

INVESTIGATING SPATIAL HETEROGENEITY IN MYOCADARDIAL WOUND ENVIRONMENTS TO IMPROVE THERAPY

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Bioengineering

---

by
Michael Jacob Potter
December 2022

Accepted by:
Dr. David Karig, Committee Chair
Dr. William Richardson
Dr. Jordon Gilmore
Dr. Ken Webb

ABSTRACT

Heart failure is a broad pathology manifestation categorized by an inability of the heart to successfully pump blood throughout the vast vessel network of the body. Within the United States, heart failure is projected to increase by approximately 46% from 2012 to 2030. Modalities of heart failure are generally related to wall mechanics that are impacted following myocardial infarction events. Interplay exists between the wall mechanics, responding cell populations, and the spatial heterogeneities in the resultant scar. This interplay directs the myocardium towards heart failure modalities governed by overly stiff or compliant states. It is essential to elucidate details underlying the progression, maturation, and remodeling of the collagenous infarct scar to uncover therapeutic avenues and improve outcomes.

We developed in vitro platforms that enable these desired insights into infarct progression. Our first platform generates continuous collagen gels possessing spatial fiber alignment heterogeneity utilizing a cost-efficient magnetic microsphere methodology. Spatial heterogeneities in the fiber architecture and cell response were evaluated. The second platform mechanically and electrically stimulates engineered heart tissue constructs and enables the evaluation of spatial heterogeneities in construct mechanics following a simulated infarction. Additionally, to avoid losing sight of the in vivo reality, we evaluated the interplay of the cellular detection of mechanics and spatial strain distributions throughout an infarcted myocardial wall. Histologically evaluated mechanosensor intensities were correlated to in vivo strain data collected via ultrasound imaging of the myocardium of a murine infarction animal model.

DEDICATION

This work is dedicated to my mother, father, brother, grandfather, and uncles. I am thankful for the lessons provided along the way, both direct and indirect. Without your examples, guidance, and sacrifice I would not be in the position I am today. I thank you all for showing just how important family can be.

ACKNOWLEDGMENTS

I would like to acknowledge the guidance and understanding provided by my PI and mentor, Dr. Will Richardson, a man of great intellect and insight. I must also acknowledge the support and generosity provided by the Clemson Department of Bioengineering and the pleasure it has been working within the Rhodes Complex. Additionally, the talented members of the Clemson Light Imaging Facility, Dr. Terri Bruce and Rhonda Powell, have been instrumental in the success of more than one of my experimental explorations and deserve thanks. My fellow lab members who have shared this journey understand the ups and downs more than most. I would like to thank them all for both their camaraderie and collaboration. Finally, I would like to thank my committee members for their open and responsive communication whenever the need arose. They have contributed greatly to the positive environment and research experience.

TABLE OF CONTENTS

Page

LIST OF FIGURES

List of Figures (Continued)

Figure Page

CHAPTER ONE

Introduction

1.1. Study Significance

The tissues of the human body vary greatly in their form and function. Not only is there tissue-to-tissue variation, but there also exists much intra-tissue variation that either occurs naturally or arises as a result of disease states. One brand of tissue variation is that of spatial heterogeneity. As the name suggests, analyzing spatial heterogeneity involves evaluating the differential occurrence of phenomena across space within the subject under question. Such phenomena can include protein expression, chemokine gradients, cellular density, cellular morphology, structural architecture, drug diffusion, and tissue mechanics just to name a few. These types of fundamental analyses have been utilized to elucidate better understandings of biological phenomena such as prostate biomarker expression, [1] liver gene activation, [2] molecule uptake in tumor aggression, [3] drug diffusion in cancer treatment, [4] evaluation of tissue engineered constructs, [5] and even the assessment of cervix remodeling during pregnancy. [6] One area where spatial heterogeneity in all its flavors is of interest is that of heart failure.

Ischemic heart disease, which is often a precursor, immediate cause, or contributing factor in heart failure, increased in global prevalence by nearly a quarter from 2007 to 2017. This translated to approximately 126.5 million affected persons that year. [7] It is widely understood that heart failure is an ever-present plague within the United States population. It remains the leading cause of death within the U.S., outpacing

cancer, respiratory disease, stroke, and Alzheimer disease to name a few.[8] An estimated 6.2 million Americans over the age of 20 experienced heart failure from 2013 to 2016. The average annual cost of heart disease from 2014 to 2015 was estimated to be $218.7 billion within the United States. This cost will only increase as heart failure prevalence is projected to rise by 46% from 2012 to 2030, bringing the percentage of the total US population experiencing heart failure to 2.97%.[7] In 2017, heart failure was attributed to 80,480 deaths, a 42.3% increase from a decade prior in 2007.[7] Despite the growth in the number of deaths, survival after the onset of heart failure in older populations has increased only slightly. Post-hospitalization 30-day, 1-year, and 5-year fatality rates for these older populations sits at approximately 10.4%, 22%, and 42.3% respectively.[7] These improvements can be attributed to general advances in heart failure treatment; however, heart failure is a broad classification of many pathology manifestations related to an impairment of myocardial efficiency. Understanding which mechanism of development is at play would improve treatment specificity and open the possibility of arresting further degradation of heart function after an initial event. An obvious component implicated in many failure modalities is the post-myocardial infarction resultant scar. The prevalence of myocardial infarction in the US from 2013 to 2016 was estimated to be approximately 8.4 million.[7] Developing an understanding of the intricacies of scar tissue formation, maturation, and remodeling is essential to improving treatment efficacy and long-term survival rates for large numbers of patients who have experienced or will experience an initial ischemic event.

Heart failure modalities related to the post-infarct scar generally relate to preload(wall passivity and stretch) and contractility. Preload-related heart failure usually leads to diastolic dysfunction and heart failure with preserved ejection fraction(HFpEF).[9] Stiffening of the myocardial wall impairs ventricular filling thus reducing volume. This leads to heightened ventricular pressures and an upward shift in the lower portion of the PV loop. When progression to heart failure results from contractility issues, it is typically heart failure with reduced ejection fraction(HFrEF).[10] In contrast to HFpEF, HFrEF results in abnormal pressure that is increased during diastolic function and decreased during systolic function. In this scenario, volume is increased throughout the cycle. Progression towards these categories of functionality depends on scar formation and collagen structure as these are determinants of the mechanical properties of the infarct and surrounding non-infarcted regions. Collagen is heavily impacted during the initial inflammatory phase following infarction. This can lead to dilation or even wall rupture. Once past this phase, collagen deposition and cross-linking stiffen the infarct scar and remote tissue, affecting the mechanics of the ventricular wall at large. Depending on how collagen content is handled throughout the timecourse of the event, from initial degradation to ultimate maturation and remodeling of the scar and surrounding myocardium, cardiac functionality can veer towards a heart failure modality ruled by wall overcompliance or lack thereof.[11] **It is essential to elucidate details of post-myocardial infarction scar progression, maturation, and remodeling to uncover potential therapeutic avenues enabling not only improved outcomes, but eventually full recovery.**

1.2. Specific Aims

About six million Americans 20 years of age or older had heart failure from 2015 to 2018 with prevalence projected to increase to greater than eight million Americans by 2030.[12] Many heart failure pathologies can be related back to the scar tissue formed in response to left ventricular myocardial infarction. Mechanical insufficiencies(reduced contractility, impaired relaxation, and susceptibility to dilation and rupture) can be attributed to various infarct scar properties such as geometry and location among others.[11] However, to elucidate the fundamental determinants of scar integrity and the associated mechanical implications, detailed analysis of the processes that give rise to the structure and the scar structure itself must be performed. Richardson et al found heterogeneity within rat infarct scar collagen composition and architecture and utilized agent-based computational modeling to predict the role of locally reinforcing cell-matrix interactions in giving rise to this observed heterogeneity.[13] This is an important insight as collagen is a ubiquitous structural protein possessing an inherent structure-function relationship that is propagated up to the macro-scale tissue mechanics of the tissues the collagen composes. Mature scar structure is directly related to macroscopic mechanics and cardiac function or failure. Therefore, understanding the emergent phenomena of spatial heterogeneity within infarct scar progression leading to the mature scar and the remodeling that occurs in the following weeks is essential to informing intervention to improve outcomes.

Investigations into spatial heterogeneity emergence will require controlled experimental platforms that recapitulate the environment under question. Currently there

exist methods to generate aligned collagenous matrices; however, not all methods can produce a continuous heterogeneous environment. The methods that can produce such an environment produce them on a smaller scale not conducive to evaluating long-range spatial heterogeneities, or the methods themselves are overly complex. Notable methods that produce constructs with some level of alignment include electrospinning, microfluidic fabrication, drop casting, and bioprinting. While each possess their uses, each have their shortfalls as well. Microfluidic fabrication typically lacks architecture heterogeneity. [14,15] Electrospinning is limited by the number of compatible solutions due to necessitated polymer inclusions.[16] Drop casting is limited by the nature of generating alignment radially.[17] Bioprinting holds the most control over architecture generation within larger and more complicated constructs; however, bioprinting methodologies require specialized fabrication tools and intricate levels of control.[18] There exists a need for a simple and cost-efficient platform capable of generating spatial heterogeneity in collagen fiber alignment to enable insights into infarct scar heterogeneity progression.

Looking outside of the underlying substrate architecture, recapitulating relevant physiological conditions also involves generating platforms with appropriate stimulation capabilities. There exists great interest in organ-on-a-chip in vitro platforms geared toward improving physiological relevance of in vitro studies. The same interest exists towards developing improved myocardial platforms simulating in vivo-like conditions. The complexity of these platforms varies; however, a uniting factor appears to be the analysis of contractile forces as a measure of embedded cardiomyocyte health. This is especially utilized in the evaluation of platforms that focus on the maturation of human

induced pluripotent stem cells into cardiomyocyte-like lineages. The methodologies of force measurement vary depending on the platform. There exist studies where single-cell contractile force is measured via deflection of a polymeric microarray composing the attachment substrate. [19] A slight variant of this methodology utilizes the deformation of embedded fluorescent beads to measure forces generated at the construct level.[20] Other platforms evaluate construct level force output via deflection of polymeric mounts. [21–23] These platforms typically apply an electrical pacing regimen to stimulate the tissues and generate a more mature cardiomyocyte phenotype from the embedded cellular populations. While these studies produce great insights into cardiomyocyte maturation and function, there exists a need for platforms enabling the study of regional differences in post-infarction injury response and mechanics progression. **The overarching theme of this dissertation is to develop and validate experimental platforms and measurement methods that enable the relevant study of the spatial heterogeneity found within myocardial scars.**

Aim 1: Produce Platform Capable of Generating and Analyzing Heterogeneous Collagen Alignment

Our platform was created to enable the swift and cost-efficient study of collagen gels possessing inherent spatial architecture heterogeneity. This was achieved through employing magnetic microspheres within a region of collagen solution prior to entire gel incubation to achieve differential levels of alignment within the gel regions. Additionally, novel image analysis methods were employed to quantify the generation of spatially

heterogeneous architectures within the collagen gel constructs as well as relate aspects of the embedded cell population to their surrounding architectures. Measurements taken included fiber and cell orientations and positions within constructs as well as cellular morphologies. These measurements were taken over a seven-day timecourse.

Aim 2: Characterize Spatial Heterogeneity in Mechanics and Cell Response within In Vitro Infarct Platform

We generated an in vitro engineered heart tissue platform capable of electrically and mechanically stimulating engineered constructs prior to and following application of a simulated infarction. Mechanics of engineered heart tissues pre- and post-simulated injury were evaluated utilizing custom analysis methodologies. Mechanical analyses quantifying spatial heterogeneities in strain fields were implemented alongside calculated stresses. These analyses evaluated, verified, and validated the mechanical alterations of the tissue samples pre- and post- wound. Pre-wound mechanics were utilized as a baseline for post-wound analysis metrics. Post-wound analyses were conducted along a seven-day timecourse post-injury.

Aim 3: Determine Relation Between Post-Myocardial Infarct Scar Mechanics and Mechanotransduction Pathways

While the preceding aims related to the generation of in vitro platforms capable of simulating different developmental stages of the infarct scar, it is important to not lose touch with the in vivo reality. Therefore, murine myocardial cross-sections were

immunohistochemically evaluated to determine spatial heterogeneities in expressed mechanosensors. This was done to enhance our understanding of the mechanics and cellular response in our environment of interest. Known mechanosensors were targeted in fixed tissue samples. The signals generated were spatially mapped to the samples of interest, and the resulting spatial distributions and heterogeneities were compared to collaborator produced mechanical characterizations. Characterizations were derived via noninvasive imaging and computational analyses. More specifically, marker signal intensity was mapped to corresponding strains within the samples to relate the cell signaling, scar progression, and mechanics that exist within the myocardium after infarction events.

1.3. References

1. Guo, T. *et al.* Multi-region proteome analysis quantifies spatial heterogeneity of prostate tissue biomarkers. *Life Sci Alliance* **1**, e201800042 (2018).
2. Ben-Moshe, S. & Itzkovitz, S. Spatial heterogeneity in the mammalian liver. *Nat Rev Gastroenterol Hepatol* **16**, 395–410 (2019).
3. Eary, J. F., O'Sullivan, F., O'Sullivan, J. & Conrad, E. U. Spatial Heterogeneity in Sarcoma ^{18} F-FDG Uptake as a Predictor of Patient Outcome. *Journal of Nuclear Medicine* **49**, 1973–1979 (2008).
4. de Maar, J. S. *et al.* Spatial heterogeneity of nanomedicine investigated by multiscale imaging of the drug, the nanoparticle and the tumour environment. *Theranostics* **10**, 1884–1909 (2020).
5. Allenby, M. C., Misener, R., Panoskaltsis, N. & Mantalaris, A. A Quantitative Three-Dimensional Image Analysis Tool for Maximal Acquisition of Spatial Heterogeneity Data. *Tissue Eng Part C Methods* **23**, 108–117 (2017).
6. Guerrero, Q. W. *et al.* Anisotropy and Spatial Heterogeneity in Quantitative Ultrasound Parameters: Relevance to the Study of the Human Cervix. *Ultrasound Med Biol* **44**, 1493–1503 (2018).
7. Virani, S. S. *et al.* Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association. *Circulation* **141**, (2020).
8. Kochanek, K. D., Xu, J. & Arias, E. Mortality in the United States, 2019. *NCHS Data Brief* 1–8 (2020).
9. Redfield, M. M. Heart Failure with Preserved Ejection Fraction. *New England Journal of Medicine* **375**, 1868–1877 (2016).
10. Braunwald, E. Heart Failure. *JACC Heart Fail* **1**, 1–20 (2013).
11. Richardson, W. J., Clarke, S. A., Alexander Quinn, T. & Holmes, J. W. Physiological implications of myocardial scar structure. *Compr Physiol* (2015) doi:10.1002/cphy.c140067.
12. Tsao, C. W. *et al.* Heart Disease and Stroke Statistics—2022 Update: A Report From the American Heart Association. *Circulation* **145**, (2022).
13. Richardson, W. J. & Holmes, J. W. Emergence of Collagen Orientation Heterogeneity in Healing Infarcts and an Agent-Based Model. *Biophys J* **110**, 2266–2277 (2016).
14. Correa, S. O., Luo, X. & Raub, C. B. Microfluidic fabrication of stable collagen microgels with aligned microstructure using flow-driven co-deposition and ionic gelation. *Journal of Micromechanics and Microengineering* **30**, (2020).
15. Lanfer, B. *et al.* Aligned fibrillar collagen matrices obtained by shear flow deposition. *Biomaterials* **29**, 3888–3895 (2008).

16. Kishan, A. P. *et al.* Fabrication of macromolecular gradients in aligned fiber scaffolds using a combination of in-line blending and air-gap electrospinning. *Acta Biomater* **56**, 118–128 (2017).

17. Nerger, B. A., Brun, P.-T. & Nelson, C. M. *Marangoni flows drive the alignment of fibrillar cell-laden hydrogels*. *Sci. Adv* vol. 6 http://advances.sciencemag.org/ (2020).

18. Nerger, B. A., Brun, P. T. & Nelson, C. M. Microextrusion printing cell-laden networks of type i collagen with patterned fiber alignment and geometry. *Soft Matter* **15**, 5728–5738 (2019).

19. Rodriguez, M. L. *et al.* Measuring the Contractile Forces of Human Induced Pluripotent Stem Cell-Derived Cardiomyocytes With Arrays of Microposts. *J Biomech Eng* **136**, (2014).

20. Aung, A. *et al.* 3D cardiac μtissues within a microfluidic device with real-time contractile stress readout. *Lab Chip* **16**, 153–162 (2016).

21. Feric, N. T. *et al.* Engineered Cardiac Tissues Generated in the Biowire II: A Platform for Human-Based Drug Discovery. *Toxicological Sciences* **172**, 89–97 (2019).

22. Ronaldson-Bouchard, K. *et al.* Engineering of human cardiac muscle electromechanically matured to an adult-like phenotype. *Nat Protoc* **14**, 2781–2817 (2019).

23. Boudou, T. *et al.* A Microfabricated Platform to Measure and Manipulate the Mechanics of Engineered Cardiac Microtissues. *Tissue Eng Part A* **18**, 910–919 (2012).

CHAPTER TWO

Fabrication and Characterization Methods for Investigating Cell-Matrix Interactions in Environments Possessing Spatial Orientation Heterogeneity

2.1. Introduction

Fibrillar collagen is a highly abundant matrix component in many tissues and a central determinant of tissue mechanical properties.[1] Fiber orientation and alignment provide a structural basis for mechanical anisotropy, which is essential for the function of many load-bearing tissues, including tendons, ligaments, heart valves, and blood vessels.[2–5] Additionally, underlying collagen architectures observed in tissues regulate a wide range of varied cellular responses including endothelial cell morphology, migration dynamics in cancer cell populations, and integrin mediated myofibroblast activation.[6–9]

In some of these tissues, matrix structure and the resulting mechanical properties vary significantly across local subregions, and the effects of such heterogeneity can be important for tissue function or failure. For example, mechanical stress gradients have been shown to produce heterogeneous structural hallmarks during headfold morphogenesis;[10,11] a hammock-like heterogeneity of matrix orientation from commissure to commissure in heart valves has been a key target of tissue engineering approaches;[12,13] the spatial transition from highly aligned tendon matrix to randomly oriented and mineralized bone matrix at the tendon-bone enthesis is critical to proper load transfer and musculoskeletal function;[14,15] spatial heterogeneity in the structural composition of aneurysms has been associated with high variation in tensile moduli and

failure locations;[16,17] and collagen orientation heterogeneity in dermal wounds has been associated with improved cell infiltration and healing rates.[18] Another notable example of heterogeneous collagen orientation is post-myocardial infarction, where failure modes can be related to the mechanical properties of the resultant scar.[19] Myocardial wall rupture post-MI can result from too much degradation (or too little accumulation) of collagen at the infarct location and potentially from misalignment of the fiber architecture.[20] In recent work, we found that stark collagen orientation heterogeneity emerges early post-MI in rat infarcts and remains through at least 6 weeks of the healing timecourse.[21] This was true even in cases with low global alignment (i.e., mostly isotropic). More recently, these localized pockets of fiber alignment have predicted increased risk for tissue rupture.[22]

In order to circumvent many of the threats presented by mechanically insufficient collagen architecture in scar tissue post-infarction, it is essential to understand how the scar is remodeled by local cell populations. While it is clear that local mechanical stresses play a major role in directing fiber manipulation by local fibroblast populations, many questions remain unanswered regarding cell-matrix interactions in heterogeneous environments such as the following: how much of a role do neighboring fibroblasts have in directing the deposition and fiber manipulation of their neighbors, how does localized regional alignment arise within largely random environments, at what distances can fibroblasts within the collagenous scar environment detect the alignment of the matrix it is embedded within, what effect does the fibroblast have on its neighbors through its own contractile inputs into the matrix network, and many others. As a first step in probing

cell-matrix interactions within heterogenous environments, we have developed a method to fabricate heterogeneous collagen gels in vitro with a region of randomly oriented fibers directly adjacent to an interconnected region of anisotropic alignment. In addition, we developed several image processing and automated analysis methods to quantify fiber and cell spatial heterogeneities from fluorescent microscopic live-cell images.

2.2. Methods

2.2.1. Gel fabrication

Briefly, type I rat tail collagen at a concentration of 4.1 mg/ml was purchased commercially (Advanced Biomatrix, San Diego, CA) and mixed on ice with the accompanying kit-provided collagen neutralization solution, 10XMEM, cell-containing media, and Pierce™ streptavidin-coated magnetic microspheres(Thermo Fisher Scientific, Waltham, MA). The microspheres possessed an average diameter of 1 μm and were bought at a solution concentration of 10 mg/ml. After mixing, solutions were pipetted into Nunc™ LabTek™ chambered coverglass slides. First, 41.25 μl of the neutralization solution was pipetted into a microcentrifuge tube followed by 91.8 μl of 10XMEM. For the bead-containing solution, 16 μl of streptavidin-coated magnetic microsphere solution was pipetted into the total solution which was then mixed thoroughly. The addition of the beads is skipped for the bead-less 'random' solution. At this point, 371 μl of the rat tail collagen solution was added and mixed with care to avoid microbubble formation. Warm cell-containing media at a volume of 60 μl is then added at a cellular concentration of 16.9 M/ml. Prior to pipetting, dividers were placed within

approximately 10.3 mm x 8.3 mm chamber wells to maintain solution separation based on what will become the two regions of interest. Dividers were fabricated as a cross to provide stability after placement within the chamber wells and are illustrated within the first step of Fig. 2.1A. Within each quadrant separated by the dividers illustrated in Fig. 2.1A, 60 μl of the final mixed solution was pipetted prior to divider removal. The bead-containing 'aligned' solution was placed within the outer quadrants while the bead-less 'random' solution was pipetted into quadrants located towards the center of the chambered coverglass slide. Aligned solution regions are indicated by exteriorly located blue shading of the chamber slide wells in the second step of Fig. 2.1A. Immediately following pipetting, dividers were removed and a compaction restricting insert was placed within the wells to inhibit gel contraction throughout the time course. The compaction restricting insert is illustrated in step 3 of Fig. 2.1A. Each well of the chamber slide contained 120 μl of bead-containing solution and 120 μl of bead-less solution. Upon divider removal, the solutions meet within the well to form a 240 μl heterogeneous gel solution. The chamber slides containing collagen solution were inserted into a custom-fabricated magnet holder assembly between two 100lb-force neodymium bar magnets (K&J Magnetics, Pipersville, PA) where the gels were polymerized at room temperature for one hour to allow for bead migration and fiber formation. The magnet holder assembly is integral to the fabrication process as it maintains the position of the gel solutions within the magnetic field produced between the two bar magnets inserted within the structure and flanking the gel. Bead migration

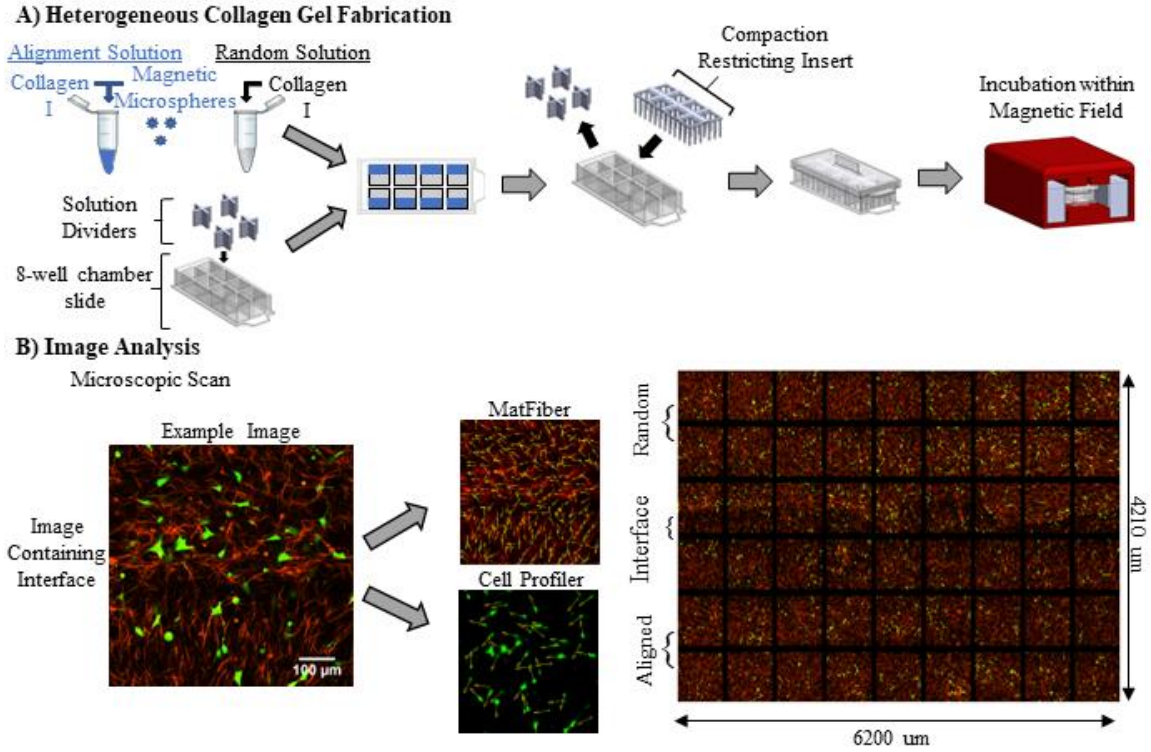

Fig. 2.1. Methodologies Overview: A) Representation of the fabrication process. The two solutions(aligned and random) are prepared and solution dividers are placed within the chamber wells. Aligned solution is pipetted into the darker shaded region at the periphery of the well to maintain positioning closer to the external magnets and experience a stronger portion of the magnetic field. Dividers are removed after pipetting, and an insert is placed within the gels to restrict compaction. The assembly is then covered and inserted into the magnet holder assembly. B) An example image is presented to highlight the orientation analyses conducted for fibers and cells within each image. Fiber analysis was conducted on red channel data while cell analysis was conducted on green channel data. The image stitch highlights gel dimensions captured by the 6x9 image grid and exhibits the range of the global coordinates for the gel sample.

along the magnetic field lines imparts the alignment found within the gel as demonstrated in work performed by Guo et al.[23] Following the one-hour room temperature polymerization, gels were placed in standard 37° C, 5% CO_2 incubators. GFP-transfected NIH3T3 cells(Cell Biolabs Inc cat: 50–672–593) were cultured in standard culture media (DMEM, 10% FBS, and 1% pen-strep) and suspended within collagen gels with a final collagen concentration of 2.5 mg/ml and a cellular concentration of 1.75 M/ml. One

solution contained the addition of streptavidin-coated magnetic microparticles, while the other solution was left without the addition of magnetic microparticles. Gels were maintained with standard culture media which was replaced daily at a volume 1.5x greater than gel solution volume. Gels were then imaged by confocal reflectance microscopy on days 1, 3, and 7 utilizing a Nikon Eclipse Ti-E motorized inverted microscope with a 20X oil-immersion objective and its accompanying EZ-C1 software(v3.80) for image capture.

2.2.2. Image acquisition and processing

Images were taken along a 6 × 9 grid path that fully captured the center section of each gel sample measuring approximately 6200 μm × 4210 μm. Images were then processed independently using a previously published image processing method (MatFiber[24]) implemented in MATLAB and CellProfiler (v3.1.5) to obtain the orientations (α) of fibers and cells within each image. An example image displaying orientation analysis as well as a stitch exhibiting the 6 × 9 grid path is shown in Fig. 2.1B. MatFiber quantifies collagen alignment by segmenting the image into a grid of approximately 10 μm × 10 μm subregions, calculating intensity gradients, and returning an angle and image coordinates for each subregion visualized in Fig. 2.1B via vector field overlay. The implemented CellProfiler pipeline operates by separating foreground from background utilizing a global threshold value. Calculation of the threshold value was performed by the built-in 'RobustBackground' thresholding method which assumes an approximate gaussian background distribution and sets the threshold value as the mean

plus user determined number of standard deviations of the included pixel range. For our analysis, the included pixel intensity window for threshold calculation ranged from the 20th to the 95th percentile and placed the threshold two standard deviations above the mean of included values. After thresholding, primary objects are identified by size and shape while excluding objects outside of a 12 to 123 μm diameter range. Object coordinates, area, major and minor axis lengths, and orientation are produced. The obtained fiber subregion orientations and coordinate positions as well as the cell coordinate positions, area, major and minor axis lengths, and orientations were placed into tables. Global coordinates were calculated for each cell and fiber subsection based on image and intra-image object position. Calculating global coordinate positions and orientations of cells and fibers enabled us to perform a variety of alignment correlations and regional comparisons across varying spatial ranges and zones as described below.

2.2.3. Fiber analysis

In order to characterize the fiber architecture transition across the random to aligned regions, we quantified fiber orientation distribution histograms across 10-degree orientation bins from 0 to 180°(n = 18 bins). Alignment values were found by calculating the mean vector length (MVL) of fiber angle distributions given by Eq. (2.1.) (as performed previously):[21,25–27]

$$MVL = \sqrt{\left(\left(\frac{\sum_1^n \sin\left(\frac{2*\alpha(n)*\pi}{180}\right)}{n}\right)^2 + \left(\frac{\sum_1^n \cos\left(\frac{2*\alpha(n)*\pi}{180}\right)}{n}\right)^2\right)} \quad (2.1.)$$

To characterize the spatial distribution of fiber alignment, we grouped fiber angles by their y-position coordinate (i.e., distance relative to the bead-no bead interface) and calculated alignment values for each y-position grouping. To find these averages, the assigned fiber alignment values (MVL) of individual zones were organized by experimental run, treatment, sample, and timepoint before being discretized and binned based on position along gradient. These analysis zones were constructed by segmenting the observed gel section into 210 μm × 210 μm square regions that spanned the coordinate range of data collected. Intra-sample fiber alignment values and zone positions were averaged based on discretized position bins. Alignment value averages were plotted against corresponding gradient position. A two-way ANOVA was conducted for gel type (heterogeneous vs homogeneous) and gel position (n = 8). Additionally, to evaluate classification of one gel region as 'aligned' and the other as 'random,' heterogeneous and homogeneous gels were halved, and fiber alignment values were averaged by respective gel region prior to conduction of student t-tests within each gel comparing gel halves. Evaluation of alignment persistence throughout the timecourse was desired to assist in understanding initial local architecture permanence with an embedded cell population. The fiber zones were sorted by experimental run, treatment, sample, and timepoint before discretization and binning by day 1 initial fiber alignment value into eight alignment value ranges each spanning a value range of 0.125. Fiber alignment values within these bins were averaged across samples for each treatment at each timepoint to obtain timecourse measurements.

2.2.4. Cell analysis

In order to characterize the cell recognition of the architecture transition from random to more aligned, we quantified cell orientation distribution histograms across 10° orientation bins from 0 to 180°(n = 18 bins) . We also quantified cell elongation and cell spread area after grouping cells based on the local alignment of collagen fibers. Specifically, cell centric analysis zones were constructed for each individual cell as circular regions with radii length of 210 μm centered on each cell. Cells were grouped based on the local fiber alignment within these zones on day 1, and cell parameters were averaged across these groups for subsequent timepoints. Only cells with a major axis to minor axis ratio of 1.5 were included within analyses utilizing cell orientation to minimize the occurrence of misattributing cellular orientation.

2.2.5. Quantifying cell and matrix heterogeneous interactions

While our gel fabrication method produced fibrous constructs with clear, macroscale heterogeneity (two starkly different regions), we also sought to test quantification metrics for analyzing more subtle microscale heterogeneities that might exist within each region. To do so, we employed three different approaches. First, we quantified the correlation between cell alignment angles and fiber alignment angles within local pockets of varying sizes. This was accomplished by finding the mean angle (MA) of all cells within a particular range of a central cell, which is calculated by Eqn. (2.2.) :

$$MA = \ arctan\left(\left(\frac{180}{pi}\right) * \left(\frac{1}{2}\right) * \left(\frac{\sum_1^n \sin\left(\frac{2*\alpha(n)*\pi}{180}\right)}{n}, \frac{\sum_1^n \cos\left(\frac{2*\alpha(n)*\pi}{180}\right)}{n}\right)\right). \quad (2.2.)$$

The MA is calculated for all fibers within that same range, then subtracted from the cell MA to quantify a delta mean angle(MA) that represents how closely aligned the cells and fibers are within the region. This was repeated for all cells in each sample, and the MVL of the resulting distribution of MAs was used as an overall measure of localized cell-fiber alignments for a given local zone size. Our second quantification of heterogeneity used a similar approach but analyzed one cell at a time rather than local pockets of cell groups. Specifically, the MA of all fibers within a particular range of a given cell was subtracted from that individual cell's orientation angle, producing a MA that represents how closely aligned the cell is with the average fiber orientation across a given range. This was repeated for all cells, and the MVL of the resulting distribution of MAs was used as an overall measure of localized cell-fiber alignments for a given local zone size. Lastly, our third quantification of heterogeneity followed the previous method developed by Richardson and Holmes using alignment vs. distance plots between individual pairs of cells and fibers.[22] Specifically, we calculated the dot product of a cell or fiber's orientation vector with every other fiber's orientation vector and then plotted those dot product alignment values against the distance between each pair. We repeated this for thousands of random fiber-fiber and cell-fiber pairings and averaged alignment values within incremental distance bins.

2.2.6. Statistical methods

Data represent the mean of eight independent replicates and error bars illustrate the standard error of the mean. Statistical analysis performed utilized two-sample t-tests and two-way ANOVAs with n = 8. A p-value was deemed statistically significant when less than 0.05. Frequency plots were constructed by pooling data from the eight independent replicates into a single analysis pool for each respective timepoint and condition.

2.2.7. Code and data availability

Analysis code and data freely available at: https://github.com/ SysMechBioLab.

2.3. Results

2.3.1. Fiber properties

Due to the importance and prevalence of heterogeneous architecture in collagen-rich environments, we sought to generate continuous tissue constructs possessing intra-gel alignment heterogeneity. In order to evaluate the degree of generated alignment within gel regions at a fundamental level, orientation distributions were constructed with the fiber units of each region of alignment of the heterogeneous gels and illustrated in Fig. 2.2B. Fibers within the aligned region display a clear peak around the desired 90◦ orientation bin while the random region exhibits no significant peak. This peak within the aligned region coupled with the lack of distinct peak within the random region again supports the claim that differential alignment within the gel was created. It also appears

that the distribution range is maintained through day 7 in both gel sections despite a slight reduction in the peak of the aligned region distribution moving from day 1 to day 3. Next, to appropriately assess the generation of a heterogeneous architecture, we evaluated the average subregion degree of alignment across the gel along the intended gradient. Instead of evaluating hand selected pockets from each greater region, observing alignment across the entirety of the gel enabled a clearer picture of whether the desired spatial trend was generated. An increase in degree of alignment is discernible from the alignment evaluation illustrated in Fig. 2.2C. Progressing across the gel reveals a clear deviation away from the alignment values exhibited by analysis of purely random control gels while the random region of the gel possesses alignment values within the range of a purely random gel. A two-way ANOVA was conducted comparing discretized cross-gradient alignment. The analysis revealed no difference based on which gel, heterogeneous or homogeneous, was analyzed. This is likely due to the heterogeneous gel possessing a region of random orientation as well as a region of greater alignment. This translates into recognizing no real difference between samples at the full gel level. However, there was significance at the $p = 0.01$ level when considering position within the gel along the axis of the gradient, suggesting a confirmation of the generation of a region of more aligned fibers. To determine if there was not only a gradient generated, but also sufficient alignment to categorize gel halves as separate alignment regions, gel halves were categorized as either a ‘Random’ or ‘Aligned’ region. Then 210um analysis zones within each region were averaged to obtain an overall value for the respective gel halves. T-tests were then performed to compare the regions and the resulting p-values are

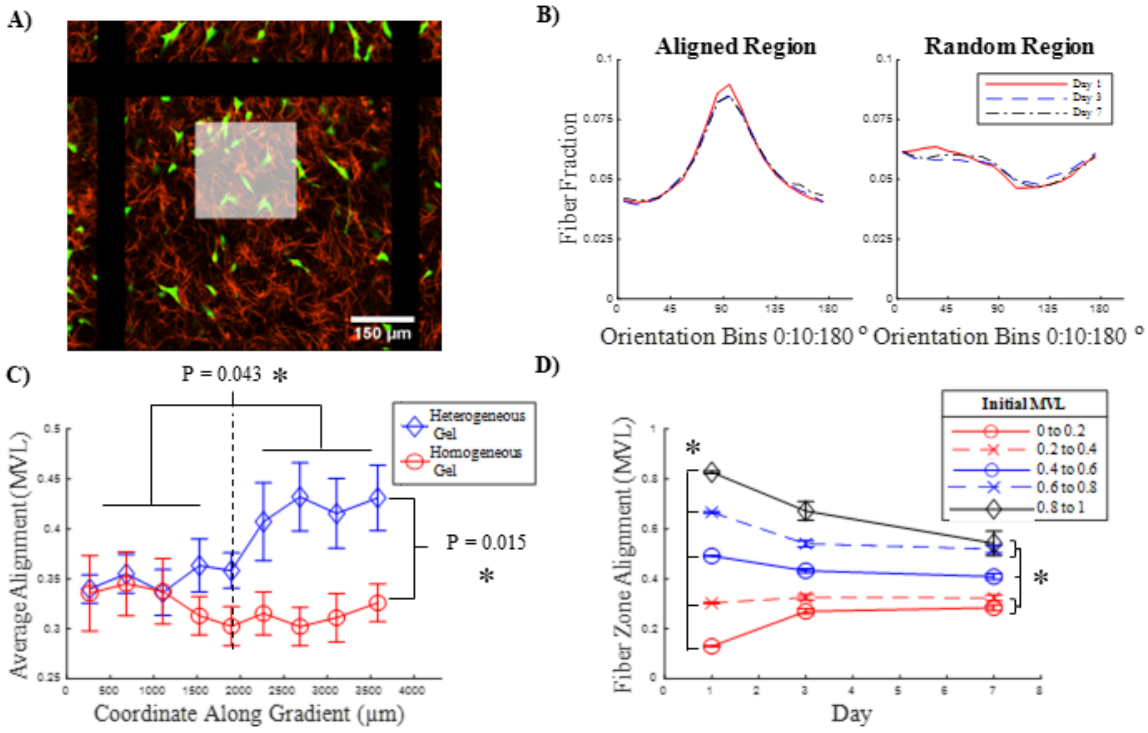

Fig. 2.2. Evaluating Generated Fiber Alignment Heterogeneity. A) A white zone overlay denoting an example zone of analysis that was utilized to evaluate the generation of an alignment gradient. B) Orientation distributions highlighting the differential fiber alignment exhibited by the two interconnected regions of heterogeneous gel constructs. C) Analysis of degree of alignment change across intended gel alignment axis. Statistical values displayed compare aligned region of heterogeneous gel to the random region of the heterogeneous gel and the corresponding homogeneous gel region.(n=8, $\pm SE$.*p<0.05). D) Evaluation of fiber architecture maintenance over timecourse. Two-way ANOVA revealed significance between all individual day 1 initial alignment groupings while day 7 analysis revealed a loss of significance between the two lowest initial alignment groupings and the two highest initial alignment value groupings. Significance was maintained between the remaining initial alignment groupings.

displayed in Fig. 2.2C. Significance at the p = 0.05 level was observed when comparing the two regions of the heterogeneous gel as well as when comparing the aligned region of the gradient gel to both regions of the random gel. No significance was observed between the random region of the heterogeneous gel and either region of the random gel. This supports the evidence that a gradient was generated across the heterogeneous gels

through the streptavidin magnetic microparticle migration method. In addition to evaluating whether the desired architecture was generated, maintenance of architecture was an additional topic of interest as embedded cell populations can manipulate their environments when they engage with them.[28] Fiber subregions were grouped by initial alignment value and average alignment was found for each grouping. These groupings were maintained for analysis of day 3 and day 7 degree of alignment as well. Resultant values are shown in Fig. 2.2D and exhibit a trend toward centrality of alignment values. The greatest changes in average alignment are observed day 1 to day 3 while day 3 to day 7 is relatively stable. Two-way ANOVA revealed statistical significance between all day 1 values while day 7 analysis revealed a loss of significance between the two lowest initial alignment groupings and between the two greatest alignment groupings while significance is maintained concerning comparison to the other initial alignment value groupings. We attribute the change in alignment values to actions of the embedded fibroblast populations onto their surrounding environment. The trend toward centrality may suggest cellular detection of surrounding architecture through imparting tension onto and detecting tension on the matrix.

2.3.2. Cell response to regional architecture

In addition to characterizing the architecture of the gel, cell response to architecture is also of interest as cell morphologies are directed by their environments.[29–32] The distribution of cellular orientation was generated by region for each day of observation in Fig. 2.3B to highlight any trends in cellular orientation related to gel

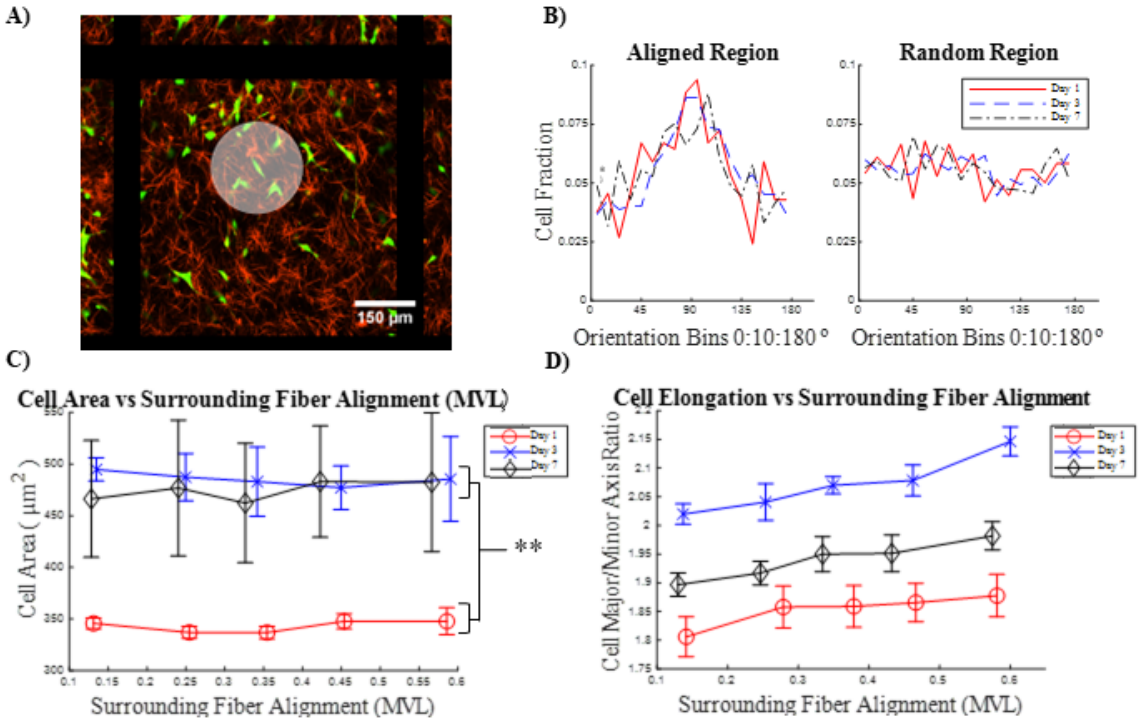

Fig. 2.3. Demonstrating Cell-Centric Analysis Through Orientation and Parameter Response to Matrix Environment. A) A white zone overlay denoting an example zone of analysis that was utilized to support cell-centric analysis methods. B) Orientation distributions highlighting the differential cellular alignment exhibited within the two interconnected regions of heterogeneous gel constructs. C) Analysis of cell area by day compared to the level of alignment exhibited by local collagen fiber architecture within heterogeneous gel constructs(n=8, $\pm SE$. *p<0.05, **p<0.01). D) Analysis of cell elongation by day compared to the level of alignment exhibited by local collagen fiber architecture within heterogeneous gel constructs.

region. A peak is observed throughout the experimental timecourse within the aligned region which is maintained through day 7. Within the random region, the distribution appears less consistent through the range of angles and does not exhibit a clear orientation preference. These orientation distributions resemble the underlying fiber orientation distributions shown in Fig. 2.2B for each respective region of the gel. Fig. 2.3C highlights the difference in cell area between day 1 and the remainder of the

experiment. Low cell area at day 1 suggests that cells appear to have not spread and begun interacting with their environment until some point between the initial day 1 and following day 3 observations as day 1 spreading values across all quintiles exhibited significant difference at the $p = 0.01$ level compared to all day 3 and day 7 values. Full spreading appears to have occurred by day 3 and was maintained through day 7. Quantifying cell elongation in Fig. 2.3D indicated a significant increase in elongation from day 1 to day 3 and day 7. While a slight positive relationship appeared between elongation and local collagen fiber alignment for days 1, 3, and 7, no intra-day significant difference was detected by two-way ANOVA.

2.3.3. Fiber-Cell interactions

A central motivation for our work was to assess the distance at which embedded cell populations can detect and direct regional alignment and coordinate with neighboring cells. Fig. 2.4B highlights a clustering of delta values around 0 showing a general agreement amongst cell and fiber mean angles between zones. Concerning the smaller analysis zone sizes, over 43% of the orientation deltas fall between −10° and 10° This agreement is seemingly enhanced in the larger zone size as a greater frequency is noticeable around the smaller delta value bins. Greater than 67% of the orientation deltas for the larger zone analysis falls between −10° and 10° This means a higher portion of analysis zones have small differences between fiber and cell mean angles when considering large analysis zones. When applying this analysis to the central cell upon which the zones are anchored and finding the delta between the central cell's orientation

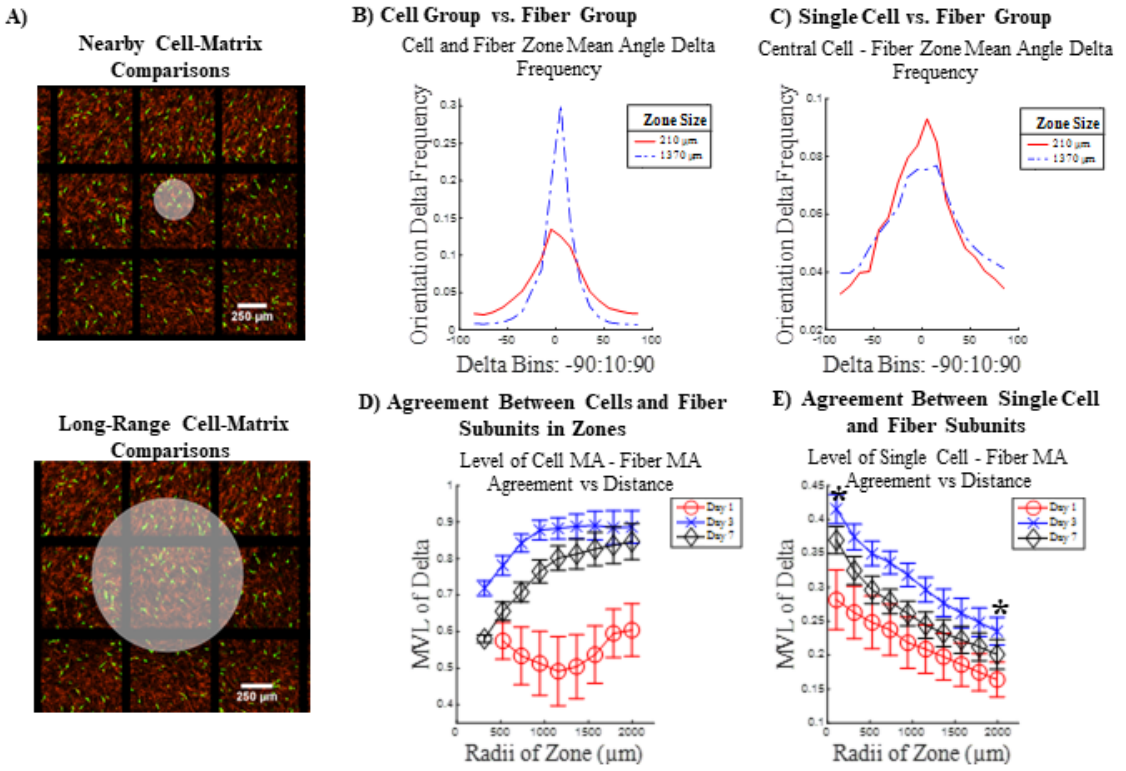

Fig. 2.4. Determining Cell-Matrix Interactions and Recognition Across Varied Distances. A) Example white overlay zones highlighting varied cell-centric analysis zone sizes. B) Analysis of mean angle orientation delta frequency of cells and fibers included in zone between two distinct zone sizes. C) Central cell orientation and surrounding fiber mean angle delta frequency for two distinct zone sizes. D) Evaluating the uniformity of orientation agreement of cells and fibers included in zone in relation to size of analysis zone. E) Evaluating the uniformity of orientation agreement of central cell and surrounding fiber mean angle in relation to size of analysis zone. Significance was observed when comparing the smallest and largest zone sizes of the analysis for day 3 and day 7(n = 8, $\pm SE$. *p<0.05).

and the mean angle of the surrounding fibers, the resulting distribution once again peaked at the lower delta values corresponding to rough agreement of orientation as seen in Fig. 2.4C. However, increasing the zone size has a marginal effect on agreement between cell orientation and fiber zone orientation. The proportion of deltas falling between −10° and 10° for the smaller and larger zone sizes respectively are 31.5% and 26.6%, meaning the opposite trend occurred when comparing to the delta distributions for mean angles of cell and fiber populations within the zones. In addition to analyzing distributions, calculating

the mean vector length of the delta distributions of increasing zone sizes provides better insight into the uniformity of the orientation deltas as you increase the zone size. As zone size is increased, there is more likely to be a uniform deviation between cell zone mean angle and fiber zone mean angle. This is observed in the gradual positive slope exhibited by the day 3 and day 7 plots in Fig. 2.4D. A two-way ANOVA revealed a significant difference between days analyzed, but no significance for radii of zone within each day. For single cell analysis, as zone size is expanded, the agreement of delta values found decreases. With larger zone sizes, there is less likely to be a uniform deviation between central cell orientation and surrounding fiber orientation. Multiple comparisons ad-hoc analysis following two-way ANOVA evaluating the dataset by distance and day did reveal a significant difference between MVL values found for the smallest zone size and the largest zone size for both days 3 and 7 with p-values less than 0.05. To further evaluate the generation of heterogeneity within our collagenous gel constructs and the cellular recognition of this architecture, individual fiber subunits and cells were compared to other fibers within a certain distance by computing the dot product of the central analysis object's (cell or fiber) orientation with each individual zone fiber's orientation within the analysis range. These dot products were averaged and plotted against the respective distance revealing a greater level of orientation agreement for close-range objects and a decline in agreement as distance was increased. Cell agreement with individual fibers is shown in Fig. 2.5B while fiber to fiber agreement is shown in Fig.

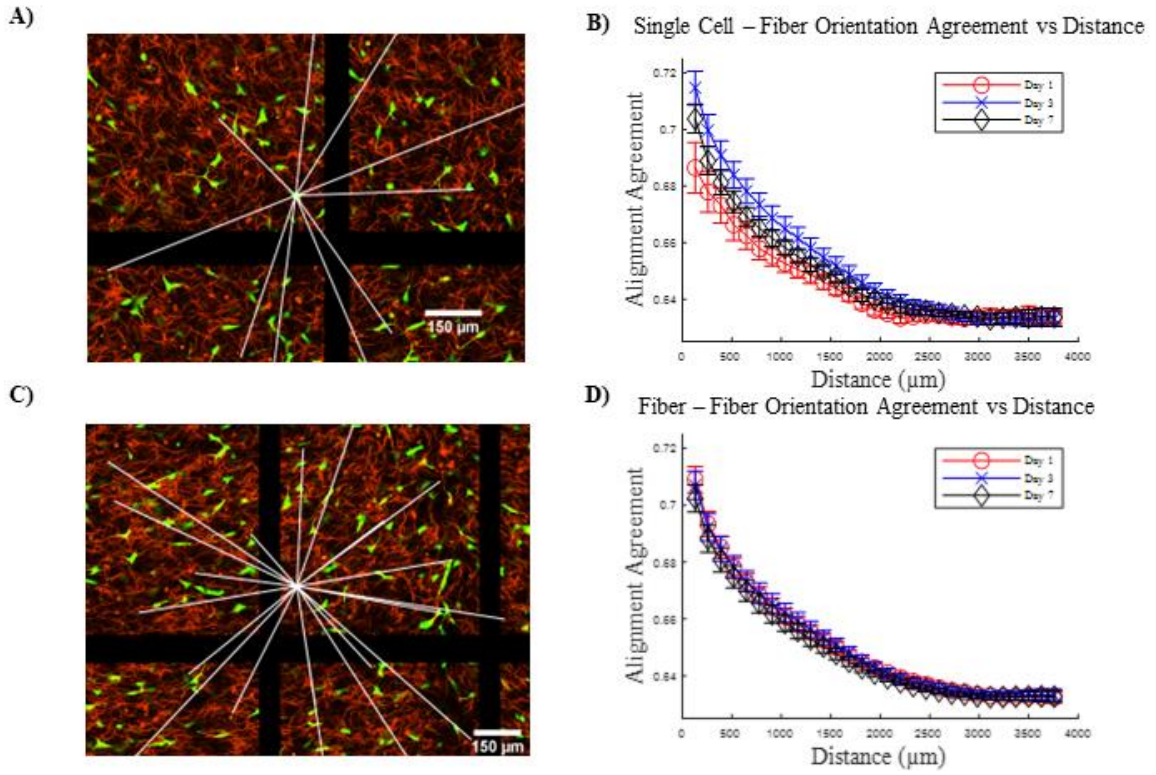

Fig. 2.5. Object to Object Anisotropic Orientation Agreement Decreases with Distance. A,C) Example image with white line overlay illustration denoting object to object analysis; cell to fiber and fiber to fiber respectively. B)Degree of alignment agreement calculated via dot product between single cells and fibers in relation to distance between analysis objects. D) Degree of alignment agreement calculated via dot product between single fiber and other fibers within construct in relation to distance between analysis fiber to fiber analyses with p-values drastically lower than p = 0.01.

2.5D. Two-way ANOVA revealed significance by distance and day for both cell to fiber and fiber to fiber analyses with p-values drastically lower than p = 0.01. This particular analysis also enables an approximation for how far cells and fibers are locally aligned as all curves tend to flatten around 3000 µm.

2.4. Discussion

In this work, we present a simple methodology to generate collagen gel constructs possessing neighboring regions of isotropic and anisotropic fiber orientations. This

platform can be utilized to probe spatial and temporal interactions of cells with fibrous matrix architectures in vitro that could inform further experimentation aimed at understanding cell behavior across neighboring regions possessing inherent differences in architecture. Through our zone-based analysis methodology, it was determined that not only did the gels possess a clear gradient of alignment across the desired dimension, but the angular distribution of fibers within each region was representative of the desired preferred alignment. Over the course of a week, subregion orientation analysis revealed a general trend toward centrality throughout the construct with the largest changes occurring between day 1 and day 3 as the embedded fibroblast populations spread, engaged with, and acted upon their surrounding environment. This process is easily discernible within the analysis of cell orientation, elongation, and spreading. While very little spreading and elongation was observed on day 1, analysis revealed a discernible orientation preference coinciding with the orientation of bead migration and the resultant underlying fibrous architecture. While a recognition of their surrounding environment is evident from a simplified cell parameter analysis, directly analyzing the degree of agreement between a central cell and fibers contained within varied analysis zone sizes revealed a rapid decrease in relevancy for mean fiber angle and distance from the central cell. However, when evaluating the mean orientation of the population of cells within the zone against the mean orientation of fibers, the agreement was improved with increasing zone size. This highlights the complicated nature of translating microscopic relationships to macroscopic behaviors and the importance of assessing matrix organization at the particular scales most relevant to a specific question of interest. Combining heterogeneity

fabrication techniques with heterogeneity analysis metrics could enable many future studies evaluating the role fibroblasts play in directing neighboring cells through time-lapse analysis of cell morphology and various molecular markers compared to neighboring cell expression of target proteins of interest. Analysis from a matrix-focused perspective can lead to insights regarding the architecture of the matrix. Closely analyzing the timecourse of local pocket degree of alignment arising as a result of embedded cell population stimulation and underlying fiber architecture provides this type of matrix-minded analysis. These concepts of neighbor-informed and regional remodeling are partly inspired by our previously conducted computational modeling studies analyzing the spontaneous emergence of fiber orientation heterogeneity through long-, medium-, and short-range sensing of fiber orientations by embedded cell populations. In these computational analyses, long-range sensing did result in cell and fiber orientation heterogeneity resembling experimental infarct data.[22] Exploring these computational results through experimental means utilizing our simple fabrication platform can enable the elucidation of the underlying mechanisms of the rise of heterogeneous fibrous architecture in relevant systems. While other alignment methodologies exist for producing aligned constructs, such as microfluidic fabrication, drop casting, electrospinning, and bioprinting, not all of them can produce a continuous heterogeneous construct, and the methodologies that can are typically complicated, intricate, time consuming, or produce a limited range of smaller construct sizes. Microfluidic fabrication of aligned collagen matrices typically incorporates additional polymer inclusions and results in highly aligned constructs that lack architecture

heterogeneity. The resulting matrices are useful for the in vitro simulation of highly aligned tissues but not highly heterogeneous tissues.[33,34] Electrospinning methods have recently been shown capable of generating gradients of alignment through fibrous constructs.[35] This methodology is however limited to certain compatible polymer solutions limiting overall application to experimental methods desiring heterogeneous collagen gels possessing no additional polymer inclusions. Drop casting was demonstrated to generate collagen gels possessing varying degrees of alignment throughout the gel constructs in the radial direction from the placed collagen droplet.[36] This methodology appears to be useful for the generation of many replicates of like-collagen gels, but the radial nature of the alignment could confound and limit analysis methods intended to elucidate cooperative cellular response and the fabrication requires intricate environmental control. Bioprinting methodologies, like other methodologies mentioned, are capable of generating constructs with controlled anisotropy and are very promising for intricate design of larger and complicated constructs.[37] However, bioprinting methodologies require specialized fabrication tools possessing inherently intricate levels of control. The methodology we have presented here simply requires typical collagen gel preparation, magnetic microparticle inclusion, a support structure, and a brief room temperature delay within a magnetic field prior to incubation. Fabrication yields intra-gel alignment heterogeneity of large volumetric constructs. A limitation to this methodology is the lack of intricate spatial control. While a transition threshold is generated between isotropic to anisotropic architecture, more elaborate and smaller scale heterogeneities were not fabricated. The nature of native heterogeneity

gradients varies with different tissues of concern, with some tissues exhibiting smooth heterogeneity gradients like those found at the tendon-bone enthesis while other transitions are abrupt such as those found in infarct scars and dermal wounds.[14,18,21] Our fabricated gels demonstrate an abrupt change much like the latter examples. Previous computational work has predicted that stark interfaces as those found in infarct scars are prone to failure when subjected to mechanical stresses.[22] Future experiments could leverage our methodology to test these mechanical properties directly. An additional limitation to our approach the uppermost levels of alignment achieved. While certain smaller regions possessed alignment levels above the 0.8 level on the scale of 0 to 1, bulk alignment of the aggregate were much more moderate. If extreme anisotropy in architecture is desired, then it may be more appropriate to utilize one of the other previously mentioned alignment methodologies. It may be possible to generate more complex patterns of alignment within fabricated collagen gels through the employment of electromagnets and more complex rigs that adjust the position, strength, and direction of the magnetic field in relation to the fabricated gels. There are many more possible experimental variations to pursue through the utilization of easily fabricated heterogeneous collagen constructs paired with our various analysis considerations. Future works will revolve around utilization of the platform to tease out the roles of matrix heterogeneity and cell-cell signaling. Previous studies utilizing various cell types have illustrated differences in cellular morphology and migration related to matrix heterogeneity and imparted tension.[38–40] Varying cell densities across the generated heterogeneities and evaluating cellular response could lead to insights concerning the

importance of matrix heterogeneity and cell-imparted tension in macro-level coordinated cell behavior. Additionally, collagen turnover within the platform should be investigated as the balance of degradation and deposition plays a role in the development of heterogeneities.[41]

2.5. Conclusion

Spatial orientation heterogeneity of collagen plays an important role in the relationship between micro-architecture and macro-functionality of tissues. While many methodologies capable of generating anisotropic alignment exist, there are trade-offs concerning time, cost, complexity, spatial control, and purity of included matrix components. This work presents a simple and effective heterogeneous collagen gel fabrication and automated analysis methodology. Spatial orientation heterogeneity was verified through regional evaluation of fiber orientation and cross-construct degree of alignment. Additionally, cell-centric methodologies employed highlight the importance of evaluating cellular response in the context of not only its immediate surrounding environment but also in the context of its environment at greater distances. This platform enables efficient and low-cost experimentation probing fundamental cell-matrix interactions within spatially heterogeneous environments.

2.6. References

1. Chen, J., Ahn, T., Colón-Bernal, I. D., Kim, J. & Banaszak Holl, M. M. The Relationship of Collagen Structural and Compositional Heterogeneity to Tissue Mechanical Properties: A Chemical Perspective. *ACS Nano* vol. 11 10665–10671 (2017).

2. Perez-Puyana, V., Jiménez-Rosado, M., Guerrero, A. & Romero, A. Anisotropic properties of PCL/gelatin scaffolds obtained via electrospinning. *International Journal of Fracture* **224**, 269–276 (2020).

3. Dwyer, K. D. & Coulombe, K. L. K. Cardiac mechanostructure: Using mechanics and anisotropy as inspiration for developing epicardial therapies in treating myocardial infarction. *Bioactive Materials* **6**, 2198–2220 (2021).

4. Lynch, H. A., Johannessen, W., Wu, J. P., Jawa, A. & Elliott, D. M. Effect of Fiber Orientation and Strain Rate on the Nonlinear Uniaxial Tensile Material Properties of Tendon. *Journal of Biomechanical Engineering* **125**, 726–731 (2003).

5. Joyce, E. M., Liao, J., Schoen, F. J., Mayer, J. E. & Sacks, M. S. Functional Collagen Fiber Architecture of the Pulmonary Heart Valve Cusp. *The Annals of Thoracic Surgery* **87**, 1240–1249 (2009).

6. Huang, N. F. *et al.* The modulation of endothelial cell morphology, function, and survival using anisotropic nanofibrillar collagen scaffolds. *Biomaterials* **34**, 4038–4047 (2013).

7. Ray, A., Morford, R. K., Ghaderi, N., Odde, D. J. & Provenzano, P. P. Dynamics of 3D carcinoma cell invasion into aligned collagen. *Integrative Biology (United Kingdom)* **10**, 100–112 (2018).

8. Han, W. *et al.* Oriented collagen fibers direct tumor cell intravasation. *Proc Natl Acad Sci U S A* **113**, 11208–11213 (2016).

9. Huang, C. *et al.* The involvement of integrin β1 signaling in the migration and myofibroblastic differentiation of skin fibroblasts on anisotropic collagen-containing nanofibers. *Biomaterials* **33**, 1791–1800 (2012).

10. Varner, V. D., Voronov, D. A. & Taber, L. A. Mechanics of head fold formation: Investigating tissue-level forces during early development. *Development* **137**, 3801–3811 (2010).

11. Filas, B. A., Varner, V. D., Voronov, D. A. & Taber, L. A. Tracking morphogenetic tissue deformations in the early chick embryo. *Journal of Visualized Experiments* (2011) doi:10.3791/3129.

12. Driessen, N. J. B., Beuten, C. V. C. & Baaijens, F. P. T. Improved prediction of the collagen fiber architecture in the aortic heart valve. *Journal of Biomechanical Engineering* **127**, 329–336 (2005).

13. Tamiello, C., Bouten, C. V. C. & Baaijens, F. P. T. Competition between cap and basal actin fiber orientation in cells subjected to contact guidance and cyclic strain. *Scientific Reports* **5**, (2015).

14. Thomopoulos, S., Marquez, J. P., Weinberger, B., Birman, V. & Genin, G. M. Collagen fiber orientation at the tendon to bone insertion and its influence on stress concentrations. *Journal of Biomechanics* **39**, 1842–1851 (2006).

15. Killian, M. L., Cavinatto, L., Galatz, L. M. & Thomopoulos, S. The role of mechanobiology in tendon healing. *Journal of Shoulder and Elbow Surgery* vol. 21 228–237 (2012).

16. Vallabhaneni, S. R. *et al.* **Heterogeneity of Tensile Strength and Matrix Metalloproteinase Activity in the Wall of Abdominal Aortic Aneurysms**. *Journal of Endovascular Therapy* **11**, (2004).

17. Hurks, R. *et al.* Circumferential heterogeneity in the abdominal aortic aneurysm wall composition suggests lateral sides to be more rupture prone. *Journal of Vascular Surgery* **55**, 203–209 (2012).

18. Montgomery, J. *et al.* The connexin 43 carboxyl terminal mimetic peptide αCT1 prompts differentiation of a collagen scar matrix resembling unwounded skin. *bioRxiv* (2020) doi:10.1101/2020.07.07.191742.

19. Clarke, S. A., Richardson, W. J. & Holmes, J. W. Modifying the mechanics of healing infarcts: Is better the enemy of good? *Journal of Molecular and Cellular Cardiology* vol. 93 115–124 (2016).

20. Richardson, W. J., Clarke, S. A., Alexander Quinn, T. & Holmes, J. W. Physiological implications of myocardial scar structure. *Compr Physiol* (2015) doi:10.1002/cphy.c140067.

21. Richardson, W. J. & Holmes, J. W. Emergence of Collagen Orientation Heterogeneity in Healing Infarcts and an Agent-Based Model. *Biophysical Journal* **110**, 2266–2277 (2016).

22. Korenczuk, C. E., Barocas, V. H. & Richardson, W. J. Effects of Collagen Heterogeneity on Myocardial Infarct Mechanics in a Multiscale Fiber Network Model. *Journal of Biomechanical Engineering* **141**, (2019).

23. Guo, C. & Kaufman, L. J. Flow and magnetic field induced collagen alignment. *Biomaterials* **28**, 1105–1114 (2007).

24. Fomovsky, G. M. & Holmes, J. W. Evolution of scar structure, mechanics, and ventricular function after myocardial infarction in the rat. *American Journal of Physiology-Heart and Circulatory Physiology* **298**, H221–H228 (2010).

25. COTTON, J. W. Review of Statistical Analysis, Fourth Edition. *Contemporary Psychology: A Journal of Reviews* **19**, 488–488 (1974).

26. Richardson, W. J., Kegerreis, B., Thomopoulos, S. & Holmes, J. W. Potential strain-dependent mechanisms defining matrix alignment in healing tendons. *Biomechanics and Modeling in Mechanobiology* **17**, 1569–1580 (2018).

27. Rouillard, A. D. & Holmes, J. W. Mechanical regulation of fibroblast migration and collagen remodelling in healing myocardial infarcts. *The Journal of Physiology* **590**, 4585–4602 (2012).

28. Petroll, W. M., Ma, L., Kim, A., Ly, L. & Vishwanath, M. Dynamic assessment of fibroblast mechanical activity during rac-induced cell spreading in 3-D culture. *Journal of Cellular Physiology* **217**, 162–171 (2008).

29. Jeon, H., Hidai, H., Hwang, D. J., Healy, K. E. & Grigoropoulos, C. P. The effect of micronscale anisotropic cross patterns on fibroblast migration. *Biomaterials* **31**, 4286–4295 (2010).

30. Thé, M. *et al. Anisotropy of cell adhesive microenvironment governs cell internal organization and orientation of polarity CELL BIOLOGY ENGINEERING*. *PNAS December* vol. 26 (2006).

31. Kim, D. H. *et al.* Mechanosensitivity of fibroblast cell shape and movement to anisotropic substratum topography gradients. *Biomaterials* **30**, 5433–5444 (2009).

32. Bashur, C. A., Dahlgren, L. A. & Goldstein, A. S. Effect of fiber diameter and orientation on fibroblast morphology and proliferation on electrospun poly(d,l-lactic-co-glycolic acid) meshes. *Biomaterials* **27**, 5681–5688 (2006).

33. Correa, S. O., Luo, X. & Raub, C. B. Microfluidic fabrication of stable collagen microgels with aligned microstructure using flow-driven co-deposition and ionic gelation. *Journal of Micromechanics and Microengineering* **30**, (2020).

34. Lanfer, B. *et al.* Aligned fibrillar collagen matrices obtained by shear flow deposition. *Biomaterials* **29**, 3888–3895 (2008).

35. Kishan, A. P. *et al.* Fabrication of macromolecular gradients in aligned fiber scaffolds using a combination of in-line blending and air-gap electrospinning. *Acta Biomaterialia* **56**, 118–128 (2017).

36. Nerger, B. A., Brun, P.-T. & Nelson, C. M. *Marangoni flows drive the alignment of fibrillar cell-laden hydrogels*. *Sci. Adv* vol. 6 http://advances.sciencemag.org/ (2020).

37. Nerger, B. A., Brun, P. T. & Nelson, C. M. Microextrusion printing cell-laden networks of type i collagen with patterned fiber alignment and geometry. *Soft Matter* **15**, 5728–5738 (2019).

38. Shin, Y. *et al.* Extracellular Matrix Heterogeneity Regulates Three-Dimensional Morphologies of Breast Adenocarcinoma Cell Invasion. *Advanced Healthcare Materials* **2**, 790–794 (2013).

39. Smith, L., Cho, S. & Discher, D. E. Mechanosensing of matrix by stem cells: From matrix heterogeneity, contractility, and the nucleus in pore-migration to cardiogenesis and muscle stem cells in vivo. *Seminars in Cell & Developmental Biology* **71**, 84–98 (2017).

40. Tambe, D. T. *et al.* Collective cell guidance by cooperative intercellular forces. *Nature Materials* **10**, 469–475 (2011).

41. Yeganegi, A., Rogers, J. & Richardson, W. J. Mechano-Regulation of Fibrillar Collagen Turnover by Fibroblasts. in *Mechanobiology Handbook* (2018).

CHAPTER THREE

Characterization of Spatial Heterogeneity in Engineered Heart Tissue Mechanics Following Simulated Infarction

3.1. Introduction

Coronary occlusions restrict blood flow to the downstream tissue leading to a myocardial infarction event. Local cell populations downstream from the occlusion begin to necrose and produce danger-associated molecular patterns(DAMPs) and other cytokine signals that activate an inflammatory cascade within the necrosing tissue segment.[1–4] Infiltrating inflammatory cells release matrix degrading molecules, such as matrix-metalloproteases(MMPs), in their effort to remove damaged and dying tissue components as well as additional cytokines encouraging activation and chemotaxis of neighboring fibroblast populations.[5–9] During their migration, fibroblasts further degrade the underlying matrix in their efforts to penetrate the wounded zone where they will deposit replacement and supplemental extracellular matrix components, mainly fibrillar collagen.[10,11]

This is the beginning of the collagenous scar structure that forms to maintain the integrity of the myocardial wall following an infarction event. Collagen content and maturation via hydroxy-lysyl cross-linking are closely linked to resulting myocardial wall-mechanics.[12–16] After maturation of the collagenous material has occurred, the infarcted region enters what is known as the remodeling phase. Embedded fibroblast populations begin adjusting their localized matrix substrate via degradation, deposition,

and repositioning. This remodeling is typically believed to be directed by the underlying mechanics detected by the embedded cellular agents.[17–19]

It is well established that fibroblasts respond differentially to varied substrate stiffnesses. Responses include up or downregulation of actin cytoskeletal state, cell morphology, ECM production, and signal or receptor molecule expression levels among others.[20–27] In addition to stiffness, unique binding moieties presented by different subtypes of ECM proteins can alter activation in fibroblast populations.[28,29] These binding moieties can be revealed following ECM breakdown by matrix-metalloproteases. Interestingly, MMP expression is yet another factor that has been suggested to vary based upon substrate stiffness.[30] Clearly substrate stiffness plays a significant role in cellular response to surrounding environments; however, constituent cells are governed by a large swathe of the environmental stimuli, including the dynamic nature of the mechanics. [29,31–34] Taken together, the overall mechanics of the surrounding environment play an important role in governing cellular response. Disruptions to the environmental mechanics such as those that occur throughout the complicated cascade following an infarction event are almost certainly going to play a role in a resultant pathology development.

Worsened outcomes following an infarction can generally be described by reductions in cardiac pump efficiency attributed to myocardial wall overcompliance or lack thereof. Following the initial inflammatory cascade, inflammatory cells can also infiltrate neighboring healthy tissue where they continue their process of cytokine release, matrix degradation, and removal of cellular and matrix components.[35] As this excessive

response continues, the myocardial wall is further weakened as structural integrity from surviving embedded cells and matrix architecture is removed unnecessarily.[36–38] This weakening of the wall changes the mechanics and drastically alters the projected course of the post-infarction recovery phases to follow.

On the other hand, excessive deposition of collagenous matrix too rapidly, or excessive cross-linking of the deposited collagen fibers, can lead to a myocardial wall that is overly stiff and noncompliant.[39,40] As the core function of the heart is to successfully pump blood throughout the vast vessel network of the body, any reduction in functionality is dire and begins down the path towards eventual heart failure.

As discussed here, there are numerous phases involved that require careful balancing to achieve an optimal outcome following a myocardial infarction event. Within each phase, there are various factors can tilt the overall response towards improved or worsened outcomes via underlying myocardial wall mechanics. Clear relationships between cardiac fibroblasts and environmental stiffness and dynamic mechanics have been established. Additionally, cardiac fibroblasts are critical to the physiological response that governs the trajectory of myocardial wall mechanics following an infarction event. A better understanding of myocardial mechanics and the resultant cell response during the development and progression of these pathophysiological states is critically needed to enable development of effective therapies and treatment regimens.

In order to evaluate these topics, we have developed an in vitro co-culture engineered heart tissue platform within our lab group visualized in Fig. 3.1. This platform consists of a fibrin-based hydrogel with embedded cardiofibroblast and cardiomyocyte

populations which are subjected to controlled electrical and mechanical stimulations. In this study, we subjected these engineered-heart tissue constructs to simulated infarctions via cryo-probe application and evaluated the temporal changes in intra-construct regional and macro-mechanics that developed as a result of the injury and the embedded cell population response.

3.2. Methods

3.2.1. Cell-Sourcing, Culture, and Fibrin Construct Fabrication

Primary cardiofibroblasts and cardiomyocytes were isolated directly from neonatal Sprague Dawley rats in accordance with IACUC approved protocol. Cardiac ventricles were separated from atria, minced, and stored within a 0.25% trypsin PBS solution overnight at 4°C. The following day, minced tissue was subjected to a brief one-minute water bath incubation with trypsin inhibitor solution. An L-15 media-based collagenase solution was then added to the tissue containing solution and placed on a rotator for a forty-minute incubation. Following this incubation, the tissue was broken apart via trituration and then filtered through a 70μm strainer to remove remaining tissue debris.

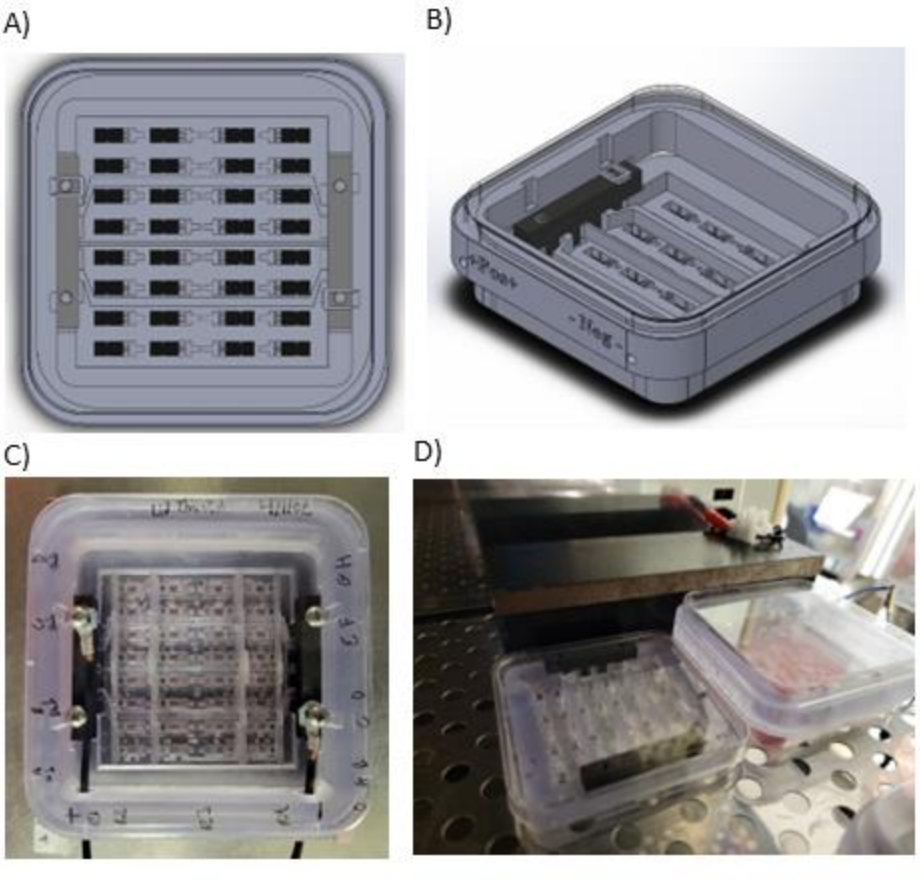

Fig. 3.1 Culture Platform. A) Top-down view of the culture platform CAD model showing the well arrangement and carbon electrode positioning. B) Isometric view of platform CAD assembly C) Top-down view of assembled platform. D) View of partially and fully assembled culture platforms placed against an electromagnet.

At this point, the solution was allowed to settle for 40-minutes at room temperature before centrifugation and resuspension in 40ml of cell culture media(DMEM, 10% FBS, 1% Pen-Strep, 0.2% Ampho-B) and seeding into two T-175 flasks. Cells were allowed to settle within the flasks for two hours at which point the flasks were lightly rinsed with media approximately ten times and the suspension was transferred into a fresh 50ml conical tube. It is this transferred solution that contained

mainly cardiomyocytes while the cardiofibroblasts remained within the flask. Cell culture media at 25ml was then added to the two T175 flasks and the fibroblasts were cultured for approximately one week until utilization in the fibrin-based engineered heart tissue fabrication.

The isolated cardiomyocytes were counted and the volume appropriate to achieve the final desired fibrin gel cardiomyocyte concentration of 3.75M/ml was transferred to a 50ml conical tube. At this point, cardiofibroblasts from the previous week were trypsinized and counted followed by a transfer to the cardiomyocyte containing conical of an appropriate volume of cardiofibroblast solution to achieve a final gel fibroblast concentration of 1.25M/ml . This suspension was thoroughly mixed and pelleted via centrifugation.

The co-culture suspension was then resuspended in thrombin solution(0.44U/ml thrombin) which was mixed with an equal volume of 5.5 mg/ml fibrinogen solution(bovine fibrinogen, HEPES-BSS). This final solution was then pipetted into the individual wells of our custom-fabricated platform followed by a one-hour 37°C incubation. After this incubation, gel media(DMEM, 10% Horse Serum, 33 µg/ml aprotinin, 50 µg/ml L-Ascorbic Acid, 10 µg/ml insulin, 1% Pen-Strep, 0.2% Ampho-B) was added to the culture and changed every other day for the duration of the experiment at consistent volumes.

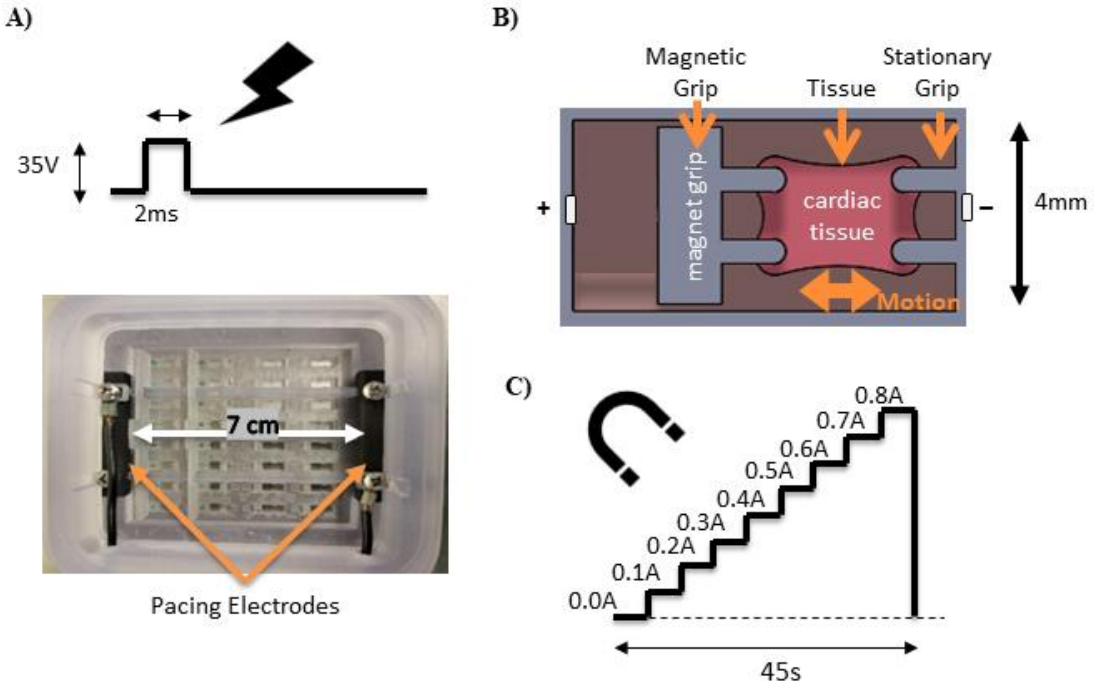

Fig. 3.2 Schematic Representation of stimulation modalities A)Illustration depicting the 2 ms unipolar 35V electrical stimulation pulse and the physical separation between the pacing electrodes within the platform. B) Illustration of the mechanical attachment of the fibrin constructs to the stationary anchor grip and the moveable magnetic grip. C)Representative schematic of the electromagnet discrete amperage ramp program utilized to stretch the engineered heart tissue constructs.

3.2.2. Culture Stimulation

Constructs were subjected to a 1 Hz electrical stimulation cadence generated by a MyoPacer Field Stimulator device(Ionoptix, Westwood, MA). The signal waveform was a positive unipolar 2 ms duration pulse at a voltage that supplied a 5V/cm potential gap between the carbon electrodes contacting the gel media schematically represented in Fig. 3.2A.

Constructs were also subjected to a consistent mechanical stretch via a generated electromagnetic field acting upon free floating construct attached to magnetic anchors depicted in Fig. 3.2B The magnetic field was generated by an

electromagnet(Cat#5684K23, McMaster-Carr, Princeton, NJ) powered via 0.3A constant current from a programmable power supply(Cat#SPD33033X-E, Siglent Technologies NA, Solon, OH). Uniaxial stretching was performed parallel to the long axis of the construct. During video acquisition, samples were mechanically stretched via discrete electromagnetic field strength step increases. This was achieved by increasing the supplied current from 0A to 0.8A by 0.1A increments. The magnet was held at each level for five seconds before advancing to the next step increase in amperage and supplying a greater pull-force onto the magnetic anchor. This discrete ramp-up regiment is schematically depicted in Fig. 3.2C.

3.2.3 Cryo-Infarct

Custom manufactured cryo-stamps were utilized to ensure a uniform cryo-wound application to each sample. The cryo-stamp contacting surface measured 3.5mm by 1mm and was applied in an orientation that generated a cryo-wound spanning the short-axis of the cell-laden fibrin gel constructs. The tips of the cryo-stamp were dipped in liquid nitrogen for 60 seconds, immediately placed onto the awaiting samples, and left on the samples for 60 seconds prior to removal. The cryo-stamp and application process are illustrated in Fig. 3.3.

3.2.4 Video Acquisition and Processing

Constructs were imaged with an EVOS FL Auto Imaging System(Life Technologies Carlsbad, CA) and videos were acquired via OBS Studio(Open Broadcaster

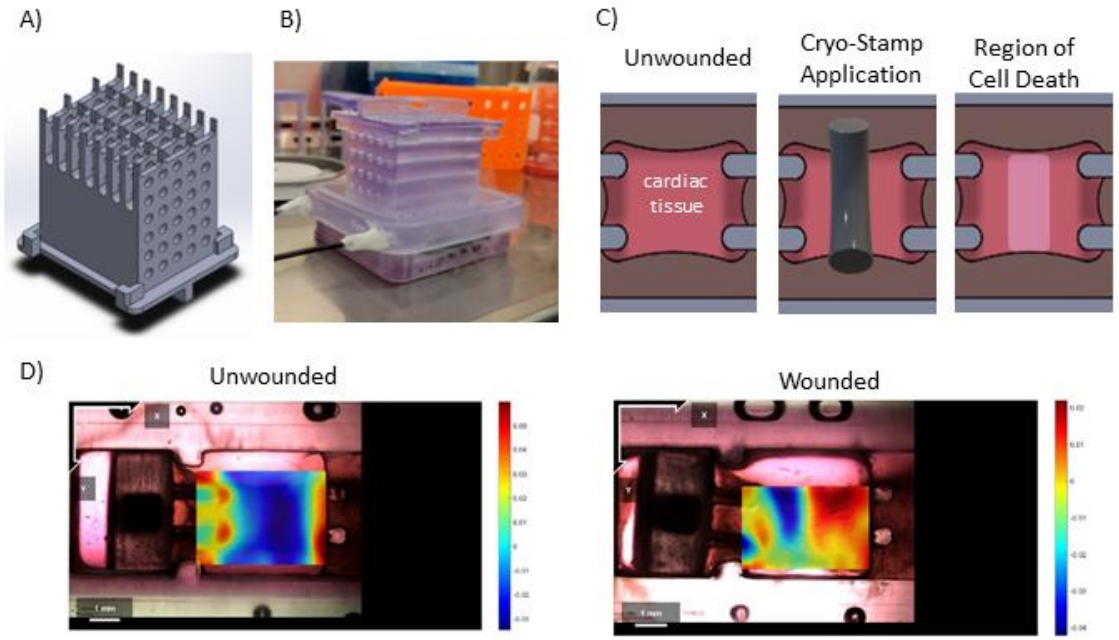

Fig. 3.3 Cryo-stamp and wounding process. A) Isometric view of cryo-stamp CAD model. B) Application of cryo-stamp to array of engineered heart tissue platform. C) Illustration of application of cryo-stamp to singular construct. D) Heatmap of construct strains occurring during an example contraction cycle prior to and following wounding.

Software) enabled screen recording of the EVOS FL Auto Imaging System display. Video files were then loaded into MATLAB(MathWorks, Natick, MA) where frames were extracted, converted to grayscale, and saved as singular JPEG image files to enable surface strain mapping analysis performed via NCORR 2D digital image correlation MATLAB software visualized in Fig. 3.3D.[41]

3.2.4 Gel Dimensions

Gel Cross-Section

Frames extracted from the portion of the video corresponding to the baseline unstretched construct were loaded into ImageJ where width measurements were taken of each gel at the two extreme ends and the middle position. Linear relationships were then constructed to interpolate discrete widths along the length of the gel occurring between the midpoint and the extreme ends of the construct. A compaction factor was also calculated by comparing the current width of the gel at the discrete points to the initial width of 4mm. The compaction factor was then used to estimate gel thickness by multiplying the original thickness of 3.25mm, determined by well height, by the compaction factor. This estimated thickness and the interpolated widths along the gel were utilized in calculating the cross-sectional area.

Force Calibration

Force values were previously experimentally generated via spring compression calibration utilizing springs with known material properties. [42]

3.2.5 Calculations

Long Axis Strain Measures

Long-axis strain values were calculated within each frame by finding the median Lagrangian strain value across the short-axis for each discrete datapoint along the long-axis of the gel construct. For each discrete calculable position along the length of the

construct, the median strain value from the surface mapped strains along the short-axis was found. Analysis was limited to the central 90% of the gel construct to avoid edge artifacts related to mechanical restraints at the terminal ends of the construct. Global mean values were calculated for each frame by averaging all of the surface strain mapped values.

Stress vs Strain

Stress values were calculated for each discrete position along the long-axis of the construct utilizing the force applied to the construct at each respective electromagnet level and the cross-sectional area calculated for each discrete position along the long-axis of the construct. Stress and strain pairings for samples of each experimental condition and timepoint were compiled and binned by the stress value of the pair into nine bins. The number of bins was selected to mirror the nine discrete force levels the constructs were subjected to during the stretch regimes. The mean and standard deviation of the contents of each bin were calculated giving nine resulting stress-strain pairings.

Macro-Modulus

The stress-strain pairs of each sample were calculated, and a linear regression was performed to attain the estimated modulus of elasticity value for each individual sample. The mean, standard deviation, and standard error were calculated for samples of each timepoint and condition.

Regional Mechanics

Constructs were segmented into sixteen subregions. Three neighboring subregions were joined to form an analysis region. Three analysis regions were selected corresponding to the infarcted region of the construct, the bordering region of the construct, and the remote region of the construct. Analysis regions were separated by three subregions. The analyses described above were performed for each analysis region separately.

Statistical Analysis

Paired t-tests were performed for all intra-condition comparisons with significance reported at $p < 0.05$. Two-sample t-tests were performed for all inter-condition comparisons with significance reported at $p < 0.05$. The unwounded control treatment group possessed a sample size of $n = 4$ for all calculations. The wounded experimental treatment group possessed a sample size of $n = 6$ for all calculations.

3.3. Results

3.3.1. Macro-Mechanics

Stress-strain relationships for each day of the timecourse were plotted as scatter plots with lines of best fit in Fig. 3.4 highlighting the general mechanics of the cell-laden fibrin constructs as a whole. Both conditions, control and wounded, exhibited a progressive strengthening stress-strain relationship after day 0 observations. The slope of the line of best fit for the control condition clearly increased from day 1 to 7. As the slope

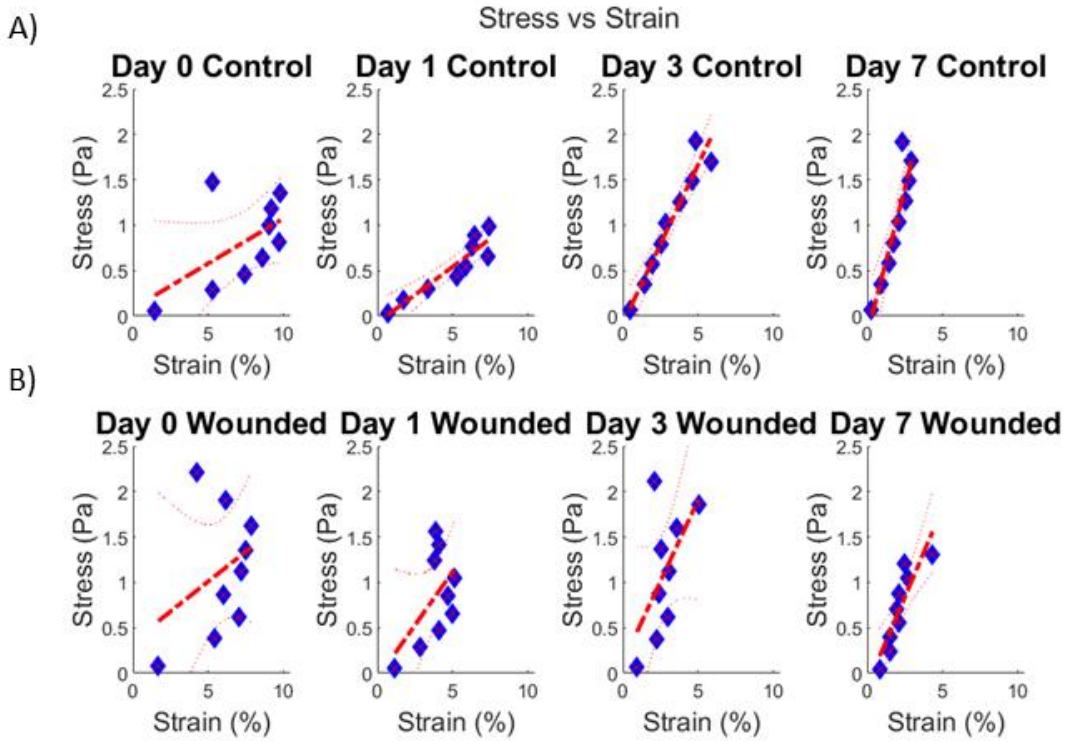

Fig. 3.4 Average binned stress vs. strain relationship of cell-laden fibrin gels. A) Unwounded control gels show a strengthening of the stress-strain relationship over the timecourse as well as a progressively increasing slope of the line of best fit. (n = 4). B) Stress-strain relationship of gels subjected to cryo-wounding after day 0 show a progressive strengthening of the relationship while the slope of the line of best fit is approximately maintained.(n = 6)

increases, the maximum strain value from the plot is reduced from an initial day 0 value near 10% to a day 7 value under 5% strain. Wounded gels exhibit a stress-strain relationship that appears to linearize without the line of best fit noticeably increasing in slope.

When evaluating the modulus of elasticity, there is a clear trend of stiffening within the unwounded control group that the wounded group does not exhibit. While the maximum modulus values of the wounded group shown in the box plots of Fig. 3.5A do

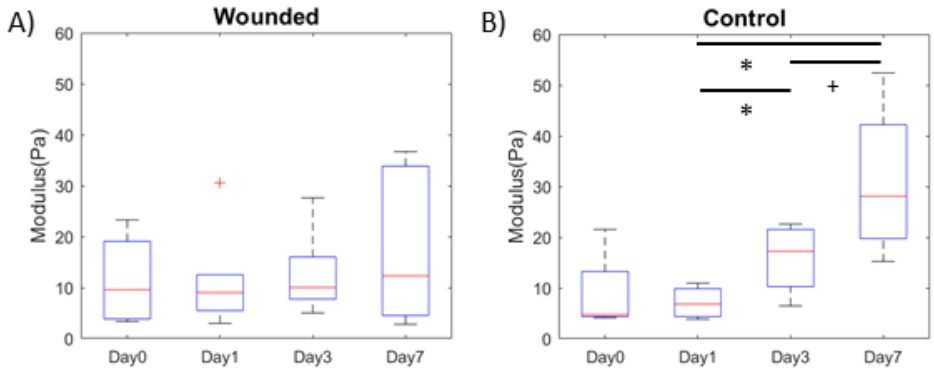

Fig. 3.5 Macro-Modulus of Wounded and Unwounded Gels. A) Box plot of wounded construct modulus shows a relatively stable modulus throughout the timecourse with day 7 values exhibiting a greater range within the dataset(n = 6). B) Box plots of unwounded control gels highlighting sample stiffening over time. (*: $p<0.05$;+$p<0.1$ | n = 4).

appear to increase day to day starting after wounding on day 1, there is no statistical significance observed between groups. The unwounded control group highlighted in Fig. 3.5B on the other hand exhibited significant levels of stiffening from day 1 to day 3 and 7. While stiffening continued from day 3 to 7, it was not as drastic as the difference observed from day 1 to 3 as exhibited by a p-value falling between 0.05 and 0.1. When comparing modulus of the control and wounded groups directly at each day, no significant differences are reported. The wounded gels started at a higher modulus than the control group counterparts, and while there was no significance, the direct comparison highlights the existing variance within the production of these fibrin constructs. By day 7, the average modulus of the unwounded group is beginning to separate itself from the wounded constructs.

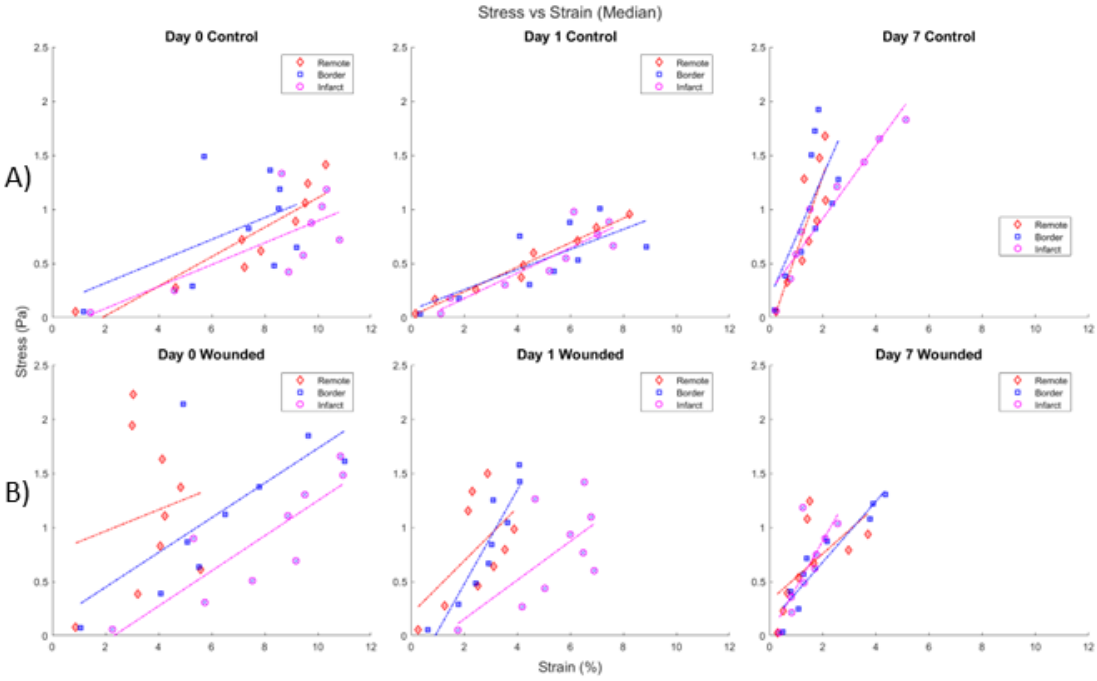

Fig. 3.6 Scatter plots displaying the average binned stress-strain relationships of regional zones within the fibrin gel constructs. A) The stress-strain relationship of the remote, border, and wounded regions of the unwounded control gels highlight a lack of divergence in behavior. The strengthening of the scatter pattern and the increase in the slope of the line of best fit exhibited by the macro-mechanics stress-strain relationship is again repeated within the analysis regions of the unwounded constructs(n = 4). B) Wounded constructs exhibit greater variance on day 0 with the remote and border zones exhibiting a convergence of their stress-strain patterns on day 1. The wounded zone lags behind. All three zones of interest show pattern convergence again by day 7 just as exhibited by the macro-mechanics(n = 6).

3.3.2. Regional-Mechanics

Breaking the construct into regions of interest yields greater insights into the response following cryo-wounding. Again, regional scatters shown in Fig. 3.6 follow similar patterns of a strengthening stress-strain relationship exemplified by a comparison of day 0 and day 7 spreads between both treatment groups. The control gels shown in Fig. 3.6A again exhibit a strengthening relationship by day 1 and a clear increase in the slope of the line of best fit for all regions by day 7. The spreads of the stress-strain relationships of each individual region within the control group are nearly indistinguishable. Wounded

gels shown in Fig. 3.6B show greater variance on initial day 0 fabrication. This variance seems to mostly be attributed to the remote region while the soon to be wounded and border zones exhibit more traditional relationships. Day 1 again shows a tightening between the border and remote stress-strain relationships; however, the wounded zone clearly lags behind in elevating the slope of the line of best fit. This was not discernible from evaluating the constructs as a whole. Within Fig. 3.4, it simply appeared that the wounded gels maintained their original stress-strain relationship. It appears that in fact the wounded zone is likely to blame for this macro-result. Again, by day 7, all regions within the wounded constructs have had their stress-strain relationships converge.

Intra-day regional comparisons of resultant moduli shown in Fig. 3.7 reveal a mostly uniform unwounded control construct throughout the timecourse that only exhibits trends towards significance between neighboring regions on the ultimate day of evaluation. While the wounded group also exhibits an apparent near uniformity of modulus among regions initially, clear differences arise after wounding. By day 1 post-wound, the remote and border regions both exhibit significantly greater stiffness than the wounded region. On day 3, significance persists between the stiffness of the remote and the wounded regions while the disparity between the bordering region and the wound is reduced to near significance. On the ultimate day of evaluation, the regions of the wounded gel have again reached statistically indistinguishable levels of stiffness.

Within the unwounded control condition, regions show little variability throughout timecourse until day 7 where the remote region appears to be stiffening to a greater value. However, all regions increase in stiffness through day 3 and 7 compared to

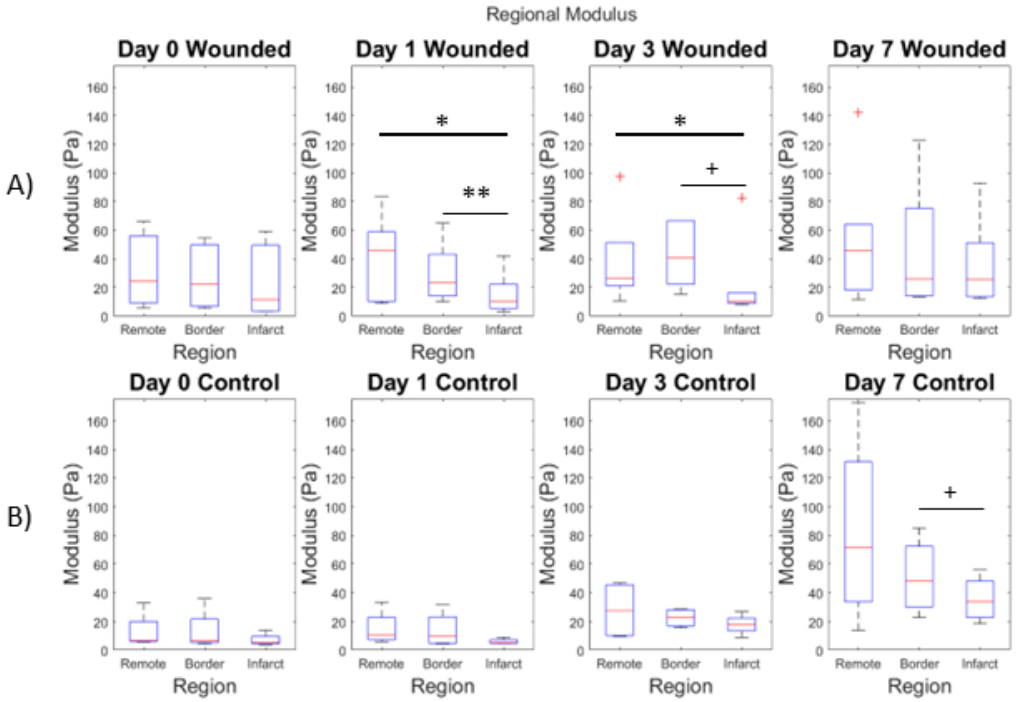

Fig. 3.7 Box plots of regional moduli. A) Wounded gel regional moduli exhibit stiffness disparity within the intermediate range of the experimental timecourse that is lost by the ultimate day. (**: $p<0.01$; *: $p<0.05$; +: $p<0.1$ | $n = 6$). B) Unwounded control gel regional moduli exhibit uniform stiffness until the ultimate day of the timecourse where a relationship approaching significance is exhibited between the labeled wound region and the border region. (+: $p<0.1$ | $n = 4$).

initial values. Experimental values are shown to be undifferentiable initially as well, albeit at higher initial stiffnesses than the control group. Day 1 measurements appear to already form a step-down pattern of stiffness with the remote region possessing the greatest stiffness, border zone possessing an intermediate level, and stepping further down to the infarcted region shows the most compliance. By day 3, the remote and border region are statistically indistinguishable. The remote region is statistically stiffer than the infarcted region, and the bordering region has been reduced to near significance. By the end of the timecourse, the disparities in stiffness have been alleviated.

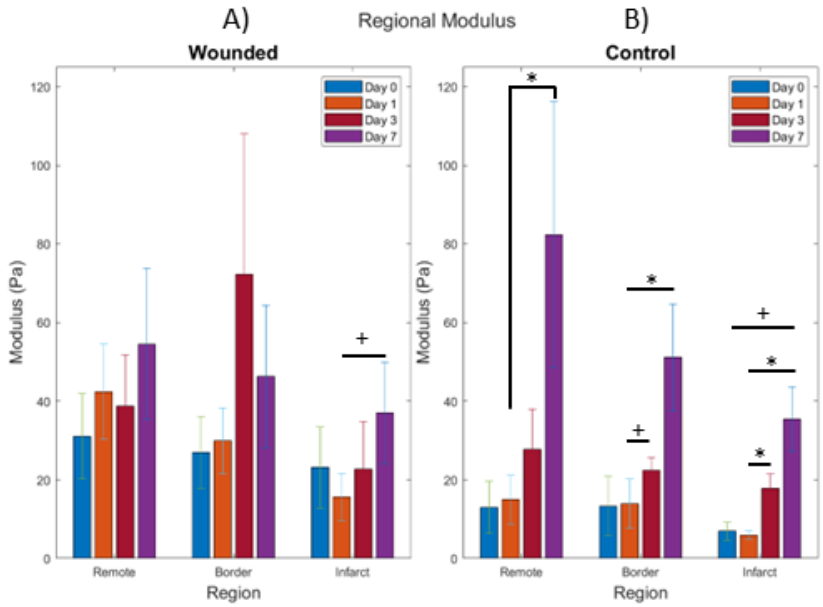

Fig. 3.8. Progression of regional modulus through the timecourse. A) Wounded gels exhibited non-significant changes throughout the timecourse.(+:$p<0.1$ | n = 6). B) Unwounded control gel regions exhibited significant increases in modulus throughout the timecourse within each region with all regions displaying significant stiffening from day 1 to day 7. (*: $p<0.05$; +: $p<0.1$ | n = 4).

The modulus change through the experiment of each region is displayed in Fig. 3.8. There exists a general trend of stiffening over time with both wounded and unwounded constructs. However, the only region exhibiting any near level of significance between days within the wounded gels is the wounded region. This occurs when comparing the day 7 modulus value to the day 1 value after wounding. The other regions within the wounded gel generally increased in stiffness; however, it was not at a statistically significant level. This is in contrast to the unwounded group which exhibited some degree of significance within all regions analyzed. The modulus of each unwounded region on day 7 was greater than the day 1 values for each. The wounded and border regions of the control gels also exhibited significantly different moduli from day 1 to day 3. These

regional differences align with the macro-moduli of the treatment groups with the regional breakdown of the wounded constructs providing greater insight confirming that the wounded region appeared to be the most impacted by the cryo-wound application.

3.4. Discussion

The healing infarct is without question a very complicated subject. It is a tumultuous environment with dynamic mechanics, varied geometries, spatial inconsistency, and overall uncertainty. Therefore, simple, effective, and consistent therapeutic strategies evade discovery and enactment. Clarke et al presented a review of literature that highlighted that experimental models of myocardial infarction vary wildly in collagen structure, mechanics, and remodeling. The authors concluded that it is difficult to ascertain possible therapeutic efficacy without testing across a range of species, infarct locations, and reperfusion protocols due to the inherent complicated biology and mechanics of the healing infarct that are often counterintuitive and extremely complex.[43] This conclusion was supported by a separate finite element analysis(FEA) study that investigated the effect that infarct location had on traditional functional output measures of cardiac efficiency. It was concluded that tradeoffs exist between therapeutic options of interest and that infarct location likely plays a significant role in what therapeutic route would most appropriately induce recovery in an infarcted myocardium.[44] Additionally, Richardson and Holmes investigated why there appears to be discrepancies between whether in-plane compaction or expansion were the likely outcomes of infarct remodeling. FEA methodologies were utilized to explore these

contradictory and possibly unintuitive findings. The authors found that that there is a careful balancing act between collagen accumulation and infarct thinning that can veer towards either infarct expansion or compaction and that changes in thickness are more likely than simple stiffening.[45] On this note, the presence of matrix components has been used in identifying patients with certain heart failure modalities, underscoring the importance of matrix turnover in the recovering myocardium. [46] These results suggest that again the healing infarct is a very complicated subject matter and that a simpler, more consistent, and more widely applicable methodology of evaluating therapeutic efficacy is desperately needed. These results in part inspired the experimental analysis conducted within this study as a preliminary investigation into the natural progression of a simplified engineered cardiac tissue construct following myocardial injury. As literature findings clearly demonstrate the importance of wall mechanics, investigating the progression of the stress-strain relationship and regional stiffness found within our mechanically active constructs is an important contribution towards the development of an improved understanding of the progression of mechanics within the early infarct.

The stress-strain relationships highlighted in Fig. 3.3 show the regions of the control groups are clustered together throughout the timecourse signifying no regional variance. This is contrasted by the experimental infarcted group. While there is seemingly more of a spread initially, the border and remote regions appear to converge in their stress-strain relationships over the time course. The infarcted region lags behind in the upward trend of the slopes on day 1 prior to rejoining the values at day 7. This observation is further confirmed via the box plots of Fig. 3.7. The regions within each

experimental condition started at very similar and almost uniform levels. While the control condition maintained this relationship through day 3, the experimental group showed a disparity in stiffness between the infarcted region and the bordering and remote regions. The delay in stiffening of the wounded tissue regions is likely due to the death of the cells within the wound zone. The unwounded zones appear to increase in stiffness on day 1 while an acellular zone remains similar to its previous values. The recovery by day 7 is expected to be linked to fibroblast populations that have migrated into the wound zone and begun the stiffening process. Curiously, by day 7, the remote region of the control group seems to be stiffening to a greater degree than the 'wound' region of the control group. This is suspected to be related to the remote region laying closer to the sample anchor of the platform. While the wound was generated near the center of the construct, the analysis regions were chosen on one side of the wound to avoid any analysis artifacts resulting from differential mechanics due to one side being stationarily anchored and the other being anchored to the freely-moving piston head driven by the electromagnet. It was assumed that the piston head may generate more artifacts that the surface strain mapping software may demonstrate trouble with analyzing.

When comparing the modulus progression of a singular region, the wounded constructs appeared to generally stiffen, yet not at a level reaching statistical significance. The wounded region of the wounded construct highlighted in Fig. 3.8 appears to decrease in modulus after application of the cryo-probe. This downshift is not shared by the other regions of the wounded gels and may be a result of some slight mechanical disruption that occurs during the wound application. In contrast to the regions of the wounded

constructs, the regions of the unwounded control constructs exhibited clear increases in modulus values supported by statistical analysis. This suggests that while the most prominent effects were observed within the wounded region, the interconnectedness of the regions clearly resulted in a disruption of the stiffening process of the remaining regions as well. This could be related to the application of the cryo-probe itself exerting damage upon embedded cell populations extending past the wound zone. It could also be related to the fibroblast populations migrating into the wounded region in the following days, indirectly reducing the local stiffening capabilities. Regardless, regional differences in mechanical properties appear to exist within the fibrin constructs introduced by wound generation that persist through the timecourse and affect the holistic mechanical evaluation of the affected gel.

Sample to sample variation within the sample fabrication process can increase the uncertainty in direct value to value comparisons across treatment groups. This variation is believed to be related to the prototype nature of the technology platform. Room for refinement within the protocols exists with some example targets including: thrombin concentration during gelation, final fibrin gel density, cellular density, myocyte to fibroblast ratio, and electrical and mechanical stimulation regimes. Due to this variation, it is more difficult to derive certain answers when comparing values of specific discrete timepoints within the experimental infarcted group to the control group. However, analyzing the trends over time within the specified treatment group and comparing those trends to that of the control group can provide insights into how the infarcted samples responded to the initial wounding over the following week. When analyzing the gels

holistically, the control groups follow a gradual stiffening over the timecourse while the experimental gels appear to maintain a similar stiffness throughout, only broken up by an increased value range on the ultimate day. These observations seem to suggest that while the wounding did not disrupt the architecture of the underlying gel, wounding is responsible for the disruption of the stiffening process. While evaluating the overall stiffening trends is useful and provides a test bed with parallels to that of the stiffening infarct, gaining insights into the spatial heterogeneity of mechanics from within the infarct to the remote tissue can elucidate the finite and incremental nuance of the recovery process.

3.5. Conclusion

The events following ischemia within the myocardial wall play a large role in determining which branch of the pathological development tree will be taken towards heart failure. Following these events, mechanics are suspected to play an outsized role in directing cellular response and either the exacerbation of myocardial wall weakening or stiffening. Due to the importance of mechanics within these processes, it is important to not only evaluate tissue mechanics, but to evaluate the spatial heterogeneity found within the tissue of note. We generated a platform capable of sustained culture of contractile pacing cell-laden fibrin constructs. The constructs contained a cardiomyocyte and cardiofibroblast co-culture. A simulated infarct zone of cell death was generated via cryo-probe wounding. The regional progression of construct mechanics was evaluated over a week-long timecourse. It was determined that while unwounded gels followed a

stiffening trajectory, the cryo-probe disrupted the stiffening of the wounded constructs via removing cells from the wound region. Fibroblast populations later migrated into the wounded region and stiffened the region to a degree that matched the surrounding regions; however, this only returned the modulus of the entire gel to that of the original unwounded gel. The regional differences revealed within the wounded constructs highlight the importance of taking into account the spatial heterogeneities of mechanics within the recovering wound environment. Evaluating the construct as a whole conveyed that the modulus remained essentially consistent across the seven days. Once the construct was observed with greater resolution, it was seen that the non-wounded regions were stiffening slightly while the wound region was maintained or softening before recovering by day 7.

This platform and analysis methodology offers promise for future studies that will expand upon the groundwork performed here. Experiments with varied stimulation regimens, cytokine supplementation, and complex wound patterns are all possibilities that will yield specific insights into the fundamentals of the interplay between the spatial heterogeneities of construct mechanics and cellular response.

3.6. References

1. Patel, S. Danger-Associated Molecular Patterns (DAMPs): the Derivatives and Triggers of Inflammation. *Curr Allergy Asthma Rep* **18**, 63 (2018).
2. van Hout, G. P., Arslan, F., Pasterkamp, G. & Hoefer, I. E. Targeting danger-associated molecular patterns after myocardial infarction. *Expert Opin Ther Targets* **20**, 223–239 (2016).
3. Frangogiannis, N. The inflammatory response in myocardial infarction. *Cardiovasc Res* **53**, 31–47 (2002).
4. Christia, P. & Frangogiannis, N. G. Targeting inflammatory pathways in myocardial infarction. *Eur J Clin Invest* **43**, 986–995 (2013).
5. Iyer, R. P., Jung, M. & Lindsey, M. L. MMP-9 signaling in the left ventricle following myocardial infarction. *American Journal of Physiology-Heart and Circulatory Physiology* **311**, H190–H198 (2016).
6. Etoh, T. *et al.* Myocardial and interstitial matrix metalloproteinase activity after acute myocardial infarction in pigs. *American Journal of Physiology-Heart and Circulatory Physiology* **281**, H987–H994 (2001).
7. Fix, C., Bingham, K. & Carver, W. Effects of interleukin-18 on cardiac fibroblast function and gene expression. *Cytokine* **53**, 19–28 (2011).
8. Bujak, M. *et al.* Induction of the CXC Chemokine Interferon-γ-Inducible Protein 10 Regulates the Reparative Response Following Myocardial Infarction. *Circ Res* **105**, 973–983 (2009).
9. Ma, Y., Iyer, R. P., Jung, M., Czubryt, M. P. & Lindsey, M. L. Cardiac Fibroblast Activation Post-Myocardial Infarction: Current Knowledge Gaps. *Trends Pharmacol Sci* **38**, 448–458 (2017).
10. Samuel, C. S. *et al.* Relaxin Modulates Cardiac Fibroblast Proliferation, Differentiation, and Collagen Production and Reverses Cardiac Fibrosis in Vivo. *Endocrinology* **145**, 4125–4133 (2004).
11. Brilla, C. G., Zhou, G., Rupp, H., Maisch, B. & Weber, K. T. Role of angiotensin II and prostaglandin E2 in regulating cardiac fibroblast collagen turnover. *Am J Cardiol* **76**, 8D-13D (1995).
12. Cai, L., Xiong, X., Kong, X. & Xie, J. The Role of the Lysyl Oxidases in Tissue Repair and Remodeling: A Concise Review. *Tissue Eng Regen Med* **14**, 15–30 (2017).
13. Brower, G. L. *et al.* The relationship between myocardial extracellular matrix remodeling and ventricular function☆. *European Journal of Cardio-Thoracic Surgery* **30**, 604–610 (2006).
14. Thiedemann, K.-U., Holubarsch, Ch., Medugorac, I. & Jacob, R. Connective tissue content and myocardial stiffness in pressure overload hypertrophy A combined study of morphologic, morphometric, biochemical, and mechanical parameters. *Basic Res Cardiol* **78**, 140–155 (1983).
15. Matsubara, L. S., Matsubara, B. B., Okoshi, M. P., Cicogna, A. C. & Janicki, J. S. Alterations in myocardial collagen content affect rat papillary muscle function. *American Journal of Physiology-Heart and Circulatory Physiology* **279**, H1534–H1539 (2000).

16. Klotz, S. *et al.* Mechanical Unloading During Left Ventricular Assist Device Support Increases Left Ventricular Collagen Cross-Linking and Myocardial Stiffness. *Circulation* **112**, 364–374 (2005).
17. Adapala, R. K., Kanugula, A. K., Paruchuri, S., Chilian, W. M. & Thodeti, C. K. TRPV4 deletion protects heart from myocardial infarction-induced adverse remodeling via modulation of cardiac fibroblast differentiation. *Basic Res Cardiol* **115**, 14 (2020).
18. MacKenna, D. Role of mechanical factors in modulating cardiac fibroblast function and extracellular matrix synthesis. *Cardiovasc Res* **46**, 257–263 (2000).
19. Baxter, S. C., Morales, M. O. & Goldsmith, E. C. Adaptive Changes in Cardiac Fibroblast Morphology and Collagen Organization as a Result of Mechanical Environment. *Cell Biochem Biophys* **51**, 33–44 (2008).
20. Herum, K. M., Choppe, J., Kumar, A., Engler, A. J. & McCulloch, A. D. Mechanical regulation of cardiac fibroblast profibrotic phenotypes. *Mol Biol Cell* **28**, 1871–1882 (2017).
21. Jones, C. & Ehrlich, H. P. Fibroblast expression of α-smooth muscle actin, α2β1 integrin and αvβ3 integrin: Influence of surface rigidity. *Exp Mol Pathol* **91**, 394–399 (2011).
22. van Putten, S., Shafieyan, Y. & Hinz, B. Mechanical control of cardiac myofibroblasts. *J Mol Cell Cardiol* **93**, 133–142 (2016).
23. Hinz, B., McCulloch, C. A. & Coelho, N. M. Mechanical regulation of myofibroblast phenoconversion and collagen contraction. *Exp Cell Res* **379**, 119–128 (2019).
24. Hinz, B. Tissue stiffness, latent TGF-β1 Activation, and mechanical signal transduction: Implications for the pathogenesis and treatment of fibrosis. *Curr Rheumatol Rep* **11**, 120–126 (2009).
25. Galie, P. A., Westfall, M. v. & Stegemann, J. P. Reduced serum content and increased matrix stiffness promote the cardiac myofibroblast transition in 3D collagen matrices. *Cardiovascular Pathology* **20**, 325–333 (2011).
26. Huang, X. *et al.* Matrix Stiffness–Induced Myofibroblast Differentiation Is Mediated by Intrinsic Mechanotransduction. *Am J Respir Cell Mol Biol* **47**, 340–348 (2012).
27. Cho, N., Razipour, S. E. & McCain, M. L. Featured Article: TGF-β1 dominates extracellular matrix rigidity for inducing differentiation of human cardiac fibroblasts to myofibroblasts. *Exp Biol Med* **243**, 601–612 (2018).
28. Wang, X. *et al.* Exogenous extracellular matrix proteins decrease cardiac fibroblast activation in stiffening microenvironment through CAPG. *J Mol Cell Cardiol* **159**, 105–119 (2021).
29. Watson, C. J. *et al.* Extracellular matrix sub-types and mechanical stretch impact human cardiac fibroblast responses to transforming growth factor beta. *Connect Tissue Res* **55**, 248–256 (2014).
30. Xie, J. *et al.* Substrate stiffness-regulated matrix metalloproteinase output in myocardial cells and cardiac fibroblasts: Implications for myocardial fibrosis. *Acta Biomater* **10**, 2463–2472 (2014).
31. Carver, W., Nagpal, M. L., Nachtigal, M., Borg, T. K. & Terracio, L. Collagen expression in mechanically stimulated cardiac fibroblasts. *Circ Res* **69**, 116–122 (1991).

32. Ruwhof, C., van Wamel, A. E. T., Egas, J. M. & van der Laarse, A. Cyclic stretch induces the release of growth promoting factors from cultured neonatal cardiomyocytes and cardiac fibroblasts. *Mol Cell Biochem* **208**, 89–98 (2000).
33. Rogers, J. D., Holmes, J. W., Saucerman, J. J. & Richardson, W. J. Mechano-chemo signaling interactions modulate matrix production by cardiac fibroblasts. *Matrix Biol Plus* **10**, 100055 (2021).
34. Atance, J., Yost, M. J. & Carver, W. Influence of the extracellular matrix on the regulation of cardiac fibroblast behavior by mechanical stretch. *J Cell Physiol* **200**, 377–386 (2004).
35. Lee, W. W. *et al.* PET/MRI of Inflammation in Myocardial Infarction. *J Am Coll Cardiol* **59**, 153–163 (2012).
36. Krishnamurthy, P. *et al.* IL-10 Inhibits Inflammation and Attenuates Left Ventricular Remodeling After Myocardial Infarction via Activation of STAT3 and Suppression of HuR. *Circ Res* **104**, (2009).
37. Zaidi, Y. *et al.* Chronic *Porphyromonas gingivalis* lipopolysaccharide induces adverse myocardial infarction wound healing through activation of $CD8^+$ T cells. *American Journal of Physiology-Heart and Circulatory Physiology* **321**, H948–H962 (2021).
38. Seropian, I. M., Toldo, S., van Tassell, B. W. & Abbate, A. Anti-Inflammatory Strategies for Ventricular Remodeling Following ST-Segment Elevation Acute Myocardial Infarction. *J Am Coll Cardiol* **63**, 1593–1603 (2014).
39. Zile, M. R. & Baicu, C. F. Biomarkers of Diastolic Dysfunction and Myocardial Fibrosis: Application to Heart Failure with a Preserved Ejection Fraction. *J Cardiovasc Transl Res* **6**, 501–515 (2013).
40. Dougherty, A. H., Naccarelli, G. v., Gray, E. L., Hicks, C. H. & Goldstein, R. A. Congestive heart failure with normal systolic function. *Am J Cardiol* **54**, 778–782 (1984).
41. Blaber, J., Adair, B. & Antoniou, A. Ncorr: Open-Source 2D Digital Image Correlation MATLAB Software. *Exp Mech* **55**, 1105–1122 (2015).
42. Coeyman, S. J. *IN VITRO BIOREACTOR FOR MECHANICAL CONTROL AND CHARACTERIZATION OF TISSUE CONSTRUCTS*. (2022).
43. Clarke, S. A., Richardson, W. J. & Holmes, J. W. Modifying the mechanics of healing infarcts: Is better the enemy of good? *Journal of Molecular and Cellular Cardiology* Preprint at https://doi.org/10.1016/j.yjmcc.2015.11.028 (2016).
44. Fomovsky, G. M., Macadangdang, J. R., Ailawadi, G. & Holmes, J. W. Model-Based Design of Mechanical Therapies for Myocardial Infarction. *J Cardiovasc Transl Res* **4**, 82–91 (2011).
45. Richardson, W. J. & Holmes, J. W. Why Is Infarct Expansion Such an Elusive Therapeutic Target? *Journal of Cardiovascular Translational Research* Preprint at https://doi.org/10.1007/s12265-015-9652-2 (2015).
46. Ward, M. *et al.* Ensemble machine learning model identifies patients with HFpEF from matrix-related plasma biomarkers. *American Journal of Physiology-Heart and Circulatory Physiology* **322**, H798–H805 (2022).

CHAPTER FOUR

Correlating Mechanosensor Expression and Myocardial Scar Mechanics In Vivo

4.1. Introduction

It is estimated that mechanics play a significant role in guiding post-infarction remodeling. Previously, in-depth studies have been conducted highlighting the importance of mechanics in the progression and remodeling of infarct scars.[1] It is inferred that these mechanics are translated to the cellular level and detected by the embedded fibroblast populations which then remodel their surrounding environment in relation to the detected mechanics. But unfortunately, it is unclear what particular mechanical stimuli (stress vs. strain vs. stiffness, circumferential vs. radial vs. longitudinal, acute vs. chronic changes, etc.) drive changes in different mechanotransduction signaling responses.

Previously, spatial heterogeneities of matrix structures in healing rat infarcts have been revealed. Computational models were applied to reveal the possible sources of this heterogeneity.[2] Within these models, similar heterogeneity emerged from initially isotropic fiber beds when fibroblast agents were enabled to detect surrounding fiber orientations at a distance as well as deposit fibers aligned with their own orientation. The detection ‘at a distance’ is of particular interest and suggests there are mechanics-based feedback mechanisms at play in fibroblast mediated generation of spatial heterogeneities. Studies have shown links between underlying mechanics, integrin mediated linkages, and fibroblast activation, migration, and extracellular matrix interactions further proving the significance of these mechanics and necessitating the development of a deeper

understanding.[3–8] Concerning cardiomyocytes, mechanotransduction is integral to a range of aspects including but not limited to heart development, signaling cascades, discrete structural component formation, cardiac organization and myofibril alignment.[8–11]

Considering this detection of mechanics is implicated in both cardiac fibroblast and myocyte populations, many studies have been conducted to elucidate specific molecular players involved in their mechanical activation and interplay. Molecules implicated in cardiac cell mechanotransduction pathways and response include: FAK,[12] integrin-ß1,[13] N-cadherin,[14] various ion-channels,[15,16] MAPK pathway,[17,18] g-proteins,[19] and ASMA[20] among others. More specifically, integrin-ß1 is essential to cell-ECM linkage and appears to have key roles in cytoskeletal arrangements, adrenergic signal response, and cardiofibroblast-ECM mediated interactions with cardiomyocytes.[3,7,8,21,22] While integrin-ß1 intermediates cell-ECM interactions, N-cadherin plays a major role in direct cell-cell mechanical communication and signal transduction. Factors such as reduced growth factor expression, slower conduction speed within cardiomyocytes, mechanical coordination, and degree of cellular alignment have been attributed to n-cadherin expression and functionality.[23–26]

The intricacies of this process elude understanding due to the nature of monitoring and evaluation of the remodeling process in situ. Therefore, as a first step in bridging this knowledge gap, it would be beneficial to provide a histochemical analysis highlighting the presence of known classical mechanotransduction markers and spatially correlate them with known strains acquired while the heart was still beating at varying stages of remodeling. A collaborator has performed four-dimensional ultrasound with three-

dimensional strain mapping to characterize in vivo left ventricular mechanics within and around a murine infarct over 28 days. The three-dimensional myocardial strain maps were reconstructed to show strain profiles from intra-infarct to remote regions.[27] We were provided strain data and paraffin-embedded samples from this study collected on the 28th day to analyze.

As a first step in bridging the knowledge gap concerning the in vivo relationship between mechanosensor expression and strain, we immunofluorescently stained for mechanosensors of interest, integrin-ß1 and n-cadherin, within the collaborator provided murine cardiac infarcts and spatially correlated signal presence to collaborator provided regional strain data collected.

4.2. Methods

4.2.1. Animal Model of myocardial infarction

Collaborator experimental protocols are described in Soepriatna et al.[27] A C57BL/6J mouse model(The Jackson Laboratory, USA) was utilized and observed via longitudinal ultrasound. Samples provided for this study fell into two experimental groups: sham(SHAM: n = 1) and myocardial infarction(MI: n = 3). Myocardial infarctions were generated via either permanent ligation or ischemia-reperfusion. Average strains for each experimental group corresponding to standardized myocardial segments described in Cerqueira *et al* were provided. [28]

4.2.2. Immunohistochemical Analyses

For immunohistochemical analysis, sections were generated from formalin-fixed, paraffin-embedded specimens, subjected to trypsin-based antigen retrieval, incubated with primary antibodies anti-integin-ß1(CAT#: MAB2405, Novus Biologicals, USA) and anti-n-cadherin(CAT#: 22018-1-AP, Proteintech, USA) followed by incubation with corresponding secondary antibodies, and stained with DAPI (4′,6-diamidino-2-phenylindole). Sections were digitally imaged with a 20x objective on a BZ-X810 All-in-One fluorescent microscope(Keyence Corp of America, USA). Negative controls were generated via the same protocol without the application of primary antibodies. Images were stitched together utilizing BZ-X800 Analyzer software(Keyence Corp of America, USA) maintaining constituent image pixel density within the stitched image. An example image stitch with representative constituent channels is shown in Fig. 4.1.

4.2.3. Image Processing

Image processing and analysis was conducted within MATLAB(MathWorks, USA). Background subtraction was performed by calculating the 97.5^{th} percentile intensity level of each RGB channel within each stitched image and setting pixels lower than the respective calculated value to zero. Representative negative, positive, and background subtracted image subsections are shown in Fig. 4.2. Binary masks corresponding to standardized myocardial segmentation methodologies were generated for each individual stitched image within the MATLAB Image Segmenter App and

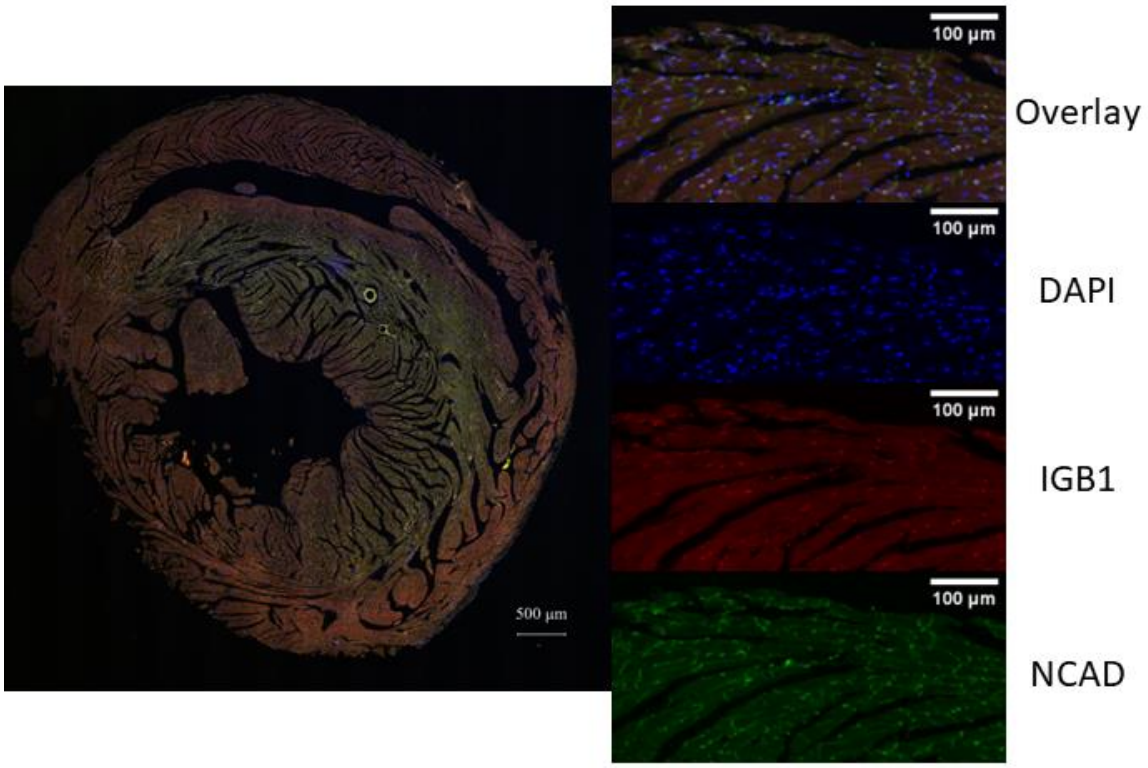

Fig. 4.1. Representative stitched image of sample cross – section and unmodified capture channels. Spatial resolution of analysis: r = 0.863 px/µm. DAPI: 4′,6-diamidino-2-phenylindole, IGB1: integrin-ß1, NCAD: n-cadherin.

applied to the stitched images to isolate each myocardial segment. The standardized myocardial segmentation methodologies are communicated by Cerqueira *et al.*[28] Representative image segmentation is shown in Fig. 4.3.

4.2.4. Calculations

Following background subtraction, remaining intensity values within each segmented region were normalized to range between 0 and 1. Normalized intensity

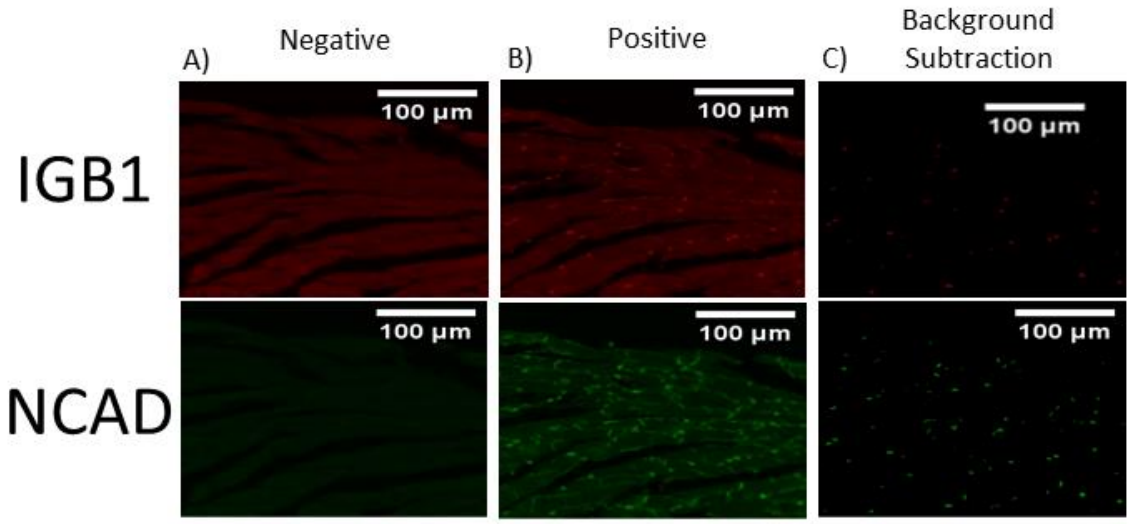

Fig. 4.2. Representative images for each mechanosensor of interest(IGB1: integrin-ß1, NCAD: n-cadherin). A) Negative controls were processed without incubation with primary antibodies. B) Positive samples prior to image processing and background subtraction. C) Processed images following background subtraction representative of final images used within analysis pipelines.

values were then averaged for integrin-ß1 and n-cadherin associated signals respectively. Regions were selectively included to match intended spatial analysis groupings. Whole heart analysis included all regions for each experimental group. Two separate regional analyses were conducted. The first analysis paired the anterior and septal regions of the analyzed myocardial segments against the inferior and lateral regions. The second analysis paired the analyzed myocardial segments by transverse planes. Regions segmented by transverse planes included the apical and midwall regions of the murine hearts.

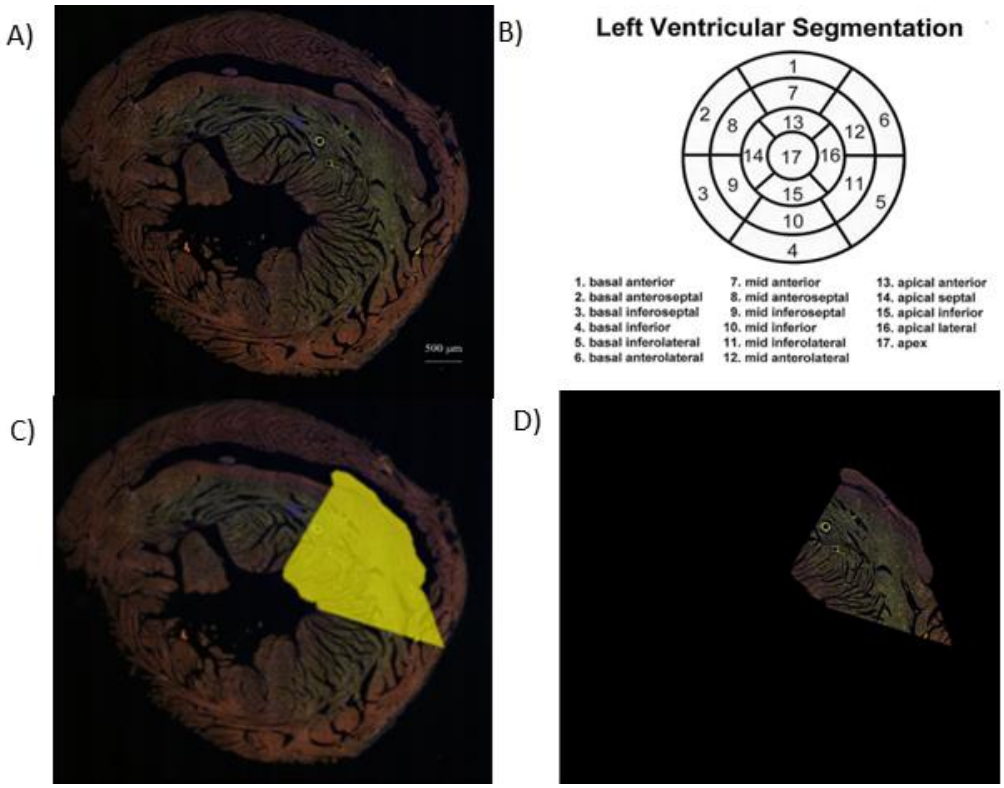

Fig. 4.3. Image segmentation utilizing standardized myocardial segmentation methodology. A) Representative sample cross-section with overlayed channels. B) Bullseye diagram depicting region codes for each myocardial segment based on standardized methodology. C) Binary mask overlayed on representative cross-section. D) Depiction of segmented region utilized in intensity analysis methodology.

4.2.5. Statistical Analysis

Data were expressed in bar charts as mean $\pm$ SEM. Statistical comparisons were made using two-sample t-tests. Linear relationships were statistically evaluated utilizing Pearson's linear correlation coefficient. All statistical analysis was conducted within MATLAB(MathWorks, USA).

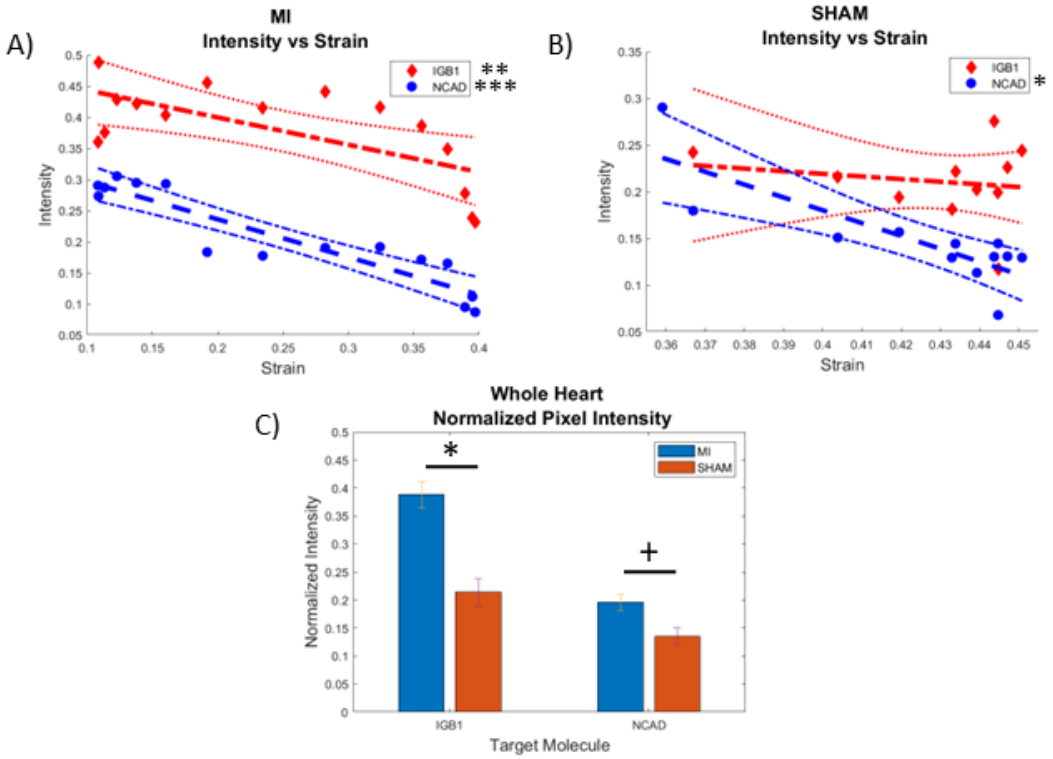

Fig. 4.4. Whole heart analysis evaluating normalized intensity in the context of strain and experimental condition. A,B) Scatter plots of normalized intensity values for mechanosensors of interest within infarcted and sham experimental conditions. 95% confidence window bounds the linear models of the relationships. (***: p<0.001; **: p<0.01; *: p<0.05) C) Bar charts evaluating differences in mechanosensory of interest normalized intensity between experimental conditions.(*: p<0.05; +: p<0.1 | MI: n = 98, SHAM: n =20).

4.3. Results

4.3.1. Whole Heart Comparisons

Analysis of normalized intensity of the target mechanosensors is shown in Fig. 4.4. Linear relationships between normalized intensity and strain exist for n-cadherin signal in both SHAM and MI experimental groups. This relationship is stronger within the MI group. Such a relationship is only exhibited for IGB1 signal within the MI

experimental group. Where relationships do exist, normalized intensity appears to be negatively correlated with strains. When comparing normalized intensity values between the MI and SHAM experimental conditions, there nearly exists statistical significance for both mechanosensors of interest. IGB1 does exhibit significance when comparing values between non-infarcted and infarcted hearts. N-cadherin value comparison trends towards significance yet only reaches a p-value less than 0.1.

4.3.2. Antero-Septal and Inferior-Lateral Regional Pairing

The spatial resolution of the analysis was improved by comparing the intensity-strain relationship based on anterior-septal and inferior lateral pairings shown in Fig. 4.5. When evaluating these groupings, there exists no linear relationship between normalized mechanosensor signal intensity and strain within uninfarcted sham samples in contrast to the relationship witnessed for n-cadherin when evaluating all regions together. Within infarcted samples, the strong negative linear relationship witnessed with whole heart evaluation reemerges for n-cadherin signal. Concerning IGB1, only a weak linear relationship is exhibited within the infarcted samples. This is in contrast to the statistically significant linear relationship exhibited during whole heart analysis. Comparing normalized intensity values of uninfarcted and infarcted samples across spatial pairings within Fig. 4.6 reveals a difference only for IGB1. There exists higher signal for IGB1 within infarcted samples compared to the sham experimental group. While n-cadherin appears to also show a higher intensity level for the infarcted groups, this difference is insignificant.

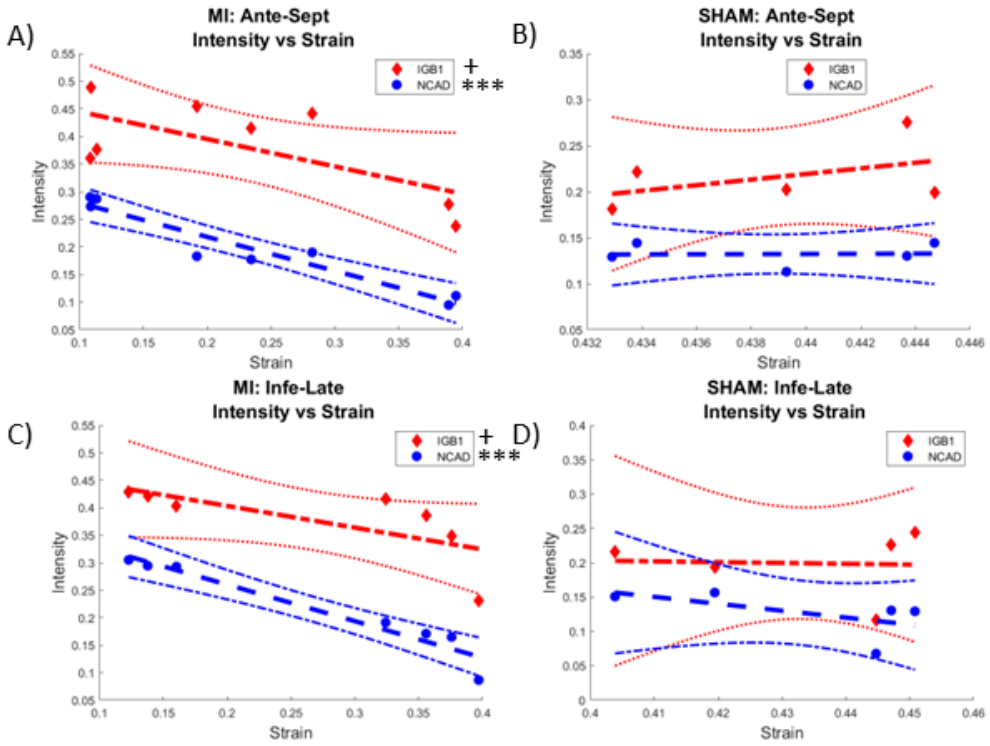

Fig. 4.5. Anterior-Septal vs Inferior-Lateral region analysis evaluating normalized intensity in the context of strain and experimental condition. Scatter plots of normalized intensity values for mechanosensors of interest within MI and SHAM experimental conditions for the A,B) anterior-septal and C,D) inferior-lateral regions. 95% confidence window bounds the linear models of the relationships. (***: $p<0.001$; +: $p<0.1$).

4.3.3 Apical and Midwall Regional Pairing

Evaluations of normalized intensity relationships to strain within the apical and midwall regions of the samples are exhibited in Fig. 4.7. In contrast to previous pairings, the regions of the apical plane of the SHAM sample exhibit a relationship concerning IGB1 normalized intensity and strain while n-cadherin within the regions is also trending towards a significant relationship. This is not shared by the regions of the SHAM midwall, is reduced for IGB1 in the apical region of infarcted samples, and lost for n-

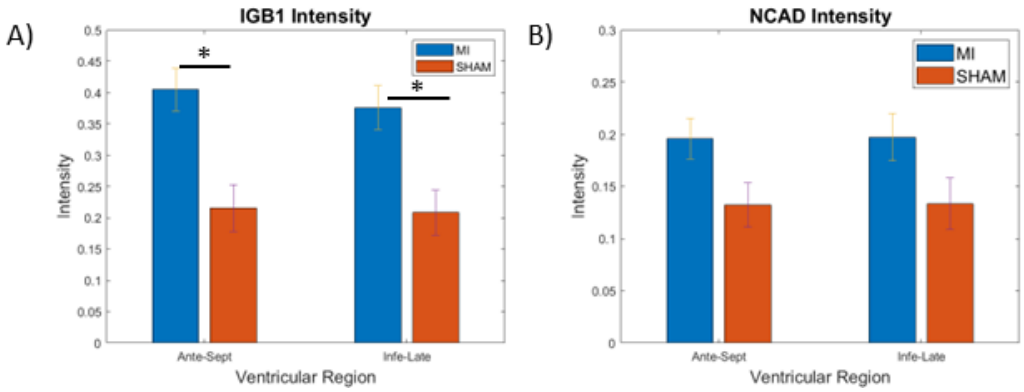

Fig. 4.6. Bar charts evaluating anterior-septal and inferior-lateral regional differences in normalized intensity of A) IGB1 and B) N-cadherin between experimental conditions.(*: $p<0.05$ | Ante-Sept MI: n = 51, SHAM: n = 10 | Infe-Late MI: n = 46, SHAM = 9).

cadherin signal. Strangely enough, infarcted midwall regions show a negative linear relationship is developed that is statistically not present within uninfarcted sham midwall regions. N-cadherin exhibits no relationship within either experimental group when analyzing relationships within the midwall transverse plane. Normalized intensity levels are evaluated in Fig. 4.8. For both mechanosensors of interest there exists statistically significant differences between apical and midwall planes for the infarcted experimental group. Comparing experimental groups, IGB1 only exhibits a difference within the midwall while n-cadherin normalized intensity values differ within each analysis region. N-cadherin also exhibits differential trends based on regional analysis. Within apical regions, n-cadherin signal intensity is lower within infarcted samples than observed intensities of the sham samples. Observed midwall signal intensity is at a greater level within infarcted samples than sham samples.

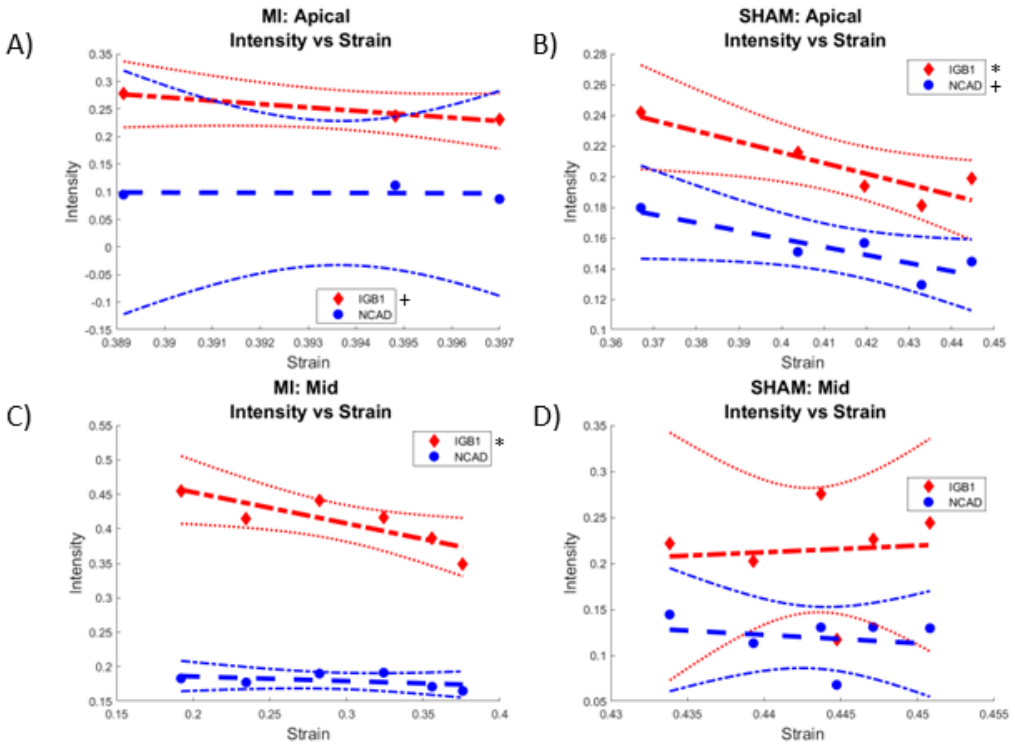

Fig. 4.7. Transverse Regional Analysis: Apical vs Midwall analysis evaluating normalized intensity in the context of strain and experimental condition. Scatter plots of normalized intensity values for mechanosensors of interest within infarcted and sham experimental conditions for the A,B) apical and C,D) midwall regions. 95% confidence window bounds the linear models of the relationships. (*: $p<0.05$; +: $p<0.1$).

4.4. Discussion

Relationships between n-cadherin and integrin-ß1 expression have previously been linked to varying mechanics as both molecules are intricately involved in the mechanotransduction process within both cardiac fibroblasts and cardiomyocytes. N-cadherin binding has been experimentally demonstrated to play an independent yet complementary role to ECM-cellular binding within cardiomyocytes. This was done by

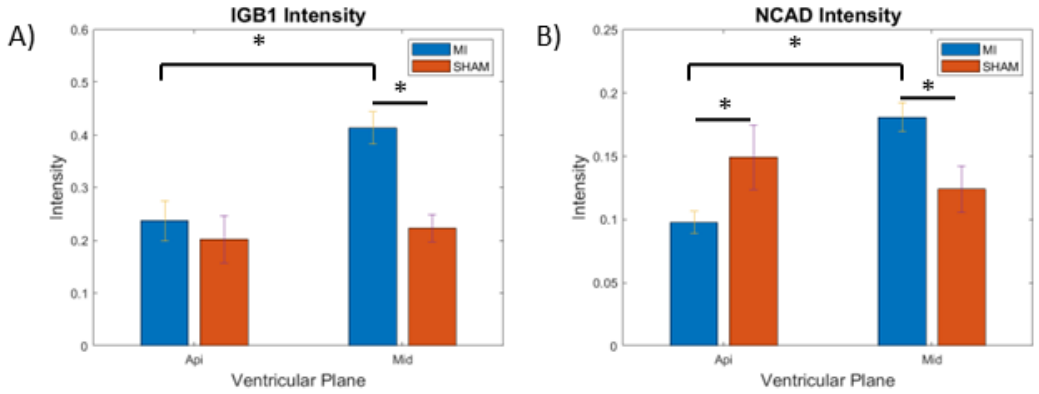

Fig 4.8. Bar charts evaluating apical and midwall transverse regional differences in normalized intensity of A) IGB1 and B) N-cadherin between experimental conditions.(*: $p<0.05$ | Apical MI: n = 14, SHAM: n = 9 | Midwall MI: n = 60, SHAM = 11).

coating substrates with either n-cadherin or matrix molecules and evaluating resultant cytoskeletal structure.[14] In a separate experiment, n-cadherin expression level in cardiomyocytes was linked to transverse principal strains via heightened immunofluorescent expression in vitro while subjected to these strains within a stretch-induced deformable elastomer scaffold.[29] Integrin-ß1 within cardiomyocytes has been shown to be essential to healthy heart function via knockout studies demonstrating that a lack of integrin-ß1 leads to reduced contractility, progressive dilated cardiomyopathy, and eventual heart failure in mouse models.[30] N-cadherin has been linked to dilated cardiomyopathy and impaired contractility via loss of myofibril anchorage to the plasma membrane.[31] Integrin-ß1 within a murine mesenchymal stem cell model has been shown to vary with underlying substrate stiffness dependent upon the constituent make-up of the substrate. Differential expression related to stiffness was observed when the underlying substrate was composed of collagen I.[32] These studies taken together highlight the

complications involved in evaluating mechanosensory response in the context of the in vivo environment and varied pathological states. Expression can vary based on substrate stiffness, mechanical stimulation, and constituent makeup of underlying substrate material. Within an infarcted myocardial wall, all of these factors are present and vary spatially throughout the ventricle. This reality necessitates improved experimental spatial resolution in investigating these factors. Within our own spatially varied analyses, differential normalized intensity response to strain was observed. Our whole heart analysis revealed a negative linear relationship between normalized intensity and regional strain within the infarcted experimental group for both mechanosensors of interest. A reduced yet still significant linear relationship was exhibited by n-cadherin within the sham experimental group that was not exhibited by integrin-ß1. When grouping regions by anterior-septal and inferior-lateral regional demarcation, the relationship exhibited by n-cadherin within the sham experimental group was lost for both regions. The negative linear relationship between n-cadherin normalized intensity and regional strain observed in the infarcted groups was maintained with this regional breakdown while the relationship for integrin-ß was weakened. Interestingly, when the regional analysis utilized regions separated by transverse planes, n-cadherin relationships were lost while IGB1 expressed differential results for each plane depending on experimental group of infarcted or sham. Within the midwall of the infarcted experimental group, IGB1 expressed a significant negative linear relationship that was not exhibited within the apical region. Just the opposite was demonstrated within the sham experimental group

where IGB1 exhibited a negative linear relationship between normalized intensity and regional strain within the apical regions that was absent within the midwall.

A previous study, conducted by Matsushita et al., analyzing n-cadherin and integrin-ß1 presence within the border zone of a murine infarct over a 60-day time period highlighted mechanosensory expression within this spatial region following applied ischemia. An initial drop in the expression of these mechanosensors was observed followed by a return to normal expression levels by the second week. While the difference was statistically insignificant at that point, the general trend showed a higher average expression level at the 15-day mark than the sham experimental conditions.[33] This coincides with our own experimental findings of greater normalized intensity values for both mechanosensors of interest within infarcted samples compared to the sham experimental condition. However, this result did differ based on our regional analysis applied. When observing whole heart trends, this relationship of greater expression held for IGB1 and trended towards significance for n-cadherin. This also coincides with integrin-ß1 results from Okada et al where IGB1 expression was increased following ischemia/reperfusion treatment application within a murine model.[34] When we grouped cardiac regions based on anterior-septal and inferior-lateral locations, this normalized intensity increase fell further from significance for n-cadherin alone while being maintained for IGB1. When cardiac regions were grouped by transverse plane separation, differences between MI and sham normalized intensity level for IGB1 were only observed within the midwall. N-cadherin normalized intensity demonstrated differential intensity levels between not only MI and sham for both regions, but it also displayed

differential intensity between regions within the MI experimental group. IGB1 also exhibited differential intensity levels between regions within the MI experimental group. These varied results highlight the importance of spatial considerations when evaluating mechanosensory expression within the context of the in vivo environment. This consideration is reinforced by studies that have demonstrated the spatial heterogeneities that exist within the myocardial wall concerning mechanics and their inextricable link to collagen architectures within post-infarction scars.[1,35–37]

Taken together, it is evident that regional differences in mechanics likely play significant roles in differential regional response by cellular agents. To generate more useful and holistic understandings of the intricate interplay of parameters within the in vivo wall, spatial heterogeneities must not be neglected. This study could be improved by increasing the biological replicates of the SHAM experimental grouping. With greater resolution of available mechanics data, our analysis would be able to more accurately correlate observed signal to local mechanics. Additionally, as n-cadherin and integrin-ß1 are both mechanosensors at the interface of either cell-ecm or cell-cell interactions, the inclusion of a range of additional mechanosensors or downstream signaling molecules implicated in niche signaling responses could elucidate a greater understanding of what mechanical cues are truly utilized on particular cellular responses.

4.5. Conclusions

Mechanics play a significant role in myocardial response following an infarction event and the ultimate trajectory of recovery. It is evident that the embedded cell populations detect and respond to these mechanics, yet there are many other factors at

play. Not only do mechanics vary spatially within the myocardial wall, but constituent tissue components vary as well. Considering that cellular response is predicated on these factors that vary spatially, it is essential to consider this factor in any worthwhile evaluation of cellular response within the myocardial wall.

In this study we analyzed normalized expression of canonical mechanosensors at the interface of cell-ecm and cell-cell interactions in the context of in vivo strains. Differential results were observed between normalized intensity levels and normalized intensity-strain correlations when evaluating data in the context of spatial distribution within the ventricle. While some trends held, others were clearly dependent on spatial location. When evaluating the normalized intensity relationship in the context of the whole heart, a clear covariance between parameters was demonstrated within the infarcted experimental condition for both mechanosensors. When observing specific regional groupings, the mechanosensors of interest responded differentially. This highlights the complexities at play and reinforces the importance of such an analysis. Future studies utilizing this methodology should involve additional downstream signal transduction molecules that can further tease out which mechanotransduction pathways are being regionally activated or ignored. Greater resolution concerning generated mechanics could also lead to interesting and unique insights that are currently infeasible with the regional analysis utilized within the current study.

4.6. References

1. Richardson, W. J., Clarke, S. A., Alexander Quinn, T. & Holmes, J. W. Physiological implications of myocardial scar structure. *Compr Physiol* (2015) doi:10.1002/cphy.c140067.

2. Richardson, W. J. & Holmes, J. W. Emergence of Collagen Orientation Heterogeneity in Healing Infarcts and an Agent-Based Model. *Biophys J* (2016) doi:10.1016/j.bpj.2016.04.014.

3. Bouzeghrane, F. α8β1 integrin is upregulated in myofibroblasts of fibrotic and scarring myocardium. *J Mol Cell Cardiol* **36**, 343–353 (2004).

4. Carracedo, S. *et al.* The Fibroblast Integrin α11β1 Is Induced in a Mechanosensitive Manner Involving Activin A and Regulates Myofibroblast Differentiation. *Journal of Biological Chemistry* **285**, 10434–10445 (2010).

5. Ng, C. P., Hinz, B. & Swartz, M. A. Interstitial fluid flow induces myofibroblast differentiation and collagen alignment in vitro. *J Cell Sci* **118**, 4731–4739 (2005).

6. Talior-Volodarsky, I., Connelly, K. A., Arora, P. D., Gullberg, D. & McCulloch, C. A. α11 integrin stimulates myofibroblast differentiation in diabetic cardiomyopathy. *Cardiovasc Res* **96**, 265–275 (2012).

7. Civitarese, R. A. *et al.* The α11 integrin mediates fibroblast–extracellular matrix–cardiomyocyte interactions in health and disease. *American Journal of Physiology-Heart and Circulatory Physiology* **311**, H96–H106 (2016).

8. Civitarese, R. A., Kapus, A., McCulloch, C. A. & Connelly, K. A. Role of integrins in mediating cardiac fibroblast–cardiomyocyte cross talk: a dynamic relationship in cardiac biology and pathophysiology. *Basic Res Cardiol* **112**, 6 (2017).

9. Kresh, J. Y. & Chopra, A. Intercellular and extracellular mechanotransduction in cardiac myocytes. *Pflugers Arch* **462**, 75–87 (2011).

10. Sheehy, S. P., Grosberg, A. & Parker, K. K. The contribution of cellular mechanotransduction to cardiomyocyte form and function. *Biomech Model Mechanobiol* **11**, 1227–1239 (2012).

11. Samarel, A. M. Costameres, focal adhesions, and cardiomyocyte mechanotransduction. *American Journal of Physiology-Heart and Circulatory Physiology* **289**, H2291–H2301 (2005).

12. Dalla Costa, A. P. *et al.* FAK mediates the activation of cardiac fibroblasts induced by mechanical stress through regulation of the mTOR complex. *Cardiovasc Res* **86**, 421–431 (2010).

13. MacKenna, D. Role of mechanical factors in modulating cardiac fibroblast function and extracellular matrix synthesis. *Cardiovasc Res* **46**, 257–263 (2000).

14. Chopra, A., Tabdanov, E., Patel, H., Janmey, P. A. & Kresh, J. Y. Cardiac myocyte remodeling mediated by N-cadherin-dependent mechanosensing. *American Journal of Physiology-Heart and Circulatory Physiology* **300**, H1252–H1266 (2011).

15. Stewart, L. & Turner, N. A. Channelling the Force to Reprogram the Matrix: Mechanosensitive Ion Channels in Cardiac Fibroblasts. *Cells* **10**, 990 (2021).

16. Reed, A., Kohl, P. & Peyronnet, R. Molecular candidates for cardiac stretch-activated ion channels. *Glob Cardiol Sci Pract* **2014**, 19 (2014).

17. MacKenna, D. A., Dolfi, F., Vuori, K. & Ruoslahti, E. Extracellular signal-regulated kinase and c-Jun NH2-terminal kinase activation by mechanical stretch is integrin-dependent and matrix-specific in rat cardiac fibroblasts. *Journal of Clinical Investigation* **101**, 301–310 (1998).

18. LAMMERDING, J., KAMM, R. D. & LEE, R. T. Mechanotransduction in Cardiac Myocytes. *Ann N Y Acad Sci* **1015**, 53–70 (2004).

19. Gudi, S. R. P., Lee, A. A., Clark, C. B. & Frangos, J. A. Equibiaxial strain and strain rate stimulate early activation of G proteins in cardiac fibroblasts. *American Journal of Physiology-Cell Physiology* **274**, C1424–C1428 (1998).

20. Wang, J., Seth, A. & McCulloch, C. A. G. Force regulates smooth muscle actin in cardiac fibroblasts. *American Journal of Physiology-Heart and Circulatory Physiology* **279**, H2776–H2785 (2000).

21. WEI, L. *et al.* βl integrin and organized actin filaments facilitate cardiomyocyte-specific RhoA-dependent activation of the skeletal α-actin promoter. *The FASEB Journal* **15**, 785–796 (2001).

22. Ieda, M. *et al.* Cardiac Fibroblasts Regulate Myocardial Proliferation through β1 Integrin Signaling. *Dev Cell* **16**, 233–244 (2009).

23. Thompson, S. A. *et al.* Acute slowing of cardiac conduction in response to myofibroblast coupling to cardiomyocytes through N-cadherin. *J Mol Cell Cardiol* **68**, 29–37 (2014).

24. Lee, E. J. *et al.* N-cadherin Determines Individual Variations in the Therapeutic Efficacy of Human Umbilical Cord Blood-derived Mesenchymal Stem Cells in a Rat Model of Myocardial Infarction. *Molecular Therapy* **20**, 155–167 (2012).

25. Vite, A. & Radice, G. L. N-Cadherin/Catenin Complex as a Master Regulator of Intercalated Disc Function. *Cell Commun Adhes* **21**, 169–179 (2014).

26. Matsuda, T. *et al.* N-cadherin-mediated cell adhesion determines the plasticity for cell alignment in response to mechanical stretch in cultured cardiomyocytes. *Biochem Biophys Res Commun* **326**, 228–232 (2004).

27. Soepriatna, A. H. *et al.* Three-dimensional myocardial strain correlates with murine left ventricular remodelling severity post-infarction. *J R Soc Interface* **16**, 20190570 (2019).

28. Cerqueira, M. D. *et al.* Standardized Myocardial Segmentation and Nomenclature for Tomographic Imaging of the Heart. *Circulation* **105**, 539–542 (2002).

29. Gopalan, S. M. *et al.* Anisotropic stretch-induced hypertrophy in neonatal ventricular myocytes micropatterned on deformable elastomers. *Biotechnol Bioeng* **81**, 578–587 (2003).

30. Shai, S.-Y. *et al.* Cardiac Myocyte-Specific Excision of the β1 Integrin Gene Results in Myocardial Fibrosis and Cardiac Failure. *Circ Res* **90**, 458–464 (2002).

31. Kostetskii, I. *et al.* Induced Deletion of the *N-Cadherin* Gene in the Heart Leads to Dissolution of the Intercalated Disc Structure. *Circ Res* **96**, 346–354 (2005).

32. Gershlak, J. R. & Black, L. D. Beta 1 integrin binding plays a role in the constant traction force generation in response to varying stiffness for cells grown on mature cardiac extracellular matrix. *Exp Cell Res* **330**, 311–324 (2015).

33. Matsushita, T. *et al.* Remodeling of Cell-Cell and Cell–Extracellular Matrix Interactions at the Border Zone of Rat Myocardial Infarcts. *Circ Res* **85**, 1046–1055 (1999).

34. Okada, H. *et al.* Integrins protect cardiomyocytes from ischemia/reperfusion injury. *Journal of Clinical Investigation* **123**, 4294–4308 (2013).

35. Richardson, W. J. & Holmes, J. W. Emergence of Collagen Orientation Heterogeneity in Healing Infarcts and an Agent-Based Model. *Biophys J* **110**, 2266–2277 (2016).

36. Korenczuk, C. E., Barocas, V. H. & Richardson, W. J. Effects of Collagen Heterogeneity on Myocardial Infarct Mechanics in a Multiscale Fiber Network Model. *J Biomech Eng* **141**, (2019).

37. Clarke, S. A., Richardson, W. J. & Holmes, J. W. Modifying the mechanics of healing infarcts: Is better the enemy of good? *Journal of Molecular and Cellular Cardiology* Preprint at https://doi.org/10.1016/j.yjmcc.2015.11.028 (2016).

CHAPTER FIVE

Conclusions and Future Works

5.1. Conclusions

The overall intent of this dissertation was the development and validation of experimental platforms and measurement methods that enable the relevant study of the spatial heterogeneities found within myocardial scars. Ischemic heart disease is continuing to increase in global prevalence with the average annual cost of heart disease within the United States exceeding $218.7 billion in years past and prevalence exceeding 8 million over three-year periods.[1] The main heart failure modalities are typically categorized as either heart failure with preserved(HFpEF) or reduced(HFrEF) ejection fraction and relate to either preload or contractility issues.[2] Both of these modalities are intricately related to myocardial mechanics which are inextricably linked to how collagen content is handled following an infarction event. At any point from initial degradation to ultimate maturation and remodeling of the scar and surrounding myocardium, cardiac functionality can veer towards a heart failure mode ruled by wall overcompliance or lack thereof.[3] Detailed analysis of the processes that give rise to the infarct scar that governs mechanics and eventual development toward pathology pathways is needed.

Literature reviews and finite element models have previously highlighted just how complicated the myocardial wall environment is throughout the recovery phases following an infarction event.[4–6] Other investigations underscore the importance of spatial heterogeneities in particular and range from how mechanics are altered due to

infarct scar location to collagen fiber architecture heterogeneities within rat infarct scars themselves.[4,6] These findings of heterogeneous collagen distributions within mature infarct scars motivated the initial study of aim 1 where the expressed goal was to produce a platform capable of generating and analyzing heterogeneous collagen alignment. Other methodologies exist to generate anisotropic alignment within collagen architectures; however, the tradeoffs are steep. We presented a simple and effective heterogeneous collagen gel fabrication and analysis methodology enabling the swift and relevant study of cellular agents embedded within collagenous boundary regions of spatially varied alignment. Heterogeneity was verified through evaluation of distributed fiber orientations across the entirety of the fabricated gels, and cell-centric analysis methodologies were employed to highlight the importance of evaluating the cell as an independent agent in the context of its surrounding environment at varying distances. Low-cost probing of fundamental cell-matrix interactions in spatially heterogeneous collagenous environments was enabled.

Additionally, cellular response within the early stages of the post-infarction environment is paramount in governing the eventual progression towards particular heart failure modalities. Therefore, to evaluate resultant cell-response during the early stages of pathophysiological response within the myocardium, we developed an in vitro-coculture engineered heart tissue platform. Utilizing this platform, we simulated a myocardial infarction event via cryo-wounding and observed the spatial heterogeneities within construct mechanics and their progression over a seven-day timecourse. Regional differences were observed within wounded constructs that revealed greater insights into

the developing mechanics following a simulated infarction event. This platform and analysis methodology hold great promise for future studies investigating varied wound patterns, stimulation regimes, and additional embedded cell-types that will tease out the relationship that exists between spatial heterogeneities of construct mechanics and cellular response.

While the generation of platforms to study the fundamental interplay between spatial heterogeneities in environmental structure, construct mechanics, and cell response are essential to elucidate further understanding from such controlled environments, it is of the utmost importance to not lose touch with the in vivo reality that inspired the other thrusts of this work. Therefore, an exploration of mechanosensor expression in the context of in vivo mechanics was performed for the final aim. Collaborator provided strain values collected utilizing digital ultrasound on mature murine infarcted myocardial walls was correlated with the normalized intensity of immunofluorescently labeled n-cadherin and integrin-ß1 canonical mechanosensors. These mechanosensors were selected during this initial evaluation to cover the bases of cell-ecm and cell-cell connections. Within this study, the presence of spatial heterogeneities arose again in the form of normalized mechanosensor intensity correlation to regional strains. Variance in correlations was observed depending on the location within the murine myocardium. It was established that the expected relationships do exist, and future explorations of said relationships must be cognizant of the complexities at play to avoid faulty conclusions if spatial variations are disregarded.

5.2. Study Limitations

While our objectives were met, there exist limitations to the work performed. Concerning the initial platform for the generation of collagen gels possessing spatial heterogeneity within degree of fibrous alignment, a primary limitation is the lack of intricate control over the generation of these heterogeneities. The platform is intended to be a rapid and cost-effective way to generate these spatial heterogeneities thus enabling the study of phenomena relevant to the produced architecture such as infarct scars. If extreme anisotropy is desired for the tissue of interest, then this platform may not meet the desired specifications. While smaller subregions within the aligned zone did exhibit strong alignment, the region at large possessed more moderate levels of alignment.

Limitations within the in vitro-coculture engineered heart tissue platform, like the heterogeneous collagen platform, revolve around intricate control. Significant sample-to-sample variability in initial mechanics was observed, as evidenced by the higher average modulus exhibited by the wounded gels from day 0. While the difference was not statistically significant, it highlights an aspect of the protocol that needs further investigation. Additionally, due to the prototype nature of the platform, there exists much room for improvement concerning the development of intricate control systems that will truly unlock the potential of the joint electrical and mechanical stimulation of the engineered heart tissue constructs. Additionally, fibrin gels were utilized in this iteration of the platform. Improvements in cell-traction and contractile forces could be had from switching to a substrate alternative such as the collagen I constructs utilized within aim 1. Limitations also existed for image acquisition in the analysis of regional mechanics.

Improving upon pixel resolution of the capture device would unlock greater levels of analysis and possibly reveal clearer insights missed with the current specifications of the image acquisition device.

The predominant limitation of the work of aim 3 is the spatial resolution of the available strain data. Strain data utilized were the average strain values for myocardial subregions corresponding to the standardized segmentation methodology. While this level of resolution still provided unique insights into the importance of considering spatial components of analyses of the myocardial wall, greater resolution that could be more closely mapped to cellular level correlations are expected to provide improved insights. Additionally, only observing the mechanosensors of n-cadherin and integrin-ß1 could obfuscate the true relationship of the environmental mechanics and the resultant cellular response. These molecules are great transducers for many downstream cascades. For more unique insights, additional participants of downstream mechanosensor signaling pathways must be included in future analyses. Limitations of fluorescence signal bleed over between channels could become problematic if the field of intended targets is expanded too drastically for a single analysis and would necessitate the exploration of additional options such as various molecular level resolution mass spectrometry evaluations. Finally, increasing the number of biological replicates, particularly for the sham experimental condition, would greatly improve the statistical power of the analysis and should be pursued.

5.3. Recommended Future Studies

Many additional avenues exist for the use of collagen gels possessing spatial heterogeneity in degree of alignment. An initial thrust should involve the fabrication of the heterogeneous architectures while only embedding cells within one portion of the architecture. Observations of differences in cellular migration rates, degree of underlying alignment remodeling, and cellular parameters such as eccentricity, major to minor axis ratio, cell area, and others could further reveal the relationship between embedded cellular populations, their immediately surrounding environment, and the architecture of the environment at large. Combining the analysis metrics used in our work with additional labeling of cytokines or regulatory markers of interest could improve the understanding of the role environmental detection has on resultant remodeling.

Concerning the in vitro co-culture engineered heart tissue platform, an obvious next step is the inclusion of stimulatory molecules to observe differences in observed mechanics and rates of stiffening related to the baseline shown in our work conducted for aim 2. Additionally, interesting combinations of electrical and mechanical stimulation regimens could be utilized to simulate pathophysiological states such as atrial fibrillation by varying the electrical stimulation cadence. Further development of control systems for the synergism of the stimulation modalities should be performed in the pursuit of increased relevance towards physiological states and the increased usefulness of the platform as a potential therapeutic test bed.

As mentioned previously, future studies building on the work of aim 3 should utilize improved spatial resolution within the strain datasets to further refine the spatially

based observational differences. Additionally, specific signal molecules further downstream of mechanosignaling pathways should be utilized to better discern the relationships between specific cellular responses and the state of the in vivo myocardial environment.

While each work has potential future uses individually, utilizing the three works presented here together presents the most exciting path forward. The analysis methodology utilized in aim 3 should be used to discover particular myocardial mechanical environments of note to be replicated by the in vitro co-culture engineered heart tissue platform of aim 2. While the generation of heterogeneous collagen gels can be utilized in conjunction with the co-culture engineered heart tissue platform to investigate more complicated mechanical environments that closely resemble the spatially heterogeneous and collagenous environments found in a myocardial wall recovering from an infarction event.

5.4. References

1. Virani, S. S. *et al.* Heart Disease and Stroke Statistics—2020 Update: A Report From the American Heart Association. *Circulation* **141**, (2020).
2. Redfield, M. M. Heart Failure with Preserved Ejection Fraction. *New England Journal of Medicine* **375**, 1868–1877 (2016).
3. Richardson, W. J., Clarke, S. A., Alexander Quinn, T. & Holmes, J. W. Physiological implications of myocardial scar structure. *Compr Physiol* (2015) doi:10.1002/cphy.c140067.
4. Richardson, W. J. & Holmes, J. W. Why Is Infarct Expansion Such an Elusive Therapeutic Target? *Journal of Cardiovascular Translational Research* Preprint at https://doi.org/10.1007/s12265-015-9652-2 (2015).
5. Fomovsky, G. M., Macadangdang, J. R., Ailawadi, G. & Holmes, J. W. Model-Based Design of Mechanical Therapies for Myocardial Infarction. *J Cardiovasc Transl Res* **4**, 82–91 (2011).
6. Clarke, S. A., Richardson, W. J. & Holmes, J. W. Modifying the mechanics of healing infarcts: Is better the enemy of good? *Journal of Molecular and Cellular Cardiology* Preprint at https://doi.org/10.1016/j.yjmcc.2015.11.028 (2016).

APPENDICES

Appendix A

Aim 1 MATLAB Scripting

A-1.1 Image Processing and Calculation of Alignment Values

```
%M. Jake Potter - Initial Image Processing

% %%Loads all tif files in directory into 'tiffiles' variable
tiffiles=dir('*.tif');
%Finds number of tif files in directory
nf=length(tiffiles);

% Size of square subregion (pixels)
S = 64;
x=tiffiles(1).name;
aimage=imread([x]); %reads image into 'aimage'
b1 = aimage(:,:,1); %convert rgb to gray
iImage = b1;   % Import the image
dImage = double(iImage)+1; % Convert into double
D = size(dImage);
N1 = floor((D(1)-10)/S);          % Number of subregions vertically;
N2 = floor((D(2)-10)/S);          % Number of subregions horizontally;
singlefibpredict=N1*N2*nf;  %Calculates number of single fiber subregions to be analyzed by multiplying subregion dim1 by
subregion dim2 by the number of files to be analyzed
%For storing information of individual fibers/cells
tableheaders={'Basename','Condition','Well','Position','GelRow','RegionLabel',...
    'Xcoord','Ycoord','DateofRun','Orientation','Eccentricity','MajorAxis','MinorAxis','Day','LoopControl','Area','Perimeter'};
tabletypes={'string','string','string','double','string','string',...
    'double','double','double','double','double','double','double','string','double','double','double'};
singlefibtable=table('Size',[singlefibpredict 13],'VariableNames',tableheaders([1:11 14:15]),...
    'VariableTypes',tabletypes([1:11 14:15]));
singlecelltable=table('Size',[height(t) 17],'VariableNames',tableheaders,...
    'VariableTypes',tabletypes);
%for storing information by image
imagetableheaders={'Basename','Condition','Day','Well','Position','GelRow','RegionLabel',...
   'MVL','Orientation','DateofRun'};
imagetabletypes={'string','string','string','string','double','string','string',...
   'double','double','double'};
fibstable=table('Size',[nf 10],'VariableNames',imagetableheaders,...
   'VariableTypes',imagetabletypes);
cellstable=table('Size',[nf 10],'VariableNames',imagetableheaders,...
   'VariableTypes',imagetabletypes);
tic

c=0; %count
%Beginning of analysis loop; progresses through all images
for n=1:nf;
    d=c; %index assistant
    c=c+1; %advances loop count
            x=tiffiles(n).name; %assigns name of file to x
            aimage=imread([x]); %reads image into 'aimage'
            b1 = aimage(:,:,1); %convert rgb to gray
            b2=histeq(b1); %normalizes pixel intensity distribution
            p=prctile(b2(find(b2>0)),15);  %finds 15th prctile
            b2(find(b2<=p))=0; %sets equalized pixels below 15th percentile to 0
            b1(find(b2==0))=0; %subtracts background of image to be analyzed

               S = 64;        % Size of the square subregion (pixels)
               Thresh = 115000;
               iImage = b1;   % Import the image
               dImage = double(iImage)+1; % Convert into double

               % Initial computations related to the image and subregions
               % Because gradient mask (next step) is 9x9, convolution result will be
```

```
% erroneous in pixels near image border. Subregions are shifted 5 pixels
% away from the image borders to avoid this error.
D = size(dImage);
N1 = floor((D(1)-10)/S);          % Number of subregions vertically;
N2 = floor((D(2)-10)/S);          % Number of subregions horizontally;

% Gradient masks computation
% for i=-4:4;
%     for j=-4:4;
%         MX(i+5,j+5) = 2*i/9*exp(-(i^2+j^2)/4);
%         MY(i+5,j+5) = 2*j/9*exp(-(i^2+j^2)/4);
%     end
% end
% Hard coded here to save computation time.
MX = ...
    [-0.00029819    -0.001716   -0.0059893  -0.012679   -0.016281   -0.012679   -0.0059893  -0.001716   -0.00029819;...
      -0.001287     -0.007406 -0.025849   -0.054723   -0.070266   -0.054723   -0.025849   -0.007406   -0.001287;...
       -0.0029946    -0.017233   -0.060149   -0.12734    -0.1635     -0.12734    -0.060149   -0.017233   -0.0029946;...
       -0.0031698    -0.018241   -0.063668   -0.13478    -0.17307    -0.13478    -0.063668   -0.018241   -0.0031698;...
        0             0          0          0           0           0           0           0           0;...
       0.0031698     0.018241 0.063668    0.13478     0.17307     0.13478     0.063668    0.018241    0.0031698;...
       0.0029946     0.017233 0.060149    0.12734     0.1635      0.12734     0.060149    0.017233    0.0029946;...
       0.001287      0.007406 0.025849    0.054723    0.070266    0.054723    0.025849    0.007406    0.001287;...
       0.00029819    0.001716 0.0059893   0.012679    0.016281    0.012679    0.0059893   0.001716    0.00029819];
MY = ...
    [-0.00029819    -0.001287   -0.0029946  -0.0031698  0   0.0031698   0.0029946   0.001287    0.00029819;...
      -0.001716     -0.007406 -0.017233   -0.018241   0   0.018241    0.017233    0.007406    0.001716;...
       -0.0059893    -0.025849   -0.060149   -0.063668   0   0.063668    0.060149    0.025849    0.0059893;...
       -0.012679     -0.054723   -0.12734    -0.13478    0   0.13478     0.12734     0.054723    0.012679;...
       -0.016281     -0.070266   -0.1635     -0.17307    0   0.17307     0.1635      0.070266    0.016281;...
       -0.012679     -0.054723   -0.12734    -0.13478    0   0.13478     0.12734     0.054723    0.012679;...
       -0.0059893    -0.025849   -0.060149   -0.063668   0   0.063668    0.060149    0.025849    0.0059893;...
       -0.001716     -0.007406   -0.017233   -0.018241   0   0.018241    0.017233    0.007406    0.001716;...
       -0.00029819 -0.001287   -0.0029946  -0.0031698  0   0.0031698   0.0029946   0.001287    0.00029819];

% Convolve masks with the image
GX = conv2(dImage,MX,'same');
GY = conv2(dImage,MY,'same');
clear MX MY

% Edge image and gradient direction
E = GX.^2+GY.^2;
phi = 180/pi*atan2(GY, GX);
clear GX GY

% Determine local orientation in each subregion
bins = 0:1:179;
FiberAngle = zeros([N1*N2 1]);
FiberPosX = zeros([N1*N2 1]);
FiberPosY = zeros([N1*N2 1]);
count = 0;

for q = 1:1:N1
for r = 1:1:N2
    lx = 5 + S*(r - 1) + 1;
      ux = 5 + S*r;
    ly = 5 + S*(q - 1) + 1;
      uy = 5 + S*q;
    AthetaW = zeros(size(bins));
for a = 1:1:numel(bins)
        C = E(ly:uy,lx:ux).*exp(2*cos(2*pi/180*(bins(a) - phi(ly:uy,lx:ux))))/exp(2);
        AthetaW(a) = sum(sum(C)); % AthetaW is a sum of alignment values between pixel gradient directions and each
direction from 'bins', weighted by pixel gradient magnitudes.
                % The sum includes all pixels within the subregion bounded by lx, ux, ly, uy.
end
    if max(AthetaW) > Thresh;
```

```
            [~, Ang] = max(AthetaW); % Sets 'Ang' as the entry from 'bins' corresponding to the maximum 'AthetaW' value.
            if Ang > 90
            Ang = Ang - 180;
            end
            count = count + 1;
            FiberAngle(count) = -1*Ang; %Negative of angle necessary for maintaining coordinate origin continuity with rest of
analysis and including cell profiler later
            FiberPosY(count) = round(5 + S*q - S/2);
            FiberPosX(count) = round(5 + S*r - S/2);
            end
            clear AthetaW Ang;
          end
          end

          if count < numel(FiberAngle)
            FiberAngle(count+1:end) = [];
            FiberPosY(count+1:end) = [];
            FiberPosX(count+1:end) = [];
          end

            FiberOrientationX = cos(FiberAngle*pi/180);
            FiberOrientationY = sin(FiberAngle*pi/180);

            h=length(FiberAngle);

            AngCelArray{c}=[FiberAngle];
            NameCelArray{c}=[repmat(tiffiles(n).name(4:end-6),[length(FiberAngle),1])];
            XCoordCelArray{c}=[FiberPosX];
            YCoordCelArray{c}=[FiberPosY];
          % Calculate MA, MVL, CSD
             c2m = mean(cos(2*FiberAngle*pi/180));
             s2m = mean(sin(2*FiberAngle*pi/180));
             MA = 180/pi*1/2*atan2(s2m, c2m);       % mean angle (deg)
             MVL = sqrt(s2m^2 + c2m^2);             % mean vector length
             CSD = 180/pi*1/2*sqrt(-2*log(MVL));    % circular standard deviation (deg)
             AD = 180/pi*1/2*sqrt(2*(1-MVL));       % angular deviation (deg)

%              clearvars phi N1 N2 E C FiberAngle MA MVL CSD AD;
%


%    formatspec= '%4.2f Pct Complete with fiber analysis\n\n';
%    fprintf(formatspec,(c/nf)*100)

        if c==1;
          est=toc/60;
        end

%       % Calculate MA, MVL, CSD
%             c2m = mean(cos(2*FiberAngle*pi/180));
%             s2m = mean(sin(2*FiberAngle*pi/180));
%             MA = 180/pi*1/2*atan2(s2m, c2m);       % mean angle (deg)
%             MVL = sqrt(s2m^2 + c2m^2);             % mean vector length
%             CSD = 180/pi*1/2*sqrt(-2*log(MVL));    % circular standard deviation (deg)
%             AD = 180/pi*1/2*sqrt(2*(1-MVL));       % angular deviation (deg)


      %Pull Basename from tiffiles structure: tiffiles(n).name(4:end-6)
fibstable(n,1)= {tiffiles(n).name(4:end-6)};
%Pull condition from tiffiles structure: tiffiles(n).name(4:5)
fibstable(n,2)={tiffiles(n).name(4:5)};
%Pull from tiffiles structure: tifiles(n).name(6:9)
fibstable(n,3)={tiffiles(n).name(6:9)};
%Pull Well from tiffiles structure: tiffiles(n).name(10) if 1:54
%                          tiffiles(n).name(11) if 55:108
if str2double(tiffiles(n).name(21:end-6))<55;
```

```
    fibstable(n,4)={tiffiles(n).name(10)};
elseif str2double(tiffiles(n).name(21:end-6))>54;
    fibstable(n,4)={tiffiles(n).name(11)};
end
%Pull Position from tiffiles structure: tiffiles(n).name(21:end)
fibstable(n,5)={str2double(tiffiles(n).name(21:end-6))};
%Assign Gel Row and Region Label based on position
  if fibstable{n,5}>=1 && ...
    fibstable{n,5}<=9 | ...
    fibstable{n,5}>=100 && ...
    fibstable{n,5}<=108;
    fibstable{n,6}={'Row1'};
    fibstable{n,7}={'Aligned'};

      elseif fibstable{n,5}>=10 && ...
      fibstable{n,5}<=18 | ...
      fibstable{n,5}>=91 && ...
      fibstable{n,5}<=99;
        fibstable{n,6}={'Row2'};
        fibstable{n,7}={'Aligned'};

        elseif fibstable{n,5}>=19 && ...
        fibstable{n,5}<=27 | ...
        fibstable{n,5}>=82 && ...
        fibstable{n,5}<=90;
          fibstable{n,6}={'Row3'};
          fibstable{n,7}={'Interface'};
%       row 1: pos 1-9 | 100-108
%       row 2: pos 10-18 | 91-99
%       row 3: pos 19-27 | 82-90
%       row 4: pos 28-36 | 73-81
%       row 5: pos 37-45 | 64-72
%       row 6: pos 46-54 | 55-63
            elseif fibstable{n,5}>=28 && ...
            fibstable{n,5}<=36 | ...
            fibstable{n,5}>=73 && ...
            fibstable{n,5}<=81;
              fibstable{n,6}={'Row4'};
              fibstable{n,7}={'Interface'};

              elseif fibstable{n,5}>=37 && ...
              fibstable{n,5}<=45 | ...
              fibstable{n,5}>=64 && ...
              fibstable{n,5}<=72;
                fibstable{n,6}={'Row5'};
                fibstable{n,7}={'Random'};

                elseif fibstable{n,5}>=46 && ...
                fibstable{n,5}<=54 | ...
                fibstable{n,5}>=55 && ...
                fibstable{n,5}<=63;
                  fibstable{n,6}={'Row6'};
                  fibstable{n,7}={'Random'};

  end
  %Pull date of run from tiffiles structure: tiffiles(n).name(12:17)
  fibstable{n,10}=str2num(tiffiles(n).name(12:17));
  %Calculate MVL for each image and store in fibstable{n,8}
  fibstable{n,8} = sqrt(s2m^2 + c2m^2);            % mean vector length
  %Calculate orientaiton for each image and store in fibstable{n,9};
  fibstable{n,9} = 180/pi*1/2*atan2(s2m, c2m);       % mean angle (deg)

            clearvars phi N1 N2 E C FiberAngle MA MVL CSD AD;

            Cellpercent=n/nf*100;
  if Cellpercent <= 5.004 & Cellpercent >= 4.996 | Cellpercent <= 10.004 &...
```

```
        Cellpercent >= 9.996 | Cellpercent <= 15.004 & Cellpercent >= 14.996 |...
        Cellpercent <= 20.004 & Cellpercent >= 19.996 |...
        Cellpercent <= 25.004 & Cellpercent >= 24.996 |...
        Cellpercent <= 30.004 & Cellpercent >= 29.996 |...
        Cellpercent <= 35.004 & Cellpercent >= 34.996 |...
        Cellpercent <= 40.004 & Cellpercent >= 39.996 |...
        Cellpercent <= 45.004 & Cellpercent >= 44.996 |...
        Cellpercent <= 55.004 & Cellpercent >= 54.996 |...
        Cellpercent <= 60.004 & Cellpercent >= 59.996 |...
        Cellpercent <= 65.004 & Cellpercent >= 64.996 |...
        Cellpercent <= 70.004 & Cellpercent >= 69.996 |...
        Cellpercent <= 75.004 & Cellpercent >= 74.996 |...
        Cellpercent <= 80.004 & Cellpercent >= 79.996 |...
        Cellpercent <= 85.004 & Cellpercent >= 84.996 |...
        Cellpercent <= 90.004 & Cellpercent >= 89.996 |...
        Cellpercent <= 95.004 & Cellpercent >= 94.996 |...
        Cellpercent <= 100.004 & Cellpercent >= 99.996;

fprintf(['Update: Fibstable Section approx. ' num2str(Cellpercent) ' Pct Complete.\n'])
    fprintf('working\n')
          rem=(nf-c)*est;
        fprintf(['Approx. ',num2str(rem),' minutes remaining in section.\n'])
  end
end


%End of commented section
%Maybe add another column at end of singlefibtable just to keep track of
%loops, add singlefibtable.LoopControl(startind:length(NameCelArray{n}))=ones(1:length(NameCelArray{n}
for n=1:nf;
    tic
    startind=find(singlefibtable.LoopControl == 0,1,'first'); %finds first 0 in column position
    singlefibtable.LoopControl(startind:startind+size(NameCelArray{n},1)-1)=ones(size(NameCelArray{n},1),1);
    singlefibtable.Basename(startind:startind+size(NameCelArray{n},1)-1)=NameCelArray{n}(:,1:end);% for m=sftH+1:sftH+h;
    singlefibtable.Orientation(startind:startind+size(NameCelArray{n},1)-1)=AngCelArray{n};
    singlefibtable.Xcoord(startind:startind+size(NameCelArray{n},1)-1)=XCoordCelArray{n};
    singlefibtable.Ycoord(startind:startind+size(NameCelArray{n},1)-1)=YCoordCelArray{n};

  if n==1;
       est=toc/60;
  end

  Cellpercent=n/nf*100;
  if   Cellpercent <= 5.004 & Cellpercent >= 4.996 | Cellpercent <= 10.004 &...
       Cellpercent >= 9.996 | Cellpercent <= 15.004 & Cellpercent >= 14.996 |...
       Cellpercent <= 20.004 & Cellpercent >= 19.996 |...
       Cellpercent <= 25.004 & Cellpercent >= 24.996 |...
       Cellpercent <= 30.004 & Cellpercent >= 29.996 |...
       Cellpercent <= 35.004 & Cellpercent >= 34.996 |...
       Cellpercent <= 40.004 & Cellpercent >= 39.996 |...
       Cellpercent <= 45.004 & Cellpercent >= 44.996 |...
       Cellpercent <= 55.004 & Cellpercent >= 54.996 |...
       Cellpercent <= 60.004 & Cellpercent >= 59.996 |...
       Cellpercent <= 65.004 & Cellpercent >= 64.996 |...
       Cellpercent <= 70.004 & Cellpercent >= 69.996 |...
       Cellpercent <= 75.004 & Cellpercent >= 74.996 |...
       Cellpercent <= 80.004 & Cellpercent >= 79.996 |...
       Cellpercent <= 85.004 & Cellpercent >= 84.996 |...
       Cellpercent <= 90.004 & Cellpercent >= 89.996 |...
       Cellpercent <= 95.004 & Cellpercent >= 94.996 |...
       Cellpercent <= 100.004 & Cellpercent >= 99.996;

fprintf(['Update: Singlefib Labeling Section approx. ' num2str(Cellpercent) ' Pct Complete.\n'])
    fprintf('working\n')
    rem=(nf-n)*est;
        fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])
```

```
  end
end
lastfib_ind=find(ismissing(singlefibtable.Basename(:)),1)-1;
singlefibtable=singlefibtable(1:lastfib_ind,:);
singlefibtable.Condition(1:end)=cellfun(@(x) x(1:2),singlefibtable.Basename(1:end),'UniformOutput',false);

tempcell=cellfun(@(x) str2num(x(18:end)),singlefibtable.Basename(1:end),'UniformOutput',false);
singlefibtable.Position(1:end)=cell2mat(tempcell);

tempcelldate=cellfun(@(x) str2num(x(9:14)),singlefibtable.Basename(1:end),'UniformOutput',false);
singlefibtable.DateofRun(1:end)=cell2mat(tempcelldate);

singlefibtable.Day(1:end)=cellfun(@(x) x(3:6),singlefibtable.Basename(1:end),'UniformOutput',false);

singlefibtable.Well(find(singlefibtable.Position(1:end)<55))=cellfun(@(x) x(7),...
    singlefibtable.Basename(find(singlefibtable.Position(1:end)<55)),'UniformOutput',false);
singlefibtable.Well(find(singlefibtable.Position(1:end)>54))=cellfun(@(x) x(8),...
    singlefibtable.Basename(find(singlefibtable.Position(1:end)>54)),'UniformOutput',false);

ind1=find(singlefibtable.Position(1:end)>=1 & ...
    singlefibtable.Position(1:end)<=9 | ...
    singlefibtable.Position(1:end)>=100 & ...
    singlefibtable.Position(1:end)<=108);
lind1=length(ind1);
singlefibtable.GelRow(ind1)=repmat({'Row1'},lind1,1);
singlefibtable.RegionLabel(ind1)=repmat({'Aligned'},lind1,1);

 ind2=find(singlefibtable.Position(1:end)>=10 & ...
       singlefibtable.Position(1:end)<=18 | ...
       singlefibtable.Position(1:end)>=91 & ...
       singlefibtable.Position(1:end)<=99);
lind2=length(ind2);
singlefibtable.GelRow(ind2)=repmat({'Row2'},lind2,1);
singlefibtable.RegionLabel(ind2)=repmat({'Aligned'},lind2,1);

ind3=find(singlefibtable.Position(1:end)>=19 & ...
          singlefibtable.Position(1:end)<=27 | ...
          singlefibtable.Position(1:end)>=82 & ...
          singlefibtable.Position(1:end)<=90);
lind3=length(ind3);
singlefibtable.GelRow(ind3)=repmat({'Row3'},lind3,1);
singlefibtable.RegionLabel(ind3)=repmat({'Interface'},lind3,1);
%       row 1: pos 1-9 | 100-108
%       row 2: pos 10-18 | 91-99
%       row 3: pos 19-27 | 82-90
%       row 4: pos 28-36 | 73-81
%       row 5: pos 37-45 | 64-72
%       row 6: pos 46-54 | 55-63
ind4=find(singlefibtable.Position(1:end)>=28 &...
             singlefibtable.Position(1:end)<=36 | ...
             singlefibtable.Position(1:end)>=73 & ...
             singlefibtable.Position(1:end)<=81);
lind4=length(ind4);
singlefibtable.GelRow(ind4)=repmat({'Row4'},lind4,1);
singlefibtable.RegionLabel(ind4)=repmat({'Interface'},lind4,1);

ind5=find(singlefibtable.Position(1:end)>=37 & ...
                singlefibtable.Position(1:end)<=45 | ...
                singlefibtable.Position(1:end)>=64 & ...
                singlefibtable.Position(1:end)<=72);
lind5=length(ind5);
singlefibtable.GelRow(ind5)=repmat({'Row5'},lind5,1);
singlefibtable.RegionLabel(ind5)=repmat({'Random'},lind5,1);

ind6=find(singlefibtable.Position(1:end)>=46 & ...
                   singlefibtable.Position(1:end)<=54 | ...
```

```
                singlefibtable.Position(1:end)>=55 & ...
                singlefibtable.Position(1:end)<=63);
lind6=length(ind6);
singlefibtable.GelRow(ind6)=repmat({'Row6'},lind6,1);
singlefibtable.RegionLabel(ind6)=repmat({'Random'},lind6,1);



save('FiberPortion_FiberSubunit_64d7.mat','singlefibtable','tableheaders','tabletypes','singlecelltable','imagetableheaders','imagetablety
pes','fibstable','cellstable','t','nf','-mat');
clear all
load('FiberPortion_FiberSubunit_64d7.mat')


% Cell Section - Cell Profiler import to t
singlecelltable{:,[1 7 8 10 11 12 13]}=[t{:,6},t{:,24},t{:,25},t{:,20},...
   t{:,12},t{:,15},t{:,19}];
singlecelltable.Basename(:)=t.Metadata_treatment(:);
singlecelltable.Xcoord(:)=t.Location_Center_X(:);
singlecelltable.Ycoord(:)=t.Location_Center_Y(:);
singlecelltable.Orientation(:)=t.AreaShape_Orientation(:);
singlecelltable.Eccentricity(:)=t.AreaShape_Eccentricity(:);
singlecelltable.MajorAxis(:)=t.AreaShape_MajorAxisLength(:);
singlecelltable.MinorAxis(:)=t.AreaShape_MinorAxisLength(:);
singlecelltable.Area(:)=t.AreaShape_Area(:);
singlecelltable.Perimeter(:)=t.AreaShape_Perimeter(:);


condtmp=cellfun(@(x) x(1:2),t{:,6},'UniformOutput',false);%table(singlecelltable.Basename);
singlecelltable.Condition(:)=condtmp;

positionnametmp=cellfun(@(x) str2num(x(18:end)),t{:,6},'UniformOutput',false);%table(singlecelltable.Position);
singlecelltable.Position(:)=cell2mat(positionnametmp);

DateofRuntmp=cellfun(@(x) str2num(x(9:14)),t{:,6},'UniformOutput',false);%table(singlecelltable.DateofRun);
singlecelltable.DateofRun(:)=cell2mat(DateofRuntmp);

Daytmp=cellfun(@(x) x(3:6),t{:,6},'UniformOutput',false);%table(singlecelltable.Day);
singlecelltable.Day(:)=Daytmp;

singlecelltable.Well(find(singlecelltable.Position(:)<55))=cellfun(@(x) x(7),...
   singlecelltable.Basename(find(singlecelltable.Position(:)<55)),'UniformOutput',false);
singlecelltable.Well(find(singlecelltable.Position(:)>54))=cellfun(@(x) x(8),...
   singlecelltable.Basename(find(singlecelltable.Position(:)>54)),'UniformOutput',false);

ind1cell=find(singlecelltable.Position(:)>=1 & ...
   singlecelltable.Position(:)<=9 | ...
   singlecelltable.Position(:)>=100 & ...
   singlecelltable.Position(:)<=108);
lind1cell=length(ind1cell);
singlecelltable.GelRow(ind1cell)=repmat({'Row1'},lind1cell,1);
singlecelltable.RegionLabel(ind1cell)=repmat({'Aligned'},lind1cell,1);

 ind2cell=find(singlecelltable.Position(:)>=10 & ...
       singlecelltable.Position(:)<=18 | ...
       singlecelltable.Position(:)>=91 & ...
       singlecelltable.Position(:)<=99);
lind2cell=length(ind2cell);
singlecelltable.GelRow(ind2cell)=repmat({'Row2'},lind2cell,1);
singlecelltable.RegionLabel(ind2cell)=repmat({'Aligned'},lind2cell,1);

ind3cell=find(singlecelltable.Position(:)>=19 & ...
          singlecelltable.Position(:)<=27 | ...
          singlecelltable.Position(:)>=82 & ...
          singlecelltable.Position(:)<=90);
lind3cell=length(ind3cell);
```

```
singlecelltable.GelRow(ind3cell)=repmat({'Row3'},lind3cell,1);
singlecelltable.RegionLabel(ind3cell)=repmat({'Interface'},lind3cell,1);
%       row 1: pos 1-9 | 100-108
%       row 2: pos 10-18 | 91-99
%       row 3: pos 19-27 | 82-90
%       row 4: pos 28-36 | 73-81
%       row 5: pos 37-45 | 64-72
%       row 6: pos 46-54 | 55-63
ind4cell=find(singlecelltable.Position(:)>=28 &...
            singlecelltable.Position(:)<=36 | ...
            singlecelltable.Position(:)>=73 & ...
            singlecelltable.Position(:)<=81);
lind4cell=length(ind4cell);
singlecelltable.GelRow(ind4cell)=repmat({'Row4'},lind4cell,1);
singlecelltable.RegionLabel(ind4cell)=repmat({'Interface'},lind4cell,1);

ind5cell=find(singlecelltable.Position(:)>=37 & ...
              singlecelltable.Position(:)<=45 | ...
              singlecelltable.Position(:)>=64 & ...
              singlecelltable.Position(:)<=72);
lind5cell=length(ind5cell);
singlecelltable.GelRow(ind5cell)=repmat({'Row5'},lind5cell,1);
singlecelltable.RegionLabel(ind5cell)=repmat({'Random'},lind5cell,1);

ind6cell=find(singlecelltable.Position(:)>=46 & ...
                singlecelltable.Position(:)<=54 | ...
                singlecelltable.Position(:)>=55 & ...
                singlecelltable.Position(:)<=63);
lind6cell=length(ind6cell);
singlecelltable.GelRow(ind6cell)=repmat({'Row6'},lind6cell,1);
singlecelltable.RegionLabel(ind6cell)=repmat({'Random'},lind6cell,1);

cellstable{:,[1:7 10]}(:)=fibstable{:,[1:7 10]}(:);
tic
for n=1:nf;
   tic
   CellAngle=singlecelltable.Orientation(find(singlecelltable.Basename(:) == cellstable.Basename(n)));

        % Calculate MA, MVL, CSD
            Cellc2m = mean(cos(2*CellAngle*pi/180));
            Cells2m = mean(sin(2*CellAngle*pi/180));
            CellMA = 180/pi*1/2*atan2(Cells2m, Cellc2m);       % mean angle (deg)
            CellMVL = sqrt(Cells2m^2 + Cellc2m^2);             % mean vector length
            CellCSD = 180/pi*1/2*sqrt(-2*log(CellMVL));    % circular standard deviation (deg)
            CellAD = 180/pi*1/2*sqrt(2*(1-CellMVL));       % angular deviation (deg)

cellstable.MVL(n)=CellMVL;
cellstable.Orientation(n)=CellMA;
clear CellAngle Cellc2m Cells2m CellMA Cell MVL CellCSD Cell AD
  if n==1;
      est=toc/60;
  end

Cellpercent=(n/nf)*100;
 if Cellpercent <= 5.004 & Cellpercent >= 4.996 | Cellpercent <= 10.004 &...
     Cellpercent >= 9.996 | Cellpercent <= 15.004 & Cellpercent >= 14.996 |...
     Cellpercent <= 20.004 & Cellpercent >= 19.996 |...
     Cellpercent <= 25.004 & Cellpercent >= 24.996 |...
     Cellpercent <= 30.004 & Cellpercent >= 29.996 |...
     Cellpercent <= 35.004 & Cellpercent >= 34.996 |...
     Cellpercent <= 40.004 & Cellpercent >= 39.996 |...
     Cellpercent <= 45.004 & Cellpercent >= 44.996 |...
     Cellpercent <= 55.004 & Cellpercent >= 54.996 |...
     Cellpercent <= 60.004 & Cellpercent >= 59.996 |...
     Cellpercent <= 65.004 & Cellpercent >= 64.996 |...
     Cellpercent <= 70.004 & Cellpercent >= 69.996 |...
```

```
        Cellpercent <= 75.004 & Cellpercent >= 74.996 |...
        Cellpercent <= 80.004 & Cellpercent >= 79.996 |...
        Cellpercent <= 85.004 & Cellpercent >= 84.996 |...
        Cellpercent <= 90.004 & Cellpercent >= 89.996 |...
        Cellpercent <= 95.004 & Cellpercent >= 94.996 |...
        Cellpercent <= 100.004 & Cellpercent >= 99.996;
fprintf(['Update: Cell Section approx. ' num2str(Cellpercent) ' Pct Complete.\n'])
    fprintf('working\n')
    rem=(nf-n)*est;
        fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])
 end
end
celltoc=toc;


save('Fiber_and_Cell_Portion_FiberSubunit_64d7.mat','singlefibtable','tableheaders','tabletypes','singlecelltable','imagetableheaders','i
magetabletypes','fibstable','cellstable','t','-mat');
clear all
load('Fiber_and_Cell_Portion_FiberSubunit_64d7.mat')

% Global Coordinate Creation
coordnums=[100:1:108]-9*[0:11]';
pixpermicron=1024/632;
%x distance between image 699um
imstepx=pixpermicron*699;
cellstorematx={};
cellstorematy={};
%y distance between image 718um
imstepy=pixpermicron*718;
%Wells A and D
for num=1:9;
    for numx=1:12;
    cellind=find(singlecelltable.Position(:)==coordnums(numx,num));
    fibind=find(singlefibtable.Position(:)==coordnums(numx,num));

    singlecelltable.GlobalPixelXcoord(cellind)=(singlecelltable.Xcoord(cellind)+imstepx*(num-1));
    singlefibtable.GlobalPixelXcoord(fibind)=(singlefibtable.Xcoord(fibind)+imstepx*(num-1));
    if numx<7;
    singlecelltable.GlobalPixelYcoord(cellind)=(singlecelltable.Ycoord(cellind)+imstepy*(numx-1));
    singlefibtable.GlobalPixelYcoord(fibind)=((singlefibtable.Ycoord(fibind))+imstepy*(numx-1));
    else
    singlecelltable.GlobalPixelYcoord(cellind)=((1024-singlecelltable.Ycoord(cellind))+imstepy*(12-numx));
    singlecelltable.Orientation(cellind)=singlecelltable.Orientation(cellind)*-1;
    singlefibtable.GlobalPixelYcoord(fibind)=((1024-singlefibtable.Ycoord(fibind))+imstepy*(12-numx));
    singlefibtable.Orientation(fibind)=singlefibtable.Orientation(fibind)*-1;
    end

    end
end

% Finds cells with axis ratio greater than or equal to 1.5 and labels them
% as polarized
singlecelltable.MajMinRat(:)=singlecelltable.MajorAxis(:)./singlecelltable.MinorAxis(:);
polarind=find(singlecelltable.MajMinRat(:)>=1.5);
singlecelltable.Polarized(polarind)=categorical({'yes'});
singlecelltable.Polarized(isundefined(singlecelltable.Polarized(:)))=categorical({'no'});
clear polarind
%Find cells with area greater than or equal to 800 cellprofiler units
spreadind=find(singlecelltable.Area(:)>=800);
singlecelltable.Spread(spreadind)=categorical({'yes'});
singlecelltable.Spread(isundefined(singlecelltable.Spread(:)))=categorical({'no'});

save('GlobalCoord_SingleFiber_SingleCell_FiberSubunit_64d7.mat','singlefibtable','singlecelltable','fibstable','cellstable','-mat');
clear all
load('GlobalCoord_SingleFiber_SingleCell_FiberSubunit_64d7.mat')
```

```
% Fiber Zone Analysis

cond={'va','ra'};
lcond=length(cond);
day={'day1';'day3';'day7'};
lday=length(day);
well={'a';'b';'c';'d'};
lwell=length(well);
DOR=[121319;200128;201007];    %date of run
lDOR=length(DOR);
numsituations=lcond*lday*lwell*lDOR;
markerconditions={'r-*','r-+','b-*','b-+','g-*','g-+','k-*','k-+','m-*','m-+'};
markerscatconds={'r*','r+','b*','b+','g*','g+','k*','k+','m*','m+'};

%determine zone sizes to implement grid is ~10000x6800
%zone sizes: ~1%, ~3%, ~5%, ~10% ~20%, ~35%, ~50%

%find size of analysis regions
rang=[max([singlecelltable.GlobalPixelXcoord;...
   singlefibtable.GlobalPixelXcoord]),...
   max([singlecelltable.GlobalPixelYcoord;...
   singlefibtable.GlobalPixelYcoord])];
%find aspect ratio long/short
rangratio=rang(1)/rang(2);
%percentage of minimum dimension to size analysis zones
approxpctzonebreakdown=[.01 .03 .05 .1 .2 .33 .5];
%size of square analysis zones
zsize=min(rang).*approxpctzonebreakdown;
%find number of zones per dimension
numzones=[floor(rangratio.*min(rang)./zsize);...
   floor(min(rang)./zsize)];
%Determine initial pixel coord to begin zone breakdown
%(x,y)
zoneinit=(rang'-([zsize;zsize].*numzones))/2;
%seed analysis zones by centroid coords
for n1=1:2;
   for n2=1:length(zoneinit);
      zonecentroids{n1,n2}=[(zoneinit(n1,n2)+(zsize(n2)/2)):zsize(n2):...
         (zsize(n2)*numzones(n1,n2)-(zsize(n2)/2)+(zoneinit(n1,n2)))];
   end
end


for n1=1:2;
   for n2=1:length(zoneinit);
         coord{n2}=[repelem(zonecentroids{1,n2}',...
            length(zonecentroids{2,n2}),1),...
            repmat(zonecentroids{2,n2}',...
            length(zonecentroids{1,n2}),1)];
   end
end
numzones(3,:)=numzones(1,:).*numzones(2,:);
%create table for each fibercentric zone set
ZoneVars={'Condition','Day','Well','DateofRun','Xmin','Xcentroid','Xmax',...
   'Ymin','Ycentroid','Ymax','FiberMVL','CellMVL','CellCount',...
   'PolarizedCellCount','SpreadCellCount',...
   'FibMeanAngle','CellMeanAngle','CellPolarizedMeanArea','CellMeanEccent',...
   'CellMeanAspectRatio','CellSpreadMeanArea','CellMeanArea',...
   'CellPolarizedMeanEccent','CellPolarizedMeanAspectRatio',...
   'CellPolarizedMeanAngle','CellUnpolarizedMVL'};
VarTypes={'string','string','string','double','double','double','double',...
   'double','double','double','double','double','double','double',...
   'double','double','double','double','double','double','double',...
   'double','double','double','double','double'};
FilteredFibZoneSize05PctTable=table('Size',[(numzones(3,3)*numsituations),...
    length(ZoneVars)],'VariableNames',ZoneVars,'VariableTypes',VarTypes);
```

```
FilteredFibZoneSize10PctTable=table('Size',[(numzones(3,4)*numsituations),...
    length(ZoneVars)],'VariableNames',ZoneVars,'VariableTypes',VarTypes);
FilteredFibZoneSize20PctTable=table('Size',[(numzones(3,5)*numsituations),...
    length(ZoneVars)],'VariableNames',ZoneVars,'VariableTypes',VarTypes);
FilteredFibZoneSize33PctTable=table('Size',[(numzones(3,6)*numsituations),...
    length(ZoneVars)],'VariableNames',ZoneVars,'VariableTypes',VarTypes);
FilteredFibZoneSize50PctTable=table('Size',[(numzones(3,7)*numsituations),...
   length(ZoneVars)],'VariableNames',ZoneVars,'VariableTypes',VarTypes);

%Assign Conditions to Rows
count=0;
beg=zeros(1,7);
for nDOR=1:lDOR;
   for ncond=1:lcond;
      if DOR(nDOR) == 121319 & cond{ncond} == 'va'
         continue
      elseif DOR(nDOR) == 200128 & cond{ncond} == 'ra'
         continue
      end
      for nwell=1:lwell;
         for nday=1:lday;
            count=count+1;

%
 %           %Zone Size 3 Descriptors Population
            FilteredFibZoneSize05PctTable(beg(3)+1:numzones(3,3)*count,[1:10])=...
               table(repmat(cond(ncond),numzones(3,3),1),...
               repmat(day(nday),numzones(3,3),1),...
               repmat(well(nwell),numzones(3,3),1),...
               repmat(DOR(nDOR),numzones(3,3),1),...
               coord{3}(:,1)-(zsize(3)/2),...
               coord{3}(:,1),...
               coord{3}(:,1)+(zsize(3)/2),...
               coord{3}(:,2)-(zsize(3)/2),...
               coord{3}(:,2),...
               coord{3}(:,2)+(zsize(3)/2));

 %           %Zone Size 4 Descriptors Population
            FilteredFibZoneSize10PctTable(beg(4)+1:numzones(3,4)*count,[1:10])=...
               table(repmat(cond(ncond),numzones(3,4),1),...
               repmat(day(nday),numzones(3,4),1),...
               repmat(well(nwell),numzones(3,4),1),...
               repmat(DOR(nDOR),numzones(3,4),1),...
               coord{4}(:,1)-(zsize(4)/2),...
               coord{4}(:,1),...
               coord{4}(:,1)+(zsize(4)/2),...
               coord{4}(:,2)-(zsize(4)/2),...
               coord{4}(:,2),...
               coord{4}(:,2)+(zsize(4)/2));

 %           %Zone Size 5 Descriptors Population
            FilteredFibZoneSize20PctTable(beg(5)+1:numzones(3,5)*count,[1:10])=...
               table(repmat(cond(ncond),numzones(3,5),1),...
               repmat(day(nday),numzones(3,5),1),...
               repmat(well(nwell),numzones(3,5),1),...
               repmat(DOR(nDOR),numzones(3,5),1),...
               coord{5}(:,1)-(zsize(5)/2),...
               coord{5}(:,1),...
               coord{5}(:,1)+(zsize(5)/2),...
               coord{5}(:,2)-(zsize(5)/2),...
               coord{5}(:,2),...
               coord{5}(:,2)+(zsize(5)/2));

 %           %Zone Size 6 Descriptors Population
            FilteredFibZoneSize33PctTable(beg(6)+1:numzones(3,6)*count,[1:10])=...
               table(repmat(cond(ncond),numzones(3,6),1),...
```

```
            repmat(day(nday),numzones(3,6),1),...
            repmat(well(nwell),numzones(3,6),1),...
            repmat(DOR(nDOR),numzones(3,6),1),...
            coord{6}(:,1)-(zsize(6)/2),...
            coord{6}(:,1),...
            coord{6}(:,1)+(zsize(6)/2),...
            coord{6}(:,2)-(zsize(6)/2),...
            coord{6}(:,2),...
            coord{6}(:,2)+(zsize(6)/2));

%             %Zone Size 7 Descriptors Population
          FilteredFibZoneSize50PctTable(beg(7)+1:numzones(3,7)*count,[1:10])=...
            table(repmat(cond(ncond),numzones(3,7),1),...
            repmat(day(nday),numzones(3,7),1),...
            repmat(well(nwell),numzones(3,7),1),...
            repmat(DOR(nDOR),numzones(3,7),1),...
            coord{7}(:,1)-(zsize(7)/2),...
            coord{7}(:,1),...
            coord{7}(:,1)+(zsize(7)/2),...
            coord{7}(:,2)-(zsize(7)/2),...
            coord{7}(:,2),...
            coord{7}(:,2)+(zsize(7)/2));

          beg=[numzones(3,1)*count,numzones(3,2)*count,...
            numzones(3,3)*count,numzones(3,4)*count,...
            numzones(3,5)*count,numzones(3,6)*count,...
            numzones(3,7)*count];
        end
      end
    end
end

%%%Pct 5 Table
tic
for n=1:height05;

          singlecellind=find(...
            singlecelltable.Condition(:)==FilteredFibZoneSize05PctTable.Condition(n) &...
            singlecelltable.Day(:)==FilteredFibZoneSize05PctTable.Day(n) &...
            singlecelltable.Well(:)==FilteredFibZoneSize05PctTable.Well(n) &...
            singlecelltable.DateofRun(:)==FilteredFibZoneSize05PctTable.DateofRun(n) &...
            singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize05PctTable.Xmin(n) &...
            singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize05PctTable.Xmax(n) &...
            singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize05PctTable.Ymin(n) &...
            singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize05PctTable.Ymax(n));

          polarizedsinglecellind=find(...
            singlecelltable.Condition(:)==FilteredFibZoneSize05PctTable.Condition(n) &...
            singlecelltable.Day(:)==FilteredFibZoneSize05PctTable.Day(n) &...
            singlecelltable.Well(:)==FilteredFibZoneSize05PctTable.Well(n) &...
            singlecelltable.DateofRun(:)==FilteredFibZoneSize05PctTable.DateofRun(n) &...
            singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize05PctTable.Xmin(n) &...
            singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize05PctTable.Xmax(n) &...
            singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize05PctTable.Ymin(n) &...
            singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize05PctTable.Ymax(n) &...
            singlecelltable.Polarized(:)=='yes');

          spreadsinglecellind=find(...
            singlecelltable.Condition(:)==FilteredFibZoneSize05PctTable.Condition(n) &...
            singlecelltable.Day(:)==FilteredFibZoneSize05PctTable.Day(n) &...
            singlecelltable.Well(:)==FilteredFibZoneSize05PctTable.Well(n) &...
            singlecelltable.DateofRun(:)==FilteredFibZoneSize05PctTable.DateofRun(n) &...
            singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize05PctTable.Xmin(n) &...
            singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize05PctTable.Xmax(n) &...
            singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize05PctTable.Ymin(n) &...
            singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize05PctTable.Ymax(n) &...
```

```
        singlecelltable.Spread(:)=='yes');


      singlefibind=find(...
        singlefibtable.Condition(:)==FilteredFibZoneSize05PctTable.Condition(n) &...
        singlefibtable.Day(:)==FilteredFibZoneSize05PctTable.Day(n) &...
        singlefibtable.Well(:)==FilteredFibZoneSize05PctTable.Well(n) &...
        singlefibtable.DateofRun(:)==FilteredFibZoneSize05PctTable.DateofRun(n) &...
        singlefibtable.GlobalPixelXcoord(:)>FilteredFibZoneSize05PctTable.Xmin(n) &...
        singlefibtable.GlobalPixelXcoord(:)<FilteredFibZoneSize05PctTable.Xmax(n) &...
        singlefibtable.GlobalPixelYcoord(:)>FilteredFibZoneSize05PctTable.Ymin(n) &...
        singlefibtable.GlobalPixelYcoord(:)<FilteredFibZoneSize05PctTable.Ymax(n));

FilteredFibZoneSize05PctTable.FiberMVL(n) =...
sqrt((mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2 +...
    (mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2);

FilteredFibZoneSize05PctTable.CellCount(n) = length(singlecellind);

FilteredFibZoneSize05PctTable.PolarizedCellCount(n)=length(polarizedsinglecellind);

FilteredFibZoneSize05PctTable.SpreadCellCount(n)=length(spreadsinglecellind);

if FilteredFibZoneSize05PctTable.PolarizedCellCount(n)>1;
  FilteredFibZoneSize05PctTable.CellMVL(n) =...
     sqrt((mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2 +...
     (mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2);
else
  FilteredFibZoneSize05PctTable.CellMVL(n) = NaN;
end

FilteredFibZoneSize05PctTable.FibMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)),...
   mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)));

FilteredFibZoneSize05PctTable.CellMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(singlecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(singlecellind)*pi/180)));

FilteredFibZoneSize05PctTable.CellPolarizedMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)));


FilteredFibZoneSize05PctTable.CellPolarizedMeanArea(n) = mean(...
   singlecelltable.Area(polarizedsinglecellind));

FilteredFibZoneSize05PctTable.CellSpreadMeanArea(n) = mean(...
   singlecelltable.Area(spreadsinglecellind));

FilteredFibZoneSize05PctTable.CellMeanArea(n) = mean(...
   singlecelltable.Area(singlecellind));

FilteredFibZoneSize05PctTable.CellMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(singlecellind));

FilteredFibZoneSize05PctTable.CellMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(singlecellind)./...
   singlecelltable.MinorAxis(singlecellind));

FilteredFibZoneSize05PctTable.CellPolarizedMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(polarizedsinglecellind));

FilteredFibZoneSize05PctTable.CellPolarizedMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(polarizedsinglecellind)./...
   singlecelltable.MinorAxis(polarizedsinglecellind));
```

```
    if n==1;
        est=toc/60;
    end
fract=100*(n/height05);
    if fract <= 5.1 & fract >= 4.9 | fract <= 10.1 &...
       fract >= 9.9 | fract <= 15.1 & fract >= 14.9 |...
       fract <= 20.1 & fract >= 19.9 |...
       fract <= 25.1 & fract >= 24.9 |...
       fract <= 30.1 & fract >= 29.9 |...
       fract <= 35.1 & fract >= 34.9 |...
       fract <= 40.1 & fract >= 39.9 |...
       fract <= 45.1 & fract >= 44.9 |...
       fract <= 55.1 & fract >= 54.9 |...
       fract <= 60.1 & fract >= 59.9 |...
       fract <= 65.1 & fract >= 64.9 |...
       fract <= 70.1 & fract >= 69.9 |...
       fract <= 75.1 & fract >= 74.9 |...
       fract <= 80.1 & fract >= 79.9 |...
       fract <= 85.1 & fract >= 84.9 |...
       fract <= 90.1 & fract >= 89.9 |...
       fract <= 95.1 & fract >= 94.9 |...
       fract <= 100.1 & fract >= 99.9;
fprintf(['5pct table Approx. ' num2str(fract) ' Pct Complete'])
    rem=(height05-n)*est;
    fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])

    end
end
toc
%%%%%Pct 10 Table
tic
for n=1:height10;

            singlecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize10PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize10PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize10PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize10PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize10PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize10PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize10PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize10PctTable.Ymax(n));

            polarizedsinglecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize10PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize10PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize10PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize10PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize10PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize10PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize10PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize10PctTable.Ymax(n) &...
               singlecelltable.Polarized(:)=='yes');

            spreadsinglecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize10PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize10PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize10PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize10PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize10PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize10PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize10PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize10PctTable.Ymax(n) &...
               singlecelltable.Spread(:)=='yes');
```

```
        singlefibind=find(...
            singlefibtable.Condition(:)==FilteredFibZoneSize10PctTable.Condition(n) &...
            singlefibtable.Day(:)==FilteredFibZoneSize10PctTable.Day(n) &...
            singlefibtable.Well(:)==FilteredFibZoneSize10PctTable.Well(n) &...
            singlefibtable.DateofRun(:)==FilteredFibZoneSize10PctTable.DateofRun(n) &...
            singlefibtable.GlobalPixelXcoord(:)>FilteredFibZoneSize10PctTable.Xmin(n) &...
            singlefibtable.GlobalPixelXcoord(:)<FilteredFibZoneSize10PctTable.Xmax(n) &...
            singlefibtable.GlobalPixelYcoord(:)>FilteredFibZoneSize10PctTable.Ymin(n) &...
            singlefibtable.GlobalPixelYcoord(:)<FilteredFibZoneSize10PctTable.Ymax(n));

FilteredFibZoneSize10PctTable.FiberMVL(n) =...
sqrt((mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2 +...
    (mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2);


FilteredFibZoneSize10PctTable.CellCount(n) = length(singlecellind);

FilteredFibZoneSize10PctTable.PolarizedCellCount(n)=length(polarizedsinglecellind);

FilteredFibZoneSize10PctTable.SpreadCellCount(n)=length(spreadsinglecellind);

if FilteredFibZoneSize10PctTable.PolarizedCellCount(n)>1;
   FilteredFibZoneSize10PctTable.CellMVL(n) =...
      sqrt((mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2 +...
      (mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2);
else
   FilteredFibZoneSize10PctTable.CellMVL(n) = NaN;
end


FilteredFibZoneSize10PctTable.FibMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)),...
   mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)));

FilteredFibZoneSize10PctTable.CellMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(singlecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(singlecellind)*pi/180)));

FilteredFibZoneSize10PctTable.CellPolarizedMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)));



FilteredFibZoneSize10PctTable.CellPolarizedMeanArea(n) = mean(...
   singlecelltable.Area(polarizedsinglecellind));

FilteredFibZoneSize10PctTable.CellSpreadMeanArea(n) = mean(...
   singlecelltable.Area(spreadsinglecellind));

FilteredFibZoneSize10PctTable.CellMeanArea(n) = mean(...
   singlecelltable.Area(singlecellind));

FilteredFibZoneSize10PctTable.CellMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(singlecellind));

FilteredFibZoneSize10PctTable.CellMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(singlecellind)./...
   singlecelltable.MinorAxis(singlecellind));

FilteredFibZoneSize10PctTable.CellPolarizedMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(polarizedsinglecellind));

FilteredFibZoneSize10PctTable.CellPolarizedMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(polarizedsinglecellind)./...
```

```
    singlecelltable.MinorAxis(polarizedsinglecellind));

  if n==1;
      est=toc/60;
  end
fract=100*(n/height10);
  if fract <= 5.1 & fract >= 4.9 | fract <= 10.1 &...
     fract >= 9.9 | fract <= 15.1 & fract >= 14.9 |...
     fract <= 20.1 & fract >= 19.9 |...
     fract <= 25.1 & fract >= 24.9 |...
     fract <= 30.1 & fract >= 29.9 |...
     fract <= 35.1 & fract >= 34.9 |...
     fract <= 40.1 & fract >= 39.9 |...
     fract <= 45.1 & fract >= 44.9 |...
     fract <= 55.1 & fract >= 54.9 |...
     fract <= 60.1 & fract >= 59.9 |...
     fract <= 65.1 & fract >= 64.9 |...
     fract <= 70.1 & fract >= 69.9 |...
     fract <= 75.1 & fract >= 74.9 |...
     fract <= 80.1 & fract >= 79.9 |...
     fract <= 85.1 & fract >= 84.9 |...
     fract <= 90.1 & fract >= 89.9 |...
     fract <= 95.1 & fract >= 94.9 |...
     fract <= 100.1 & fract >= 99.9;
fprintf(['10pct table Approx. ' num2str(fract) ' Pct Complete'])
  rem=(height10-n)*est;
  fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])

  end
end
toc
%%%%%Pct 20 Table
tic
for n=1:height20;

        singlecellind=find(...
          singlecelltable.Condition(:)==FilteredFibZoneSize20PctTable.Condition(n) &...
          singlecelltable.Day(:)==FilteredFibZoneSize20PctTable.Day(n) &...
          singlecelltable.Well(:)==FilteredFibZoneSize20PctTable.Well(n) &...
          singlecelltable.DateofRun(:)==FilteredFibZoneSize20PctTable.DateofRun(n) &...
          singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize20PctTable.Xmin(n) &...
          singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize20PctTable.Xmax(n) &...
          singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize20PctTable.Ymin(n) &...
          singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize20PctTable.Ymax(n));

        polarizedsinglecellind=find(...
          singlecelltable.Condition(:)==FilteredFibZoneSize20PctTable.Condition(n) &...
          singlecelltable.Day(:)==FilteredFibZoneSize20PctTable.Day(n) &...
          singlecelltable.Well(:)==FilteredFibZoneSize20PctTable.Well(n) &...
          singlecelltable.DateofRun(:)==FilteredFibZoneSize20PctTable.DateofRun(n) &...
          singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize20PctTable.Xmin(n) &...
          singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize20PctTable.Xmax(n) &...
          singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize20PctTable.Ymin(n) &...
          singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize20PctTable.Ymax(n) &...
          singlecelltable.Polarized(:)=='yes');

        spreadsinglecellind=find(...
          singlecelltable.Condition(:)==FilteredFibZoneSize20PctTable.Condition(n) &...
          singlecelltable.Day(:)==FilteredFibZoneSize20PctTable.Day(n) &...
          singlecelltable.Well(:)==FilteredFibZoneSize20PctTable.Well(n) &...
          singlecelltable.DateofRun(:)==FilteredFibZoneSize20PctTable.DateofRun(n) &...
          singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize20PctTable.Xmin(n) &...
          singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize20PctTable.Xmax(n) &...
          singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize20PctTable.Ymin(n) &...
          singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize20PctTable.Ymax(n) &...
          singlecelltable.Spread(:)=='yes');
```

```
        singlefibind=find(...
            singlefibtable.Condition(:)==FilteredFibZoneSize20PctTable.Condition(n) &...
            singlefibtable.Day(:)==FilteredFibZoneSize20PctTable.Day(n) &...
            singlefibtable.Well(:)==FilteredFibZoneSize20PctTable.Well(n) &...
            singlefibtable.DateofRun(:)==FilteredFibZoneSize20PctTable.DateofRun(n) &...
            singlefibtable.GlobalPixelXcoord(:)>FilteredFibZoneSize20PctTable.Xmin(n) &...
            singlefibtable.GlobalPixelXcoord(:)<FilteredFibZoneSize20PctTable.Xmax(n) &...
            singlefibtable.GlobalPixelYcoord(:)>FilteredFibZoneSize20PctTable.Ymin(n) &...
            singlefibtable.GlobalPixelYcoord(:)<FilteredFibZoneSize20PctTable.Ymax(n));

FilteredFibZoneSize20PctTable.FiberMVL(n) =...
sqrt((mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2 +...
    (mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2);


FilteredFibZoneSize20PctTable.CellCount(n) = length(singlecellind);

FilteredFibZoneSize20PctTable.PolarizedCellCount(n)=length(polarizedsinglecellind);

FilteredFibZoneSize20PctTable.SpreadCellCount(n)=length(spreadsinglecellind);

if FilteredFibZoneSize20PctTable.PolarizedCellCount(n)>1;
   FilteredFibZoneSize20PctTable.CellMVL(n) =...
      sqrt((mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2 +...
      (mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2);
else
   FilteredFibZoneSize20PctTable.CellMVL(n) = NaN;
end


FilteredFibZoneSize20PctTable.FibMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)),...
   mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)));

FilteredFibZoneSize20PctTable.CellMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(singlecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(singlecellind)*pi/180)));

FilteredFibZoneSize20PctTable.CellPolarizedMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)));


FilteredFibZoneSize20PctTable.CellPolarizedMeanArea(n) = mean(...
   singlecelltable.Area(polarizedsinglecellind));

FilteredFibZoneSize20PctTable.CellSpreadMeanArea(n) = mean(...
   singlecelltable.Area(spreadsinglecellind));

FilteredFibZoneSize20PctTable.CellMeanArea(n) = mean(...
   singlecelltable.Area(singlecellind));

FilteredFibZoneSize20PctTable.CellMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(singlecellind));

FilteredFibZoneSize20PctTable.CellMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(singlecellind)./...
   singlecelltable.MinorAxis(singlecellind));


FilteredFibZoneSize20PctTable.CellPolarizedMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(polarizedsinglecellind));
```

```
  FilteredFibZoneSize20PctTable.CellPolarizedMeanAspectRatio(n) = mean(...
     singlecelltable.MajorAxis(polarizedsinglecellind)./...
     singlecelltable.MinorAxis(polarizedsinglecellind));

    if n==1;
        est=toc/60;
    end
fract=100*(n/height20);
    if fract <= 5.1 & fract >= 4.9 | fract <= 10.1 &...
       fract >= 9.9 | fract <= 15.1 & fract >= 14.9 |...
       fract <= 20.1 & fract >= 19.9 |...
       fract <= 25.1 & fract >= 24.9 |...
       fract <= 30.1 & fract >= 29.9 |...
       fract <= 35.1 & fract >= 34.9 |...
       fract <= 40.1 & fract >= 39.9 |...
       fract <= 45.1 & fract >= 44.9 |...
       fract <= 55.1 & fract >= 54.9 |...
       fract <= 60.1 & fract >= 59.9 |...
       fract <= 65.1 & fract >= 64.9 |...
       fract <= 70.1 & fract >= 69.9 |...
       fract <= 75.1 & fract >= 74.9 |...
       fract <= 80.1 & fract >= 79.9 |...
       fract <= 85.1 & fract >= 84.9 |...
       fract <= 90.1 & fract >= 89.9 |...
       fract <= 95.1 & fract >= 94.9 |...
       fract <= 100.1 & fract >= 99.9;
fprintf(['Approx. ' num2str(fract) ' Pct Complete'])
    rem=(height20-n)*est;
    fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])

    end
end
toc
%%%%%Pct 33 Table
tic
for n=1:height33;

            singlecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize33PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize33PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize33PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize33PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize33PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize33PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize33PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize33PctTable.Ymax(n));

            polarizedsinglecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize33PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize33PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize33PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize33PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize33PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize33PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize33PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize33PctTable.Ymax(n) &...
               singlecelltable.Polarized(:)=='yes');

            spreadsinglecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize33PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize33PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize33PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize33PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize33PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize33PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize33PctTable.Ymin(n) &...
```

```
        singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize33PctTable.Ymax(n) &...
        singlecelltable.Spread(:)=='yes');


      singlefibind=find(...
        singlefibtable.Condition(:)==FilteredFibZoneSize33PctTable.Condition(n) &...
        singlefibtable.Day(:)==FilteredFibZoneSize33PctTable.Day(n) &...
        singlefibtable.Well(:)==FilteredFibZoneSize33PctTable.Well(n) &...
        singlefibtable.DateofRun(:)==FilteredFibZoneSize33PctTable.DateofRun(n) &...
        singlefibtable.GlobalPixelXcoord(:)>FilteredFibZoneSize33PctTable.Xmin(n) &...
        singlefibtable.GlobalPixelXcoord(:)<FilteredFibZoneSize33PctTable.Xmax(n) &...
        singlefibtable.GlobalPixelYcoord(:)>FilteredFibZoneSize33PctTable.Ymin(n) &...
        singlefibtable.GlobalPixelYcoord(:)<FilteredFibZoneSize33PctTable.Ymax(n));

FilteredFibZoneSize33PctTable.FiberMVL(n) =...
sqrt((mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2 +...
    (mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2);


FilteredFibZoneSize33PctTable.CellCount(n) = length(singlecellind);

FilteredFibZoneSize33PctTable.PolarizedCellCount(n)=length(polarizedsinglecellind);

FilteredFibZoneSize33PctTable.SpreadCellCount(n)=length(spreadsinglecellind);

if FilteredFibZoneSize33PctTable.PolarizedCellCount(n)>1;
   FilteredFibZoneSize33PctTable.CellMVL(n) =...
      sqrt((mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2 +...
      (mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2);
else
   FilteredFibZoneSize33PctTable.CellMVL(n) = NaN;
end


FilteredFibZoneSize33PctTable.FibMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)),...
   mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)));

FilteredFibZoneSize33PctTable.CellMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(singlecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(singlecellind)*pi/180)));

FilteredFibZoneSize33PctTable.CellPolarizedMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)));


FilteredFibZoneSize33PctTable.CellPolarizedMeanArea(n) = mean(...
   singlecelltable.Area(polarizedsinglecellind));

FilteredFibZoneSize33PctTable.CellSpreadMeanArea(n) = mean(...
   singlecelltable.Area(spreadsinglecellind));

FilteredFibZoneSize33PctTable.CellMeanArea(n) = mean(...
   singlecelltable.Area(singlecellind));

FilteredFibZoneSize33PctTable.CellMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(singlecellind));

FilteredFibZoneSize33PctTable.CellMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(singlecellind)./...
   singlecelltable.MinorAxis(singlecellind));


FilteredFibZoneSize33PctTable.CellPolarizedMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(polarizedsinglecellind));
```

```
  FilteredFibZoneSize33PctTable.CellPolarizedMeanAspectRatio(n) = mean(...
     singlecelltable.MajorAxis(polarizedsinglecellind)./...
     singlecelltable.MinorAxis(polarizedsinglecellind));
    if n==1;
        est=toc/60;
    end
fract=100*(n/height33);
    if fract <= 5.1 & fract >= 4.9 | fract <= 10.1 &...
       fract >= 9.9 | fract <= 15.1 & fract >= 14.9 |...
       fract <= 20.1 & fract >= 19.9 |...
       fract <= 25.1 & fract >= 24.9 |...
       fract <= 30.1 & fract >= 29.9 |...
       fract <= 35.1 & fract >= 34.9 |...
       fract <= 40.1 & fract >= 39.9 |...
       fract <= 45.1 & fract >= 44.9 |...
       fract <= 55.1 & fract >= 54.9 |...
       fract <= 60.1 & fract >= 59.9 |...
       fract <= 65.1 & fract >= 64.9 |...
       fract <= 70.1 & fract >= 69.9 |...
       fract <= 75.1 & fract >= 74.9 |...
       fract <= 80.1 & fract >= 79.9 |...
       fract <= 85.1 & fract >= 84.9 |...
       fract <= 90.1 & fract >= 89.9 |...
       fract <= 95.1 & fract >= 94.9 |...
       fract <= 100.1 & fract >= 99.9;
fprintf(['Approx. ' num2str(fract) ' Pct Complete'])
    rem=(height33-n)*est;
    fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])

    end
end
toc
%%%%%Pct 50 Table
tic
for n=1:height50;

            singlecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize50PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize50PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize50PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize50PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize50PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize50PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize50PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize50PctTable.Ymax(n));

            polarizedsinglecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize50PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize50PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize50PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize50PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize50PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize50PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize50PctTable.Ymin(n) &...
               singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize50PctTable.Ymax(n) &...
               singlecelltable.Polarized(:)=='yes');

            spreadsinglecellind=find(...
               singlecelltable.Condition(:)==FilteredFibZoneSize50PctTable.Condition(n) &...
               singlecelltable.Day(:)==FilteredFibZoneSize50PctTable.Day(n) &...
               singlecelltable.Well(:)==FilteredFibZoneSize50PctTable.Well(n) &...
               singlecelltable.DateofRun(:)==FilteredFibZoneSize50PctTable.DateofRun(n) &...
               singlecelltable.GlobalPixelXcoord(:)>FilteredFibZoneSize50PctTable.Xmin(n) &...
               singlecelltable.GlobalPixelXcoord(:)<FilteredFibZoneSize50PctTable.Xmax(n) &...
               singlecelltable.GlobalPixelYcoord(:)>FilteredFibZoneSize50PctTable.Ymin(n) &...
```

```
        singlecelltable.GlobalPixelYcoord(:)<FilteredFibZoneSize50PctTable.Ymax(n) &...
        singlecelltable.Spread(:)=='yes');


      singlefibind=find(...
        singlefibtable.Condition(:)==FilteredFibZoneSize50PctTable.Condition(n) &...
        singlefibtable.Day(:)==FilteredFibZoneSize50PctTable.Day(n) &...
        singlefibtable.Well(:)==FilteredFibZoneSize50PctTable.Well(n) &...
        singlefibtable.DateofRun(:)==FilteredFibZoneSize50PctTable.DateofRun(n) &...
        singlefibtable.GlobalPixelXcoord(:)>FilteredFibZoneSize50PctTable.Xmin(n) &...
        singlefibtable.GlobalPixelXcoord(:)<FilteredFibZoneSize50PctTable.Xmax(n) &...
        singlefibtable.GlobalPixelYcoord(:)>FilteredFibZoneSize50PctTable.Ymin(n) &...
        singlefibtable.GlobalPixelYcoord(:)<FilteredFibZoneSize50PctTable.Ymax(n));

FilteredFibZoneSize50PctTable.FiberMVL(n) =...
sqrt((mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2 +...
    (mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)))^2);


FilteredFibZoneSize50PctTable.CellCount(n) = length(singlecellind);

FilteredFibZoneSize50PctTable.PolarizedCellCount(n)=length(polarizedsinglecellind);

FilteredFibZoneSize50PctTable.SpreadCellCount(n)=length(spreadsinglecellind);

if FilteredFibZoneSize50PctTable.PolarizedCellCount(n)>1;
   FilteredFibZoneSize50PctTable.CellMVL(n) =...
      sqrt((mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2 +...
      (mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)))^2);
else
   FilteredFibZoneSize50PctTable.CellMVL(n) = NaN;
end


FilteredFibZoneSize50PctTable.FibMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlefibtable.Orientation(singlefibind)*pi/180)),...
   mean(cos(2*singlefibtable.Orientation(singlefibind)*pi/180)));

FilteredFibZoneSize50PctTable.CellMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(singlecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(singlecellind)*pi/180)));

FilteredFibZoneSize50PctTable.CellPolarizedMeanAngle(n) = 180/pi*1/2*atan2(...
   mean(sin(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)),...
   mean(cos(2*singlecelltable.Orientation(polarizedsinglecellind)*pi/180)));

FilteredFibZoneSize50PctTable.CellPolarizedMeanArea(n) = mean(...
   singlecelltable.Area(polarizedsinglecellind));

FilteredFibZoneSize50PctTable.CellSpreadMeanArea(n) = mean(...
   singlecelltable.Area(spreadsinglecellind));

FilteredFibZoneSize50PctTable.CellMeanArea(n) = mean(...
   singlecelltable.Area(singlecellind));

FilteredFibZoneSize50PctTable.CellMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(singlecellind));

FilteredFibZoneSize50PctTable.CellMeanAspectRatio(n) = mean(...
   singlecelltable.MajorAxis(singlecellind)./...
   singlecelltable.MinorAxis(singlecellind));


FilteredFibZoneSize50PctTable.CellPolarizedMeanEccent(n) = mean(...
   singlecelltable.Eccentricity(polarizedsinglecellind));
```

```
  FilteredFibZoneSize50PctTable.CellPolarizedMeanAspectRatio(n) = mean(...
     singlecelltable.MajorAxis(polarizedsinglecellind)./...
     singlecelltable.MinorAxis(polarizedsinglecellind));

    if n==1;
        est=toc/60;
    end
fract=100*(n/height50);
    if fract <= 5.1 & fract >= 4.9 | fract <= 10.1 &...
       fract >= 9.9 | fract <= 15.1 & fract >= 14.9 |...
       fract <= 20.1 & fract >= 19.9 |...
       fract <= 25.1 & fract >= 24.9 |...
       fract <= 30.1 & fract >= 29.9 |...
       fract <= 35.1 & fract >= 34.9 |...
       fract <= 40.1 & fract >= 39.9 |...
       fract <= 45.1 & fract >= 44.9 |...
       fract <= 55.1 & fract >= 54.9 |...
       fract <= 60.1 & fract >= 59.9 |...
       fract <= 65.1 & fract >= 64.9 |...
       fract <= 70.1 & fract >= 69.9 |...
       fract <= 75.1 & fract >= 74.9 |...
       fract <= 80.1 & fract >= 79.9 |...
       fract <= 85.1 & fract >= 84.9 |...
       fract <= 90.1 & fract >= 89.9 |...
       fract <= 95.1 & fract >= 94.9 |...
       fract <= 100.1 & fract >= 99.9;
fprintf(['Approx. ' num2str(fract) ' Pct Complete'])
    rem=(height50-n)*est;
    fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])

    end
end
toc
% Global Coordinate Creation
coordnums=[100:1:108]-9*[0:11]';
pixpermicron=1024/632;
%x distance between image 699um
imstepx=pixpermicron*699;
cellstorematx={};
cellstorematy={};
%y distance between image 718um
imstepy=pixpermicron*718;

save_str=['GlobalVARAAnalysisVars_', date, '.mat'];
save(save_str,...
    'FilteredFibZoneSize05PctTable',...
    'FilteredFibZoneSize10PctTable',...
    'FilteredFibZoneSize20PctTable',...
    'FilteredFibZoneSize33PctTable',...
    'FilteredFibZoneSize50PctTable',...
    'imstepx','imstepy','cond','lcond','day',...
    'lday','DOR','lDOR','well','lwell','pixpermicron','singlecelltable',...
    'singlefibtable','zsize','-mat')

clear all
save_str=['GlobalVARAAnalysisVars_', date, '.mat'];
load(save_str)

%% Cell Centric Zone Generation

%Only uses 'Polarized' Cells for orientation calculations
    %Calculates MVL if more than one polarized cell in zone
        %(including central cell)
%%Cell surrounding zones
%Zsizes: 03,05,10,20,33,50
```

```
%Generate tables

ZoneColumnNames={'Condition';'DateofRun';'Well';'Day';'GelRow';'Position';...
    'RegionLabel';'Xcoord';'Ycoord';'GlobalXcoord';'GlobalYcoord';...
    'Polarized';'Orientation';'Eccentricity';'MajorAxis';'MinorAxis';...
    'Area';'Spread';'Perimeter';'FiberMVLzsize03';'FiberMAzsize03';...
    'CellEccentzsize03';'CellMajAxzsize03';'CellMinAxzsize03';...
    'CellAreazsize03';'CellPerimeterzsize03';'CellMVLzsize03';...
    'CellMAzsize03';'CellMajMinRatzsize03';'CellCellDotzsize03';...
    'CellFibDotzsize03';...
    'FiberMVLzsize05';'FiberMAzsize05';...
    'CellEccentzsize05';'CellMajAxzsize05';'CellMinAxzsize05';...
    'CellAreazsize05';'CellPerimeterzsize05';'CellMVLzsize05';...
    'CellMAzsize05';'CellMajMinRatzsize05';'CellCellDotzsize05';...
    'CellFibDotzsize05';...
    'FiberMVLzsize10';'FiberMAzsize10';...
    'CellEccentzsize10';'CellMajAxzsize10';'CellMinAxzsize10';...
    'CellAreazsize10';'CellPerimeterzsize10';'CellMVLzsize10';...
    'CellMAzsize10';'CellMajMinRatzsize10';'CellCellDotzsize10';...
    'CellFibDotzsize10';...
    'FiberMVLzsize20';'FiberMAzsize20';...
    'CellEccentzsize20';'CellMajAxzsize20';'CellMinAxzsize20';...
    'CellAreazsize20';'CellPerimeterzsize20';'CellMVLzsize20';...
    'CellMAzsize20';'CellMajMinRatzsize20';'CellCellDotzsize20';...
    'CellFibDotzsize20';...
    'FiberMVLzsize33';'FiberMAzsize33';...
    'CellEccentzsize33';'CellMajAxzsize33';'CellMinAxzsize33';...
    'CellAreazsize33';'CellPerimeterzsize33';'CellMVLzsize33';...
    'CellMAzsize33';'CellMajMinRatzsize33';'CellCellDotzsize33';...
    'CellFibDotzsize33';...
    'FiberMVLzsize50';'FiberMAzsize50';...
    'CellEccentzsize50';'CellMajAxzsize50';'CellMinAxzsize50';...
    'CellAreazsize50';'CellPerimeterzsize50';'CellMVLzsize50';...
    'CellMAzsize50';'CellMajMinRatzsize50';'CellCellDotzsize50';...
    'CellFibDotzsize50'};

ZoneColumntypes={'string';'double';'string';'string';'string';'double';...
    'string';'double';'double';'double';'double';...
    'string';'double';'double';'double';'double';...
    'double';'string';'double';'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';...
    'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';...
    'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';...
    'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';...
    'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';'double';'double';...
    'double';...
    'double';'double';...
    'double';'double';'double';...
```

```
    'double';'double';'double';...
    'double';'double';'double';...
    'double'};

singlecellzonetable=table('Size',[height(singlecelltable) 91],...
    'VariableNames',ZoneColumnNames,'VariableTypes',ZoneColumntypes);

singlecellzonetable(:,1:19)=singlecelltable(:,[2, 9, 3, 14, 5, 4, 6:8, 18:19, 21, 10:13, 16, 20, 17]);
%fix singlecellzonetable to populate values that are static from
%singlecelltablezonetable(:,1:19)=singlecelltbale(:,[

%find and group fiber angles to calculate MA and MVL
nfin=height(singlecellzonetable);
tic
for n=1:nfin;
%fiber
    fibtempind=find(singlefibtable.Condition(:)==...
        singlecellzonetable.Condition(n) & ...
        singlefibtable.DateofRun(:)==singlecellzonetable.DateofRun(n) & ...
        singlefibtable.Well(:)==singlecellzonetable.Well(n) & ...
        singlefibtable.Day(:)==singlecellzonetable.Day(n));
%cell
    celltempind=find(singlecelltable.Condition(n)==...
        singlecellzonetable.Condition(n) & ...
        singlecelltable.DateofRun(:)==singlecellzonetable.DateofRun(n) & ...
        singlecelltable.Well(:)==singlecellzonetable.Well(n) & ...
        singlecelltable.Day(:)==singlecellzonetable.Day(n));

    singlefibtemptab=singlefibtable(fibtempind,:);
    singlecelltemptab=singlecelltable(celltempind,:);
%Remove analysis cell from zonecells pool
    %Find Index of analysis cell in zone pool
    indx=find(singlecelltemptab.GlobalPixelXcoord(:)==singlecelltable.GlobalPixelXcoord(n) &...
    singlecelltemptab.GlobalPixelYcoord(:)==singlecelltable.GlobalPixelYcoord(n));
    %Recreate Temporary Cell Table with Cells of Day and Well excluding
    %analysis cells
    singlecelltemptab=vertcat(singlecelltemptab(1:indx-1,:),singlecelltemptab(indx+1:end,:));
%CellFiber Distance Vector-Find distance between cell and all fibers for
%well
cellfibdistvect=sqrt(...
    (singlefibtemptab.GlobalPixelXcoord(:)-singlecellzonetable.GlobalXcoord(n)).^2 +...
    (singlefibtemptab.GlobalPixelYcoord(:)-singlecellzonetable.GlobalYcoord(n)).^2);

%CellCell Distance Vector-Find distance between cell and all cells for
%well
cellcelldistvect=sqrt(...
    (singlecelltemptab.GlobalPixelXcoord(:)-singlecellzonetable.GlobalXcoord(n)).^2 +...
    (singlecelltemptab.GlobalPixelYcoord(:)-singlecellzonetable.GlobalYcoord(n)).^2);

%
%Find fiber and cells within radial zone limits

%ZoneSize03 126 um
%Fiber relation
cellfibindzsize03=find((cellfibdistvect(:)/pixpermicron)<=126);

fibangzsize03=singlefibtemptab.Orientation(cellfibindzsize03);
                fibzsize03c2m = mean(cos(2*fibangzsize03*pi/180));
                fibzsize03s2m = mean(sin(2*fibangzsize03*pi/180));
                fibzsize03MA = 180/pi*1/2*atan2(fibzsize03s2m, fibzsize03c2m);      % mean angle (deg)
                fibzsize03MVL = sqrt(fibzsize03s2m^2 + fibzsize03c2m^2);            % mean vector length
                fibzsize03CSD = 180/pi*1/2*sqrt(-2*log(fibzsize03MVL));    % circular standard deviation (deg)
                fibzsize03AD = 180/pi*1/2*sqrt(2*(1-fibzsize03MVL));       % angular deviation (deg)
singlecellzonetable.FiberMVLzsize03(n)=fibzsize03MVL;
singlecellzonetable.FiberMAzsize03(n)=fibzsize03MA;
clear fibzsize03c2m fibzsize03s2m fibzsize03MA fibzsize03MVL...
```

```
    fibzsize03CSD fibzsize03AD cellfibindzsize03;

%Cell relation
cellcellindzsize03=find((cellcelldistvect(:)/pixpermicron)<=126);
cellcellindzsize03polar=find((cellcelldistvect(:)/pixpermicron)<=126 &...
    singlecelltemptab.Polarized(:)=='yes');
cellangzsize03=singlecelltemptab.Orientation(cellcellindzsize03polar);
if singlecellzonetable.Polarized(n)=='yes';
ac=[cos(singlecellzonetable.Orientation(n)*pi/180),sin(singlecellzonetable.Orientation(n)*pi/180)];
bc=[cos(cellangzsize03*pi/180),sin(cellangzsize03*pi/180)];
fc=[cos(fibangzsize03*pi/180),sin(fibangzsize03*pi/180)];
avgcellcelldotzsize03=abs(mean(bc*ac'));
avgcellfibdotzsize03=abs(mean(fc*ac'));
else
avgcellcelldotzsize03=NaN;
avgcellfibdotzsize03=NaN;
end

if singlecellzonetable.Polarized(n)=='yes';
    cellangzsize03=vertcat(cellangzsize03,singlecellzonetable.Orientation(n));
end
                cellzsize03c2m = mean(cos(2*cellangzsize03*pi/180));
                cellzsize03s2m = mean(sin(2*cellangzsize03*pi/180));
                cellzsize03MA = 180/pi*1/2*atan2(cellzsize03s2m, cellzsize03c2m);        % mean angle (deg)
                if length(cellangzsize03)>1;
                  cellzsize03MVL = sqrt(cellzsize03s2m^2 + cellzsize03c2m^2);             % mean vector length
                else
                   cellzsize03MVL = NaN;
                end
                cellzsize03CSD = 180/pi*1/2*sqrt(-2*log(cellzsize03MVL));     % circular standard deviation (deg)
                cellzsize03AD = 180/pi*1/2*sqrt(2*(1-cellzsize03MVL));        % angular deviation (deg)
avgcellangzsize03=cellzsize03MA;
avgcelleccenzsize03=mean(singlecelltemptab.Eccentricity(cellcellindzsize03));
avgcellmajaxzsize03=mean(singlecelltemptab.MajorAxis(cellcellindzsize03));
avgcellminaxzsize03=mean(singlecelltemptab.MinorAxis(cellcellindzsize03));
avgcellareazsize03=mean(singlecelltemptab.Area(cellcellindzsize03));
avgcellperimzsize03=mean(singlecelltemptab.Perimeter(cellcellindzsize03));
avgcellmajminratzsize03=mean(singlecelltemptab.MajorAxis(cellcellindzsize03)./...
    singlecelltemptab.MinorAxis(cellcellindzsize03));



singlecellzonetable.CellMVLzsize03(n)=cellzsize03MVL;
singlecellzonetable.CellMAzsize03(n)=cellzsize03MA;
singlecellzonetable.CellEccentzsize03(n)=avgcelleccenzsize03;
singlecellzonetable.CellMajAxzsize03(n)=avgcellmajaxzsize03;
singlecellzonetable.CellMinAxzsize03(n)=avgcellminaxzsize03;
singlecellzonetable.CellAreazsize03(n)=avgcellareazsize03;
singlecellzonetable.CellPerimeterzsize03(n)=avgcellperimzsize03;
singlecellzonetable.CellMajMinRatzsize50(n)=avgcellmajminratzsize03;
singlecellzonetable.CellCellDotzsize03(n)=avgcellcelldotzsize03;
singlecellzonetable.CellFibDotzsize03(n)=avgcellfibdotzsize03;
clear cellangzsize03 fibangzsize03 cellzsize03c2m cellzsize03s2m cellzsize03MA cellzsize03MVL...
    cellzsize03CSD cellzsize03AD cellcellindzsize03 avgcelleccenzsize03...
    avgcellmajaxzsize03 avgcellminaxzsize03 avgcellareazsize03...
    avgcellperimzsize03 avgcellmajminratzsize03 avgcellcelldotzsize03...
    avgcellfibdotzsize03 ac bc fc;
%ZoneSize05 210 um
%Fiber relation
cellfibindzsize05=find((cellfibdistvect(:)/pixpermicron)<=210);

fibangzsize05=singlefibtemptab.Orientation(cellfibindzsize05);
                fibzsize05c2m = mean(cos(2*fibangzsize05*pi/180));
                fibzsize05s2m = mean(sin(2*fibangzsize05*pi/180));
                fibzsize05MA = 180/pi*1/2*atan2(fibzsize05s2m, fibzsize05c2m);        % mean angle (deg)
                fibzsize05MVL = sqrt(fibzsize05s2m^2 + fibzsize05c2m^2);              % mean vector length
```

```
                fibzsize05CSD = 180/pi*1/2*sqrt(-2*log(fibzsize05MVL));     % circular standard deviation (deg)
                fibzsize05AD = 180/pi*1/2*sqrt(2*(1-fibzsize05MVL));        % angular deviation (deg)
singlecellzonetable.FiberMVLzsize05(n)=fibzsize05MVL;
singlecellzonetable.FiberMAzsize05(n)=fibzsize05MA;
clear fibzsize05c2m fibzsize05s2m fibzsize05MA fibzsize05MVL...
   fibzsize05CSD fibzsize05AD cellfibindzsize05;
%Cell relation
cellcellindzsize05=find((cellcelldistvect(:)/pixpermicron)<=210);
cellcellindzsize05polar=find((cellcelldistvect(:)/pixpermicron)<=210 &...
   singlecelltemptab.Polarized(:)=='yes');
cellangzsize05=singlecelltemptab.Orientation(cellcellindzsize05polar);

if singlecellzonetable.Polarized(n)=='yes';
   ac=[cos(singlecellzonetable.Orientation(n)*pi/180),sin(singlecellzonetable.Orientation(n)*pi/180)];
   bc=[cos(cellangzsize05*pi/180),sin(cellangzsize05*pi/180)];
   fc=[cos(fibangzsize05*pi/180),sin(fibangzsize05*pi/180)];

   avgcellcelldotzsize05=abs(mean(bc*ac'));
   avgcellfibdotzsize05=abs(mean(fc*ac'));
else
   avgcellcelldotzsize05=NaN;
   avgcellfibdotzsize05=NaN;
end

if singlecellzonetable.Polarized(n)=='yes';
   cellangzsize05=vertcat(cellangzsize05,singlecellzonetable.Orientation(n));
end
                cellzsize05c2m = mean(cos(2*cellangzsize05*pi/180));
                cellzsize05s2m = mean(sin(2*cellangzsize05*pi/180));
                cellzsize05MA = 180/pi*1/2*atan2(cellzsize05s2m, cellzsize05c2m);       % mean angle (deg)
                if length(cellangzsize05)>1;
                  cellzsize05MVL = sqrt(cellzsize05s2m^2 + cellzsize05c2m^2);
                else
                   cellzsize05MVL = NaN;  % mean vector length
                end
                cellzsize05CSD = 180/pi*1/2*sqrt(-2*log(cellzsize05MVL));     % circular standard deviation (deg)
                cellzsize05AD = 180/pi*1/2*sqrt(2*(1-cellzsize05MVL));        % angular deviation (deg)
avgcellangzsize05=cellzsize05MA;
avgcelleccenzsize05=mean(singlecelltemptab.Eccentricity(cellcellindzsize05));
avgcellmajaxzsize05=mean(singlecelltemptab.MajorAxis(cellcellindzsize05));
avgcellminaxzsize05=mean(singlecelltemptab.MinorAxis(cellcellindzsize05));
avgcellareazsize05=mean(singlecelltemptab.Area(cellcellindzsize05));
avgcellperimzsize05=mean(singlecelltemptab.Perimeter(cellcellindzsize05));
avgcellmajminratzsize05=mean(singlecelltemptab.MajorAxis(cellcellindzsize05)./...
   singlecelltemptab.MinorAxis(cellcellindzsize05));
   singlecellzonetable.CellMVLzsize05(n)=cellzsize05MVL;
singlecellzonetable.CellMAzsize05(n)=cellzsize05MA;
singlecellzonetable.CellEccentzsize05(n)=avgcelleccenzsize05;
singlecellzonetable.CellMajAxzsize05(n)=avgcellmajaxzsize05;
singlecellzonetable.CellMinAxzsize05(n)=avgcellminaxzsize05;
singlecellzonetable.CellAreazsize05(n)=avgcellareazsize05;
singlecellzonetable.CellPerimeterzsize05(n)=avgcellperimzsize05;
singlecellzonetable.CellMajMinRatzsize05(n)=avgcellmajminratzsize05;
singlecellzonetable.CellCellDotzsize05(n)=avgcellcelldotzsize05;
singlecellzonetable.CellFibDotzsize05(n)=avgcellfibdotzsize05;
clear cellangzsize05  fibangzsize05 cellzsize05c2m cellzsize05s2m cellzsize05MA cellzsize05MVL...
   cellzsize05CSD cellzsize05AD cellcellindzsize05 avgcelleccenzsize05...
   avgcellmajaxzsize05 avgcellminaxzsize05 avgcellareazsize05...
   avgcellperimzsize05 avgcellmajminratzsize05 avgcellcelldotzsize05...
   avgcellfibdotzsize05 ac bc fc;
%ZoneSize10 421 um
%Fiber Relation
cellfibindzsize10=find((cellfibdistvect(:)/pixpermicron)<=421);

fibangzsize10=singlefibtemptab.Orientation(cellfibindzsize10);
                fibzsize10c2m = mean(cos(2*fibangzsize10*pi/180));
```

```
                fibzsize10s2m = mean(sin(2*fibangzsize10*pi/180));
                fibzsize10MA = 180/pi*1/2*atan2(fibzsize10s2m, fibzsize10c2m);       % mean angle (deg)
                fibzsize10MVL = sqrt(fibzsize10s2m^2 + fibzsize10c2m^2);             % mean vector length
                fibzsize10CSD = 180/pi*1/2*sqrt(-2*log(fibzsize10MVL));     % circular standard deviation (deg)
                fibzsize10AD = 180/pi*1/2*sqrt(2*(1-fibzsize10MVL));        % angular deviation (deg)
singlecellzonetable.FiberMVLzsize10(n)=fibzsize10MVL;
singlecellzonetable.FiberMAzsize10(n)=fibzsize10MA;
clear fibzsize10c2m fibzsize10s2m fibzsize10MA fibzsize10MVL...
   fibzsize10CSD fibzsize10AD cellfibindzsize10;
%Cell Relation
cellcellindzsize10=find((cellcelldistvect(:)/pixpermicron)<=421);
cellcellindzsize10polar=find((cellcelldistvect(:)/pixpermicron)<=421 &...
   singlecelltemptab.Polarized(:)=='yes');

cellangzsize10=singlecelltemptab.Orientation(cellcellindzsize10polar);
if singlecellzonetable.Polarized(n)=='yes';
ac=[cos(singlecellzonetable.Orientation(n)*pi/180),sin(singlecellzonetable.Orientation(n)*pi/180)];
bc=[cos(cellangzsize10*pi/180),sin(cellangzsize10*pi/180)];
fc=[cos(fibangzsize10*pi/180),sin(fibangzsize10*pi/180)];

avgcellcelldotzsize10=abs(mean(bc*ac'));
avgcellfibdotzsize10=abs(mean(fc*ac'));
else
avgcellcelldotzsize10=NaN;
avgcellfibdotzsize10=NaN;
end

if singlecellzonetable.Polarized(n)=='yes';
   cellangzsize10=vertcat(cellangzsize10,singlecellzonetable.Orientation(n));
end

               cellzsize10c2m = mean(cos(2*cellangzsize10*pi/180));
                cellzsize10s2m = mean(sin(2*cellangzsize10*pi/180));
                cellzsize10MA = 180/pi*1/2*atan2(cellzsize10s2m, cellzsize10c2m);       % mean angle (deg)
                if length(cellangzsize10)>1;
                   cellzsize10MVL = sqrt(cellzsize10s2m^2 + cellzsize10c2m^2);             % mean vector length
                else
                   cellzsize10MVL = NaN;
                end
                cellzsize10CSD = 180/pi*1/2*sqrt(-2*log(cellzsize10MVL));     % circular standard deviation (deg)
                cellzsize10AD = 180/pi*1/2*sqrt(2*(1-cellzsize10MVL));        % angular deviation (deg)
avgcellangzsize10=cellzsize10MA;
avgcelleccenzsize10=mean(singlecelltemptab.Eccentricity(cellcellindzsize10));
avgcellmajaxzsize10=mean(singlecelltemptab.MajorAxis(cellcellindzsize10));
avgcellminaxzsize10=mean(singlecelltemptab.MinorAxis(cellcellindzsize10));
avgcellareazsize10=mean(singlecelltemptab.Area(cellcellindzsize10));
avgcellperimzsize10=mean(singlecelltemptab.Perimeter(cellcellindzsize10));
avgcellmajminratzsize10=mean(singlecelltemptab.MajorAxis(cellcellindzsize10)./...
   singlecelltemptab.MinorAxis(cellcellindzsize10));

singlecellzonetable.CellMVLzsize10(n)=cellzsize10MVL;
singlecellzonetable.CellMAzsize10(n)=cellzsize10MA;
singlecellzonetable.CellEccentzsize10(n)=avgcelleccenzsize10;
singlecellzonetable.CellMajAxzsize10(n)=avgcellmajaxzsize10;
singlecellzonetable.CellMinAxzsize10(n)=avgcellminaxzsize10;
singlecellzonetable.CellAreazsize10(n)=avgcellareazsize10;
singlecellzonetable.CellPerimeterzsize10(n)=avgcellperimzsize10;
singlecellzonetable.CellMajMinRatzsize10(n)=avgcellmajminratzsize10;
singlecellzonetable.CellCellDotzsize10(n)=avgcellcelldotzsize10;
singlecellzonetable.CellFibDotzsize10(n)=avgcellfibdotzsize10;
clear fibangzsize10 cellangzsize10 cellzsize10c2m cellzsize10s2m cellzsize10MA cellzsize10MVL...
   cellzsize10CSD cellzsize10AD cellcellindzsize10 avgcelleccenzsize10...
   avgcellmajaxzsize10 avgcellminaxzsize10 avgcellareazsize10...
   avgcellperimzsize10 avgcellmajminratzsize10 avgcellcelldotzsize10...
   avgcellfibdotzsize10 ac bc fc;
%ZoneSize20 842 um
```

```
%Fiber Relation
cellfibindzsize20=find((cellfibdistvect(:)/pixpermicron)<=842);

fibangzsize20=singlefibtemptab.Orientation(cellfibindzsize20);
                fibzsize20c2m = mean(cos(2*fibangzsize20*pi/180));
                fibzsize20s2m = mean(sin(2*fibangzsize20*pi/180));
                fibzsize20MA = 180/pi*1/2*atan2(fibzsize20s2m, fibzsize20c2m);       % mean angle (deg)
                fibzsize20MVL = sqrt(fibzsize20s2m^2 + fibzsize20c2m^2);             % mean vector length
                fibzsize20CSD = 180/pi*1/2*sqrt(-2*log(fibzsize20MVL));     % circular standard deviation (deg)
                fibzsize20AD = 180/pi*1/2*sqrt(2*(1-fibzsize20MVL));        % angular deviation (deg)
singlecellzonetable.FiberMVLzsize20(n)=fibzsize20MVL;
singlecellzonetable.FiberMAzsize20(n)=fibzsize20MA;
clear fibzsize20c2m fibzsize20s2m fibzsize20MA fibzsize20MVL...
   fibzsize20CSD fibzsize20AD cellfibindzsize20;

%Cell Relation
cellcellindzsize20=find((cellcelldistvect(:)/pixpermicron)<=842);
cellcellindzsize20polar=find((cellcelldistvect(:)/pixpermicron)<=842 &...
   singlecelltemptab.Polarized(:)=='yes');

cellangzsize20=singlecelltemptab.Orientation(cellcellindzsize20polar);
if singlecellzonetable.Polarized(n)=='yes';
ac=[cos(singlecellzonetable.Orientation(n)*pi/180),sin(singlecellzonetable.Orientation(n)*pi/180)];
bc=[cos(cellangzsize20*pi/180),sin(cellangzsize20*pi/180)];
fc=[cos(fibangzsize20*pi/180),sin(fibangzsize20*pi/180)];

avgcellcelldotzsize20=abs(mean(bc*ac'));
avgcellfibdotzsize20=abs(mean(fc*ac'));
else
avgcellcelldotzsize20=NaN;
avgcellfibdotzsize20=NaN;
end
if singlecellzonetable.Polarized(n)=='yes';
   cellangzsize20=vertcat(cellangzsize20,singlecellzonetable.Orientation(n));
end


                cellzsize20c2m = mean(cos(2*cellangzsize20*pi/180));
                cellzsize20s2m = mean(sin(2*cellangzsize20*pi/180));
                cellzsize20MA = 180/pi*1/2*atan2(cellzsize20s2m, cellzsize20c2m);       % mean angle (deg)
                if length(cellangzsize20)>1;
                   cellzsize20MVL = sqrt(cellzsize20s2m^2 + cellzsize20c2m^2);             % mean vector length
                else
                   cellzsize20MVL = NaN;
                end
                cellzsize20CSD = 180/pi*1/2*sqrt(-2*log(cellzsize20MVL));     % circular standard deviation (deg)
                cellzsize20AD = 180/pi*1/2*sqrt(2*(1-cellzsize20MVL));        % angular deviation (deg)
avgcellangzsize20=cellzsize20MA;
avgcelleccenzsize20=mean(singlecelltemptab.Eccentricity(cellcellindzsize20));
avgcellmajaxzsize20=mean(singlecelltemptab.MajorAxis(cellcellindzsize20));
avgcellminaxzsize20=mean(singlecelltemptab.MinorAxis(cellcellindzsize20));
avgcellareazsize20=mean(singlecelltemptab.Area(cellcellindzsize20));
avgcellperimzsize20=mean(singlecelltemptab.Perimeter(cellcellindzsize20));
avgcellmajminratzsize20=mean(singlecelltemptab.MajorAxis(cellcellindzsize20)./...
   singlecelltemptab.MinorAxis(cellcellindzsize20));

singlecellzonetable.CellMVLzsize20(n)=cellzsize20MVL;
singlecellzonetable.CellMAzsize20(n)=cellzsize20MA;
singlecellzonetable.CellEccentzsize20(n)=avgcelleccenzsize20;
singlecellzonetable.CellMajAxzsize20(n)=avgcellmajaxzsize20;
singlecellzonetable.CellMinAxzsize20(n)=avgcellminaxzsize20;
singlecellzonetable.CellAreazsize20(n)=avgcellareazsize20;
singlecellzonetable.CellPerimeterzsize20(n)=avgcellperimzsize20;
singlecellzonetable.CellMajMinRatzsize20(n)=avgcellmajminratzsize20;
singlecellzonetable.CellCellDotzsize20(n)=avgcellcelldotzsize20;
singlecellzonetable.CellFibDotzsize20(n)=avgcellfibdotzsize20;
```

```
clear cellangzsize20 fibangzsize20 cellzsize20c2m cellzsize20s2m cellzsize20MA cellzsize20MVL...
   cellzsize20CSD cellzsize20AD cellcellindzsize20 avgcelleccenzsize20...
   avgcellmajaxzsize20 avgcellminaxzsize20 avgcellareazsize20...
   avgcellperimzsize20 avgcellmajminratzsize20 avgcellcelldotzsize20...
   avgcellfibdotzsize20 ac bc fc;
%ZoneSize33 1390 um
%Fiber Relation
cellfibindzsize33=find((cellfibdistvect(:)/pixpermicron)<=1390);

fibangzsize33=singlefibtemptab.Orientation(cellfibindzsize33);
                fibzsize33c2m = mean(cos(2*fibangzsize33*pi/180));
                fibzsize33s2m = mean(sin(2*fibangzsize33*pi/180));
                fibzsize33MA = 180/pi*1/2*atan2(fibzsize33s2m, fibzsize33c2m);       % mean angle (deg)
                fibzsize33MVL = sqrt(fibzsize33s2m^2 + fibzsize33c2m^2);             % mean vector length
                fibzsize33CSD = 180/pi*1/2*sqrt(-2*log(fibzsize33MVL));    % circular standard deviation (deg)
                fibzsize33AD = 180/pi*1/2*sqrt(2*(1-fibzsize33MVL));       % angular deviation (deg)
singlecellzonetable.FiberMVLzsize33(n)=fibzsize33MVL;
singlecellzonetable.FiberMAzsize33(n)=fibzsize33MA;
clear fibzsize33c2m fibzsize33s2m fibzsize33MA fibzsize33MVL...
   fibzsize33CSD fibzsize33AD cellfibindzsize33;
%Cell relation
cellcellindzsize33=find((cellcelldistvect(:)/pixpermicron)<=1390);
cellcellindzsize33polar=find((cellcelldistvect(:)/pixpermicron)<=1390 &...
   singlecelltemptab.Polarized(:)=='yes');
cellangzsize33=singlecelltemptab.Orientation(cellcellindzsize33polar);
if singlecellzonetable.Polarized(n)=='yes';
ac=[cos(singlecellzonetable.Orientation(n)*pi/180),sin(singlecellzonetable.Orientation(n)*pi/180)];
bc=[cos(cellangzsize33*pi/180),sin(cellangzsize33*pi/180)];
fc=[cos(fibangzsize33*pi/180),sin(fibangzsize33*pi/180)];

avgcellcelldotzsize33=abs(mean(bc*ac'));
avgcellfibdotzsize33=abs(mean(fc*ac'));
else
avgcellcelldotzsize33=NaN;
avgcellfibdotzsize33=NaN;
end

if singlecellzonetable.Polarized(n)=='yes';
   cellangzsize33=vertcat(cellangzsize33,singlecellzonetable.Orientation(n));
end

                cellzsize33c2m = mean(cos(2*cellangzsize33*pi/180));
                cellzsize33s2m = mean(sin(2*cellangzsize33*pi/180));
                cellzsize33MA = 180/pi*1/2*atan2(cellzsize33s2m, cellzsize33c2m);       % mean angle (deg)
                if length(cellangzsize33)>1;
                   cellzsize33MVL = sqrt(cellzsize33s2m^2 + cellzsize33c2m^2);          % mean vector length
                else
                   cellzsize33MVL = NaN;
                end
                cellzsize33CSD = 180/pi*1/2*sqrt(-2*log(cellzsize33MVL));    % circular standard deviation (deg)
                cellzsize33AD = 180/pi*1/2*sqrt(2*(1-cellzsize33MVL));       % angular deviation (deg)
avgcellangzsize33=cellzsize33MA;
avgcelleccenzsize33=mean(singlecelltemptab.Eccentricity(cellcellindzsize33));
avgcellmajaxzsize33=mean(singlecelltemptab.MajorAxis(cellcellindzsize33));
avgcellminaxzsize33=mean(singlecelltemptab.MinorAxis(cellcellindzsize33));
avgcellareazsize33=mean(singlecelltemptab.Area(cellcellindzsize33));
avgcellperimzsize33=mean(singlecelltemptab.Perimeter(cellcellindzsize33));
avgcellmajminratzsize33=mean(singlecelltemptab.MajorAxis(cellcellindzsize33)./...
   singlecelltemptab.MinorAxis(cellcellindzsize33));
singlecellzonetable.CellMVLzsize33(n)=cellzsize33MVL;
singlecellzonetable.CellMAzsize33(n)=cellzsize33MA;
singlecellzonetable.CellEccentzsize33(n)=avgcelleccenzsize33;
singlecellzonetable.CellMajAxzsize33(n)=avgcellmajaxzsize33;
singlecellzonetable.CellMinAxzsize33(n)=avgcellminaxzsize33;
singlecellzonetable.CellAreazsize33(n)=avgcellareazsize33;
singlecellzonetable.CellPerimeterzsize33(n)=avgcellperimzsize33;
```

```
singlecellzonetable.CellMajMinRatzsize33(n)=avgcellmajminratzsize33;
singlecellzonetable.CellCellDotzsize33(n)=avgcellcelldotzsize33;
singlecellzonetable.CellFibDotzsize33(n)=avgcellfibdotzsize33;
clear cellangzsize33 fibangzsize33 cellzsize33c2m cellzsize33s2m cellzsize33MA cellzsize33MVL...
   cellzsize33CSD cellzsize33AD cellcellindzsize33 avgcelleccenzsize33...
   avgcellmajaxzsize33 avgcellminaxzsiz33 avgcellareazsize33...
   avgcellperimzsize33 avgcellmajminratzsize33 avgcellcelldotzsize33...
   avgcellfibdotzsize33 ac bc fc;
%ZoneSize50 2100 um
%Fiber Relation
cellfibindzsize50=find((cellfibdistvect(:)/pixpermicron)<=2100);

fibangzsize50=singlefibtemptab.Orientation(cellfibindzsize50);
                fibzsize50c2m = mean(cos(2*fibangzsize50*pi/180));
                fibzsize50s2m = mean(sin(2*fibangzsize50*pi/180));
                fibzsize50MA = 180/pi*1/2*atan2(fibzsize50s2m, fibzsize50c2m);       % mean angle (deg)
                fibzsize50MVL = sqrt(fibzsize50s2m^2 + fibzsize50c2m^2);             % mean vector length
                fibzsize50CSD = 180/pi*1/2*sqrt(-2*log(fibzsize50MVL));    % circular standard deviation (deg)
                fibzsize50AD = 180/pi*1/2*sqrt(2*(1-fibzsize50MVL));       % angular deviation (deg)
singlecellzonetable.FiberMVLzsize50(n)=fibzsize50MVL;
singlecellzonetable.FiberMAzsize50(n)=fibzsize50MA;
clear fibzsize50c2m fibzsize50s2m fibzsize50MA fibzsize50MVL...
   fibzsize50CSD fibzsize50AD cellfibindzsize50;
%Cell Relation
cellcellindzsize50=find((cellcelldistvect(:)/pixpermicron)<=2100);
cellcellindzsize50polar=find((cellcelldistvect(:)/pixpermicron)<=2100 &...
   singlecelltemptab.Polarized(:)=='yes');
cellangzsize50=singlecelltemptab.Orientation(cellcellindzsize50polar);
if singlecellzonetable.Polarized(n)=='yes';
   ac=[cos(singlecellzonetable.Orientation(n)*pi/180),sin(singlecellzonetable.Orientation(n)*pi/180)];
   bc=[cos(cellangzsize50*pi/180),sin(cellangzsize50*pi/180)];
   fc=[cos(fibangzsize50*pi/180),sin(fibangzsize50*pi/180)];

   avgcellcelldotzsize50=abs(mean(bc*ac'));
   avgcellfibdotzsize50=abs(mean(fc*ac'));
else
   avgcellcelldotzsize50=NaN;
   avgcellfibdotzsize50=NaN;
end

if singlecellzonetable.Polarized(n)=='yes';
   cellangzsize50=vertcat(cellangzsize50,singlecellzonetable.Orientation(n));
end


                cellzsize50c2m = mean(cos(2*cellangzsize50*pi/180));
                cellzsize50s2m = mean(sin(2*cellangzsize50*pi/180));
                cellzsize50MA = 180/pi*1/2*atan2(cellzsize50s2m, cellzsize50c2m);       % mean angle (deg)
                if length(cellangzsize50)>1;
                   cellzsize50MVL = sqrt(cellzsize50s2m^2 + cellzsize50c2m^2);          % mean vector length
                else
                   cellzsize50MVL = NaN;
                end
                cellzsize50CSD = 180/pi*1/2*sqrt(-2*log(cellzsize50MVL));    % circular standard deviation (deg)
                cellzsize50AD = 180/pi*1/2*sqrt(2*(1-cellzsize50MVL));       % angular deviation (deg)
avgcellangzsize50=cellzsize50MA;
avgcelleccenzsize50=mean(singlecelltemptab.Eccentricity(cellcellindzsize50));
avgcellmajaxzsize50=mean(singlecelltemptab.MajorAxis(cellcellindzsize50));
avgcellminaxzsize50=mean(singlecelltemptab.MinorAxis(cellcellindzsize50));
avgcellareazsize50=mean(singlecelltemptab.Area(cellcellindzsize50));
avgcellperimzsize50=mean(singlecelltemptab.Perimeter(cellcellindzsize50));
avgcellmajminratzsize50=mean(singlecelltemptab.MajorAxis(cellcellindzsize50)./...
   singlecelltemptab.MinorAxis(cellcellindzsize50));

singlecellzonetable.CellMVLzsize50(n)=cellzsize50MVL;
singlecellzonetable.CellMAzsize50(n)=cellzsize50MA;
```

```
singlecellzonetable.CellEccentzsize50(n)=avgcelleccenzsize50;
singlecellzonetable.CellMajAxzsize50(n)=avgcellmajaxzsize50;
singlecellzonetable.CellMinAxzsize50(n)=avgcellminaxzsize50;
singlecellzonetable.CellAreazsize50(n)=avgcellareazsize50;
singlecellzonetable.CellPerimeterzsize50(n)=avgcellperimzsize50;
singlecellzonetable.CellMajMinRatzsize50(n)=avgcellmajminratzsize50;
singlecellzonetable.CellCellDotzsize50(n)=avgcellcelldotzsize50;
singlecellzonetable.CellFibDotzsize50(n)=avgcellfibdotzsize50;
clear cellangzsize50 fibangzsize50 cellzsize50c2m cellzsize50s2m cellzsize50MA cellzsize50MVL...
   cellzsize50CSD cellzsize50AD cellcellindzsize50 avgcelleccenzsize50...
   avgcellmajaxzsize50 avgcellminaxzsize50 avgcellareazsize50...
   avgcellperimzsize50 avgcellmajminratzsize50 avgcellcelldotzsize50...
   avgcellfibdotzsize50 ac bc fc;
%
%
%




   if n==1;
      est=toc/60;
   end

   Cellpercent=(n/nfin)*100;
   if Cellpercent <= 5.1 & Cellpercent >= 4.9 | Cellpercent <= 10.1 &...
      Cellpercent >= 9.9 | Cellpercent <= 15.1 & Cellpercent >= 14.9 |...
      Cellpercent <= 20.1 & Cellpercent >= 19.9 |...
      Cellpercent <= 25.1 & Cellpercent >= 24.9 |...
      Cellpercent <= 30.1 & Cellpercent >= 29.9 |...
      Cellpercent <= 35.1 & Cellpercent >= 34.9 |...
      Cellpercent <= 40.1 & Cellpercent >= 39.9 |...
      Cellpercent <= 45.1 & Cellpercent >= 44.9 |...
      Cellpercent <= 55.1 & Cellpercent >= 54.9 |...
      Cellpercent <= 60.1 & Cellpercent >= 59.9 |...
      Cellpercent <= 65.1 & Cellpercent >= 64.9 |...
      Cellpercent <= 70.1 & Cellpercent >= 69.9 |...
      Cellpercent <= 75.1 & Cellpercent >= 74.9 |...
      Cellpercent <= 80.1 & Cellpercent >= 79.9 |...
      Cellpercent <= 85.1 & Cellpercent >= 84.9 |...
      Cellpercent <= 90.1 & Cellpercent >= 89.9 |...
      Cellpercent <= 95.1 & Cellpercent >= 94.9 |...
      Cellpercent <= 100.1 & Cellpercent >= 99.9;
   fprintf(['Update: Cell Section approx. ' num2str(Cellpercent) ' Pct Complete.\n'])
   fprintf('working\n')
   rem=(nfin-n)*est;
   fprintf(['Approx. ',num2str(rem),' minutes remaining\n'])
   end

end
SingleCellZoneVarGenerationDate=datetime;

%Find 'missing' condition rows and remove
% Missing rows are likely rows not meeting user input criteria for conditions,
% days, etc to be analyzed and are excluded in analysis
%miss_zsize03_ind=find(ismissing(FilteredFibZoneSize03PctTable.Condition(:)));
miss_zsize05_ind=find(ismissing(FilteredFibZoneSize05PctTable.Condition(:)));
miss_zsize10_ind=find(ismissing(FilteredFibZoneSize10PctTable.Condition(:)));
miss_zsize20_ind=find(ismissing(FilteredFibZoneSize20PctTable.Condition(:)));
miss_zsize33_ind=find(ismissing(FilteredFibZoneSize33PctTable.Condition(:)));
miss_zsize50_ind=find(ismissing(FilteredFibZoneSize50PctTable.Condition(:)));

%FilteredFibZoneSize03PctTable(miss_zsize03_ind,:)=[];
FilteredFibZoneSize05PctTable(miss_zsize05_ind,:)=[];
```

```
FilteredFibZoneSize10PctTable(miss_zsize10_ind,:)=[];
FilteredFibZoneSize20PctTable(miss_zsize20_ind,:)=[];
FilteredFibZoneSize33PctTable(miss_zsize33_ind,:)=[];
FilteredFibZoneSize50PctTable(miss_zsize50_ind,:)=[];


%09/21/20 Run Date
singlecellzonetable_FiberSubunit_64 = singlecellzonetable;
FilteredFibZoneSize05PctTable_FiberSubunit_64 = FilteredFibZoneSize05PctTable;
FilteredFibZoneSize10PctTable_FiberSubunit_64 = FilteredFibZoneSize10PctTable;
FilteredFibZoneSize20PctTable_FiberSubunit_64 = FilteredFibZoneSize20PctTable;
FilteredFibZoneSize33PctTable_FiberSubunit_64 = FilteredFibZoneSize33PctTable;
FilteredFibZoneSize50PctTable_FiberSubunit_64 = FilteredFibZoneSize50PctTable;
singlecelltable_FiberSubunit_64 = singlecelltable;
singlefibtable_FiberSubunit_64 = singlefibtable;


save_str=['ZoneAnalysis_',date,'.mat'];
save_str=['ZoneAnalysis_FiberSubunit_64_day137_',date,'.mat'];
save(save_str,...
    'singlecellzonetable_FiberSubunit_64',...
    'FilteredFibZoneSize05PctTable_FiberSubunit_64',...
    'FilteredFibZoneSize10PctTable_FiberSubunit_64',...
    'FilteredFibZoneSize20PctTable_FiberSubunit_64',...
    'FilteredFibZoneSize33PctTable_FiberSubunit_64',...
    'FilteredFibZoneSize50PctTable_FiberSubunit_64',...
    'imstepx','imstepy','cond','lcond','day',...
    'lday','DOR','lDOR','well','lwell','pixpermicron','singlecelltable_FiberSubunit_64',...
    'singlefibtable_FiberSubunit_64','zsize','-mat')

clear all
save_str=['ZoneAnalysis_FiberSubunit_64_day137_',date,'.mat'];
load(save_str)
toc/60
```

A-1.2 Supplemental Alignment Calculation

```
function [MVL,MA] = MVL_calc(ang)
%Mean Vector Length Calculator of Angle Pool.
%ang input should contain:
   %vector of all angles to be used in calculating single MVL.
   %Angles contained in ang should be in units of degrees.

%MVL output ranges from 0 to 1. A value of 1 signifies perfect alignment
   %A value of 0 signifies complete randomness.
c2m = mean(cos(2*ang*pi/180));
s2m = mean(sin(2*ang*pi/180));

MVL = sqrt((s2m^2)+(c2m^2));

MA = 180/pi*(1/2)*atan2(s2m,c2m);
End
```

Appendix B

Aim 2 MATLAB Scripting

B-1.1 Initial Processing of NCORR MATLAB Software Data

```
%M. Jake Potter
%Initial processing file utilized to compile strain data generated from
%NCORR surface strain mapping MATLAB software and manually collected
%variables stored in external tables

%%% Load data from file for analysis
clear all
close all

%% Plot the strain med
%% Creating Table with Area and Stress Values
%
%Define Table conditions
mydir = dir('*_15fps.mat');
%Determine number of files in directory
numfiles = length(mydir);
%variable names
vnam_littab =
{'Condition','Day','Row_Pos','Col_Pos','Date','Plate','FPS','Pist_Cat','SampThick','Area_CS','Area_PistCS','Area_PistBordCS','Area_C
olCS','Area_ColBordCS','PistWid','ColWid','MidWid',...

'Force','Stress','Stress_Pist','Stress_PistBord','Stress_Col','Stress_ColBord','LongAx_Strain_Med','LongAx_Strain_Mean','Glob_Strain
_Med','Glob_Strain_Mean'};
%variable types
vtyp_littab =
{'categorical','categorical','categorical','categorical','categorical','categorical','categorical','categorical','double','double','double','double','
double','double','double','double','double',...
    'cell','cell','cell','cell','cell','cell','cell','cell','cell','cell'};
%Table Size
tsz_littab = [numfiles size(vtyp_littab,2)];

littab_xxeul = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_xxlag = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_yyeul = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_yylag = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_xyeul = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_xylag = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_xxeul_disp = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_xxlag_disp = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_yyeul_disp = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);
littab_yylag_disp = table('Size',tsz_littab,'VariableTypes',vtyp_littab,'VariableNames',vnam_littab);

areatab = readtable('Gel Cross Section.xlsx');
    t = categorical(areatab.Condition);
    areatab.Condition = t;

    t = categorical(areatab.Day);
    areatab.Day = t;

    t = categorical(areatab.Row_Pos);
    areatab.Row_Pos = t;

    t = categorical(areatab.Col_Pos);
    areatab.Col_Pos = t;

    t = categorical(areatab.Date);
    areatab.Date = t;
```

```
    t = categorical(areatab.Plate);
    areatab.Plate = t;

    t = categorical(areatab.Pist_Cat);
    areatab.Pist_Cat = t;



%Create and fill force table
fvnam = {'Amp','Row1','Row2'};
fvtyp = {'double','double','double'};
fsz = [12,3];
forcetab = table('Size',fsz,'VariableTypes',fvtyp,'VariableNames',fvnam);
forcetab{:,1} = [0:0.1:1,0]';
%Entry force values are in mN; calculate and convert to N
forcetab{2:end-1,2} = ((forcetab{2:end-1,1}*3.1816)-0.0934)/1000;
forcetab{3:end-1,3} = ((forcetab{3:end-1,1}*1.8763)-0.26)/1000;


fps = 60;
%FPS reduction factor; 3 leads to 60fps to 20
s = 4;
frmbrk = fps/s;
%Number of amp windows analyzed
ampfin = 10;
totfram_range = [1:(frmbrk*5);(frmbrk*5)+1:(frmbrk*5)*2;...
    ((frmbrk*5)*2)+1:(frmbrk*5)*3;((frmbrk*5)*3)+1:(frmbrk*5)*4;...
    ((frmbrk*5)*4)+1:(frmbrk*5)*5;((frmbrk*5)*5)+1:(frmbrk*5)*6;...
    ((frmbrk*5)*6)+1:(frmbrk*5)*7;((frmbrk*5)*7)+1:(frmbrk*5)*8;...
    ((frmbrk*5)*8)+1:(frmbrk*5)*9;((frmbrk*5)*9)+1:(frmbrk*5)*10;...
    ((frmbrk*5)*10)+1:(frmbrk*5)*11;((frmbrk*5)*11)+1:(frmbrk*5)*12];

fram_range = totfram_range(1:ampfin,:);


%Loop through and and unify datasets
for n = 1:numfiles
%Assign name of set to work with
nam = mydir(n).name;
% %Load dataset
load(nam)
%%Determine Size of Structures(correlates to number of frames analyzed)
% numframs = length(data_dic_save.strains);
%%Assign appropriate values to table
littab_xxeul = metadata_fillout(littab_xxeul,nam,n,forcetab,areatab);
littab_xxlag = metadata_fillout(littab_xxlag,nam,n,forcetab,areatab);
littab_yyeul = metadata_fillout(littab_yyeul,nam,n,forcetab,areatab);
littab_yylag = metadata_fillout(littab_yylag,nam,n,forcetab,areatab);
littab_xyeul = metadata_fillout(littab_xyeul,nam,n,forcetab,areatab);
littab_xylag = metadata_fillout(littab_xylag,nam,n,forcetab,areatab);
littab_xxeul_disp = metadata_fillout(littab_xxeul_disp,nam,n,forcetab,areatab);
littab_xxlag_disp = metadata_fillout(littab_xxlag_disp,nam,n,forcetab,areatab);
littab_yyeul_disp = metadata_fillout(littab_yyeul_disp,nam,n,forcetab,areatab);
littab_yylag_disp = metadata_fillout(littab_yylag_disp,nam,n,forcetab,areatab);


%Loop for pulling data from each frame
lamed = {};
medvect = [];
lamean = {};
meanvect = [];
maxvect = [];
minvect = [];
strdelt = [];
runmed = [];
```

```
numframs = length(data_dic_save.strains);

%Load particular datasets of concern
%This loads all frames of intended sample indicated by index i
   data1_xxeul = struct('plot_exx_cur_formatted', {data_dic_save.strains(1:numframs).plot_exx_cur_formatted});
   data1_xxlag = struct('plot_exx_ref_formatted', {data_dic_save.strains(1:numframs).plot_exx_ref_formatted});
   data1_yyeul = struct('plot_eyy_cur_formatted', {data_dic_save.strains(1:numframs).plot_eyy_cur_formatted});
   data1_yylag = struct('plot_eyy_ref_formatted', {data_dic_save.strains(1:numframs).plot_eyy_ref_formatted});
   data1_xyeul = struct('plot_exy_cur_formatted', {data_dic_save.strains(1:numframs).plot_exy_cur_formatted});
   data1_xylag = struct('plot_exy_ref_formatted', {data_dic_save.strains(1:numframs).plot_exy_ref_formatted});
   data1_xxeul_disp = struct('plot_u_cur_formatted', {data_dic_save.displacements(1:numframs).plot_u_cur_formatted});
   data1_xxlag_disp = struct('plot_u_ref_formatted', {data_dic_save.displacements(1:numframs).plot_u_ref_formatted});
   data1_yyeul_disp = struct('plot_v_cur_formatted', {data_dic_save.displacements(1:numframs).plot_v_cur_formatted});
   data1_yylag_disp = struct('plot_v_ref_formatted', {data_dic_save.displacements(1:numframs).plot_v_ref_formatted});
% numfram = length(data1_xxeul{1});


   for i2 = 1:numframs;
   %Basic Data Extraction. Clean up and refine later.This pulls frame by frame
   %intended datasets
   %This pulls specifc subset to work with;
   if littab_xxeul.Pist_Cat(n) == 'Right';
   data3_xxeul = data1_xxeul(i2).plot_exx_cur_formatted;
   data3_xxlag = data1_xxlag(i2).plot_exx_ref_formatted;
   data3_yyeul = data1_yyeul(i2).plot_eyy_cur_formatted;
   data3_yylag = data1_yylag(i2).plot_eyy_ref_formatted;
   data3_xyeul = data1_xyeul(i2).plot_exy_cur_formatted;
   data3_xylag = data1_xylag(i2).plot_exy_ref_formatted;
   data3_xxeul_disp = data1_xxeul_disp(i2).plot_u_cur_formatted;
   data3_xxlag_disp = data1_xxlag_disp(i2).plot_u_ref_formatted;
   data3_yyeul_disp = data1_yyeul_disp(i2).plot_v_cur_formatted;
   data3_yylag_disp = data1_yylag_disp(i2).plot_v_ref_formatted;
   else
   data3_xxeul = fliplr(data1_xxeul(i2).plot_exx_cur_formatted);
   data3_xxlag = fliplr(data1_xxlag(i2).plot_exx_ref_formatted);
   data3_yyeul = fliplr(data1_yyeul(i2).plot_eyy_cur_formatted);
   data3_yylag = fliplr(data1_yylag(i2).plot_eyy_ref_formatted);
   data3_xyeul = fliplr(data1_xyeul(i2).plot_exy_cur_formatted);
   data3_xylag = fliplr(data1_xylag(i2).plot_exy_ref_formatted);
   data3_xxeul_disp = fliplr(data1_xxeul_disp(i2).plot_u_cur_formatted);
   data3_xxlag_disp = fliplr(data1_xxlag_disp(i2).plot_u_ref_formatted);
   data3_yyeul_disp = fliplr(data1_yyeul_disp(i2).plot_v_cur_formatted);
   data3_yylag_disp = fliplr(data1_yylag_disp(i2).plot_v_ref_formatted);
   end

%%XX_eul
   [lamed_xxeul{i2},lamean_xxeul{i2},medvect_xxeul(i2),...
      meanvect_xxeul(i2),maxvect_xxeul(i2),minvect_xxeul(i2)] =...
      str_proc(data3_xxeul);
%%XX_lag
   [lamed_xxlag{i2},lamean_xxlag{i2},medvect_xxlag(i2),...
      meanvect_xxlag(i2),maxvect_xxlag(i2),minvect_xxlag(i2)] =...
      str_proc(data3_xxlag);

%%YY_eul
   [lamed_yyeul{i2},lamean_yyeul{i2},medvect_yyeul(i2),...
      meanvect_yyeul(i2),maxvect_yyeul(i2),minvect_yyeul(i2)] =...
      str_proc(data3_yyeul);
%%YY_lag
   [lamed_yylag{i2},lamean_yylag{i2},medvect_yylag(i2),...
      meanvect_yylag(i2),maxvect_yylag(i2),minvect_yylag(i2)] =...
      str_proc(data3_yylag);
%%XY_eul
   [lamed_xyeul{i2},lamean_xyeul{i2},medvect_xyeul(i2),...
      meanvect_xyeul(i2),maxvect_xyeul(i2),minvect_xyeul(i2)] =...
```

```
      str_proc(data3_xyeul);
%%XY_lag
   [lamed_xylag{i2},lamean_xylag{i2},medvect_xylag(i2),...
      meanvect_xylag(i2),maxvect_xylag(i2),minvect_xylag(i2)] =...
      str_proc(data3_xylag);

%%XX_eul_disp
   [lamed_xxeul_disp{i2},lamean_xxeul_disp{i2},medvect_xxeul_disp(i2),...
      meanvect_xxeul_disp(i2),maxvect_xxeul_disp(i2),minvect_xxeul_disp(i2)] =...
      str_proc(data3_xxeul_disp);
%%XX_lag_disp
   [lamed_xxlag_disp{i2},lamean_xxlag_disp{i2},medvect_xxlag_disp(i2),...
      meanvect_xxlag_disp(i2),maxvect_xxlag_disp(i2),minvect_xxlag_disp(i2)] =...
      str_proc(data3_xxlag_disp);

%%YY_eul_disp
   [lamed_yyeul_disp{i2},lamean_yyeul_disp{i2},medvect_yyeul_disp(i2),...
      meanvect_yyeul_disp(i2),maxvect_yyeul_disp(i2),minvect_yyeul_disp(i2)] =...
      str_proc(data3_yyeul_disp);
%%YY_lag_disp
   [lamed_yylag_disp{i2},lamean_yylag_disp{i2},medvect_yylag_disp(i2),...
      meanvect_yylag_disp(i2),maxvect_yylag_disp(i2),minvect_yylag_disp(i2)] =...
      str_proc(data3_yylag_disp);

   end
   %XXEUL
   littab_xxeul = tabpop(littab_xxeul,n,lamed_xxeul,lamean_xxeul,medvect_xxeul,meanvect_xxeul,...
      maxvect_xxeul,minvect_xxeul);
   clear lamed_xxeul lamean_xxeul medvect_xxeul meanvect_xxeul...

   %XXLAG
   littab_xxlag = tabpop(littab_xxlag,n,lamed_xxlag,lamean_xxlag,medvect_xxlag,meanvect_xxlag,...
      maxvect_xxlag,minvect_xxlag);
   clear lamed_xxlag lamean_xxlag medvect_xxlag meanvect_xxlag

   %%YYEUL
   littab_yyeul = tabpop(littab_yyeul,n,lamed_yyeul,lamean_yyeul,medvect_yyeul,meanvect_yyeul,...
      maxvect_yyeul,minvect_yyeul);
   clear lamed_yyeul lamean_yyeul medvect_yyeul meanvect_yyeul...

   %%YYLAG
   littab_yylag = tabpop(littab_yylag,n,lamed_yylag,lamean_yylag,medvect_yylag,meanvect_yylag,...
      maxvect_yylag,minvect_yylag);
   clear lamed_yylag lamean_yylag medvect_yylag meanvect_yylag

   %%XYEUL
   littab_xyeul = tabpop(littab_xyeul,n,lamed_xyeul,lamean_xyeul,medvect_xyeul,meanvect_xyeul,...
      maxvect_xyeul,minvect_xyeul);
   clear lamed_xyeul lamean_xyeul medvect_xyeul meanvect_xyeul...

   %%XYLAG
   littab_xylag = tabpop(littab_xylag,n,lamed_xylag,lamean_xylag,medvect_xylag,meanvect_xylag,...
      maxvect_xylag,minvect_xylag);
   clear lamed_xylag lamean_xylag medvect_xylag meanvect_xylag

   %%XXEUL_Disp
   littab_xxeul_disp = tabpop(littab_xxeul_disp,n,lamed_xxeul_disp,lamean_xxeul_disp,medvect_xxeul_disp,...
      meanvect_xxeul_disp,maxvect_xxeul_disp,minvect_xxeul_disp);
   clear lamed_xxeul_disp lamean_xxeul_disp medvect_xxeul_disp meanvect_xxeul_disp...

   %%XXLAG_Disp
   littab_xxlag_disp = tabpop(littab_xxlag_disp,n,lamed_xxlag_disp,lamean_xxlag_disp,medvect_xxlag_disp,...
      meanvect_xxlag_disp,maxvect_xxlag_disp,minvect_xxlag_disp);
   clear lamed_xxlag_disp lamean_xxlag_disp medvect_xxlag_disp meanvect_xxlag_disp

   %%XXEUL_Disp
```

```
    littab_yyeul_disp = tabpop(littab_yyeul_disp,n,lamed_yyeul_disp,lamean_yyeul_disp,medvect_yyeul_disp,...
        meanvect_yyeul_disp,maxvect_yyeul_disp,minvect_yyeul_disp);
    clear lamed_yyeul_disp lamean_yyeul_disp medvect_yyeul_disp meanvect_yyeul_disp...

    %%XXLAG_Disp
    littab_yylag_disp = tabpop(littab_yylag_disp,n,lamed_yylag_disp,lamean_yylag_disp,medvect_yylag_disp,...
        meanvect_yylag_disp,maxvect_yylag_disp,minvect_yylag_disp);
    clear lamed_yylag_disp lamean_yylag_disp medvect_yylag_disp meanvect_yylag_disp data_dic_save

end



fps = 15;
%Loop for pulling data from each frame
lamed = {};
medvect = [];
lamean = {};
meanvect = [];
maxvect = [];
minvect_xxeul = [];
strdelt = [];
runmed = [];

littab = littab_xxeul;
save('full_analysis_unified_mismatch_15fps_xxeul.mat','littab','fram_range','-mat');
clear littab

littab = littab_xxlag;
save('full_analysis_unified_mismatch_15fps_xxlag.mat','littab','fram_range','-mat');
clear littab

littab = littab_yyeul;
save('full_analysis_unified_mismatch_15fps_yyeul.mat','littab','fram_range','-mat');
clear littab

littab = littab_yylag;
save('full_analysis_unified_mismatch_15fps_yylag.mat','littab','fram_range','-mat');
clear littab

littab = littab_xyeul;
save('full_analysis_unified_mismatch_15fps_xyeul.mat','littab','fram_range','-mat');
clear littab

littab = littab_xylag;
save('full_analysis_unified_mismatch_15fps_xylag.mat','littab','fram_range','-mat');
clear littab

littab = littab_xxeul_disp;
save('full_analysis_unified_mismatch_15fps_xxeul_disp.mat','littab','fram_range','-mat');
clear littab

littab = littab_xxlag_disp;
save('full_analysis_unified_mismatch_15fps_xxlag_disp.mat','littab','fram_range','-mat');
clear littab

littab = littab_yyeul_disp;
save('full_analysis_unified_mismatch_15fps_yyeul_disp.mat','littab','fram_range','-mat');
clear littab

littab = littab_yylag_disp;
save('full_analysis_unified_mismatch_15fps_yylag_disp.mat','littab','fram_range','-mat');
clear littab
```

B-1.2 Supplemental Functions to Initial Processing of NCORR MATLAB Software Data

```
function littab = metadata_fillout(littab,nam,n,forcetab,areatab)
%LITTAB Table Setup
%   Fills in metadata information, measured area, and force for sample
%%Assign appropriate values to table
littab.Condition(n) = nam(7:9);
littab.Day(n)=nam(5);
littab.Row_Pos(n)=nam(4);
littab.Col_Pos(n)=nam(3);
littab.Date(n)=nam(10:13);
littab.Plate(n)=nam(1:2);
littab.FPS(n)=nam(19:20);

% Fill Area cross section. In mm^2, convert to m^2
area_ind = find(areatab.Condition(:,:) == littab.Condition(n,:) &...
      areatab.Day(:,:) == littab.Day(n,:) &...
      areatab.Row_Pos(:,:) == littab.Row_Pos(n,:) &...
      areatab.Col_Pos(:,:) == littab.Col_Pos(n,:) &...
      areatab.Date(:,:) == littab.Date(n,:) &...
      areatab.Plate(:,:) == littab.Plate(n,:));

   if length(area_ind)>1;
      area_ind = area_ind(2);
   end

littab.SampThick(n,:) = (10^-3)*areatab.Mid_H(area_ind,:);
littab.Area_CS(n,:) = (10^-6)*areatab.Area_mm2(area_ind,:);
littab.Area_PistCS(n,:) = (10^-6)*areatab.Pist_Area_mm2(area_ind,:);
littab.Area_PistBordCS(n,:) = (10^-6)*areatab.PistBorder_Area_mm2(area_ind,:);
littab.Area_ColCS(n,:) = (10^-6)*areatab.Col_Area_mm2(area_ind,:);
littab.Area_ColBordCS(n,:) = (10^-6)*areatab.ColBorder_Area_mm2(area_ind,:);
littab.PistWid(n,:) = (10^-3)*areatab.Pist_Side(area_ind,:);
littab.ColWid(n,:) = (10^-3)*areatab.Col_Side(area_ind,:);
littab.MidWid(n,:) = (10^-3)*areatab.Mid(area_ind,:);
littab.Pist_Cat(n) = areatab.Pist_Cat(area_ind);

if littab.Row_Pos(n) == '1' | littab.Row_Pos(n) == '4';
   littab.Force{n} = forcetab.Row1;
else
   littab.Force{n} = forcetab.Row2;
end

littab.Stress{n} = littab.Force{n}/littab.Area_CS(n);
littab.Stress_Pist{n} = littab.Force{n}/littab.Area_PistCS(n);
littab.Stress_PistBord{n} = littab.Force{n}/littab.Area_PistBordCS(n);
littab.Stress_Col{n} = littab.Force{n}/littab.Area_ColCS(n);
littab.Stress_ColBord{n} = littab.Force{n}/littab.Area_ColBordCS(n);

end



function [lamed,lamean,medvect,meanvect,maxvect,minvect] = str_proc(data3)
%Finds basic dataset defining features such as mean, median, min, max
%   Detailed explanation goes here
   %Find nonzero elements of the dataset
   nzelems= find(data3);
   usemat = NaN(size(data3));
   usemat(nzelems) = data3(nzelems);

   %Pull mediai2 strains from each column across longaxis
   lamed = median(usemat,1,'omitnan');
   %Pull overall media strain from observed values
   medvect = median(usemat,'all','omitnan');
   %Pull mean strains from each column across long axis
   lamean = mean(usemat,1,'omitnan');
```

```
    %Pull overall mean strain from observed values
    meanvect = mean(usemat,'all','omitnan');
    %Pull overall max strain from observed
    maxvect = max(usemat,[],'all','omitnan');
    %Pull min strain from each column from observed values
    minvect = min(usemat,[],'all','omitnan');

end

function table1 = tabpop(table1,n,lamed,lamean,medvect,meanvect,maxvect,minvect)
%Assign final values to table
    table1.LongAx_Strain_Med{n} = lamed;
    table1.LongAx_Strain_Mean{n} = lamean;
    table1.Glob_Strain_Med{n} = medvect;
    table1.Glob_Strain_Mean{n} = meanvect;
    table1.Glob_Max{n} = maxvect;
    table1.Glob_Min{n} = minvect;
end
```

B-1.3 Main Script and Supplemental Functions for the Calculation of Macro-Modulus Values

```
clear all
close all
load('C:\Users\mjpotte\Desktop\Local Processing\Strain Mapping Analysis\new15fpsanalysis_220921\New
Datamats\full_analysis_unified_mismatch_15fps_xxlag.mat');
indx_find;

%Load dataset
%load('P3C11dctl0506_all_15fps.mat');
%% Calculating FPS
fps = 60;
%FPS reduction factor; 3 leads to 60fps to 20
s = 4;
frmbrk = fps/s;
%Number of amp windows analyzed
ampfin = 9;
totfram_range = [1:(frmbrk*5);(frmbrk*5)+1:(frmbrk*5)*2;...
((frmbrk*5)*2)+1:(frmbrk*5)*3;((frmbrk*5)*3)+1:(frmbrk*5)*4;...
((frmbrk*5)*4)+1:(frmbrk*5)*5;((frmbrk*5)*5)+1:(frmbrk*5)*6;...
((frmbrk*5)*6)+1:(frmbrk*5)*7;((frmbrk*5)*7)+1:(frmbrk*5)*8;...
((frmbrk*5)*8)+1:(frmbrk*5)*9;((frmbrk*5)*9)+1:(frmbrk*5)*10;...
((frmbrk*5)*10)+1:(frmbrk*5)*11;((frmbrk*5)*11)+1:(frmbrk*5)*12];
fram_range = totfram_range(1:ampfin,:);

numsamps = height(littab);
stre_v_stra_med = {};
stre_v_stra_mean = {};
frame_max = [];
modvect_med = [];
modvect_mean = [];
pct_usable = 0.9;
for n = 1:numsamps;
%%Take length of strain matrix longaxis
%Pull Longaxis strain
   temp_med = littab.LongAx_Strain_Med{n};
   temp_mean = littab.LongAx_Strain_Mean{n};
   %numfram = length(littab.LongAx_Strain_Med{n});
   numfram = fram_range(end,end);

%% Segment Sample based on longaxis position
   %Output vectors containing indices, stress, and strains belonging to each section for that
   %sample
   %Indcell 1x5 cell array containing 1xn vectors where leftmost cell contains indices for leftmost region,
   %rightmost contains indices for rightmost region
   %Indices correspond to frame
   %stra_stre_cell 2x5 cell array containing 1xn vectors where leftmost
      %cell contains strain values in row 1 and stress values in row 2
      %corresponding to leftmost region
[indcell_med,stra_stre_cell_med] =
sampseg_eighth(temp_med,numfram,littab.ColWid(n),littab.MidWid(n),littab.PistWid(n),littab.Force{n},littab.SampThick(n),pct_usa
ble,fram_range);
[indcell_mean,stra_stre_cell_mean] =
sampseg_eighth(temp_mean,numfram,littab.ColWid(n),littab.MidWid(n),littab.PistWid(n),littab.Force{n},littab.SampThick(n),pct_us
able,fram_range);

%% Calculate the modulus of elasticity for each region of the sample
   %Input cells containing 1x5 bins of regional values
   %Output modulus for each region in 1x5 vect class double
   %Stack output vectors into matrix matching height dimensions of table
modvect_med(n,:) = singsampmod_rev221013(stra_stre_cell_med);
modvect_mean(n,:) = singsampmod_rev221013(stra_stre_cell_mean);

end
```

```
%% Calculate Stats and Plot
   %Input modulus matrix and which day and condition for output
   %Output Box Plots and Bar Charts
   %Figure 1: Box Plot for experimental condition with 4 subplots(1 per
      %day)
   %Figure 2: Box Plot for control condition with 4 subplots(1 per
      %day)
   %Figure 3: Bar Chart with Bars for Experimental vs Control within each
      %day. 10 bars, 5 pairs of exp vs control. Each Day is subplot
%Generating Box Plots

%Normalized nrm = 1;
%Non-normalized nrm = 0;
nrm = 0;
if nrm == 1;
%Experimental
%Median
[d0expmed_vals,d0expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,1},ind_exp{1,1});
[d1expmed_vals,d1expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,2},ind_exp{1,1});
[d3expmed_vals,d3expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,3},ind_exp{1,1});
[d7expmed_vals,d7expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,4},ind_exp{1,1});
%Mean
[d0expmean_vals,d0expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,1},ind_exp{1,1});
[d1expmean_vals,d1expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,2},ind_exp{1,1});
[d3expmean_vals,d3expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,3},ind_exp{1,1});
[d7expmean_vals,d7expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,4},ind_exp{1,1});

%Control
%Median
[d0ctlmed_vals,d0ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,1},ind_ctl{1,1});
[d1ctlmed_vals,d1ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,2},ind_ctl{1,1});
[d3ctlmed_vals,d3ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,3},ind_ctl{1,1});
[d7ctlmed_vals,d7ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,4},ind_ctl{1,1});
%Mean
[d0ctlmean_vals,d0ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,1},ind_ctl{1,1});
[d1ctlmean_vals,d1ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,2},ind_ctl{1,1});
[d3ctlmean_vals,d3ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,3},ind_ctl{1,1});
[d7ctlmean_vals,d7ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,4},ind_ctl{1,1});
boxplot_script_nrmlz
else
%Experimental
%Median
[d0expmed_vals,d0expmed_labs] = boxprep(modvect_med,ind_exp{1,1});
[d1expmed_vals,d1expmed_labs] = boxprep(modvect_med,ind_exp{1,2});
[d3expmed_vals,d3expmed_labs] = boxprep(modvect_med,ind_exp{1,3});
[d7expmed_vals,d7expmed_labs] = boxprep(modvect_med,ind_exp{1,4});
%Mean
[d0expmean_vals,d0expmean_labs] = boxprep(modvect_mean,ind_exp{1,1});
[d1expmean_vals,d1expmean_labs] = boxprep(modvect_mean,ind_exp{1,2});
[d3expmean_vals,d3expmean_labs] = boxprep(modvect_mean,ind_exp{1,3});
[d7expmean_vals,d7expmean_labs] = boxprep(modvect_mean,ind_exp{1,4});

%Control
%Median
[d0ctlmed_vals,d0ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,1});
[d1ctlmed_vals,d1ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,2});
[d3ctlmed_vals,d3ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,3});
[d7ctlmed_vals,d7ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,4});
%Mean
[d0ctlmean_vals,d0ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,1});
[d1ctlmean_vals,d1ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,2});
[d3ctlmean_vals,d3ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,3});
[d7ctlmean_vals,d7ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,4});
boxplot_1plotscript_mismatch

end
```

```
%Perform Stats to plot bar charts
%Experimental
%Median
d0expmed_barvals = basic_stats(modvect_med,ind_exp{1,1});
d1expmed_barvals = basic_stats(modvect_med,ind_exp{1,2});
d3expmed_barvals = basic_stats(modvect_med,ind_exp{1,3});
d7expmed_barvals = basic_stats(modvect_med,ind_exp{1,4});

%Mean
d0expmean_barvals = basic_stats(modvect_mean,ind_exp{1,1});
d1expmean_barvals = basic_stats(modvect_mean,ind_exp{1,2});
d3expmean_barvals = basic_stats(modvect_mean,ind_exp{1,3});
d7expmean_barvals = basic_stats(modvect_mean,ind_exp{1,4});

%Control
%Median
d0ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,1});
d1ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,2});
d3ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,3});
d7ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,4});

%Mean
d0ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,1});
d1ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,2});
d3ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,3});
d7ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,4});

errbarchart_script
%% Bin values based on stress or strain
   % Input cells containing 1x5 bins of regional values
   %Output bins for each cell
   %stra_stre_cell is a 1x5 cell array containing 2xn matrices with row 1
      %strain data and row 2 stress data
   %numbins determines how many bins to segment data into. Integer value.
   %binrow determines which row of input stra_stre_cell cell-contained
      %matrices to bin by(stress-2 or strain-1)
% bincell = sampbin(stra_stre_cell_med,numbins,binrow);

%% Ttests
stats_script_mismatch

function [indcell,stra_stre_cell] =
sampseg(temp_strain,numfram,colwid,midwid,pistwid,forcemat,SampThick,pct_usable,fram_range);

   %Output vectors containing indices, stress, and strains belonging to each section for that
   %sample
   %Indcell 1x5 cell array containing 1xn vectors where leftmost cell contains indices for leftmost region,
   %rightmost contains indices for rightmost region
   %Indices correspond to frame
   %stra_stre_cell 1x5 cell array containing 2xn vectors where leftmost
      %cell contains strain values in row 1 and stress values in row 2
      %corresponding to leftmost region

%Initialize values
indcell = {};
stra_stre_cell = {};
n_sect = 1;
for n0 = 1:numfram;

%%Calculate normal linear trend from perimeter width value to midwidth
%%value
%Piston Width is in PistWid Column
%Column Width is in ColWid Column
%Middle Width is in MidWid Column
%Frames are setup for a right piston
```

```
%Column to Middle
   %Finds dimensions of non-nan values within frame
   %'First non-nan and last non-nan to find range of sample strain signal'
   %Find range of strain signal
   longax_str = temp_strain{n0};
   %Find length of input matrix;
   im_len = length(longax_str);
   %Find actual signal to bound sample.
   nanind = find(isnan(longax_str));
   not_nanind = ~isnan(longax_str);
   longax_str(find(longax_str<0))=0;
   %Column side is left side of matrix; Piston side is right side
   range_ind0 = [find(not_nanind,1,'first'),find(not_nanind,1,'last')];
   %Find implemented region size if regions are 1/8 of usable range
   reg_sz = (range_ind0(2)-range_ind0(1))/n_sect;
   midframe = round((mean(range_ind0,'all')));
   %Find p% length of usable range
   usable_len = round((range_ind0(2)-range_ind0(1))*pct_usable);
   range_ind = [round(midframe-(usable_len/2)),round(midframe+(usable_len/2))];

   %Set frame boundary indices from left to right.
   inf_frams = [midframe-(reg_sz/2),midframe+(reg_sz/2)];
   bord_frams = [inf_frams(1)-reg_sz,inf_frams(1)-1;...
                 inf_frams(2)+1, inf_frams(2)+reg_sz];
   rem_frams = [bord_frams(1,1)-reg_sz,bord_frams(1,1)-1;...
                 bord_frams(2,2)+1,bord_frams(2,2)+reg_sz];
   if rem_frams(1,1)<=0;
      rem_frams(1,1) = 1;
   end
   if rem_frams(2,2)>im_len;
      rem_frams(2,2)=im_len;
   end
   samp_len = rem_frams(2,2)-rem_frams(1,1);
   %Divide into regions; split into 5 regions. Col_Side, Col_Bord,
   %Infarct, Pis_Bord, Pis_Side
   %Number of sections to divide into
   nsecth = round((range_ind(2)-range_ind(1))/n_sect);%Finds value to split range into fifths
   div_fram = unique([range_ind(1):nsecth:range_ind(2),range_ind(2)]); % Generates dividing frame markers

   %% For uniform division of region into fifths
   %Region range: Inclusive of first value up to border value;
      %exception: final window is inclusive of first and last value
      if n0 == 1;
         for n = 1:n_sect;
            if n < n_sect;
               indcell{1,n} = [div_fram(n):div_fram(n+1)-1];
               stra_stre_cell{1,n} = [longax_str(div_fram(n):div_fram(n+1)-1)];
            else
               indcell{1,n} = [div_fram(n):div_fram(n+1)];
               stra_stre_cell{1,n} = [longax_str(div_fram(n):div_fram(n+1))];
            end
         end
      else
         for n = 1:n_sect;
            if n < n_sect;
               indcell{1,n} = [indcell{1,n},div_fram(n):div_fram(n+1)-1];
               stra_stre_cell{1,n} = [stra_stre_cell{1,n},longax_str(div_fram(n):div_fram(n+1)-1)];
            else
               indcell{1,n} = [indcell{1,n},div_fram(n):div_fram(n+1)];
               stra_stre_cell{1,n} = [stra_stre_cell{1,n},longax_str(div_fram(n):div_fram(n+1))];
            end
         end
      end


   %% When dividing in segments
```

```
    %Calculate Estimated Width for each position
    %Column Side
    wcol_delt = (colwid-midwid)/length(range_ind0(1):midframe);
%     wcol_widmat = unique([colwid:-wcol_delt:midwid,midwid]);
%     wcol_widmat = linspace(colwid,midwid,length(range_ind0(1):midframe));
    %Piston Side
    wpis_delt = (pistwid-midwid)/length(midframe:range_ind0(2));
%     wpis_widmat = unique([midwid:wpis_delt:pistwid,pistwid]);
%     wpis_widmat = linspace(midwid,pistwid,length(midframe+1:range_ind0(2)));
    %Concatenate Widths
%     wid_sub = horzcat(wcol_widmat,wpis_widmat(2:end));

%    mid_indx = round(length(wid_sub)/2);
    start_ind = round(midframe-(usable_len/2));
    fin_ind = round(midframe+(usable_len/2))

    for n = start_ind:fin_ind;
        inddelt = n-midframe;
        if inddelt < 0;
            area_sub(n) = midwid-(inddelt*wcol_delt);
        elseif inddelt == 0;
            area_sub(n) = midwid;
        elseif inddelt > 0;
            area_sub(n) = midwid+(inddelt*wpis_delt);
        end
    end

    %Calculate Area for subsection
%     area_sub = wid_sub*SampThick;

    %Find Force on frame:
    [stp,col] = find(fram_range==n0);
    frc = forcemat(stp);

    %Calculate Stress for subsection
    stress_sub = frc./area_sub;
    if n0 == 1;

            for n = 1:n_sect;
                if n < n_sect;
                    stra_stre_cell{2,n} = [stress_sub(div_fram(n):div_fram(n+1)-1)];
                else
                    stra_stre_cell{2,n} = [stress_sub(div_fram(n):div_fram(n+1))];
                end
            end
    else
            for n = 1:n_sect;
                if n < n_sect;
                    stra_stre_cell{2,n} = [stra_stre_cell{2,n},stress_sub(div_fram(n):div_fram(n+1)-1)];
                else
                    stra_stre_cell{2,n} = [stra_stre_cell{2,n},stress_sub(div_fram(n):div_fram(n+1))];
                end
            end
    end
end
end



function modvect = sampmod_rev221004(stra_stre_cell_med);

%% Calculate the modulus of elasticity for each region of the sample
    %Input cells containing 1x5 bins of regional values
    %Output modulus for each region in 1x5 vect class double
    %Stack output vectors into matrix matching height dimensions of table
    modvect = [stra_stre_cell_med{1,1}']\...
        [stra_stre_cell_med{2,1}'];
```

```
end

function [valvect,labvect] = boxprep(modmat,indcnd)
%% Initialize Values
valvect = [];
labvect = [];
% labvect = [];
%% Pull values from matrix matching row indices for condition and day
usemat = modmat(indcnd,:);
[nr,nc] = size(usemat);
%% Generate column vector with all values stacked
for n = 1:nc;
   valvect = [valvect;usemat(:,n)];
   if n<10;
      labvect = [labvect;repmat(horzcat('Reg0',num2str(n)),nr,1)];
   elseif n>=10;
      labvect = [labvect;repmat(horzcat('Reg',num2str(n)),nr,1)];
   end
end
%% Generate Label Column Vector

% labvect = [repmat('CS ',nr,1);repmat('CBS',nr,1);repmat('IS ',nr,1);repmat('PBS',nr,1);repmat('PS ',nr,1)];
end


%Box Plot Script
%% Experimental
%fontsizes
titfsz = 14;
lgdfsz = 10;
labfsz = 12;
fgwd = 10.1;%cm
fght = 8.1;%cm
fgun = 'centimeters';

%Convert Modulus Units to Pa
strconv = 1;%10^-3;
ymn = 0;
ymx = 60;%.055;
%Median Strain
f1 = figure(1)
f1.Units = fgun;
f1.Position = [2.54 2.54 fgwd fght];
[expmedvals,expmedlabs] = boxmismatchfun(d0expmed_vals,d1expmed_vals,d3expmed_vals,d7expmed_vals);
boxplot(expmedvals*strconv,expmedlabs);
title('Wounded','fontsize',titfsz)
%xlabel('Treatment','fontsize',labfsz)
ylabel('Modulus(Pa)','fontsize',labfsz)
ylim([ymn ymx]);

%% Control
%Median Strain
f3 = figure(3)
f3.Units = fgun;
f3.Position = [2.54 2.54 fgwd fght];
[ctlmedvals,ctlmedlabs] = boxmismatchfun(d0ctlmed_vals,d1ctlmed_vals,d3ctlmed_vals,d7ctlmed_vals);
boxplot(ctlmedvals*strconv,ctlmedlabs);
title('Control','fontsize',titfsz)
%xlabel('Treatment','fontsize',labfsz)
ylabel('Modulus(Pa)','fontsize',labfsz)
ylim([ymn ymx]);


function [output_mat] = basic_stats(inmat,ind)
%Pull values for day and condition
mat1 = inmat(ind,:);
```

```
%Find Averages and std errors
avg = mean(mat1,1,'omitnan');
std_dev1 = std(mat1,0,1,'omitnan');
std_err1 = std_dev1/sqrt(size(mat1,1));
samp_cnts = size(mat1,1);

output_mat = [avg;std_dev1;std_err1]%;samp_cnts];
end


%% Stats for Medians
%EXP
compars = ["0vs1";"0vs3";"0vs7";"1vs3";"1vs7";"3vs7"];
[expmed_h1,expmed_p1] = ttest(modvect_med(ind_exp{1,1}),modvect_med(ind_exp{1,2}));...
[expmed_h2,expmed_p2] = ttest(modvect_med(ind_exp{1,1}),modvect_med(ind_exp{1,3}));...
[expmed_h3,expmed_p3] = ttest(modvect_med(ind_exp{1,1}),modvect_med(ind_exp{1,4}));...
[expmed_h4,expmed_p4] = ttest(modvect_med(ind_exp{1,2}),modvect_med(ind_exp{1,3}));...
[expmed_h5,expmed_p5] = ttest(modvect_med(ind_exp{1,2}),modvect_med(ind_exp{1,4}));...
[expmed_h6,expmed_p6] = ttest(modvect_med(ind_exp{1,3}),modvect_med(ind_exp{1,4}));
Expmed = [expmed_p1;expmed_p2;expmed_p3;expmed_p4;expmed_p5;expmed_p6];

%Ctl
[ctlmed_h1,ctlmed_p1] = ttest(modvect_med(ind_ctl{1,1}),modvect_med(ind_ctl{1,2}));...
[ctlmed_h2,ctlmed_p2] = ttest(modvect_med(ind_ctl{1,1}),modvect_med(ind_ctl{1,3}));...
[ctlmed_h3,ctlmed_p3] = ttest(modvect_med(ind_ctl{1,1}),modvect_med(ind_ctl{1,4}));...
[ctlmed_h4,ctlmed_p4] = ttest(modvect_med(ind_ctl{1,2}),modvect_med(ind_ctl{1,3}));...
[ctlmed_h5,ctlmed_p5] = ttest(modvect_med(ind_ctl{1,2}),modvect_med(ind_ctl{1,4}));...
[ctlmed_h6,ctlmed_p6] = ttest(modvect_med(ind_ctl{1,3}),modvect_med(ind_ctl{1,4}));
Ctlmed = [ctlmed_p1;ctlmed_p2;ctlmed_p3;ctlmed_p4;ctlmed_p5;ctlmed_p6];

samcompmeds = table(compars,Expmed,Ctlmed);

expvsctl = ["Day0";"Day1";"Day3";"Day7"];
[compmed_h1,compmed_p1] = ttest2(modvect_med(ind_exp{1,1}),modvect_med(ind_ctl{1,1}));
[compmed_h2,compmed_p2] = ttest2(modvect_med(ind_exp{1,2}),modvect_med(ind_ctl{1,2}));
[compmed_h2,compmed_p3] = ttest2(modvect_med(ind_exp{1,3}),modvect_med(ind_ctl{1,3}));
[compmed_h2,compmed_p4] = ttest2(modvect_med(ind_exp{1,4}),modvect_med(ind_ctl{1,4}));

tstsmed = [compmed_p1;compmed_p2;compmed_p3;compmed_p4];
crosscompmeds = table(expvsctl,tstsmed);


%save('ModStats_mismatch.mat','samcompmeans','crosscompmeans','samcompmeds','crosscompmeds','-mat');
```

B-1.4 Main Script and Supplemental Functions for the Calculation of Regional-Modulus Values

```
clear all
close all
load( datamat filepath );
indx_find;

%Load dataset
%% Calculating FPS
fps = 60;
%FPS reduction factor; 3 leads to 60fps to 20
s = 4;
frmbrk = fps/s;
%Number of amp windows analyzed
ampfin = 9;
totfram_range = [1:(frmbrk*5);(frmbrk*5)+1:(frmbrk*5)*2;...
((frmbrk*5)*2)+1:(frmbrk*5)*3;((frmbrk*5)*3)+1:(frmbrk*5)*4;...
((frmbrk*5)*4)+1:(frmbrk*5)*5;((frmbrk*5)*5)+1:(frmbrk*5)*6;...
((frmbrk*5)*6)+1:(frmbrk*5)*7;((frmbrk*5)*7)+1:(frmbrk*5)*8;...
((frmbrk*5)*8)+1:(frmbrk*5)*9;((frmbrk*5)*9)+1:(frmbrk*5)*10;...
((frmbrk*5)*10)+1:(frmbrk*5)*11;((frmbrk*5)*11)+1:(frmbrk*5)*12];
fram_range = totfram_range(1:ampfin,:);

numsamps = height(littab);
stre_v_stra_med = {};
stre_v_stra_mean = {};
frame_max = [];
modvect_med = [];
modvect_mean = [];
pct_usable = 0.9;
for n = 1:numsamps;
%%Take length of strain matrix longaxis
%Pull Longaxis strain
   temp_med = littab.LongAx_Strain_Med{n};
   temp_mean = littab.LongAx_Strain_Mean{n};
   %numfram = length(littab.LongAx_Strain_Med{n});
   numfram = fram_range(end,end);

%% Segment Sample based on longaxis position
   %Output vectors containing indices, stress, and strains belonging to each section for that
   %sample
   %Indcell 1x5 cell array containing 1xn vectors where leftmost cell contains indices for leftmost region,
   %rightmost contains indices for rightmost region
   %Indices correspond to frame
   %stra_stre_cell 2x5 cell array containing 1xn vectors where leftmost
      %cell contains strain values in row 1 and stress values in row 2
      %corresponding to leftmost region
[indcell_med,stra_stre_cell_med] =
sampseg_eighth(temp_med,numfram,littab.ColWid(n),littab.MidWid(n),littab.PistWid(n),littab.Force{n},littab.SampThick(n),pct_usa
ble,fram_range);
[indcell_mean,stra_stre_cell_mean] =
sampseg_eighth(temp_mean,numfram,littab.ColWid(n),littab.MidWid(n),littab.PistWid(n),littab.Force{n},littab.SampThick(n),pct_us
able,fram_range);

%% Calculate the modulus of elasticity for each region of the sample
   %Input cells containing 1x5 bins of regional values
   %Output modulus for each region in 1x5 vect class double
   %Stack output vectors into matrix matching height dimensions of table
modvect_med(n,:) = sampmod_rev221004(stra_stre_cell_med);
modvect_mean(n,:) = sampmod_rev221004(stra_stre_cell_mean);
svsmat_med{n,1} = sampsvs_prep_rev221019(stra_stre_cell_med);
svsmat_mean{n,1} = sampsvs_prep_rev221019(stra_stre_cell_mean);

end
%% Calculate Stats and Plot
```

```
%    Input modulus matrix and which day and condition for output
%    Output Box Plots and Bar Charts
%    Figure 1: Box Plot for experimental condition with 4 subplots(1 per
%       day)
%    Figure 2: Box Plot for control condition with 4 subplots(1 per
%       day)
%    Figure 3: Bar Chart with Bars for Experimental vs Control within each
%       day. 10 bars, 5 pairs of exp vs control. Each Day is subplot
% Generating Box Plots

% Normalized nrm = 1;
% Non-normalized nrm = 0;
nrm = 0;
if nrm == 1;
%Experimental
%Median
[d0expmed_vals,d0expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,1},ind_exp{1,1});
[d1expmed_vals,d1expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,2},ind_exp{1,1});
[d3expmed_vals,d3expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,3},ind_exp{1,1});
[d7expmed_vals,d7expmed_labs] = boxprep_nrmlz(modvect_med,ind_exp{1,4},ind_exp{1,1});
%Mean
[d0expmean_vals,d0expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,1},ind_exp{1,1});
[d1expmean_vals,d1expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,2},ind_exp{1,1});
[d3expmean_vals,d3expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,3},ind_exp{1,1});
[d7expmean_vals,d7expmean_labs] = boxprep_nrmlz(modvect_mean,ind_exp{1,4},ind_exp{1,1});

%Control
%Median
[d0ctlmed_vals,d0ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,1},ind_ctl{1,1});
[d1ctlmed_vals,d1ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,2},ind_ctl{1,1});
[d3ctlmed_vals,d3ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,3},ind_ctl{1,1});
[d7ctlmed_vals,d7ctlmed_labs] = boxprep_nrmlz(modvect_med,ind_ctl{1,4},ind_ctl{1,1});
%Mean
[d0ctlmean_vals,d0ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,1},ind_ctl{1,1});
[d1ctlmean_vals,d1ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,2},ind_ctl{1,1});
[d3ctlmean_vals,d3ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,3},ind_ctl{1,1});
[d7ctlmean_vals,d7ctlmean_labs] = boxprep_nrmlz(modvect_mean,ind_ctl{1,4},ind_ctl{1,1});
else
%Experimental
%Median
[d0expmed_vals,d0expmed_labs] = boxprep(modvect_med,ind_exp{1,1});
[d1expmed_vals,d1expmed_labs] = boxprep(modvect_med,ind_exp{1,2});
[d3expmed_vals,d3expmed_labs] = boxprep(modvect_med,ind_exp{1,3});
[d7expmed_vals,d7expmed_labs] = boxprep(modvect_med,ind_exp{1,4});
%Mean
[d0expmean_vals,d0expmean_labs] = boxprep(modvect_mean,ind_exp{1,1});
[d1expmean_vals,d1expmean_labs] = boxprep(modvect_mean,ind_exp{1,2});
[d3expmean_vals,d3expmean_labs] = boxprep(modvect_mean,ind_exp{1,3});
[d7expmean_vals,d7expmean_labs] = boxprep(modvect_mean,ind_exp{1,4});

%Control
%Median
[d0ctlmed_vals,d0ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,1});
[d1ctlmed_vals,d1ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,2});
[d3ctlmed_vals,d3ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,3});
[d7ctlmed_vals,d7ctlmed_labs] = boxprep(modvect_med,ind_ctl{1,4});
%Mean
[d0ctlmean_vals,d0ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,1});
[d1ctlmean_vals,d1ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,2});
[d3ctlmean_vals,d3ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,3});
[d7ctlmean_vals,d7ctlmean_labs] = boxprep(modvect_mean,ind_ctl{1,4});
end
boxplot_script

%Perform Stats to plot bar charts
%Experimental
```

```
%Median
d0expmed_barvals = basic_stats(modvect_med,ind_exp{1,1});
d1expmed_barvals = basic_stats(modvect_med,ind_exp{1,2});
d3expmed_barvals = basic_stats(modvect_med,ind_exp{1,3});
d7expmed_barvals = basic_stats(modvect_med,ind_exp{1,4});

%Mean
d0expmean_barvals = basic_stats(modvect_mean,ind_exp{1,1});
d1expmean_barvals = basic_stats(modvect_mean,ind_exp{1,2});
d3expmean_barvals = basic_stats(modvect_mean,ind_exp{1,3});
d7expmean_barvals = basic_stats(modvect_mean,ind_exp{1,4});

%Control
%Median
d0ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,1});
d1ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,2});
d3ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,3});
d7ctlmed_barvals = basic_stats(modvect_med,ind_ctl{1,4});

%Mean
d0ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,1});
d1ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,2});
d3ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,3});
d7ctlmean_barvals = basic_stats(modvect_mean,ind_ctl{1,4});

errbarchart_script
%% Bin values based on stress or strain
   % Input cells containing 1x5 bins of regional values
   %Output bins for each cell
   %stra_stre_cell is a 1x5 cell array containing 2xn matrices with row 1
      %strain data and row 2 stress data
   %numbins determines how many bins to segment data into. Integer value.
   %binrow determines which row of input stra_stre_cell cell-contained
      %matrices to bin by(stress-2 or strain-1)
% bincell = sampbin(stra_stre_cell_med,numbins,binrow);
stats_script_med

%%Stress vs strain binning and plotting
for n = 1:4;
   %Pull Vals from samples for day and condition, extend together
   svsexpmean{1,n} = vect_extend_rev221019(svsmat_mean,ind_exp{1,n});
   svsctlmean{1,n} = vect_extend_rev221019(svsmat_mean,ind_ctl{1,n});
   svsexpmed{1,n} = vect_extend_rev221019(svsmat_med,ind_exp{1,n});
   svsctlmed{1,n} = vect_extend_rev221019(svsmat_med,ind_ctl{1,n});

   %then bin and and find averages
   %Cells 1-4 correspond to days 0,1,3,7
   numbins = 9;
   svsexpmean_binned{1,n} = bin_and_stats_rev_221019(svsexpmean{1,n},numbins);
   svsctlmean_binned{1,n} = bin_and_stats_rev_221019(svsctlmean{1,n},numbins);
   svsexpmed_binned{1,n} = bin_and_stats_rev_221019(svsexpmed{1,n},numbins);
   svsctlmed_binned{1,n} = bin_and_stats_rev_221019(svsctlmed{1,n},numbins);
end

stre_v_stra_reg3_medtileplotting_rev221103_fig4

function [indcell,stra_stre_cell] =
sampseg(temp_strain,numfram,colwid,midwid,pistwid,forcemat,SampThick,pct_usable,fram_range);

   %Output vectors containing indices, stress, and strains belonging to each section for that
   %sample
   %Indcell 1x5 cell array containing 1xn vectors where leftmost cell contains indices for leftmost region,
   %rightmost contains indices for rightmost region
   %Indices correspond to frame
   %stra_stre_cell 1x5 cell array containing 2xn vectors where leftmost
      %cell contains strain values in row 1 and stress values in row 2
```

```
        %corresponding to leftmost region

%Initialize values
indcell = {};
stra_stre_cell = {};
n_sect = 16;
for n0 = 1:numfram;

%%Calculate normal linear trend from perimeter width value to midwidth
%%value
%Piston Width is in PistWid Column
%Column Width is in ColWid Column
%Middle Width is in MidWid Column
%Frames are setup for a right piston
%Column to Middle
   %Finds dimensions of non-nan values within frame
   %'First non-nan and last non-nan to find range of sample strain signal'
   %Find range of strain signal
   longax_str = temp_strain{n0};
   %Find length of input matrix;
   im_len = length(longax_str);
   %Find actual signal to bound sample.
   nanind = find(isnan(longax_str));
   not_nanind = ~isnan(longax_str);
   longax_str(find(longax_str<0))=0;
   %Column side is left side of matrix; Piston side is right side
   range_ind0 = [find(not_nanind,1,'first'),find(not_nanind,1,'last')];
   %Find implemented region size if regions are 1/8 of usable range
   reg_sz = (range_ind0(2)-range_ind0(1))/n_sect;
   midframe = round((mean(range_ind0,'all')));
   %Find p% length of usable range
   usable_len = round((range_ind0(2)-range_ind0(1))*pct_usable);
   range_ind = [round(midframe-(usable_len/2)),round(midframe+(usable_len/2))];

   %Set frame boundary indices from left to right.
   inf_frams = [midframe-(reg_sz/2),midframe+(reg_sz/2)];
   bord_frams = [inf_frams(1)-reg_sz,inf_frams(1)-1;...
                 inf_frams(2)+1, inf_frams(2)+reg_sz];
   rem_frams = [bord_frams(1,1)-reg_sz,bord_frams(1,1)-1;...
                 bord_frams(2,2)+1,bord_frams(2,2)+reg_sz];
   if rem_frams(1,1)<=0;
      rem_frams(1,1) = 1;
   end
   if rem_frams(2,2)>im_len;
      rem_frams(2,2)=im_len;
   end
   samp_len = rem_frams(2,2)-rem_frams(1,1);
   %Divide into regions; split into 5 regions. Col_Side, Col_Bord,
   %Infarct, Pis_Bord, Pis_Side
   %Number of sections to divide into
   nsecth = round((range_ind(2)-range_ind(1))/n_sect);%Finds value to split range into fifths
   div_fram = unique([range_ind(1):nsecth:range_ind(2),range_ind(2)]); % Generates dividing frame markers

   %% For uniform division of region into fifths
   %Region range: Inclusive of first value up to border value;
      %exception: final window is inclusive of first and last value
      if n0 == 1;
         for n = 1:n_sect-1;
            if n < n_sect-1;
               indcell{1,n} = [div_fram(n):div_fram(n+1)-1];
               stra_stre_cell{1,n} = [longax_str(div_fram(n):div_fram(n+1)-1)];
            else
               indcell{1,n} = [div_fram(n):div_fram(n+1)];
               stra_stre_cell{1,n} = [longax_str(div_fram(n):div_fram(n+1))];
            end
         end
```

```
    end
        for n = 1:n_sect-1;
            if n < n_sect-1;
                indcell{1,n} = [indcell{1,n},div_fram(n):div_fram(n+1)-1];
                stra_stre_cell{1,n} = [stra_stre_cell{1,n},longax_str(div_fram(n):div_fram(n+1)-1)];
            else
                indcell{1,n} = [indcell{1,n},div_fram(n):div_fram(n+1)];
                stra_stre_cell{1,n} = [stra_stre_cell{1,n},longax_str(div_fram(n):div_fram(n+1))];
            end
        end


    %% When dividing in segments
    %Calculate Estimated Width for each position
    %Column Side
    wcol_delt = (colwid-midwid)/length(range_ind0(1):midframe);
%     wcol_widmat = unique([colwid:-wcol_delt:midwid,midwid]);
%     wcol_widmat = linspace(colwid,midwid,length(range_ind0(1):midframe));
    %Piston Side
    wpis_delt = (pistwid-midwid)/length(midframe:range_ind0(2));
%     wpis_widmat = unique([midwid:wpis_delt:pistwid,pistwid]);
%     wpis_widmat = linspace(midwid,pistwid,length(midframe+1:range_ind0(2)));
    %Concatenate Widths
%     wid_sub = horzcat(wcol_widmat,wpis_widmat(2:end));

%    mid_indx = round(length(wid_sub)/2);
    start_ind = round(midframe-(usable_len/2));
    fin_ind = round(midframe+(usable_len/2))

    for n = start_ind:fin_ind;
        inddelt = n-midframe;
        if inddelt < 0;
            area_sub(n) = midwid-(inddelt*wcol_delt);
        elseif inddelt == 0;
            area_sub(n) = midwid;
        elseif inddelt > 0;
            area_sub(n) = midwid+(inddelt*wpis_delt);
        end
    end

    %Calculate Area for subsection
%     area_sub = wid_sub*SampThick;

    %Find Force on frame:
    [stp,col] = find(fram_range==n0);
    frc = forcemat(stp);

    %Calculate Stress for subsection
    stress_sub = frc./area_sub;
    if n0 == 1;

        for n = 1:n_sect-1;
            if n < n_sect-1;
                stra_stre_cell{2,n} = [stress_sub(div_fram(n):div_fram(n+1)-1)];
            else
                stra_stre_cell{2,n} = [stress_sub(div_fram(n):div_fram(n+1))];
            end
        end
    end
        for n = 1:n_sect-1;
            if n < n_sect-1;
                stra_stre_cell{2,n} = [stra_stre_cell{2,n},stress_sub(div_fram(n):div_fram(n+1)-1)];
            else
                stra_stre_cell{2,n} = [stra_stre_cell{2,n},stress_sub(div_fram(n):div_fram(n+1))];
            end
        end
```

```
end
end

function modvect = sampmod_rev221004(stra_stre_cell_med);

%% Calculate the modulus of elasticity for each region of the sample
   %Input cells containing 1x5 bins of regional values
   %Output modulus for each region in 1x5 vect class double
   %Stack output vectors into matrix matching height dimensions of table
%    vectsz = size(stra_stre_cell_med,2);
   vectsz = 5;
   modvect(1) = [stra_stre_cell_med{1,4}';stra_stre_cell_med{1,5}';stra_stre_cell_med{1,6}']\...
      [stra_stre_cell_med{2,4}';stra_stre_cell_med{2,5}';stra_stre_cell_med{2,6}'];
   modvect(2) = [stra_stre_cell_med{1,8}';stra_stre_cell_med{1,9}';stra_stre_cell_med{1,10}']\...
      [stra_stre_cell_med{2,8}';stra_stre_cell_med{2,9}';stra_stre_cell_med{2,10}'];
   modvect(3) = [stra_stre_cell_med{1,11}';stra_stre_cell_med{1,12}';stra_stre_cell_med{1,13}']\...
      [stra_stre_cell_med{2,11}';stra_stre_cell_med{2,12}';stra_stre_cell_med{2,13}'];

%    for n = 1:vectsz;
%       modvect(n) = stra_stre_cell_med{1,n}'\stra_stre_cell_med{2,n}';
%    end
      %    modvect = [stra_stre_cell_med{1,1}'\stra_stre_cell_med{2,1}',...
%          stra_stre_cell_med{1,2}'\stra_stre_cell_med{2,2}',...
%          stra_stre_cell_med{1,3}'\stra_stre_cell_med{2,3}',...
%          stra_stre_cell_med{1,4}'\stra_stre_cell_med{2,4}',...
%          stra_stre_cell_med{1,5}'\stra_stre_cell_med{2,5}'];

end



function [valvect,labvect] = boxprep(modmat,indcnd)
%% Initialize Values
valvect = [];
labvect = [];
% labvect = [];
%% Pull values from matrix matching row indices for condition and day
usemat = modmat(indcnd,:);
[nr,nc] = size(usemat);
%% Generate column vector with all values stacked
for n = 1:nc;
   valvect = [valvect;usemat(:,n)];
   if n == 1;
      labvect = [labvect;repmat(horzcat('Remote '),nr,1)];%num2str(n)),nr,1)];
   elseif n == 2;
      labvect = [labvect;repmat(horzcat('Border '),nr,1)];%,num2str(n)),nr,1)];
   elseif n == 3;
      labvect = [labvect;repmat(horzcat('Infarct'),nr,1)];%,num2str(n)),nr,1)];
   end
%    if n<10;
%       labvect = [labvect;repmat(horzcat('Reg0',num2str(n)),nr,1)];
%    elseif n>=10;
%       labvect = [labvect;repmat(horzcat('Reg',num2str(n)),nr,1)];
%    end
end
%% Generate Label Column Vector

% labvect = [repmat('CS ',nr,1);repmat('CBS',nr,1);repmat('IS ',nr,1);repmat('PBS',nr,1);repmat('PS ',nr,1)];
end


%Box Plot Script
%% Woundederimental
titfsz = 16;
lgdfsz = 10;
```

```
labfsz = 14;

%Convert Modulus Units to kPa
strconv = 1;%10^-3;
ymn = 0;
ymx = 175*strconv;%0.125;
ylab = 'Modulus (Pa)';
xlab = 'Region';
fgwd = 28;
fght = 18;
%Median Strain
f1 = figure(1)
f1.Units = 'centimeters';
f1.Position = [2.54 2.54 fgwd fght];
t = tiledlayout(2,4,'TileSpacing','none');
title(t,'Regional Modulus','fontsize',titfsz);

nexttile
%subplot(2,2,1)
boxplot(d0expmed_vals*strconv,d0expmed_labs);
title('Day 0 Wounded','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

nexttile
%subplot(2,2,2)
boxplot(d1expmed_vals*strconv,d1expmed_labs);
title('Day 1 Wounded','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

nexttile
%subplot(2,2,3)
boxplot(d3expmed_vals*strconv,d3expmed_labs);
title('Day 3 Wounded','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

nexttile
%subplot(2,2,4)
boxplot(d7expmed_vals*strconv,d7expmed_labs);
title('Day 7 Wounded','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);
%Control
%Median Strain
% f3 = figure(3)
% f3.Units = 'centimeters';
% f3.Position = [2.54 2.54 fgwd fght];

nexttile
%subplot(2,2,1)
boxplot(d0ctlmed_vals*strconv,d0ctlmed_labs);
title('Day 0 Control','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

nexttile
%subplot(2,2,2)
boxplot(d1ctlmed_vals*strconv,d1ctlmed_labs);
title('Day 1 Control','fontsize',titfsz)
```

```
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

nexttile
%subplot(2,2,3)
boxplot(d3ctlmed_vals*strconv,d3ctlmed_labs);
title('Day 3 Control','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

nexttile
%subplot(2,2,4)
boxplot(d7ctlmed_vals*strconv,d7ctlmed_labs);
title('Day 7 Control','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

%Experimental
%Mean strain
f2 = figure(2)
subplot(2,2,1)
boxplot(d0expmean_vals*strconv,d0expmean_labs);
title('Day 0 Wounded (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

subplot(2,2,2)
boxplot(d1expmean_vals*strconv,d1expmean_labs);
title('Day 1 Wounded (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

subplot(2,2,3)
boxplot(d3expmean_vals*strconv,d3expmean_labs);
title('Day 3 Wounded (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

subplot(2,2,4)
boxplot(d7expmean_vals*strconv,d7expmean_labs);
title('Day 7 Wounded (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

%% Control
%Mean strain
f4 = figure(4)
subplot(2,2,1)
boxplot(d0ctlmean_vals*strconv,d0ctlmean_labs);
title('Day 0 Control (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

subplot(2,2,2)
boxplot(d1ctlmean_vals*strconv,d1ctlmean_labs);
title('Day 1 Control (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
```

```
ylim([ymn ymx]);

subplot(2,2,3)
boxplot(d3ctlmean_vals*strconv,d3ctlmean_labs);
title('Day 3 Control (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);

subplot(2,2,4)
boxplot(d7ctlmean_vals*strconv,d7ctlmean_labs);
title('Day 7 Control (Using Mean Strain)','fontsize',titfsz)
xlabel(xlab,'fontsize',labfsz)
ylabel(ylab,'fontsize',labfsz)
ylim([ymn ymx]);


function [output_mat] = basic_stats(inmat,ind)
%Pull values for day and condition
mat1 = inmat(ind,:);

%Find Averages and std errors
avg = mean(mat1,1,'omitnan');
std_dev1 = std(mat1,0,1,'omitnan');
std_err1 = std_dev1/sqrt(size(mat1,1));
samp_cnts = size(mat1,1);

output_mat = [avg;std_dev1;std_err1]%;samp_cnts];
end

clear xtipserr
%Bar Chart Scripts
titfsz = 16;
lgdfsz = 10;
labfsz = 14;
ylb = 'Modulus (Pa)';
xlb = 'Region';
fgwd = 28;
fght = 18;
lgd = {'Wounded';'Control'};
xtlabs = {'Remote';'Border';'Infarct'};
%Modulus Unit Conversion
uniconv = 1;%10^-3;
%Bar Prep
%initialize vars
d0med_bldmat = [];
d1med_bldmat = [];
d3med_bldmat = [];
d7med_bldmat = [];
expmed_bldmat = []
expmederr_bldmat = []
ctlmed_bldmat = [];
ctlmederr_bldmat = [];
d0mederr_bldmat = [];
d1mederr_bldmat = [];
d3mederr_bldmat = [];
d7mederr_bldmat = [];

d0mean_bldmat = [];
d1mean_bldmat = [];
d3mean_bldmat = [];
d7mean_bldmat = [];

d0meanerr_bldmat = [];
d1meanerr_bldmat = [];
d3meanerr_bldmat = [];
```

```
d7meanerr_bldmat = [];

ymn = 0;
ymx = 125*uniconv;%0.125;

nmrws = size(d0expmed_barvals,2);

for n = 1:nmrws;
   d0med_bldmat = [d0med_bldmat;d0expmed_barvals(1,n),d0ctlmed_barvals(1,n)];
   d1med_bldmat = [d1med_bldmat;d1expmed_barvals(1,n),d1ctlmed_barvals(1,n)];
   d3med_bldmat = [d3med_bldmat;d3expmed_barvals(1,n),d3ctlmed_barvals(1,n)];
   d7med_bldmat = [d7med_bldmat;d7expmed_barvals(1,n),d7ctlmed_barvals(1,n)];
   d0mean_bldmat = [d0mean_bldmat;d0expmean_barvals(1,n),d0ctlmean_barvals(1,n)];
   d1mean_bldmat = [d1mean_bldmat;d1expmean_barvals(1,n),d1ctlmean_barvals(1,n)];
   d3mean_bldmat = [d3mean_bldmat;d3expmean_barvals(1,n),d3ctlmean_barvals(1,n)];
   d7mean_bldmat = [d7mean_bldmat;d7expmean_barvals(1,n),d7ctlmean_barvals(1,n)];
   expmed_bldmat = [expmed_bldmat;...
               d0expmed_barvals(1,n),d1expmed_barvals(1,n),...
               d3expmed_barvals(1,n),d7expmed_barvals(1,n)];
   ctlmed_bldmat = [ctlmed_bldmat;...
               d0ctlmed_barvals(1,n),d1ctlmed_barvals(1,n),...
               d3ctlmed_barvals(1,n),d7ctlmed_barvals(1,n)];
   if n == 1;
      xlabs{1,n} = 'Remote ';%horzcat('Reg',num2str(n));
   elseif n == 2;
      xlabs{1,n} = 'Border ';%horzcat('Reg',num2str(n));
   elseif n == 3;
      xlabs{1,n} = 'Infarct';%horzcat('Reg',num2str(n));

   end
   end
x = categorical(xlabs);


for n = 1:nmrws;
   d0mederr_bldmat = [d0mederr_bldmat;d0expmed_barvals(3,n),d0ctlmed_barvals(3,n)];
   d1mederr_bldmat = [d1mederr_bldmat;d1expmed_barvals(3,n),d1ctlmed_barvals(3,n)];
   d3mederr_bldmat = [d3mederr_bldmat;d3expmed_barvals(3,n),d3ctlmed_barvals(3,n)];
   d7mederr_bldmat = [d7mederr_bldmat;d7expmed_barvals(3,n),d7ctlmed_barvals(3,n)];
   d0meanerr_bldmat = [d0meanerr_bldmat;d0expmean_barvals(3,n),d0ctlmean_barvals(3,n)];
   d1meanerr_bldmat = [d1meanerr_bldmat;d1expmean_barvals(3,n),d1ctlmean_barvals(3,n)];
   d3meanerr_bldmat = [d3meanerr_bldmat;d3expmean_barvals(3,n),d3ctlmean_barvals(3,n)];
   d7meanerr_bldmat = [d7meanerr_bldmat;d7expmean_barvals(3,n),d7ctlmean_barvals(3,n)];
   expmederr_bldmat = [expmederr_bldmat;...
               d0expmed_barvals(3,n),d1expmed_barvals(3,n),...
               d3expmed_barvals(3,n),d7expmed_barvals(3,n)];
   ctlmederr_bldmat = [ctlmederr_bldmat;...
               d0ctlmed_barvals(3,n),d1ctlmed_barvals(3,n),...
               d3ctlmed_barvals(3,n),d7ctlmed_barvals(3,n)];
end
 x = categorical({'Remote ','Border ','Infarct'});
 x = reordercats(x,{'Remote ','Border ','Infarct'});
f5 = figure(5)
f5.Units = 'centimeters';
f5.Position = [2.54 2.54 fgwd fght];
t = tiledlayout(2,2,'TileSpacing','none');
title(t,'Regional Modulus','fontsize',titfsz);
nexttile
b = bar(d0med_bldmat*uniconv,'grouped')
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr(1:2,:) = [b(1).XEndPoints;b(2).XEndPoints];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
hold on
er = errorbar(xtipserr',d0med_bldmat*uniconv,d0mederr_bldmat*uniconv);%,d0mederr_bldmat*uniconv);
title('Day 0','fontsize',titfsz)
xlabel(xlb,'fontsize',labfsz)
```

```
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
legend(lgd,'fontsize',lgdfsz)
xticklabels(xtlabs);
hold off
nexttile
b = bar(d1med_bldmat*uniconv,'grouped')
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
hold on
title('Day 1','fontsize',titfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],d1med_bldmat*uniconv,d1mederr_bldmat*uniconv,d1mederr_bldmat*uniconv);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
legend(lgd,'fontsize',lgdfsz)
xticklabels(xtlabs);
hold off

nexttile
b = bar(d3med_bldmat*uniconv,'grouped')
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
hold on
title('Day 3','fontsize',titfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],d3med_bldmat*uniconv,d3mederr_bldmat*uniconv,d3mederr_bldmat*uniconv);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
legend(lgd,'fontsize',lgdfsz)
xticklabels(xtlabs);
hold off

nexttile
b = bar(d7med_bldmat*uniconv,'grouped')
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
hold on
title('Day 7','fontsize',titfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],d7med_bldmat*uniconv,d7mederr_bldmat*uniconv,d7mederr_bldmat*uniconv);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
legend(lgd,'fontsize',lgdfsz)
xticklabels(xtlabs);
hold off

reglgd = {'Day 0';'Day 1';'Day 3';'Day 7'};
f7 = figure(7)
f7.Units = 'centimeters';
f7.Position = [2.54 2.54 fgwd fght];
t = tiledlayout(1,2,'TileSpacing','none');
title(t,'Regional Modulus','fontsize',titfsz);
nexttile
b = bar(expmed_bldmat*uniconv,'grouped')
```

```
b(3).FaceColor = [0.6350 0.0780 0.1840];
b(4).FaceColor = [0.4940 0.1840 0.5560];
xtips = [b(1).XEndPoints,b(2).XEndPoints,b(3).XEndPoints,b(4).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints',b(3).XEndPoints',b(4).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints,b(3).YEndPoints,b(4).YEndPoints];
hold on
title('Wounded','fontsize',titfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],expmed_bldmat*uniconv,expmederr_bldmat*uniconv,expmederr_bldmat*uniconv);
for lp = 1:4;
er(lp).LineStyle = 'none';
end
legend(reglgd,'fontsize',lgdfsz)
xticklabels(xtlabs);
hold off

nexttile
b = bar(ctlmed_bldmat*uniconv,'grouped')
b(3).FaceColor = [0.6350 0.0780 0.1840];
b(4).FaceColor = [0.4940 0.1840 0.5560];
xtips = [b(1).XEndPoints,b(2).XEndPoints,b(3).XEndPoints,b(4).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints',b(3).XEndPoints',b(4).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
hold on
title('Control','fontsize',titfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],ctlmed_bldmat*uniconv,ctlmederr_bldmat*uniconv,ctlmederr_bldmat*uniconv);
for lp = 1:4;
er(lp).LineStyle = 'none';
end
legend(reglgd,'fontsize',lgdfsz)
xticklabels(xtlabs);
hold off


f6 = figure(6)
subplot(2,2,1)
b = bar(x,d0mean_bldmat*uniconv)
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
labs = [string(sprintf('%.4f',(round(b(1).YData(1),4))),...
    string(sprintf('%.4f',round(b(1).YData(2),4)))...
    string(sprintf('%.4f',(round(b(1).YData(3),4))),...
    string(sprintf('%.4f',round(b(2).YData(1),4)))...
    string(sprintf('%.4f',(round(b(2).YData(2),4))),...
    string(sprintf('%.4f',round(b(2).YData(3),4)))...
    ]
text(xtips,ytips,labs,'HorizontalAlignment','right','VerticalAlignment','bottom')
hold on
er = errorbar([xtipserr],d0mean_bldmat*uniconv,d0meanerr_bldmat*uniconv,d0meanerr_bldmat*uniconv);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
title('D0 Modulus MeanStr')
legend(lgd,'fontsize',lgdfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
hold off

subplot(2,2,2)
```

```
b = bar(x,d1mean_bldmat*uniconv)
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
labs = [string(sprintf('%.4f',(round(b(1).YData(1),4))),...
    string(sprintf('%.4f',round(b(1).YData(2),4)))...
    string(sprintf('%.4f',(round(b(1).YData(3),4)))),...
    string(sprintf('%.4f',round(b(2).YData(1),4)))...
    string(sprintf('%.4f',(round(b(2).YData(2),4)))),...
    string(sprintf('%.4f',round(b(2).YData(3),4)))...
    ]
text(xtips,ytips,labs,'HorizontalAlignment','right','VerticalAlignment','bottom')
hold on
title('D1 Modulus MeanStr')
legend(lgd,'fontsize',lgdfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],d1mean_bldmat*uniconv,d1meanerr_bldmat*uniconv,d1meanerr_bldmat*uniconv);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
hold off

subplot(2,2,3)
b = bar(x,d3mean_bldmat*uniconv)
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
labs = [string(sprintf('%.4f',(round(b(1).YData(1),4))),...
    string(sprintf('%.4f',round(b(1).YData(2),4)))...
    string(sprintf('%.4f',(round(b(1).YData(3),4)))),...
    string(sprintf('%.4f',round(b(2).YData(1),4)))...
    string(sprintf('%.4f',(round(b(2).YData(2),4)))),...
    string(sprintf('%.4f',round(b(2).YData(3),4)))...
    ]
text(xtips,ytips,labs,'HorizontalAlignment','right','VerticalAlignment','bottom')
hold on
title('D3 Modulus MeanStr')
legend(lgd,'fontsize',lgdfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
er = errorbar([xtipserr],d3mean_bldmat*uniconv,d3meanerr_bldmat*uniconv,d3meanerr_bldmat*uniconv);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
hold off

subplot(2,2,4)
b = bar(x,d7mean_bldmat*uniconv)
xtips = [b(1).XEndPoints,b(2).XEndPoints];
xtipserr = [b(1).XEndPoints',b(2).XEndPoints'];
ytips = [b(1).YEndPoints,b(2).YEndPoints];
labs = [string(sprintf('%.4f',(round(b(1).YData(1),4))),...
    string(sprintf('%.4f',round(b(1).YData(2),4)))...
    string(sprintf('%.4f',(round(b(1).YData(3),4)))),...
    string(sprintf('%.4f',round(b(2).YData(1),4)))...
    string(sprintf('%.4f',(round(b(2).YData(2),4)))),...
    string(sprintf('%.4f',round(b(2).YData(3),4)))...
    ]
text(xtips,ytips,labs,'HorizontalAlignment','right','VerticalAlignment','bottom')
hold on
title('D7 Modulus MeanStr')
legend(lgd,'fontsize',lgdfsz)
xlabel(xlb,'fontsize',labfsz)
ylabel(ylb,'fontsize',labfsz)
ylim([ymn ymx])
```

```
er = errorbar([xtipserr],d7mean_bldmat*uniconv,d7meanerr_bldmat*uniconv,d7meanerr_bldmat*uniconv);
er(1).LineStyle = 'none'; er(2).LineStyle = 'none';
hold off


%%Stats
expind = [ind_exp{1,1};ind_exp{1,2};ind_exp{1,3};ind_exp{1,4}]
exptab = littab(expind,:);
ht = height(exptab);
regstor = [repmat(categorical("Reg1"),ht,1);...
    repmat(categorical("Reg2"),ht,1);repmat(categorical("Reg3"),ht,1)]';
daystor = [exptab.Day(:);exptab.Day(:);exptab.Day(:)]';
datstor = [modvect_med(expind,1);modvect_med(expind,2);modvect_med(expind,3)]';
[p,t,stats] = anovan(datstor,{regstor daystor},...
    'model','full','varnames',{'Region','Day'});
figure
[c,m,h,gnames] = multcompare(stats);


ctlind = [ind_ctl{1,1};ind_ctl{1,2};ind_ctl{1,3};ind_ctl{1,4}]
ctltab = littab(ctlind,:);
ht = height(ctltab);
regstor = [repmat(categorical("Reg1"),ht,1);...
    repmat(categorical("Reg2"),ht,1);repmat(categorical("Reg3"),ht,1)]';
daystor = [ctltab.Day(:);ctltab.Day(:);ctltab.Day(:)]';
datstor = [modvect_med(ctlind,1);modvect_med(ctlind,2);modvect_med(ctlind,3)]';
[p,t,stats] = anovan(datstor,{regstor daystor},...
    'model','full','varnames',{'Region','Day'});
figure
[c,m,h,gnames] = multcompare(stats);


%
% %EXP Day0
 compars = ["1vs2";"1vs3";"2vs3"];
[exp_h1,exp_p1] = ttest(modvect_med(ind_exp{1,1},1),modvect_med(ind_exp{1,1},2));
[exp_h2,exp_p2] = ttest(modvect_med(ind_exp{1,1},1),modvect_med(ind_exp{1,1},3));
[exp_h3,exp_p3] = ttest(modvect_med(ind_exp{1,1},2),modvect_med(ind_exp{1,1},3));
%EXP Day 1
[exp_h4,exp_p4] = ttest(modvect_med(ind_exp{1,2},1),modvect_med(ind_exp{1,2},2));
[exp_h5,exp_p5] = ttest(modvect_med(ind_exp{1,2},1),modvect_med(ind_exp{1,2},3));
[exp_h6,exp_p6] = ttest(modvect_med(ind_exp{1,2},2),modvect_med(ind_exp{1,2},3));
%EXP Day 3
[exp_h7,exp_p7] = ttest(modvect_med(ind_exp{1,3},1),modvect_med(ind_exp{1,3},2));
[exp_h8,exp_p8] = ttest(modvect_med(ind_exp{1,3},1),modvect_med(ind_exp{1,3},3));
[exp_h9,exp_p9] = ttest(modvect_med(ind_exp{1,3},2),modvect_med(ind_exp{1,3},3));
%EXP Day 7
[exp_h10,exp_p10] = ttest(modvect_med(ind_exp{1,4},1),modvect_med(ind_exp{1,4},2));
[exp_h11,exp_p11] = ttest(modvect_med(ind_exp{1,4},1),modvect_med(ind_exp{1,4},3));
[exp_h12,exp_p12] = ttest(modvect_med(ind_exp{1,4},2),modvect_med(ind_exp{1,4},3));

Exp = [exp_p1,exp_p2,exp_p3;exp_p4,exp_p5,exp_p6;exp_p7,exp_p8,exp_p9;exp_p10,exp_p11,exp_p12];

%CTL Day 0
[ctl_h1,ctl_p1] = ttest(modvect_med(ind_ctl{1,1},1),modvect_med(ind_ctl{1,1},2));
[ctl_h2,ctl_p2] = ttest(modvect_med(ind_ctl{1,1},1),modvect_med(ind_ctl{1,1},3));
[ctl_h3,ctl_p3] = ttest(modvect_med(ind_ctl{1,1},2),modvect_med(ind_ctl{1,1},3));
%CTL Day 1
[ctl_h4,ctl_p4] = ttest(modvect_med(ind_ctl{1,2},1),modvect_med(ind_ctl{1,2},2));
[ctl_h5,ctl_p5] = ttest(modvect_med(ind_ctl{1,2},1),modvect_med(ind_ctl{1,2},3));
[ctl_h6,ctl_p6] = ttest(modvect_med(ind_ctl{1,2},2),modvect_med(ind_ctl{1,2},3));
%CTL Day 3
[ctl_h7,ctl_p7] = ttest(modvect_med(ind_ctl{1,3},1),modvect_med(ind_ctl{1,3},2));
[ctl_h8,ctl_p8] = ttest(modvect_med(ind_ctl{1,3},1),modvect_med(ind_ctl{1,3},3));
[ctl_h9,ctl_p9] = ttest(modvect_med(ind_ctl{1,3},2),modvect_med(ind_ctl{1,3},3));
%CTL Day 7
```

```
[ctl_h10,ctl_p10] = ttest(modvect_med(ind_ctl{1,4},1),modvect_med(ind_ctl{1,4},2));
[ctl_h11,ctl_p11] = ttest(modvect_med(ind_ctl{1,4},1),modvect_med(ind_ctl{1,4},3));
[ctl_h12,ctl_p12] = ttest(modvect_med(ind_ctl{1,4},2),modvect_med(ind_ctl{1,4},3));

Ctl = [ctl_p1,ctl_p2,ctl_p3;ctl_p4,ctl_p5,ctl_p6;ctl_p7,ctl_p8,ctl_p9;ctl_p10,ctl_p11,ctl_p12];

%% TTESTS for same region at different points of timecourse
%EXP
%REG1
regvars = ["0vs1";"0vs3";"0vs7";"1vs3";"1vs7";"3vs7"];
[expreg1_h1,expreg1_p1] = ttest(modvect_med(ind_exp{1,1},1),modvect_med(ind_exp{1,2},1));
[expreg1_h2,expreg1_p2] = ttest(modvect_med(ind_exp{1,1},1),modvect_med(ind_exp{1,3},1));
[expreg1_h3,expreg1_p3] = ttest(modvect_med(ind_exp{1,1},1),modvect_med(ind_exp{1,4},1));
[expreg1_h4,expreg1_p4] = ttest(modvect_med(ind_exp{1,2},1),modvect_med(ind_exp{1,3},1));
[expreg1_h5,expreg1_p5] = ttest(modvect_med(ind_exp{1,2},1),modvect_med(ind_exp{1,4},1));
[expreg1_h6,expreg1_p6] = ttest(modvect_med(ind_exp{1,3},1),modvect_med(ind_exp{1,4},1));

[expreg2_h1,expreg2_p1] = ttest(modvect_med(ind_exp{1,1},2),modvect_med(ind_exp{1,2},2));
[expreg2_h2,expreg2_p2] = ttest(modvect_med(ind_exp{1,1},2),modvect_med(ind_exp{1,3},2));
[expreg2_h3,expreg2_p3] = ttest(modvect_med(ind_exp{1,1},2),modvect_med(ind_exp{1,4},2));
[expreg2_h4,expreg2_p4] = ttest(modvect_med(ind_exp{1,2},2),modvect_med(ind_exp{1,3},2));
[expreg2_h5,expreg2_p5] = ttest(modvect_med(ind_exp{1,2},2),modvect_med(ind_exp{1,4},2));
[expreg2_h6,expreg2_p6] = ttest(modvect_med(ind_exp{1,3},2),modvect_med(ind_exp{1,4},2));

[expreg3_h1,expreg3_p1] = ttest(modvect_med(ind_exp{1,1},3),modvect_med(ind_exp{1,2},3));
[expreg3_h2,expreg3_p2] = ttest(modvect_med(ind_exp{1,1},3),modvect_med(ind_exp{1,3},3));
[expreg3_h3,expreg3_p3] = ttest(modvect_med(ind_exp{1,1},3),modvect_med(ind_exp{1,4},3));
[expreg3_h4,expreg3_p4] = ttest(modvect_med(ind_exp{1,2},3),modvect_med(ind_exp{1,3},3));
[expreg3_h5,expreg3_p5] = ttest(modvect_med(ind_exp{1,2},3),modvect_med(ind_exp{1,4},3));
[expreg3_h6,expreg3_p6] = ttest(modvect_med(ind_exp{1,3},3),modvect_med(ind_exp{1,4},3));

EXPDayStats = [expreg1_p1,expreg1_p2,expreg1_p3,expreg1_p4,expreg1_p5,expreg1_p6;...
               expreg2_p1,expreg2_p2,expreg2_p3,expreg2_p4,expreg2_p5,expreg2_p6;...
               expreg3_p1,expreg3_p2,expreg3_p3,expreg3_p4,expreg3_p5,expreg3_p6];


[ctlreg1_h1,ctlreg1_p1] = ttest(modvect_med(ind_ctl{1,1},1),modvect_med(ind_ctl{1,2},1));
[ctlreg1_h2,ctlreg1_p2] = ttest(modvect_med(ind_ctl{1,1},1),modvect_med(ind_ctl{1,3},1));
[ctlreg1_h3,ctlreg1_p3] = ttest(modvect_med(ind_ctl{1,1},1),modvect_med(ind_ctl{1,4},1));
[ctlreg1_h4,ctlreg1_p4] = ttest(modvect_med(ind_ctl{1,2},1),modvect_med(ind_ctl{1,3},1));
[ctlreg1_h5,ctlreg1_p5] = ttest(modvect_med(ind_ctl{1,2},1),modvect_med(ind_ctl{1,4},1));
[ctlreg1_h6,ctlreg1_p6] = ttest(modvect_med(ind_ctl{1,3},1),modvect_med(ind_ctl{1,4},1));

[ctlreg2_h1,ctlreg2_p1] = ttest(modvect_med(ind_ctl{1,1},2),modvect_med(ind_ctl{1,2},2));
[ctlreg2_h2,ctlreg2_p2] = ttest(modvect_med(ind_ctl{1,1},2),modvect_med(ind_ctl{1,3},2));
[ctlreg2_h3,ctlreg2_p3] = ttest(modvect_med(ind_ctl{1,1},2),modvect_med(ind_ctl{1,4},2));
[ctlreg2_h4,ctlreg2_p4] = ttest(modvect_med(ind_ctl{1,2},2),modvect_med(ind_ctl{1,3},2));
[ctlreg2_h5,ctlreg2_p5] = ttest(modvect_med(ind_ctl{1,2},2),modvect_med(ind_ctl{1,4},2));
[ctlreg2_h6,ctlreg2_p6] = ttest(modvect_med(ind_ctl{1,3},2),modvect_med(ind_ctl{1,4},2));

[ctlreg3_h1,ctlreg3_p1] = ttest(modvect_med(ind_ctl{1,1},3),modvect_med(ind_ctl{1,2},3));
[ctlreg3_h2,ctlreg3_p2] = ttest(modvect_med(ind_ctl{1,1},3),modvect_med(ind_ctl{1,3},3));
[ctlreg3_h3,ctlreg3_p3] = ttest(modvect_med(ind_ctl{1,1},3),modvect_med(ind_ctl{1,4},3));
[ctlreg3_h4,ctlreg3_p4] = ttest(modvect_med(ind_ctl{1,2},3),modvect_med(ind_ctl{1,3},3));
[ctlreg3_h5,ctlreg3_p5] = ttest(modvect_med(ind_ctl{1,2},3),modvect_med(ind_ctl{1,4},3));
[ctlreg3_h6,ctlreg3_p6] = ttest(modvect_med(ind_ctl{1,3},3),modvect_med(ind_ctl{1,4},3));

CTLDayStats = [ctlreg1_p1,ctlreg1_p2,ctlreg1_p3,ctlreg1_p4,ctlreg1_p5,ctlreg1_p6;...
               ctlreg2_p1,ctlreg2_p2,ctlreg2_p3,ctlreg2_p4,ctlreg2_p5,ctlreg2_p6;...
               ctlreg3_p1,ctlreg3_p2,ctlreg3_p3,ctlreg3_p4,ctlreg3_p5,ctlreg3_p6];

%% TTESTS for EXP vs CTL on same day
ovcom = ["D0-REM","D0-BRD","D0-INF";...
      "D1-REM","D1-BRD","D1-INF";...
      "D3-REM","D3-BRD","D3-INF";...
      "D7-REM","D7-BRD","D7-INF"];
```

```
%D0
[cmpd0_h1,cmpd0_p1] = ttest2(modvect_med(ind_exp{1,1},1),modvect_med(ind_ctl{1,1},1));
[cmpd0_h2,cmpd0_p2] = ttest2(modvect_med(ind_exp{1,1},2),modvect_med(ind_ctl{1,1},2));
[cmpd0_h3,cmpd0_p3] = ttest2(modvect_med(ind_exp{1,1},3),modvect_med(ind_ctl{1,1},3));

%D1
[cmpd1_h1,cmpd1_p1] = ttest2(modvect_med(ind_exp{1,2},1),modvect_med(ind_ctl{1,2},1));
[cmpd1_h2,cmpd1_p2] = ttest2(modvect_med(ind_exp{1,2},2),modvect_med(ind_ctl{1,2},2));
[cmpd1_h3,cmpd1_p3] = ttest2(modvect_med(ind_exp{1,2},3),modvect_med(ind_ctl{1,2},3));

%D3
[cmpd3_h1,cmpd3_p1] = ttest2(modvect_med(ind_exp{1,3},1),modvect_med(ind_ctl{1,3},1));
[cmpd3_h2,cmpd3_p2] = ttest2(modvect_med(ind_exp{1,3},2),modvect_med(ind_ctl{1,3},2));
[cmpd3_h3,cmpd3_p3] = ttest2(modvect_med(ind_exp{1,3},3),modvect_med(ind_ctl{1,3},3));

%D7
[cmpd7_h1,cmpd7_p1] = ttest2(modvect_med(ind_exp{1,4},1),modvect_med(ind_ctl{1,4},1));
[cmpd7_h2,cmpd7_p2] = ttest2(modvect_med(ind_exp{1,4},2),modvect_med(ind_ctl{1,4},2));
[cmpd7_h3,cmpd7_p3] = ttest2(modvect_med(ind_exp{1,4},3),modvect_med(ind_ctl{1,4},3));

CMPStats = [cmpd0_p1,cmpd0_p2,cmpd0_p3;...
        cmpd1_p1,cmpd1_p2,cmpd1_p3;...
        cmpd3_p1,cmpd3_p2,cmpd3_p3;...
        cmpd7_p1,cmpd7_p2,cmpd7_p3];
```

Appendix C

Aim 3 MATLAB Scripting

C-1.1 Background Subtraction and Sorting of Images and Supplemental Functions

```
%% Main Background Reducer Script
clear all
close all

%Home Directory
homdirnam = redacted;

%Processed Positive Sample Directory
prcsddir = redacted;
prcsddir_sig = redacted;
prcsddir_pct = redacted;
%Assign Base Text for Pos-Neg Pair
%Sample Base Names Lookup
lkup_bs_nams = ['MI22TRP_EXP';...
            'MI23TRP_EXP';...
            'MI24TRP_EXP';...
            'SHAMTRP_EXP'];
namopts = 4;

%Assign if it is the A or B grouping
%AB Lookup
lkup_ab = ['A';'B'];
letops = 2;

%Assign Directory of Sample Folders
sampdirnam = redacted;

%Begin Loop for rotating through possible combinations
for n0 = 1:namopts;
   bs_txt = lkup_bs_nams(n0,:);
   for n1 = 1:letops;
      ab = lkup_ab(n1,:);
      %Assign Negative and Positive Sample directories of Group
      negdirnam = horzcat(sampdirnam,bs_txt,'_NEG_',ab);
      posdirnam = horzcat(sampdirnam,bs_txt,'_POS_',ab);

      %% Read folders of Positive Directory to determine how many sections there
      %are
      posdir_ind = dir(posdirnam); %Index directory of positive sample
      h = height(posdir_ind); %Find number of items in directory
      %Loop through to find folders of actual samples and record
      samplist = [];
      c = 1;
      for n = 1:h;
         if length(posdir_ind(n).name) == 4;

            if all(posdir_ind(n).name(1:2) == 'XY');
               samplist(c,:) = char(posdir_ind(n).name);
               c = c + 1;
            else
               continue
            end
         end
      end

      %Evaluate Number of Samps
      sh = size(samplist,1);

      %Loop through sample pairings to calculate background reduction and apply
```

```
%to positive pair
for ns = 1:sh;
   sampfoldneg = horzcat(negdirnam,'\',samplist(ns,:),'\',bs_txt,'_NEG_',ab,'_',samplist(ns,:));
   sampfoldpos = horzcat(posdirnam,'\',samplist(ns,:),'\',bs_txt,'_POS_',ab,'_',samplist(ns,:));
   %Trying subtracting from positive
   ch1_nam = horzcat(sampfoldpos,'_IGB1_bgtf.tif');
   ch2_nam = horzcat(sampfoldpos,'_NCAD_bgtf.tif');
   ch3_nam = horzcat(sampfoldpos,'_DAPI_bgtf.tif');
   ovly_nam = horzcat(sampfoldpos,'_OVLY_bgtf.tif');

   ch1_im = imread(ch1_nam);
   ch2_im = imread(ch2_nam);
   ch3_im = imread(ch3_nam);
   ovly_im = imread(ovly_nam);

   %% Convert 0 values to NaN to exclude from analysis; if needed go back and pull only nonzero vals
   ch1_nzelems= find(ch1_im);
   ch1_usemat = NaN(size(ch1_im));
   ch1_usemat(ch1_nzelems) = ch1_im(ch1_nzelems);

   ch2_nzelems= find(ch2_im);
   ch2_usemat = NaN(size(ch2_im));
   ch2_usemat(ch2_nzelems) = ch2_im(ch2_nzelems);

   ch3_nzelems= find(ch3_im);
   ch3_usemat = NaN(size(ch3_im));
   ch3_usemat(ch3_nzelems) = ch3_im(ch3_nzelems);

  ovly_nzelems= find(ovly_im);
  ovly_usemat = NaN(size(ovly_im));
  ovly_usemat(ovly_nzelems) = ovly_im(ovly_nzelems);


%% Decision in pipeline
%Either find distribution by image or add nonzero values of image to
%matrix and add successive image nonzero values to matrix then create
%ultimate distribution at the end

%First step is add all to single matrix and calculate distribution


   ch1_stormat = ch1_usemat;
   ch2_stormat = ch2_usemat;
   ch3_stormat = ch3_usemat;
   ovly_stormat = ovly_usemat;

%% Find value that is xxth percentile to set bar for background
%Starting at 2 std deviations; 97.6%ile
sig = 2;
%Ch1
ch1_vals = bckgr_distr(ch1_stormat,sig);

%Ch2
ch2_vals = bckgr_distr(ch2_stormat,sig);

%Ch3
ch3_vals = bckgr_distr(ch3_stormat,sig);

%Ovly
ovly_vals = bckgr_distr(ovly_stormat,sig);

%Value stack
valstck = cat(3,ch1_vals,ch2_vals,ch3_vals,ovly_vals);
bckvals_recnam = horzcat(prcsddir,bs_txt,ab,samplist(sh,:),'_bckgrdval.mat');

%       foldname = ch1_dir(1).folder;
```

```
        save(bckvals_recnam,'valstck');

        %Read in Positive Channels
        ch1_nam = horzcat(sampfoldpos,'_IGB1_bgtf.tif');
        ch1_savnam = horzcat(bs_txt,'_POS_',ab,'_',samplist(ns,:),'_IGB1_bgtf_prcsd.tif');
        ch2_nam = horzcat(sampfoldpos,'_NCAD_bgtf.tif');
        ch2_savnam = horzcat(bs_txt,'_POS_',ab,'_',samplist(ns,:),'_NCAD_bgtf_prcsd.tif');
        ch3_nam = horzcat(sampfoldpos,'_DAPI_bgtf.tif');
        ch3_savnam = horzcat(bs_txt,'_POS_',ab,'_',samplist(ns,:),'_DAPI_bgtf_prcsd.tif');
        ovly_nam = horzcat(sampfoldpos,'_OVLY_bgtf.tif');
        ovly_savnam = horzcat(bs_txt,'_POS_',ab,'_',samplist(ns,:),'_OVLY_bgtf_prcsd.tif');


        ch1_im = imread(ch1_nam);
        ch2_im = imread(ch2_nam);
        ch3_im = imread(ch3_nam);
        ovly_im = imread(ovly_nam);


        %% Subtract determined value from existing signal in positive samples
          %If result of subtraction is negative, set to zero.
        ch1_im_sig = bckgrd_sub(ch1_im,ch1_vals,1);
        ch2_im_sig = bckgrd_sub(ch2_im,ch2_vals,1);
        ch3_im_sig = bckgrd_sub(ch3_im,ch3_vals,1);
        ovly_im_sig = bckgrd_sub(ovly_im,ovly_vals,1);

        ch1_im_pct = bckgrd_sub(ch1_im,ch1_vals,2);
        ch2_im_pct = bckgrd_sub(ch2_im,ch2_vals,2);
        ch3_im_pct = bckgrd_sub(ch3_im,ch3_vals,2);
        ovly_im_pct = bckgrd_sub(ovly_im,ovly_vals,2);

        %% Save processed image to new folder
        %Consideration for saving as same name or update name to 'prcsd_...'
        %Unsure how stitch software will handle name adjustments

        imwrite(ch1_im_sig,horzcat(prcsddir_sig,ch1_savnam));
        imwrite(ch2_im_sig,horzcat(prcsddir_sig,ch2_savnam));
        imwrite(ch3_im_sig,horzcat(prcsddir_sig,ch3_savnam));
        imwrite(ovly_im_sig,horzcat(prcsddir_sig,ovly_savnam));

        imwrite(ch1_im_pct,horzcat(prcsddir_pct,ch1_savnam));
        imwrite(ch2_im_pct,horzcat(prcsddir_pct,ch2_savnam));
        imwrite(ch3_im_pct,horzcat(prcsddir_pct,ch3_savnam));
        imwrite(ovly_im_pct,horzcat(prcsddir_pct,ovly_savnam));
        end
    end
end


function chvals = bckgr_distr(ch_stormat,s)
%Find upper 2sigma value of normal distribution from dataset
%   ch_stormat is rgb image matrix for negative control
pctl = 97.5;
try
    ch_1 = ch_stormat(:,:,1);
    ch_1vect = ch_1(:);
%     ch_1distr = fitdist(ch_1vect,'normal');
    ch_1val = [mean(ch_1vect,'omitnan')+s*std(ch_1vect,0,'all','omitnan');prctile(ch_1vect,pctl)];
catch
    ch_1val = [0;0];
end

try
    ch_2 = ch_stormat(:,:,2);
    ch_2vect = ch_2(:);
%     ch_2distr = fitdist(ch_2vect,'normal');
```

```
    ch_2val = [mean(ch_2vect,'omitnan')+s*std(ch_2vect,0,'all','omitnan');prctile(ch_2vect,pctl)];
catch
    ch_2val = [0;0];
end

try
    ch_3 = ch_stormat(:,:,3);
    ch_3vect = ch_3(:);
%     ch_3distr = fitdist(ch_3vect,'normal');
    ch_3val = [mean(ch_3vect,'omitnan')+s*std(ch_3vect,0,'all','omitnan');prctile(ch_3vect,pctl)];
catch
    ch_3val = [0;0];
end

    chvals = [ch_1val,ch_2val,ch_3val];
end

function im_mat = bckgrd_sub(im_mat,bck_val,v)
%subtracts appropriate value from channel
%   Detailed explanation goes here
%     im_mat(:,:,1) = im_mat(:,:,1)-bck_val(1,1);
%     im_mat(:,:,2) = im_mat(:,:,2)-bck_val(1,2);
%     im_mat(:,:,3) = im_mat(:,:,3)-bck_val(1,3);
%   Select if using sig val or pct val
%Sig val: v = 1     Pct Val: v = 2;

    im_mat1 = im_mat(:,:,1);
    im_mat1(find(im_mat1<bck_val(v,1))) = 0;
    im_mat2 = im_mat(:,:,2);
    im_mat2(find(im_mat2<bck_val(v,2))) = 0;
    im_mat3 = im_mat(:,:,3);
    im_mat3(find(im_mat3<bck_val(v,3))) = 0;

%     im_mat(:,:,1) = im_mat(:,:,1) - bck_val(1,1);
%     im_mat(:,:,2) = im_mat(:,:,2) - bck_val(1,2);
%     im_mat(:,:,3) = im_mat(:,:,3) - bck_val(1,3);

%     im_mat(find(im_mat<bck_val)) = 0;
    im_mat = cat(3,im_mat1,im_mat2,im_mat3);
end



%%Script for Analysis via pulling mask data from lookup table and applying
%%to image
close all
clear all
load('aim3masktab.mat');
load('aim3vals.mat','MI_mean','MI_std','ppvals_MI','ppvals_SH','Sham_mean','Sham_std');
%Generate New Table for Storing Calculations
calctab = aim3masktab;
%% Assign Directories
prcsd_dir = ...
redacted';
savdir = redacted';
imdir = redacted';


%% Begin Loop
h = height(aim3masktab);

for n = 1:h;
    %% Pull Sample information from mask table
    samp = char(aim3masktab.Sample(n));
    posneg = char(aim3masktab.PosNeg(n));
    ab = char(aim3masktab.AB(n));
```

```
    sampnam = horzcat(samp,'TRP_EXP_',posneg,'_',ab);
    sect = char(aim3masktab.XY(n));
    prcsd_samp_igb1 = horzcat(sampnam,'_',sect,'_IGB1_bgtf_prcsd.tif');
    prcsd_samp_ncad = horzcat(sampnam,'_',sect,'_NCAD_bgtf_prcsd.tif');
    prcsd_samp_dapi = horzcat(sampnam,'_',sect,'_DAPI_bgtf_prcsd.tif');
%Maybe don't need
%     txt = horzcat(imdir,sampnam,'\',sect,'\',sampnam,'_',sect,'_OVLY_bgtf.tif');
%     txt_sc = horzcat(imdir,sampnam,'\',sect,'\',sampnam,'_',sect,'_OVLY_48tf_scale.tif');
    %% Read in Images
    im_igb1 = imread(horzcat(prcsd_dir,prcsd_samp_igb1));
    im_ncad = imread(horzcat(prcsd_dir,prcsd_samp_ncad));
    im_dapi = imread(horzcat(prcsd_dir,prcsd_samp_dapi));
    %% Load Mask and apply to orignal image
    mask = aim3masktab.Mask{n};
    reg_igb1 = im_igb1(:,:,1);
    reg_igb1(find(~mask)) = 0;
    %reg_igb1 = cat(3,reg_igb1,im_igb1(:,:,2),im_igb1(:,:,3));
    reg_ncad = im_ncad(:,:,2);
    reg_ncad(find(~mask)) = 0;
    %reg_ncad = cat(3,imncad(:,:,1),reg_ncad,imncad(:,:,3));
    reg_dapi = im_dapi(:,:,3);
    reg_dapi(find(~mask)) = 0;
    %reg_dapi = cat(3,imdapi(:,:,1),im_dapi(:,:,2),reg_dapi);

    %Calculate Area of Segment in pixel counts
    area_sect = length(find(mask==1));
    calctab.Area(n) = area_sect;

    %Find number of pixels overthreshhold for channel
    igb1_thrshcnt = length(find(reg_igb1~=0));
    ncad_thrshcnt = length(find(reg_ncad~=0));
    dapi_thrshcnt = length(find(reg_dapi~=0));
    calctab.ThreshCnt{n} = [igb1_thrshcnt;ncad_thrshcnt;dapi_thrshcnt];

    %Calculate Intensity Average when normalized 0 to 1
    igb1_intavg = mean(rescale(reg_igb1(find(reg_igb1~=0))));
    ncad_intavg = mean(rescale(reg_ncad(find(reg_ncad~=0))));
    dapi_intavg = mean(rescale(reg_dapi(find(reg_dapi~=0))));
    calctab.IntenseAvg{n} = [igb1_intavg;ncad_intavg;dapi_intavg];

    %Calculate number of pixels over threshhold per area
    %Pixel counts normalized to area
    igb1_thrshcntavg = igb1_thrshcnt/area_sect;
    ncad_thrshcntavg = ncad_thrshcnt/area_sect;
    dapi_thrshcntavg = dapi_thrshcnt/area_sect;
    calctab.pxthreshfract{n} =...
       [igb1_thrshcntavg;ncad_thrshcntavg;dapi_thrshcntavg];
end
    save('aim3vals221113.mat','-mat','calctab','MI_mean','MI_std','ppvals_MI','ppvals_SH','Sham_mean','Sham_std');
```

Appendix C-2 Calculations, Statistics, and Plotting of Aim 3 Values

```
%Loop through sections and average available data
clear all
load('aim3vals221113.mat')
for n = 1:17;
   %Add section tag to first row of matrix

   %Find entries matching number
   ind_mi = find(calctab.RegCode(:) == categorical(n) & calctab.MIType(:) == 'MI');
   ind_sh = find(calctab.RegCode(:) == categorical(n) & calctab.MIType(:) == 'NA');
   %Average values
   if n == 1;
      %MI Samps
      thrshavgint_igb1_mi = [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,1),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,1),0,'all','omitnan')];
      thrshavgint_ncad_mi = [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,2),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,2),0,'all','omitnan')];
      thrshavgint_dapi_mi = [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,3),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,3),0,'all','omitnan')];
      %Sham Samps
      thrshavgint_igb1_sh = [n;...
         mean(numxtract(calctab.IntenseAvg,ind_sh,1),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_sh,1),0,'all','omitnan')];
      thrshavgint_ncad_sh = [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,2),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,2),0,'all','omitnan')];
      thrshavgint_dapi_sh = [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,3),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,3),0,'all','omitnan')];
   else
      %MI Samps
      thrshavgint_igb1_mi = [thrshavgint_igb1_mi,...
         [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,1),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,1),0,'all','omitnan')]];
      thrshavgint_ncad_mi = [thrshavgint_ncad_mi,...
         [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,2),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,2),0,'all','omitnan')]];
      thrshavgint_dapi_mi = [thrshavgint_dapi_mi,...
         [n;...
         mean(numxtract(calctab.IntenseAvg,ind_mi,3),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_mi,3),0,'all','omitnan')]];
      %Sham Samps
      thrshavgint_igb1_sh = [thrshavgint_igb1_sh,...
         [n;...
         mean(numxtract(calctab.IntenseAvg,ind_sh,1),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_sh,1),0,'all','omitnan')]];
      thrshavgint_ncad_sh = [thrshavgint_ncad_sh,...
         [n;...
         mean(numxtract(calctab.IntenseAvg,ind_sh,2),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_sh,2),0,'all','omitnan')]];
      thrshavgint_dapi_sh = [thrshavgint_dapi_sh,...
         [n;...
         mean(numxtract(calctab.IntenseAvg,ind_sh,3),'all','omitnan');...
         std(numxtract(calctab.IntenseAvg,ind_sh,3),0,'all','omitnan')]];
   end
end


%Store mean and std values from collab paper
```

```
ppvals_mi = [MI_mean;MI_std];
ppvals_sh = [Sham_mean;Sham_std];

%Find length of collaborator vectors
len_mi = length(MI_mean(1,:));
len_sh = length(Sham_mean(1,:));

%Find non-Nan indices of calculated data. NaNs correspond to regions which
%we had no data for
ind_igb1_mi = find(~isnan(thrshavgint_igb1_mi(2,:)));
ind_ncad_mi = find(~isnan(thrshavgint_ncad_mi(2,:)));
ind_dapi_mi = find(~isnan(thrshavgint_dapi_mi(2,:)));
ind_igb1_sh = find(~isnan(thrshavgint_igb1_sh(2,:)));
ind_ncad_sh = find(~isnan(thrshavgint_ncad_sh(2,:)));
ind_dapi_sh = find(~isnan(thrshavgint_dapi_sh(2,:)));


scatvect_igb1_mi_all = thrshavgint_igb1_mi(2:3,ind_igb1_mi);
scatvect_ncad_mi_all = thrshavgint_ncad_mi(2:3,ind_ncad_mi);
scatvect_dapi_mi_all = thrshavgint_dapi_mi(2:3,ind_dapi_mi);
scatvect_igb1_sh_all = thrshavgint_igb1_sh(2:3,ind_igb1_sh);
scatvect_ncad_sh_all = thrshavgint_ncad_sh(2:3,ind_ncad_sh);
scatvect_dapi_sh_all = thrshavgint_dapi_sh(2:3,ind_dapi_sh);



% save('aim3vals.mat','-mat','calctab','MI_mean','MI_std','Sham_mean','Sham_std','ppvals_MI','ppvals_SH');

%% Scatter Plots for All values
close all
titfsz = 16;
lgdfsz = 10;
labfsz = 14;
%MI Plot of intensity vs strain w/ IGB1 and NCAD present
figure(1)
hold on
ivs_igb1_mi_all = [ppvals_mi(1,1:15)',scatvect_igb1_mi_all(1,1:15)'];
mdl_igb1_mi_all = fitlm(ivs_igb1_mi_all(:,1),ivs_igb1_mi_all(:,2));
txt_igb1_mi_all = char(horzcat(sprintf('IGB1 \n R^{2} = '),...
    sprintf('%.2f',mdl_igb1_mi_all.Rsquared.Ordinary)));
igb1_mi_r2 = mdl_igb1_mi_all.Rsquared.Ordinary;
[igb1_mi_pears_rho,igb1_mi_pears] = corr(ivs_igb1_mi_all(:,1),ivs_igb1_mi_all(:,2),'Type','Pearson');
[rho,igb1_mi_spear] = corr(ivs_igb1_mi_all(:,1),ivs_igb1_mi_all(:,2),'Type','Spearman');
ivs_ncad_mi_all = [ppvals_mi(1,1:15)',scatvect_ncad_mi_all(1,1:15)'];
mdl_ncad_mi_all = fitlm(ivs_ncad_mi_all(:,1),ivs_ncad_mi_all(:,2));
ncad_mi_r2 = mdl_ncad_mi_all.Rsquared.Ordinary;
[ncad_mi_pears_rho,ncad_mi_pears] = corr(ivs_ncad_mi_all(:,1),ivs_ncad_mi_all(:,2),'Type','Pearson');
[rho,ncad_mi_spear] = corr(ivs_ncad_mi_all(:,1),ivs_ncad_mi_all(:,2),'Type','Spearman');
txt_ncad_mi_all = char(horzcat(sprintf('NCAD \n R^{2} = '),...
    sprintf('%.2f',mdl_ncad_mi_all.Rsquared.Ordinary)));
ivs_dapi_mi_all = [ppvals_mi(1,1:15)',scatvect_dapi_mi_all(1,1:15)'];
mdl_dapi_mi_all = fitlm(ivs_dapi_mi_all(:,1),ivs_dapi_mi_all(:,2));
dapi_mi_r2 = mdl_dapi_mi_all.Rsquared.Ordinary;
[dapi_mi_pears_rho,dapi_mi_pears] = corr(ivs_dapi_mi_all(:,1),ivs_dapi_mi_all(:,2),'Type','Pearson');
[rho,dapi_mi_spear] = corr(ivs_dapi_mi_all(:,1),ivs_dapi_mi_all(:,2),'Type','Spearman');
txt_dapi_mi_all = char(horzcat(sprintf('DAPI \nR^{2} = '),...
    sprintf('%.2f',mdl_dapi_mi_all.Rsquared.Ordinary)));

scatter(ivs_igb1_mi_all(:,1),ivs_igb1_mi_all(:,2),...
      'd','MarkerFaceColor','r','MarkerEdgeColor','r','LineWidth',2.5);
scatter(ivs_ncad_mi_all(:,1),ivs_ncad_mi_all(:,2),...
      'o','MarkerFaceColor','b','MarkerEdgeColor','b','LineWidth',2.5);
% scatter(ivs_dapi_mi_all(:,1),ivs_dapi_mi_all(:,2),...
%       's','MarkerFaceColor','b','MarkerEdgeColor','k');
p1 = plot(mdl_igb1_mi_all);
```

```
%text(mean(ivs_igb1_mi_all(:,1)),max(ivs_igb1_mi_all(:,2))*.95,txt_igb1_mi_all);
p2 = plot(mdl_ncad_mi_all);
%text(mean(ivs_ncad_mi_all(:,1)),max(ivs_ncad_mi_all(:,2))*.95,txt_ncad_mi_all);
% plot(mdl_dapi_mi_all);
% %text(mean(ivs_dapi_mi_all(:,1)),max(ivs_dapi_mi_all(:,2))*.95,txt_dapi_mi_all);
[p1,p2] = scatmdl_paramadjust(p1,p2);
hold off
title(sprintf('MI \n Intensity vs Strain'),'fontsize',titfsz);
xlabel('Strain','fontsize',labfsz)
ylabel('Intensity','fontsize',labfsz)
legend('IGB1','NCAD','fontsize',lgdfsz)%,'DAPI')
xlim([0.1,0.4])
%SHAM Plot of intensity vs strain w/ IGB1 and NCAD present
figure(2)
hold on
ivs_igb1_sh_all = [ppvals_sh(1,7:17)',scatvect_igb1_sh_all(1,1:11)'];
mdl_igb1_sh_all = fitlm(ivs_igb1_sh_all(:,1),ivs_igb1_sh_all(:,2));
igb1_sh_r2 = mdl_igb1_sh_all.Rsquared.Ordinary;
[igb1_sh_pears_rho,igb1_sh_pears] = corr(ivs_igb1_sh_all(:,1),ivs_igb1_sh_all(:,2),'Type','Pearson');
[rho,igb1_sh_spear] = corr(ivs_igb1_sh_all(:,1),ivs_igb1_sh_all(:,2),'Type','Spearman');
txt_igb1_sh_all = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl_igb1_sh_all.Rsquared.Ordinary)));
ivs_ncad_sh_all = [ppvals_sh(1,[1,7:17])',scatvect_ncad_sh_all(1,1:12)'];
mdl_ncad_sh_all = fitlm(ivs_ncad_sh_all(:,1),ivs_ncad_sh_all(:,2));
ncad_sh_r2 = mdl_ncad_sh_all.Rsquared.Ordinary;
[ncad_sh_pears_rho,ncad_sh_pears] = corr(ivs_ncad_sh_all(:,1),ivs_ncad_sh_all(:,2),'Type','Pearson');
[rho,ncad_sh_spear] = corr(ivs_ncad_sh_all(:,1),ivs_ncad_sh_all(:,2),'Type','Spearman');
txt_ncad_sh_all = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl_ncad_sh_all.Rsquared.Ordinary)));
% ivs_dapi_sh_all = [ppvals_sh(1,[1,7:17])',scatvect_dapi_sh_all(1,1:12)'];
% mdl_dapi_sh_all = fitlm(ivs_dapi_sh_all(:,1),ivs_dapi_sh_all(:,2));
% dapi_sh_r2 = mdl_dapi_sh_all.Rsquared.Ordinary;
% [dapi_sh_pears_rho,dapi_sh_pears] = corr(ivs_dapi_sh_all(:,1),ivs_dapi_sh_all(:,2),'Type','Pearson');
% [rho,dapi_sh_spear] = corr(ivs_dapi_sh_all(:,1),ivs_dapi_sh_all(:,2),'Type','Spearman');
% txt_dapi_sh_all = char(horzcat(sprintf('R^{2} = '),...
%     sprintf('%.2f',mdl_dapi_sh_all.Rsquared.Ordinary)));

scatter(ivs_igb1_sh_all(:,1),ivs_igb1_sh_all(:,2),...
      'd','MarkerFaceColor','r','MarkerEdgeColor','r','LineWidth',2.5);
scatter(ivs_ncad_sh_all(:,1),ivs_ncad_sh_all(:,2),...
      'o','MarkerFaceColor','b','MarkerEdgeColor','b','LineWidth',2.5);
% scatter(ivs_dapi_sh_all(:,1),ivs_dapi_sh_all(:,2),...
%       's','MarkerFaceColor','b','MarkerEdgeColor','k');

p1 = plot(mdl_igb1_sh_all);
%text(mean(ivs_igb1_sh_all(:,1)),max(ivs_igb1_sh_all(:,2))*.95,txt_igb1_sh_all);
p2 = plot(mdl_ncad_sh_all);
%text(mean(ivs_ncad_sh_all(:,1)),max(ivs_ncad_sh_all(:,2))*.95,txt_ncad_sh_all);
%plot(mdl_dapi_sh_all);
%text(mean(ivs_dapi_sh_all(:,1)),max(ivs_dapi_sh_all(:,2))*.95,txt_dapi_sh_all);
scatmdl_paramadjust(p1,p2);
hold off
title(sprintf('SHAM \n Intensity vs Strain'),'fontsize',titfsz);
xlabel('Strain','fontsize',labfsz)
ylabel('Intensity','fontsize',labfsz)
legend('IGB1','NCAD','fontsize',lgdfsz)%,'DAPI')
xlim([0.355,0.455])
%Complie R^2 vals, and pvals of pearson and spearman correlations
%   Export to Excel Sheet
statvalmat = [igb1_mi_r2,igb1_sh_r2,...
            ncad_mi_r2,ncad_sh_r2,...
            dapi_mi_r2,dapi_sh_r2;...
            igb1_mi_pears,igb1_sh_pears,...
            ncad_mi_pears,ncad_sh_pears,...
            dapi_mi_pears,dapi_sh_pears;...
            igb1_mi_spear,igb1_sh_spear,...
```

```
            ncad_mi_spear,ncad_sh_spear,...
            dapi_mi_spear,dapi_sh_spear;...
            igb1_mi_pears_rho,igb1_sh_pears_rho,...
            ncad_mi_pears_rho,ncad_sh_pears_rho,...
            dapi_mi_pears_rho,dapi_sh_pears_rho];
xlsht_nam = 'Stats_Table_Compar.xlsx';
sht_nam = 'Whole_Int_vs_Str';
rng = 'B5';
writematrix(statvalmat,xlsht_nam,'Sheet',sht_nam,'Range',rng);


%IGB1 Plot of intensity vs strain w/ MI and SHAM
figure(3)
hold on
scatter(ivs_igb1_mi_all(:,1),ivs_igb1_mi_all(:,2),...
      'd','MarkerFaceColor','r','MarkerEdgeColor','k');
scatter(ivs_igb1_sh_all(:,1),ivs_igb1_sh_all(:,2),...
      'o','MarkerFaceColor','b','MarkerEdgeColor','k');
plot(mdl_igb1_mi_all);
%text(mean(ivs_igb1_mi_all(:,1)),max(ivs_igb1_mi_all(:,2))*.95,txt_igb1_mi_all);
plot(mdl_igb1_sh_all);
%text(mean(ivs_igb1_sh_all(:,1)),max(ivs_igb1_sh_all(:,2))*.95,txt_igb1_sh_all);

hold off
title(sprintf('IGB1 \n Intensity vs Strain'))
xlabel('Strain (Fract)')
ylabel('Intensity')
legend('MI','SHAM')

%NCAD Plot of intensity vs strain w/ MI and SHAM
figure(4)
hold on
scatter(ivs_ncad_mi_all(:,1),ivs_ncad_mi_all(:,2),...
      'd','MarkerFaceColor','r','MarkerEdgeColor','k');
scatter(ivs_ncad_sh_all(:,1),ivs_ncad_sh_all(:,2),...
      'o','MarkerFaceColor','b','MarkerEdgeColor','k');
plot(mdl_ncad_mi_all);
%text(mean(ivs_ncad_mi_all(:,1)),max(ivs_ncad_mi_all(:,2))*.95,txt_ncad_mi_all);
plot(mdl_ncad_sh_all);
%text(mean(ivs_ncad_sh_all(:,1)),max(ivs_ncad_sh_all(:,2))*.95,txt_ncad_sh_all);

hold off
title(sprintf('NCAD \n Intensity vs Strain'))
xlabel('Strain (Fract)')
ylabel('Intensity')
legend('MI','SHAM')


%DAPI Plot of intensity vs strain w/ MI and SHAM
figure(5)
hold on
scatter(ivs_dapi_mi_all(:,1),ivs_dapi_mi_all(:,2),...
      'd','MarkerFaceColor','r','MarkerEdgeColor','k');
scatter(ivs_dapi_sh_all(:,1),ivs_dapi_sh_all(:,2),...
      'o','MarkerFaceColor','b','MarkerEdgeColor','k');
plot(mdl_dapi_mi_all);
%text(mean(ivs_dapi_mi_all(:,1)),max(ivs_dapi_mi_all(:,2))*.95,txt_dapi_mi_all);
plot(mdl_dapi_sh_all);
%text(mean(ivs_dapi_sh_all(:,1)),max(ivs_dapi_sh_all(:,2))*.95,txt_dapi_sh_all);

hold off
title(sprintf('DAPI \n Intensity vs Strain'))
xlabel('Strain (Fract)')
ylabel('Intensity')
legend('MI','SHAM')
```

```
%% Bar Chart for All Values(MI vs SHAM)
clear all
close all

load('aim3vals221113.mat');
% load('aim3vals.mat');
ind_mi = find(calctab.MIType(:) == 'MI');
ind_sh = find(calctab.MIType(:) == 'NA');
%% Average Values
%Avg Signal Intensity
intavg_igb1_mi_vect = numxtract(calctab.IntenseAvg,ind_mi,1);
intavg_ncad_mi_vect = numxtract(calctab.IntenseAvg,ind_mi,2);
intavg_dapi_mi_vect = numxtract(calctab.IntenseAvg,ind_mi,3);
intavg_igb1_mi_mean = mean(intavg_igb1_mi_vect,'all','omitnan');
intavg_ncad_mi_mean = mean(intavg_ncad_mi_vect,'all','omitnan');
intavg_dapi_mi_mean = mean(intavg_dapi_mi_vect,'all','omitnan');

%Std Err
intavg_igb1_mi_mean_sterr = std(intavg_igb1_mi_vect,0,'all','omitnan')/...
                            sqrt(length(intavg_igb1_mi_vect));
intavg_ncad_mi_mean_sterr = std(intavg_ncad_mi_vect,0,'all','omitnan')/...
                            sqrt(length(intavg_ncad_mi_vect));
intavg_dapi_mi_mean_sterr = std(intavg_dapi_mi_vect,0,'all','omitnan')/...
                            sqrt(length(intavg_dapi_mi_vect));


intavg_igb1_sh_vect = numxtract(calctab.IntenseAvg,ind_sh,1);
intavg_ncad_sh_vect = numxtract(calctab.IntenseAvg,ind_sh,2);
intavg_dapi_sh_vect = numxtract(calctab.IntenseAvg,ind_sh,3);
intavg_igb1_sh_mean = mean(intavg_igb1_sh_vect,'all','omitnan');
intavg_ncad_sh_mean = mean(intavg_ncad_sh_vect,'all','omitnan');
intavg_dapi_sh_mean = mean(intavg_dapi_sh_vect,'all','omitnan');


%Std Err
intavg_igb1_sh_mean_sterr = std(intavg_igb1_sh_vect,0,'all','omitnan')/...
                            sqrt(length(intavg_igb1_sh_vect));
intavg_ncad_sh_mean_sterr = std(intavg_ncad_sh_vect,0,'all','omitnan')/...
                            sqrt(length(intavg_ncad_sh_vect));
intavg_dapi_sh_mean_sterr = std(intavg_dapi_sh_vect,0,'all','omitnan')\...
                            sqrt(length(intavg_dapi_sh_vect));

%Set Marker Size
mksz = 42;
%Average signal intensity
barmat_intavg = [intavg_igb1_mi_mean,intavg_igb1_sh_mean;...
            intavg_ncad_mi_mean,intavg_ncad_sh_mean];
barmat_intavg_sterr = [intavg_igb1_mi_mean_sterr,intavg_igb1_sh_mean_sterr;...
            intavg_ncad_mi_mean_sterr,intavg_ncad_sh_mean_sterr];

[h_intavg_igb1,p_intavg_igb1] = ttest2(intavg_igb1_mi_vect,intavg_igb1_sh_vect);
[h_intavg_ncad,p_intavg_ncad] = ttest2(intavg_ncad_mi_vect,intavg_ncad_sh_vect);
[h_intavg_dapi,p_intavg_dapi] = ttest2(intavg_dapi_mi_vect,intavg_dapi_sh_vect);
figure(1)
x = categorical({'IGB1','NCAD'});
x = reordercats(x,{'IGB1','NCAD'});
b = bar(barmat_intavg,'grouped')
hold on
xtipserr = [b(1).XEndPoints;b(2).XEndPoints];
ytips = b(1).YEndPoints;
er = errorbar(xtipserr',barmat_intavg,barmat_intavg_sterr);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
xticklabels(x)
% if h_intavg_igb1 == 1;
%     scatter(x(1),max(barmat_intavg(1,1:2)),'*','k','SizeData',mksz);
```

```
% end
% if h_intavg_ncad == 1;
%     scatter(x(2),max(barmat_intavg(2,1:2)),'*','k','SizeData',mksz);
% end
hold off
title(sprintf('Whole Heart \n Normalized Pixel Intensity'),'fontsize',16)
legend('MI','SHAM','fontsize',10)
xlabel('Target Molecule','fontsize',14)
ylabel('Normalized Intensity','fontsize',14);
ylim([0 0.5])

%% Scatters and Linear Regressions Planar
%Find indices, pull vectors with corresponding vals
clear all
close all

load('aim3vals221113.mat');
%load indices of each plane separated by MI and SHAM
%ind_mi_api; ind_mi_mid; ind_mi_bas; ind_sh_api; ind_sh_mid; ind_sh_bas;
planar_inds

%% Average Values in Prep for Bar Chart
%Avg Signal Intensity
intavg_mi_api = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_api);
intavg_mi_mid = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_mid);
intavg_mi_bas = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_bas);

scatvect_mi_api_igb1_intavg = [intavg_mi_api.igb1.vct,str2double(string(calctab.RegCode(ind_mi_api)))];
scatvect_mi_mid_igb1_intavg = [intavg_mi_mid.igb1.vct,str2double(string(calctab.RegCode(ind_mi_mid)))];
scatvect_mi_bas_igb1_intavg = [intavg_mi_bas.igb1.vct,str2double(string(calctab.RegCode(ind_mi_bas)))];

scatvect_mi_api_ncad_intavg = [intavg_mi_api.ncad.vct,str2double(string(calctab.RegCode(ind_mi_api)))];
scatvect_mi_mid_ncad_intavg = [intavg_mi_mid.ncad.vct,str2double(string(calctab.RegCode(ind_mi_mid)))];
scatvect_mi_bas_ncad_intavg = [intavg_mi_bas.ncad.vct,str2double(string(calctab.RegCode(ind_mi_bas)))];

scatvect_mi_api_dapi_intavg = [intavg_mi_api.dapi.vct,str2double(string(calctab.RegCode(ind_mi_api)))];
scatvect_mi_mid_dapi_intavg = [intavg_mi_mid.dapi.vct,str2double(string(calctab.RegCode(ind_mi_mid)))];
scatvect_mi_bas_dapi_intavg = [intavg_mi_bas.dapi.vct,str2double(string(calctab.RegCode(ind_mi_bas)))];

intavg_sh_api = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_api);
intavg_sh_mid = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_mid);
intavg_sh_bas = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_bas);


scatvect_sh_api_igb1_intavg = [intavg_sh_api.igb1.vct,str2double(string(calctab.RegCode(ind_sh_api)))];
scatvect_sh_mid_igb1_intavg = [intavg_sh_mid.igb1.vct,str2double(string(calctab.RegCode(ind_sh_mid)))];
scatvect_sh_bas_igb1_intavg = [intavg_sh_bas.igb1.vct,str2double(string(calctab.RegCode(ind_sh_bas)))];

scatvect_sh_api_ncad_intavg = [intavg_sh_api.ncad.vct,str2double(string(calctab.RegCode(ind_sh_api)))];
scatvect_sh_mid_ncad_intavg = [intavg_sh_mid.ncad.vct,str2double(string(calctab.RegCode(ind_sh_mid)))];
scatvect_sh_bas_ncad_intavg = [intavg_sh_bas.ncad.vct,str2double(string(calctab.RegCode(ind_sh_bas)))];

scatvect_sh_api_dapi_intavg = [intavg_sh_api.dapi.vct,str2double(string(calctab.RegCode(ind_sh_api)))];
scatvect_sh_mid_dapi_intavg = [intavg_sh_mid.dapi.vct,str2double(string(calctab.RegCode(ind_sh_mid)))];
scatvect_sh_bas_dapi_intavg = [intavg_sh_bas.dapi.vct,str2double(string(calctab.RegCode(ind_sh_bas)))];

%% Paper Values are stored in 2xn matrices where the first row contains the strain data
%MI Data in ppvals_MI, Sham data in ppvals_SH
ppval_mi_api = ppvals_MI(1,13:end);
ppval_mi_mid = ppvals_MI(1,7:12);
ppval_mi_bas = ppvals_MI(1,1:6);

ppval_sh_api = ppvals_SH(1,13:end);
ppval_sh_mid = ppvals_SH(1,7:12);
ppval_sh_bas = ppvals_SH(1,1:6);
```

```
%% MI Intensity Vs Strain IGB1 and NCAD
%Apical
tit_mi_api = sprintf('MI: Apical \n Intensity vs Strain');
xlb_mi_api = 'Strain';
ylb_mi_api = 'Intensity';
lgd_mi_api = {'IGB1','NCAD'};%,'DAPI'};
mi_api_int_statmat = scatscript_3input(ppval_mi_api,scatvect_mi_api_igb1_intavg,...
    ppval_mi_api,scatvect_mi_api_ncad_intavg,...
    ppval_mi_api,scatvect_mi_api_dapi_intavg,...
    tit_mi_api,xlb_mi_api,ylb_mi_api,lgd_mi_api,'api');

%Assign Excel File and Sheet names
xlsht_nam = 'Stats_Table_Compar.xlsx';
sht_nam = 'Plane_Bth_vs_Str';
%Write Stat Measure Vals to Excel
rng = 'B6';
writematrix(mi_api_int_statmat,xlsht_nam,'Sheet',sht_nam,'Range',rng);

%Mid
tit_mi_mid = sprintf('MI: Mid \n Intensity vs Strain');
xlb_mi_mid = 'Strain';
ylb_mi_mid = 'Intensity';
lgd_mi_mid = {'IGB1','NCAD'};%'DAPI'};
mi_mid_int_statmat = scatscript_3input(ppval_mi_mid,scatvect_mi_mid_igb1_intavg,...
    ppval_mi_mid,scatvect_mi_mid_ncad_intavg,...
    ppval_mi_mid,scatvect_mi_mid_dapi_intavg,...
    tit_mi_mid,xlb_mi_mid,ylb_mi_mid,lgd_mi_mid,'mid');
%Write Stat Measure Vals to Excel
rng = 'B14';
writematrix(mi_mid_int_statmat,xlsht_nam,'Sheet',sht_nam,'Range',rng);

%% Sham Intensity Vs Strain IGB1 and NCAD
%Apical
tit_sh_api = sprintf('SHAM: Apical \n Intensity vs Strain');
xlb_sh_api = 'Strain';
ylb_sh_api = 'Intensity';
lgd_sh_api = {'IGB1','NCAD'};%'DAPI'};
sh_api_int_statmat = scatscript_3input(ppval_sh_api,scatvect_sh_api_igb1_intavg,...
    ppval_sh_api,scatvect_sh_api_ncad_intavg,...
    ppval_sh_api,scatvect_sh_api_dapi_intavg,...
    tit_sh_api,xlb_sh_api,ylb_sh_api,lgd_sh_api,'api');
%Write Stat Measure Vals to Excel
rng = 'E6';
writematrix(sh_api_int_statmat,xlsht_nam,'Sheet',sht_nam,'Range',rng);

%Mid
tit_sh_mid = sprintf('SHAM: Mid \n Intensity vs Strain');
xlb_sh_mid = 'Strain';
ylb_sh_mid = 'Intensity';
lgd_sh_mid = {'IGB1','NCAD'};%'DAPI'};
sh_mid_int_statmat = scatscript_3input(ppval_sh_mid,scatvect_sh_mid_igb1_intavg,...
    ppval_sh_mid,scatvect_sh_mid_ncad_intavg,...
    ppval_sh_mid,scatvect_sh_mid_dapi_intavg,...
    tit_sh_mid,xlb_sh_mid,ylb_sh_mid,lgd_sh_mid,'mid');
%Write Stat Measure Vals to Excel
rng = 'E14';
writematrix(sh_mid_int_statmat,xlsht_nam,'Sheet',sht_nam,'Range',rng);

%% IGB1 Intensity Vs Strain MI and Sham
%Apical
tit_igb1_api = sprintf('IGB1:Apical \n Intensity vs Strain');
xlb_igb1_api = 'Strain';
ylb_igb1_api = 'Intensity';
lgd_igb1_api = {'MI','Sham'};
scatscript_2input(ppval_mi_api,scatvect_mi_api_igb1_intavg,...
    ppval_sh_api,scatvect_sh_api_igb1_intavg,...
```

```
    tit_igb1_api,xlb_igb1_api,ylb_igb1_api,lgd_igb1_api,'api');

%Mid
tit_igb1_mid = sprintf('IGB1:Mid \n Intensity vs Strain');
xlb_igb1_mid = 'Strain';
ylb_igb1_mid = 'Intensity';
lgd_igb1_mid = {'MI','SHAM'};
scatscript_2input(ppval_mi_mid,scatvect_mi_mid_igb1_intavg,...
    ppval_sh_mid,scatvect_sh_mid_igb1_intavg,...
    tit_igb1_mid,xlb_igb1_mid,ylb_igb1_mid,lgd_igb1_mid,'mid');


%% NCAD Intensity Vs Strain MI and Sham
%Apical
tit_ncad_api = sprintf('NCAD:Apical \n Intensity vs Strain');
xlb_ncad_api = 'Strain';
ylb_ncad_api = 'Intensity';
lgd_ncad_api = {'MI','Sham'};
scatscript_2input(ppval_mi_api,scatvect_mi_api_ncad_intavg,...
    ppval_sh_api,scatvect_sh_api_ncad_intavg,...
    tit_ncad_api,xlb_ncad_api,ylb_ncad_api,lgd_ncad_api,'api');

%Mid
tit_ncad_mid = sprintf('NCAD:Mid \n Intensity vs Strain');
xlb_ncad_mid = 'Strain';
ylb_ncad_mid = 'Intensity';
lgd_ncad_mid = {'MI','SHAM'};
scatscript_2input(ppval_mi_mid,scatvect_mi_mid_ncad_intavg,...
    ppval_sh_mid,scatvect_sh_mid_ncad_intavg,...
    tit_ncad_mid,xlb_ncad_mid,ylb_ncad_mid,lgd_ncad_mid,'mid');


%% Bar Chart for Values by Plane(MI vs SHAM)
clear all
close all
titfsz = 16;
lgdfsz = 10;
labfsz = 14;

load('aim3vals221113.mat');
%load indices of each plane separated by MI and SHAM
%ind_mi_api; ind_mi_mid; ind_mi_bas; ind_sh_api; ind_sh_mid; ind_sh_bas;
planar_inds

%% Average Values in Prep for Bar Chart
%Avg Signal Intensity
intavg_mi_api = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_api);
intavg_mi_mid = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_mid);
intavg_mi_bas = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_bas);

intavg_sh_api = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_api);
intavg_sh_mid = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_mid);
intavg_sh_bas = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_bas);

%% Create Bar Charts for MI-Sham IGB1 and NCAD for each Plane
%% Create Bar Charts for Intensity
%%%IGB1
%Set Marker Size
mksz = 42;

barmat_igb1_intavg = [intavg_mi_api.igb1.avg,intavg_sh_api.igb1.avg;...
                    intavg_mi_mid.igb1.avg,intavg_sh_mid.igb1.avg];%...
barmat_igb1_intavg_sterr = [intavg_mi_api.igb1.sterr,intavg_sh_api.igb1.sterr;...
                    intavg_mi_mid.igb1.sterr,intavg_sh_mid.igb1.sterr];%...
%                       intavg_mi_bas.igb1.avg,intavg_sh_bas.igb1.avg];
statmat_igb1_intavg = stattest_apimid(...
```

```
    intavg_mi_api.igb1.vct,intavg_mi_mid.igb1.vct,...%intavg_mi_bas.igb1.vct,...
    intavg_sh_api.igb1.vct,intavg_sh_mid.igb1.vct);%,intavg_sh_bas.igb1.vct);
% statrep_igb1_intavg = sigchk(statmat_igb1_intavg);
figure(1)
x = categorical({'Api','Mid'});%,'Bas'});
x = reordercats(x,{'Api','Mid'});%,'Bas'});
b = bar(barmat_igb1_intavg,'grouped');
hold on
xtips = [b(1).XEndPoints',b(2).XEndPoints'];
er = errorbar(xtips,barmat_igb1_intavg,barmat_igb1_intavg_sterr);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
xticklabels(x);

hold off

title('IGB1 Intensity','fontsize',titfsz)
legend('MI','SHAM','Location','northeast','fontsize',lgdfsz)
xlabel('Ventricular Plane','fontsize',labfsz)
ylabel('Intensity','fontsize',labfsz);
ylim([0 0.6])
%%%NCAD
%Set Marker Size
mksz = 42;

barmat_ncad_intavg = [intavg_mi_api.ncad.avg,intavg_sh_api.ncad.avg;...
                      intavg_mi_mid.ncad.avg,intavg_sh_mid.ncad.avg];%...
barmat_ncad_intavg_sterr = [intavg_mi_api.ncad.sterr,intavg_sh_api.ncad.sterr;...
                      intavg_mi_mid.ncad.sterr,intavg_sh_mid.ncad.sterr];%...
%                         intavg_mi_bas.ncad.avg,intavg_sh_bas.ncad.avg];
statmat_ncad_intavg = stattest_apimid(...
    intavg_mi_api.ncad.vct,intavg_mi_mid.ncad.vct,...intavg_mi_bas.ncad.vct,...
    intavg_sh_api.ncad.vct,intavg_sh_mid.ncad.vct);%,intavg_sh_bas.ncad.vct);
% statrep_ncad_intavg = sigchk(statmat_ncad_intavg);
figure(2)
x = categorical({'Api','Mid'});%,'Bas'});
x = reordercats(x,{'Api','Mid'});%,'Bas'});
b = bar(barmat_ncad_intavg,'grouped');
hold on
xtips = [b(1).XEndPoints',b(2).XEndPoints'];
er = errorbar(xtips,barmat_ncad_intavg,barmat_ncad_intavg_sterr);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
xticklabels(x);

hold off

title('NCAD Intensity','fontsize',titfsz)
legend('MI','SHAM','Location','northeast','fontsize',lgdfsz)
xlabel('Ventricular Plane','fontsize',labfsz)
ylabel('Intensity','fontsize',labfsz);
ylim([0 0.25])

%% Scatters and Linear Regressions Planar
clear all
close all
load('aim3vals221113.mat')

spatial_inds
%% Average Values in Prep for Bar Chart
%Avg Signal Intensity
intavg_mi_anse = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_anse);
intavg_mi_inla = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_inla);

scatvect_mi_anse_igb1_intavg = [intavg_mi_anse.igb1.vct,str2double(string(calctab.RegCode(ind_mi_anse)))];
scatvect_mi_inla_igb1_intavg = [intavg_mi_inla.igb1.vct,str2double(string(calctab.RegCode(ind_mi_inla)))];
```

```
scatvect_mi_anse_ncad_intavg = [intavg_mi_anse.ncad.vct,str2double(string(calctab.RegCode(ind_mi_anse)))];
scatvect_mi_inla_ncad_intavg = [intavg_mi_inla.ncad.vct,str2double(string(calctab.RegCode(ind_mi_inla)))];


intavg_sh_anse = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_anse);
intavg_sh_inla = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_inla);

scatvect_sh_anse_igb1_intavg = [intavg_sh_anse.igb1.vct,str2double(string(calctab.RegCode(ind_sh_anse)))];
scatvect_sh_inla_igb1_intavg = [intavg_sh_inla.igb1.vct,str2double(string(calctab.RegCode(ind_sh_inla)))];

scatvect_sh_anse_ncad_intavg = [intavg_sh_anse.ncad.vct,str2double(string(calctab.RegCode(ind_sh_anse)))];
scatvect_sh_inla_ncad_intavg = [intavg_sh_inla.ncad.vct,str2double(string(calctab.RegCode(ind_sh_inla)))];

scatvect_sh_anse_dapi_intavg = [intavg_sh_anse.dapi.vct,str2double(string(calctab.RegCode(ind_sh_anse)))];
scatvect_sh_inla_dapi_intavg = [intavg_sh_inla.dapi.vct,str2double(string(calctab.RegCode(ind_sh_inla)))];


%% Paper Values are stored in 2xn matrices where the first row contains the strain data
%MI Data in ppvals_MI, Sham data in ppvals_SH
ppval_mi_anse = ppvals_MI(1,[1 2 3 7 8 9 13 14]);
ppval_mi_inla = ppvals_MI(1,[4 5 6 10 11 12 15]);

ppval_sh_anse = ppvals_SH(1,[1 2 3 7 8 9 13 14]);
ppval_sh_inla = ppvals_SH(1,[4 5 6 10 11 12 15 16]);
%% MI Intensity Vs Strain IGB1 and NCAD
%Ante-Sept
tit_mi_anse = sprintf('MI: Ante-Sept \n Intensity vs Strain');
xlb_mi_anse = 'Strain';
ylb_mi_anse = 'Intensity';
lgd_mi_anse = {'IGB1','NCAD'};%,'DAPI'};
mi_anse_int_statmat = scatscript_anse_inla_3input(ppval_mi_anse,scatvect_mi_anse_igb1_intavg,...
    ppval_mi_anse,scatvect_mi_anse_ncad_intavg,...
    ppval_mi_anse,scatvect_mi_anse_dapi_intavg,...
    tit_mi_anse,xlb_mi_anse,ylb_mi_anse,lgd_mi_anse,'anse');

%Infe-Late
tit_mi_inla = sprintf('MI: Infe-Late \n Intensity vs Strain');
xlb_mi_inla = 'Strain';
ylb_mi_inla = 'Intensity';
lgd_mi_inla = {'IGB1','NCAD'};%,'DAPI'};
mi_inla_int_statmat = scatscript_anse_inla_3input(ppval_mi_inla,scatvect_mi_inla_igb1_intavg,...
    ppval_mi_inla,scatvect_mi_inla_ncad_intavg,...
    ppval_mi_inla,scatvect_mi_inla_dapi_intavg,...
    tit_mi_inla,xlb_mi_inla,ylb_mi_inla,lgd_mi_inla,'inla');

%% Sham Intensity Vs Strain IGB1 and NCAD
%Ante-Sept
tit_sh_anse = sprintf('SHAM: Ante-Sept \n Intensity vs Strain');
xlb_sh_anse = 'Strain';
ylb_sh_anse = 'Intensity';
lgd_sh_anse = {'IGB1','NCAD'};%,'DAPI'};
sh_anse_int_statmat = scatscript_anse_inla_3input(ppval_sh_anse,scatvect_sh_anse_igb1_intavg,...
    ppval_sh_anse,scatvect_sh_anse_ncad_intavg,...
    ppval_sh_anse,scatvect_sh_anse_dapi_intavg,...
    tit_sh_anse,xlb_sh_anse,ylb_sh_anse,lgd_sh_anse,'anse');


%Infe-Late
tit_sh_inla = sprintf('SHAM: Infe-Late \n Intensity vs Strain');
xlb_sh_inla = 'Strain';
ylb_sh_inla = 'Intensity';
lgd_sh_inla = {'IGB1','NCAD'};%,'DAPI'};
sh_inla_int_statmat = scatscript_anse_inla_3input(ppval_sh_inla,scatvect_sh_inla_igb1_intavg,...
    ppval_sh_inla,scatvect_sh_inla_ncad_intavg,...
    ppval_sh_inla,scatvect_sh_inla_dapi_intavg,...
```

```
    tit_sh_inla,xlb_sh_inla,ylb_sh_inla,lgd_sh_inla,'inla');

%% Bar Chart for Values by Plane(MI vs SHAM)
clear all
close all
titfsz = 16;
lgdfsz = 10;
labfsz = 14;


load('aim3vals221113.mat');
%load indices of each plane separated by MI and SHAM
%ind_mi_anse; ind_mi_inla; ind_mi_bas; ind_sh_anse; ind_sh_inla; ind_sh_bas;
spatial_inds

%% Average Values in Prep for Bar Chart
%Avg Signal Intensity
intavg_mi_anse = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_anse);
intavg_mi_inla = numxtract_plus_stats(calctab.IntenseAvg,ind_mi_inla);

intavg_sh_anse = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_anse);
intavg_sh_inla = numxtract_plus_stats(calctab.IntenseAvg,ind_sh_inla);


%% Create Bar Charts for MI-Sham IGB1 and NCAD for each Plane
%% Create Bar Charts for Intensity
%%%IGB1
%Set Marker Size
mksz = 42;

barmat_igb1_intavg = [intavg_mi_anse.igb1.avg,intavg_sh_anse.igb1.avg;...
                      intavg_mi_inla.igb1.avg,intavg_sh_inla.igb1.avg];
barmat_igb1_intavg_sterr = [intavg_mi_anse.igb1.sterr,intavg_sh_anse.igb1.sterr;...
                      intavg_mi_inla.igb1.sterr,intavg_sh_inla.igb1.sterr];
statmat_igb1_intavg = stattest_anse_inla(...
    intavg_mi_anse.igb1.vct,intavg_mi_inla.igb1.vct,...%intavg_mi_bas.igb1.vct,...
    intavg_sh_anse.igb1.vct,intavg_sh_inla.igb1.vct)%,intavg_sh_bas.igb1.vct);
%statrep_igb1_intavg = sigchk(statmat_igb1_intavg);
figure(1)
x = categorical({'Ante-Sept','Infe-Late'})%,'Bas'});
x = reordercats(x,{'Ante-Sept','Infe-Late'})%,'Bas'});
b = bar(barmat_igb1_intavg,'grouped');
hold on
xtips = [b(1).XEndPoints',b(2).XEndPoints'];
er = errorbar(xtips,barmat_igb1_intavg,barmat_igb1_intavg_sterr);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
xticklabels(x);

hold off

title('IGB1 Intensity','fontsize',titfsz)
legend('MI','SHAM','Location','northeast','fontsize',lgdfsz)
xlabel('Ventricular Region','fontsize',labfsz)
ylabel('Intensity','fontsize',labfsz);
ylim([0 0.5])
%%%NCAD
%Set Marker Size
mksz = 42;

barmat_ncad_intavg = [intavg_mi_anse.ncad.avg,intavg_sh_anse.ncad.avg;...
                      intavg_mi_inla.ncad.avg,intavg_sh_inla.ncad.avg];%...
barmat_ncad_intavg_sterr = [intavg_mi_anse.ncad.sterr,intavg_sh_anse.ncad.sterr;...
                      intavg_mi_inla.ncad.sterr,intavg_sh_inla.ncad.sterr];%...
%                         intavg_mi_bas.ncad.avg,intavg_sh_bas.ncad.avg];
statmat_ncad_intavg = stattest_anse_inla(...
```

```
    intavg_mi_anse.ncad.vct,intavg_mi_inla.ncad.vct,...%intavg_mi_bas.ncad.vct,...
    intavg_sh_anse.ncad.vct,intavg_sh_inla.ncad.vct);%,intavg_sh_bas.ncad.vct);
%statrep_ncad_intavg = sigchk(statmat_ncad_intavg);
figure(2)
x = categorical({'Ante-Sept','Infe-Late'});%,'Bas'});
x = reordercats(x,{'Ante-Sept','Infe-Late'});%,'Bas'});
b = bar(barmat_ncad_intavg,'grouped');
hold on
xtips = [b(1).XEndPoints',b(2).XEndPoints'];
er = errorbar(xtips,barmat_ncad_intavg,barmat_ncad_intavg_sterr);
er(1).LineStyle = 'none';
er(2).LineStyle = 'none';
xticklabels(x);

hold off

title('NCAD Intensity','fontsize',titfsz)
legend('MI','SHAM','Location','northeast','fontsize',labfsz)
xlabel('Ventricular Region','fontsize',labfsz)
ylabel('Intensity','fontsize',labfsz);
ylim([0 0.3])

function outarray = numxtract(tabcol,ind,rw);
outarray = cellfun(@(x,c) x(c),...
    tabcol(ind),num2cell(repmat(rw,[length(ind),1])));

end
function outarray = numxtract_plus_stats(tabcol,ind);
%Pull Vector of Values Corresponding to input indices
outarray.igb1.vct = cellfun(@(x,c) x(c),...
    tabcol(ind),num2cell(repmat(1,[length(ind),1])));
outarray.ncad.vct = cellfun(@(x,c) x(c),...
    tabcol(ind),num2cell(repmat(2,[length(ind),1])));
outarray.dapi.vct = cellfun(@(x,c) x(c),...
    tabcol(ind),num2cell(repmat(3,[length(ind),1])));

%Find Average of Pulled Vectors
outarray.igb1.avg = mean(outarray.igb1.vct,'all','omitnan');
outarray.ncad.avg = mean(outarray.ncad.vct,'all','omitnan');
outarray.dapi.avg = mean(outarray.dapi.vct,'all','omitnan');

%Find Std of Pulled Vectors
outarray.igb1.stdev = std(outarray.igb1.vct,0,'all','omitnan');
outarray.ncad.stdev = std(outarray.ncad.vct,0,'all','omitnan');
outarray.dapi.stdev = std(outarray.dapi.vct,0,'all','omitnan');

%Find Std Err of Pulled Vectors
outarray.igb1.sterr = outarray.igb1.stdev/sqrt(length(outarray.igb1.vct));
outarray.ncad.sterr = outarray.ncad.stdev/sqrt(length(outarray.ncad.vct));
outarray.dapi.sterr = outarray.dapi.stdev/sqrt(length(outarray.dapi.vct));

end

%% Generate vectors for planes of: Apical, Mid, Basal
regnmvect = str2double(string(calctab.RegCode(:)));
ind_mi_api = find(calctab.MIType(:) == 'MI' &...
            regnmvect >= 13 &...
            regnmvect <= 17);
ind_mi_mid = find(calctab.MIType(:) == 'MI' &...
            regnmvect >= 7 &...
            regnmvect <= 12);
ind_mi_bas = find(calctab.MIType(:) == 'MI' &...
            regnmvect >= 1 &...
            regnmvect <= 6);
```

```
ind_sh_api = find(calctab.MIType(:) == 'NA' &...
            regnmvect >= 13 &...
            regnmvect <= 17);

ind_sh_mid = find(calctab.MIType(:) == 'NA' &...
            regnmvect >= 7 &...
            regnmvect <= 12);

ind_sh_bas = find(calctab.MIType(:) == 'NA' &...
            regnmvect >= 1 &...
            regnmvect <= 6);

function [p1,p2] = scatmdl_paramadjust(p1,p2)
%Input lines generated from plot(mdl) and adjust lines
%(1,1,1) = markeroutline
%(2,1,1) = central line
%(3,1,1) = bottom bounding line
%(4,1,1) = top bounding line
p1(1,1,1).Color = [1 0 0];
p1(2,1,1).Color = [1 0 0];
p1(3,1,1).Color = [1 0 0];
p1(4,1,1).Color = [1 0 0];

%p1(1,1,1).LineStyle = '-';
p1(2,1,1).LineStyle = '-.';
p1(3,1,1).LineStyle = ':';
p1(4,1,1).LineStyle = ':';

%p1(1,1,1).LineWidth = 0.25;
p1(2,1,1).LineWidth = 4;
p1(3,1,1).LineWidth = 2;
p1(4,1,1).LineWidth = 2;

p2(1,1,1).Color = [0 0 1];
p2(2,1,1).Color = [0 0 1];
p2(3,1,1).Color = [0 0 1];
p2(4,1,1).Color = [0 0 1];

%p2(1,1,1).LineStyle = '-';
p2(2,1,1).LineStyle = '--';
p2(3,1,1).LineStyle = '-.';
p2(4,1,1).LineStyle = '-.';

%p2(1,1,1).LineWidth = 0.25;
p2(2,1,1).LineWidth = 4;
p2(3,1,1).LineWidth = 2;
p2(4,1,1).LineWidth = 2;

end


function statmat = scatscript_3input(indvect1,depvect1,indvect2,depvect2,indvect3,depvect3,tit,xl,yl,leg,planchk)
%Script to Intake dependent value of interest and plot against
%corresponding strain value
titfsz = 16;
lgdfsz = 10;
labfsz = 14;


if planchk == 'api';
   nlm = 5;
   adjfact = 12;
elseif planchk == 'mid';
   nlm = 6;
   adjfact = 6;
elseif planchk == 'bas';
```

```
    nlm = 6;
    adjfact = 0;
end
for n0 = 1:nlm;
    ind1 = find(depvect1(:,2) == (n0+adjfact));
    ind2 = find(depvect2(:,2) == (n0+adjfact));
    ind3 = find(depvect3(:,2) == (n0+adjfact));


%place into nx2 matrix with independent var in 1st column
try
    bldmat1(n0,:) = [indvect1(n0),mean(depvect1(ind1),'all','omitnan')];
    bldmat2(n0,:) = [indvect2(n0),mean(depvect2(ind2),'all','omitnan')];
    bldmat3(n0,:) = [indvect3(n0),mean(depvect3(ind3),'all','omitnan')];
catch
    continue
end
end
%Find non-nan indices
    indvals_1 = find(~isnan(bldmat1(:,2)));
    indvals_2 = find(~isnan(bldmat2(:,2)));
    indvals_3 = find(~isnan(bldmat3(:,2)));

    %Fit linear regression model  to the relationship
    mdl1 = fitlm(bldmat1(indvals_1,1),bldmat1(indvals_1,2));
    mdl2 = fitlm(bldmat2(indvals_2,1),bldmat2(indvals_2,2));
    mdl3 = fitlm(bldmat3(indvals_3,1),bldmat3(indvals_3,2));

%Generate Stat Mat for output
%Contains R^2, Pearson pval, Spearman pval
    mdl1_r2 = mdl1.Rsquared.Ordinary;
    [mdl1_prho,mdl1_pears] = corr(bldmat1(indvals_1,1),bldmat1(indvals_1,2),...
      'Type','Pearson');
    [rho,mdl1_spear] = corr(bldmat1(indvals_1,1),bldmat1(indvals_1,2),...
      'Type','Spearman');
    mdl2_r2 = mdl2.Rsquared.Ordinary;
    [mdl2_prho,mdl2_pears] = corr(bldmat2(indvals_1,1),bldmat2(indvals_1,2),...
      'Type','Pearson');
    [rho,mdl2_spear] = corr(bldmat2(indvals_1,1),bldmat2(indvals_1,2),...
      'Type','Spearman');
    mdl3_r2 = mdl3.Rsquared.Ordinary;
    [mdl3_prho,mdl3_pears] = corr(bldmat3(indvals_1,1),bldmat3(indvals_1,2),...
      'Type','Pearson');
    [rho,mdl3_spear] = corr(bldmat3(indvals_1,1),bldmat3(indvals_1,2),...
      'Type','Spearman');

   statmat = [mdl1_r2,mdl2_r2,...%mdl3_r2;...
              mdl1_pears,mdl2_pears,...%mdl3_pears;...
              mdl1_spear,mdl2_spear,...%mdl3_spear;...
              mdl1_prho,mdl2_prho];,...%mdl3_prho];

%Generate Text string containing R^2 value of relationship
    txt1 = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl1.Rsquared.Ordinary)));
    txt2 = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl2.Rsquared.Ordinary)));
    txt3 = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl3.Rsquared.Ordinary)));

%Open new figure
figvar = figure;
hold on

%Scatter Group 1
scatter(bldmat1(indvals_1,1),bldmat1(indvals_1,2),...
       'd','MarkerFaceColor','r','MarkerEdgeColor','r','LineWidth',2.5);
```

```
%Scatter Group 2
scatter(bldmat2(indvals_2,1),bldmat2(indvals_2,2),...
      'o','MarkerFaceColor','b','MarkerEdgeColor','b','LineWidth',2.5');

% %Scatter Group 3
% scatter(bldmat3(indvals_3,1),bldmat3(indvals_3,2),...
%       's','MarkerFaceColor','b','MarkerEdgeColor','k');

%Plot linear regression line
p1 = plot(mdl1);
p2 = plot(mdl2);
[p1,p2] = scatmdl_paramadjust(p1,p2);
%plot(mdl3);
%Add R^2 text to plot at coordinates of first point
% text(mean(bldmat1(indvals_1(:),1)),mean(bldmat1(indvals_1(:),2)),txt1);
% text(mean(bldmat2(indvals_2(:),1)),mean(bldmat2(indvals_2(:),2)),txt2);
% text(mean(bldmat3(indvals_3(:),1)),mean(bldmat3(indvals_3(:),2)),txt3);
hold off

%Add detail
title(tit,'fontsize',titfsz);
xlabel(xl,'fontsize',labfsz);
ylabel(yl,'fontsize',labfsz);
legend(leg,'fontsize',lgdfsz);
end


function statmat = scatscript_anse_inla_3input(indvect1,depvect1,indvect2,depvect2,indvect3,depvect3,tit,xl,yl,leg,planchk)
%Script to Intake dependent value of interest and plot against
%corresponding strain value
titfsz = 16;
lgdfsz = 10;
labfsz = 14;

if planchk == 'anse';
   indlist = [1 2 3 7 8 9 13 14];
   nlm = length(indlist);

elseif planchk == 'inla';
   indlist = [4 5 6 10 11 12 15 16];
   nlm = length(indlist);


end
for n0 = 1:nlm;
   ind1 = find(depvect1(:,2) == indlist(n0));
   ind2 = find(depvect2(:,2) == indlist(n0));
   ind3 = find(depvect3(:,2) == indlist(n0));


%place into nx2 matrix with independent var in 1st column
try
   bldmat1(n0,:) = [indvect1(n0),mean(depvect1(ind1),'all','omitnan')];
   bldmat2(n0,:) = [indvect2(n0),mean(depvect2(ind2),'all','omitnan')];
   bldmat3(n0,:) = [indvect3(n0),mean(depvect3(ind3),'all','omitnan')];
catch
   continue
end
end
%Find non-nan indices
   indvals_1 = find(~isnan(bldmat1(:,2)));
   indvals_2 = find(~isnan(bldmat2(:,2)));
   indvals_3 = find(~isnan(bldmat3(:,2)));

   %Fit linear regression model  to the relationship
```

```
    mdl1 = fitlm(bldmat1(indvals_1,1),bldmat1(indvals_1,2));
    mdl2 = fitlm(bldmat2(indvals_2,1),bldmat2(indvals_2,2));
    mdl3 = fitlm(bldmat3(indvals_3,1),bldmat3(indvals_3,2));

%Generate Stat Mat for output
%Contains R^2, Pearson pval, Spearman pval
    mdl1_r2 = mdl1.Rsquared.Ordinary;
    [mdl1_prho,mdl1_pears] = corr(bldmat1(indvals_1,1),bldmat1(indvals_1,2),...
      'Type','Pearson');
    [mdl1_srho,mdl1_spear] = corr(bldmat1(indvals_1,1),bldmat1(indvals_1,2),...
      'Type','Spearman');
    mdl2_r2 = mdl2.Rsquared.Ordinary;
    [mdl2_prho,mdl2_pears] = corr(bldmat2(indvals_1,1),bldmat2(indvals_1,2),...
      'Type','Pearson');
    [mdl2_srho,mdl2_spear] = corr(bldmat2(indvals_1,1),bldmat2(indvals_1,2),...
      'Type','Spearman');
    mdl3_r2 = mdl3.Rsquared.Ordinary;
    [mdl3_prho,mdl3_pears] = corr(bldmat3(indvals_1,1),bldmat3(indvals_1,2),...
      'Type','Pearson');
    [mdl3_srho,mdl3_spear] = corr(bldmat3(indvals_1,1),bldmat3(indvals_1,2),...
      'Type','Spearman');

   statmat = [mdl1_r2,mdl2_r2,mdl3_r2;...
              mdl1_pears,mdl2_pears,mdl3_pears;...
              mdl1_spear,mdl2_spear,mdl3_spear;...
              mdl1_prho,mdl2_prho,mdl3_prho];

%Generate Text string containing R^2 value of relationship
    txt1 = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl1.Rsquared.Ordinary)));
    txt2 = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl2.Rsquared.Ordinary)));
    txt3 = char(horzcat(sprintf('R^{2} = '),...
    sprintf('%.2f',mdl3.Rsquared.Ordinary)));

%Open new figure
figvar = figure;
hold on

%Scatter Group 1
scatter(bldmat1(indvals_1,1),bldmat1(indvals_1,2),...
       'd','MarkerFaceColor','r','MarkerEdgeColor','r','LineWidth',2.5);

%Scatter Group 2
scatter(bldmat2(indvals_2,1),bldmat2(indvals_2,2),...
       'o','MarkerFaceColor','b','MarkerEdgeColor','b','LineWidth',2.5);

%Scatter Group 3
% scatter(bldmat3(indvals_3,1),bldmat3(indvals_3,2),...
%         's','MarkerFaceColor','b','MarkerEdgeColor','k');

%Plot linear regression line
p1 = plot(mdl1);
p2 = plot(mdl2);
[p1,p2] = scatmdl_paramadjust(p1,p2);
% plot(mdl3);
%Add R^2 text to plot at coordinates of first point
% text(mean(bldmat1(indvals_1(:),1)),mean(bldmat1(indvals_1(:),2)),txt1);
% text(mean(bldmat2(indvals_2(:),1)),mean(bldmat2(indvals_2(:),2)),txt2);
% text(mean(bldmat3(indvals_3(:),1)),mean(bldmat3(indvals_3(:),2)),txt3);
hold off

%Add detail
title(tit,'fontsize',titfsz);
xlabel(xl,'fontsize',labfsz);
ylabel(yl,'fontsize',labfsz);
```

```
legend(leg,'fontsize',lgdfsz);
end


%% Spatial Indices
%Spatial Separated by Plane
regnmvect = str2double(string(calctab.RegCode(:)));

ind_mi_api_ante = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 13);
ind_mi_api_sept = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 14);
ind_mi_api_late = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 16);
ind_mi_api_infe = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 15);

ind_mi_mid_ante = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 7);
ind_mi_mid_sept = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 8 &...
                regnmvect <= 9);
ind_mi_mid_late = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 11 &...
                regnmvect <= 12);
ind_mi_mid_infe = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 10);

ind_mi_bas_ante = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 1);
ind_mi_bas_sept = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 2 &...
                regnmvect <= 3);
ind_mi_bas_late = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 5 &...
                regnmvect <= 6);
ind_mi_bas_infe = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 4);

ind_sh_api_ante = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 13);
ind_sh_api_sept = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 14);
ind_sh_api_late = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 16);
ind_sh_api_infe = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 15);

ind_sh_mid_ante = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 7);
ind_sh_mid_sept = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 8 &...
                regnmvect <= 9);
ind_sh_mid_late = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 11 &...
                regnmvect <= 12);
ind_sh_mid_infe = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 10);

ind_sh_bas_ante = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 1);
ind_sh_bas_sept = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 2 &...
                regnmvect <= 3);
ind_sh_bas_late = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 5 &...
```

```
                regnmvect <= 6);
ind_sh_bas_infe = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 4);

%Spatial All
ind_mi_ante = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 1 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect == 7 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect == 13);
ind_mi_sept = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 14 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 8 &...
                regnmvect <= 9 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 2 &...
                regnmvect <= 3);
ind_mi_late = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 16 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 11 &...
                regnmvect <= 12 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 5 &...
                regnmvect <= 6);
ind_mi_infe = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 15 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect == 10 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect == 4);

ind_sh_ante = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 1 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect == 7 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect == 13);
ind_sh_sept = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 14 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 8 &...
                regnmvect <= 9 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 2 &...
                regnmvect <= 3);
ind_sh_late = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 16 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 11 &...
                regnmvect <= 12 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 5 &...
                regnmvect <= 6);
ind_sh_infe = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 15 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect == 10 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect == 4);
%% Spatial FrontBack
% tags: frt bck
% frt = [1 2 6 7 8 12 13 14];
% bck = [3 4 5 9 10 11 15 16];
```

```
ind_mi_frnt = find(calctab.MIType(:) == 'MI' &...
            regnmvect >= 1 &...
            regnmvect <= 2 |...
            calctab.MIType(:) == 'MI' &...
            regnmvect >= 6 &...
            regnmvect <= 8 |...
            calctab.MIType(:) == 'MI' &...
            regnmvect >= 12 &...
            regnmvect <= 14);
ind_mi_back = find(calctab.MIType(:) == 'MI' &...
            regnmvect >= 3 &...
            regnmvect <= 5 |...
            calctab.MIType(:) == 'MI' &...
            regnmvect >= 9 &...
            regnmvect <= 11 |...
            calctab.MIType(:) == 'MI' &...
            regnmvect >= 15 &...
            regnmvect <= 16);

ind_sh_frnt = find(calctab.MIType(:) == 'NA' &...
            regnmvect >= 1 &...
            regnmvect <= 2 |...
            calctab.MIType(:) == 'NA' &...
            regnmvect >= 6 &...
            regnmvect <= 8 |...
            calctab.MIType(:) == 'NA' &...
            regnmvect >= 12 &...
            regnmvect <= 14);

ind_sh_back = find(calctab.MIType(:) == 'NA' &...
            regnmvect >= 3 &...
            regnmvect <= 5 |...
            calctab.MIType(:) == 'NA' &...
            regnmvect >= 9 &...
            regnmvect <= 11 |...
            calctab.MIType(:) == 'NA' &...
            regnmvect >= 15 &...
            regnmvect <= 16);

%% Spatial Anterior-Septal(AnSe) vs Inferior-Lateral(InLa)
% AnSe = [1 2 3 7 8 9 13 14];
% InLa = [4 5 6 10 11 12 15 16];
ind_mi_anse = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 1 &...
                regnmvect <= 3 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 7 &...
                regnmvect <= 9 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 13 &...
                regnmvect <= 14);
ind_mi_inla = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 4 &...
                regnmvect <= 6 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 10 &...
                regnmvect <= 12 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 15 &...
                regnmvect <= 16);
ind_sh_anse = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 1 &...
                regnmvect <= 3 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 7 &...
```

```
                regnmvect <= 9 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 13 &...
                regnmvect <= 14);
ind_sh_inla = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 4 &...
                regnmvect <= 6 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 10 &...
                regnmvect <= 12 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 15 &...
                regnmvect <= 16);

%% Spatial Anterior-Lateral(AnLa) vs Inferior-Septal(InSe)
%Anterior-Septal(AnSe) vs Inferior-Lateral(InLa)
% AnLa = [1 5 6 7 11 12 13 16];
% InSe = [2 3 4 8 9 10 14 15];
ind_mi_anla = find(calctab.MIType(:) == 'MI' &...
                regnmvect == 1 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 5 &...
                regnmvect <= 7 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 11 &...
                regnmvect <= 13 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect == 16);
ind_mi_inse = find(calctab.MIType(:) == 'MI' &...
                regnmvect >= 2 &...
                regnmvect <= 4 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 8 &...
                regnmvect <= 10 |...
                calctab.MIType(:) == 'MI' &...
                regnmvect >= 14 &...
                regnmvect <= 15);
ind_sh_anla = find(calctab.MIType(:) == 'NA' &...
                regnmvect == 1 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 5 &...
                regnmvect <= 7 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 11 &...
                regnmvect <= 13 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect == 16);
ind_sh_inse = find(calctab.MIType(:) == 'NA' &...
                regnmvect >= 2 &...
                regnmvect <= 4 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 8 &...
                regnmvect <= 10 |...
                calctab.MIType(:) == 'NA' &...
                regnmvect >= 14 &...
                regnmvect <= 15);

function outmat = stattest(mi_api,mi_mid,mi_bas,sh_api,sh_mid,sh_bas)
%Two-Sample TTest script to output matrices containing h and p values for
%   sample comparisons of interest
%   The two main conditions are MI and NA with different planes of concern
%       also differentiating data
%   The output of this function will be a (3x2x3) matrix.
%       3x2x1 will contain an initial column of h-values and a second
%       column of p-values comparing MI vs NA where each row will be the
%       comparison within a plane of concern.
```

```
%      3x2x2 WIll compare the planes of interest within the MI condition
%      3x3x2 Will compare the planes of interest within the NA condition

%MI vs SH same plane
[h_mish_api,p_mish_api] = ttest2(mi_api,sh_api);
[h_mish_mid,p_mish_mid] = ttest2(mi_mid,sh_mid);
[h_mish_bas,p_mish_bas] = ttest2(mi_bas,sh_bas);

%MI Planar Comparison
[h_apimid_mi,p_apimid_mi] = ttest2(mi_api,mi_mid);
[h_apibas_mi,p_apibas_mi] = ttest2(mi_api,mi_bas);
[h_midbas_mi,p_midbas_mi] = ttest2(mi_mid,mi_bas);

%SH Planar Comparison
[h_apimid_sh,p_apimid_sh] = ttest2(sh_api,sh_mid);
[h_apibas_sh,p_apibas_sh] = ttest2(sh_api,sh_bas);
[h_midbas_sh,p_midbas_sh] = ttest2(sh_mid,sh_bas);


outmat = cat(3,...
   [h_mish_api,p_mish_api;h_mish_mid,p_mish_mid;h_mish_bas,p_mish_bas],...
   [h_apimid_mi,p_apimid_mi;h_apibas_mi,p_apibas_mi;h_midbas_mi,p_midbas_mi],...
   [h_apimid_sh,p_apimid_sh;h_apibas_sh,p_apibas_sh;h_midbas_sh,p_midbas_sh]);


end


function outmat = stattest_anse_inla(mi_api,mi_mid,sh_api,sh_mid)
%Two-Sample TTest script to output matrices containing h and p values for
%  sample comparisons of interest
%  The two main conditions are MI and NA with different planes of concern
%     also differentiating data
%  The output of this function will be a (3x2x3) matrix.
%     3x2x1 will contain an initial column of h-values and a second
%     column of p-values comparing MI vs NA where each row will be the
%     comparison within a plane of concern.
%     3x2x2 WIll compare the planes of interest within the MI condition
%     3x3x2 Will compare the planes of interest within the NA condition

%MI vs SH same plane
[h_mish_api,p_mish_api] = ttest2(mi_api,sh_api);
[h_mish_mid,p_mish_mid] = ttest2(mi_mid,sh_mid);
% [h_mish_bas,p_mish_bas] = ttest2(mi_bas,sh_bas);

%MI Planar Comparison
[h_apimid_mi,p_apimid_mi] = ttest2(mi_api,mi_mid);
% [h_apibas_mi,p_apibas_mi] = ttest2(mi_api,mi_bas);
% [h_midbas_mi,p_midbas_mi] = ttest2(mi_mid,mi_bas);

%SH Planar Comparison
[h_apimid_sh,p_apimid_sh] = ttest2(sh_api,sh_mid);
% [h_apibas_sh,p_apibas_sh] = ttest2(sh_api,sh_bas);
% [h_midbas_sh,p_midbas_sh] = ttest2(sh_mid,sh_bas);


outmat = cat(3,...
   [h_mish_api,p_mish_api;h_mish_mid,p_mish_mid],...%h_mish_bas,p_mish_bas],...
   [h_apimid_mi,p_apimid_mi;NaN,NaN],...%h_apibas_mi,p_apibas_mi;h_midbas_mi,p_midbas_mi],...
   [h_apimid_sh,p_apimid_sh;NaN,NaN]);%h_apibas_sh,p_apibas_sh;h_midbas_sh,p_midbas_sh]);


end


function outmat = stattest_apimid(mi_api,mi_mid,sh_api,sh_mid);%mi_bas,sh_api,sh_mid,sh_bas)
```

```
%Two-Sample TTest script to output matrices containing h and p values for
%   sample comparisons of interest
%   The two main conditions are MI and NA with different planes of concern
%       also differentiating data
%   The output of this function will be a (3x2x3) matrix.
%       3x2x1 will contain an initial column of h-values and a second
%       column of p-values comparing MI vs NA where each row will be the
%       comparison within a plane of concern.
%       3x2x2 WIll compare the planes of interest within the MI condition
%       3x3x2 Will compare the planes of interest within the NA condition

%MI vs SH same plane
[h_mish_api,p_mish_api] = ttest2(mi_api,sh_api);
[h_mish_mid,p_mish_mid] = ttest2(mi_mid,sh_mid);
% [h_mish_bas,p_mish_bas] = ttest2(mi_bas,sh_bas);

%MI Planar Comparison
[h_apimid_mi,p_apimid_mi] = ttest2(mi_api,mi_mid);
% [h_apibas_mi,p_apibas_mi] = ttest2(mi_api,mi_bas);
% [h_midbas_mi,p_midbas_mi] = ttest2(mi_mid,mi_bas);

%SH Planar Comparison
[h_apimid_sh,p_apimid_sh] = ttest2(sh_api,sh_mid);
% [h_apibas_sh,p_apibas_sh] = ttest2(sh_api,sh_bas);
% [h_midbas_sh,p_midbas_sh] = ttest2(sh_mid,sh_bas);


outmat = cat(3,...
    [h_mish_api,p_mish_api;h_mish_mid,p_mish_mid],...;h_mish_bas,p_mish_bas],...
    [h_apimid_mi,p_apimid_mi;NaN,NaN],...%h_apibas_mi,p_apibas_mi;h_midbas_mi,p_midbas_mi],...
    [h_apimid_sh,p_apimid_sh;NaN,NaN]);%h_apibas_sh,p_apibas_sh;h_midbas_sh,p_midbas_sh]);


end
```