Engineering 2D Cardiac Tissues Using Biomimetic Protein Micropatterns Based on the Extracellular Matrix in the Embryonic Heart

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Materials Science & Engineering

Ivan Batalov

B.S., Applied Mathematics and Physics, Moscow Institute of Physics and Technology
M.S., Applied Mathematics and Physics, Moscow Institute of Physics and Technology
M.S., Materials Science & Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

April 27, 2017

Acknowledgements

Despite of the fact that most Ph.D. students graduate, earning a doctoral degree is still a long and difficult process that requires large amount of effort, determination, and psychological endurance. I am immensely grateful to everyone whose support helped me make this possible.

Firstly, I'd like to thank my advisor Adam W. Feinberg for allowing me to work in his group, providing valuable advice in my research, and helping me grow as a scientist. Doing research in cardiac tissue engineering under his leadership has been exciting and fruitful.

My work wouldn't be possible without the help of my colleagues. Dr. Quentin Jallerat, Dr. John Szymanski, and Dr. Rachelle Palchesko-Simko taught me all the necessary cell culture and tissue engineering techniques I've used throughout my work in the lab. I am particularly grateful to Quentin for all the hours we spent working on establishing the stem cell culture and differentiation room and to Rachelle for proofreading and correcting all my terrible writing. I'd like to thank Jaci Bliley and Andrew Lee who have been helping me with the stem cell duties after Quentin's graduation and who will continue the everyday stem cell struggle after I'm gone. I'd like to give special thanks to my former undergraduate student Sean Kim, who worked with me for 2 years, for being surprisingly helpful for an undergraduate student, quickly learning new techniques and spending countless hours in the lab spin-coating for me. Overall, I am thankful to all current and former members of the Regenerative Biomaterials and Therapeutics Group for your jokes, fruitful discussions, and being a pleasure to work with.

I thank my committee – Christopher Bettinger, Michael Bockstaller, and Newell Washburn – for providing me with valuable feedback and support that helped me guide my research in the right direction.

I would also like to thank Prof. Wu’s lab at Stanford University and Prof. van der Meer’s lab at the University of Groningen for providing me stem cells for my research.

My work wouldn’t be possible without the generosity of my funding sources. I’d like to thanks John and Claire Bertucci for their fellowship that supported me during the last year of my program. I’d also like to thank National Institute of Health and American Heart Association for providing Prof. Feinberg grants that funded my research (NIH Innovator grant DP2HL117750, AHA Scientist Development Grant 12SDG11800036).

Finally, I’d like to thank my all my friends, your help and support have made the years I have spent in the USA one of the greatest experiences in my life.

Abstract

Cardiovascular disease is the leading cause of death worldwide. Due to the extremely low natural regeneration rate of heart muscle, development of new therapeutics directed towards heart repair is challenging. A potential approach to regenerate damaged heart is offered by cardiac tissue engineering. Specifically, it aims at engineering cardiac muscle in vitro and implanting it into the site of injury so that it can be integrated into the host tissue and restore the heart's function. To ensure the effectiveness of this technique, the engineered tissue needs to recapitulate structural and functional properties of the native myocardium. Myocardium consists of laminar sheets of uniaxially aligned cardiac muscle cells (cardiomyocytes) wrapped around the heart. Therefore, achieving high cardiomyocyte alignment in engineered muscle is crucial. In this study we aimed at stimulating cardiomyocyte alignment by mimicking their niche in the embryonic heart. We hypothesized that recapitulating the extracellular cues that guide myocardial development in the embryo can guide cardiac tissue organization *in vitro*. To test this hypothesis, we imaged the structure of fibronectin – the most abundant protein in embryonic heart's extracellular matrix (ECM) – and derived a 2D pattern from it that was then microcontact printed onto a substrate to guide cell alignment. We compared chick cardiomyocyte alignment on the biomimetic pattern and line patterns that have been extensively studied in the past. Results revealed a unique cell density-dependent response of cardiomyocytes to the biomimetic pattern that allowed us to elucidate the role of cell-cell and cell-ECM interactions in cardiomyocyte alignment on fibronectin patterns by looking at the effect of local pattern features on alignment and inhibiting N-cadherin-based cell-cell junctions. Further, to engineer more clinically relevant tissues, we differentiated human induced pluripotent stem cells (iPSCs) and embryonic stem cells (ESCs) into cardiomyocytes and seeded them onto the fibronectin patterns. Cardiac tissues produced with these cells showed significant differences compared to the chick tissues due to their immature phenotype. We showed that co-culture with cardiac fibroblasts (CFBs) as well as maturation of iPSC-derived cardiomyocytes (iPSC-CMs) increased tissue alignment, indicating the important role of both of these factors in developing novel methods to engineer functional cardiac tissues.

Table of Contents

List of Figures and Illustrations

List of Acronyms

NCD	Noncommunicable Disease
iPSC	Induced Pluripotent Stem Cell
ESC	Embryonic Stem Cell
ESC-CM	ESC-derived Cardiomyocyte
iPSC-CM	iPSC-derived Cardiomyocyte
HUES9-CM	Cardiomyocyte derived from the ESC line HUES9
PLLA	Poly(L-lactide)
CFB	Cardiac Fibroblast
HT	Heart Tube
HH	Denotes chick embryonic developmental stage according to Hamburger and Hamilton
PDMS	Poly-(dimethylsiloxane)
FBS	Fetal Bovine Serum
HI-FBS	Heat-Inactivated Fetal Bovine Serum
OOP	Orientational Order Parameter
EB	Embryoid Body
GSK3	Glycogen Synthase Kinase 3
ECM	Extracellular Matrix
T3	Tri-iodo-L-thyronine

Chapter 1: Introduction

1.1 Cardiac Disease is the Leading Cause of Death Worldwide

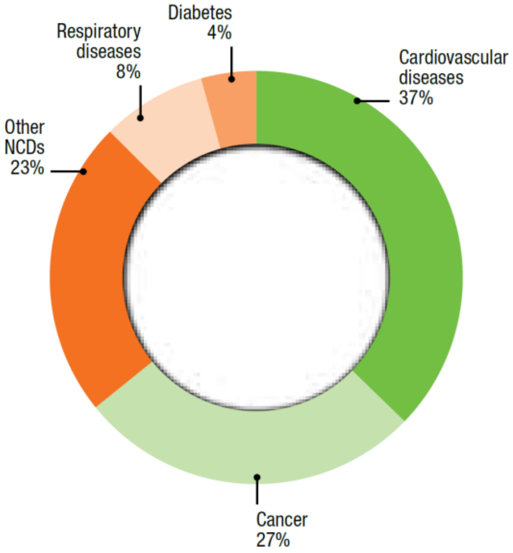

Figure 1.1 Top causes of death due to noncommunicable diseases (NCDs). Image taken with permission from the World Health Organization's 2016 World Health Statistics.

Cardiac disease is currently the leading cause of premature human death, accounting for 17.5 million deaths per year globally according to the World Health Organization's 2016 World Health Statistics (fig. 1.1). Most of these deaths occur due to heart failure resulting from myocardial infarction – death of cardiac muscle due to oxygen deficiency, usually caused by an occlusion of a coronary artery supplying the heart.[1] After the infarction, the injured area is replaced by non-contractile scar tissue that increases the load on the rest of the heart due to its higher stiffness.[2] Myocardial infarction can also cause arrhythmias due to the inability of scar tissue to propagate the action potential.[3,4] This, along with other concurrent cardiovascular conditions, such as hypertension, further weakens the heart and can eventually lead to heart failure.[5] The primary challenge in developing effective therapeutics for heart repair is the extremely low natural regeneration rate of the human heart. For some species, like zebrafish, the heart can fully regenerate on its own after a serious injury, such as 20% ventricular resection, within a few months.[6] However, the normal regeneration rate of a human heart is estimated to be around 1% per year at the age of 20 and around 0.4% at the age of 75,[7] which is insufficient for adequate heart repair. To this date, the only effective treatment available to prevent an infarcted heart from failure is heart transplantation. However, it is greatly limited by the donor availability. According to the International Society for Heart & Lung Transplantation, only ~4500 heart transplantations are performed annually around the world, ~2,500 of which occur in the United States, and ~10% of patients die while waiting for a donor heart.[8] Further, there are a large number of patients that do not qualify for heart transplant but could still benefit from therapies aimed at heart repair, which could be a significant portion of the people that die due to heart disease.

1.2 Cell-Based Therapies for Cardiac Regeneration

Cell-based therapies have recently emerged as a potential solution for heart repair.[9] Various cell sources have been proposed for stimulating myocardial regeneration, including committed cardiomyocytes, cardiac progenitor cells, and several types of stem cells (fig. 1.2A). There are three main strategies for inducing heart regeneration using cell therapies: fibroblast transdifferentiation, myocyte proliferation, and stem cell or cardiac cell injection (fig. 1.2B). Fibroblast transdifferentiation is an approach aimed at direct reprogramming of CFBs into functional cardiomyocytes. It was inspired by the recent advances in generating iPSCs from CFBs and differentiating them into cardiomyocytes. Studies reported that using 4 transcription factors, GATA4, HAND2, MEF2C, and TBX5, indigenous fibroblasts can be reprogrammed into beating cardiomyocytes that can improve myocardial function and reduce the size of post-infarcted scar tissue.[10] However, these studies are still not ready for human clinical studies as the potential side effects of fibroblast reprogramming are not understood, in addition to reproducibility issues reported by several authors.[9, 11-13]

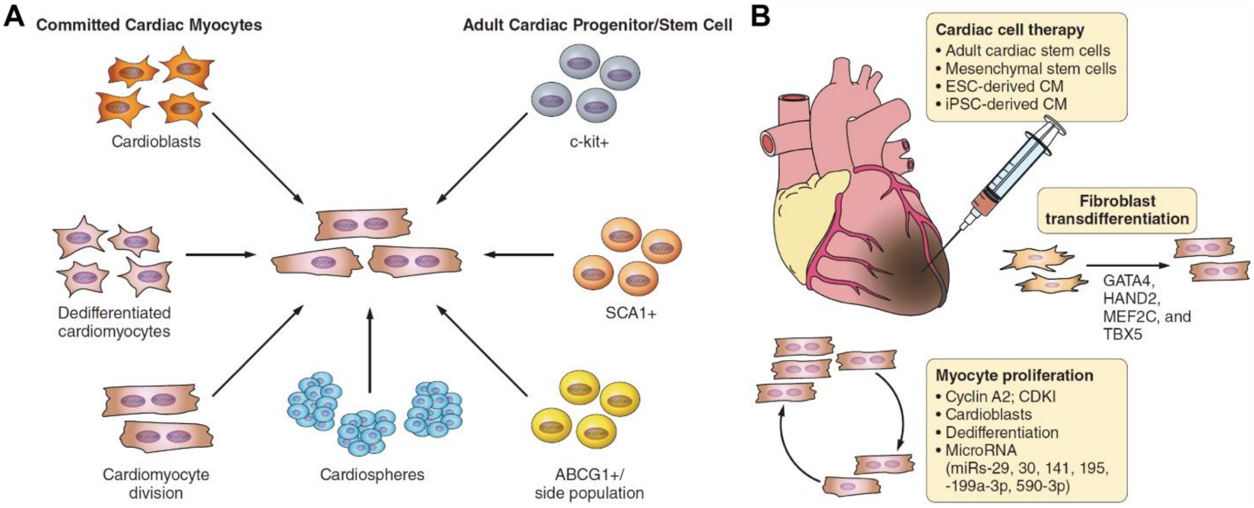

Figure 1.2 Cell-based therapies for cardiac regeneration. A: Proposed potential cell sources for cardiac repair. **B:** Cell-based strategies for myocardial repair. Adapted from Zhang, Y., J. Mignone, et al. (2015). Physiological Reviews **95**(4): 1189-1204.

The cardiomyocyte proliferation approach aims at forcing terminally differentiated cardiomyocytes to re-enter the cell cycle and divide to replenish the cardiomyocyte population and regenerate the infarcted part of the heart muscle. Early attempts utilizing overexpression of oncoproteins showed that although this approach is possible, it is also extremely difficult due to various side effects, such as tumor growth, cell apoptosis, and differential response of atrial and ventricular cardiomyocytes to the same stimuli.[14] Controlling negative and positive cell cycle regulators was also shown to induce cardiomyocyte cell cycle re-entry, although no studies have yet reported improved myocardial regeneration using this method.[15] Recent studies have found that microRNA can also be used to selectively stimulate cardiomyocyte proliferation reducing scar formation and improving post-infarct heart function.[16-18] Overall, significant progress has been achieved in inducing cardiomyocyte proliferation, but further research is required to make this technique efficient and safe for clinical therapies.

Stem cell injection therapies rely on integration of the injected cells into the host tissue and paracrine signaling provided by the injected cells to stimulate angiogenesis and heart remodeling.[19] Most of these therapies use injection of a certain type of stem cells, such as c-KIT$^+$ cardiac progenitor cells, bone marrow-derived stem cells, ESC-derived cardiomyocytes (ESC-CMs) or iPSC-CMs, to induce heart regeneration.[7] One of the main problems with cell injection is low survivability of the injected cells due to scar tissue's higher stiffness and the lack of sufficient vasculature to provide oxygen and nutrition for the injected cells.

1.3 Tissue Engineering Approaches to Heart Repair

Tissue engineering offers another potential approach to treat infarcted hearts. The goal of cardiac tissue engineering is to develop cardiac muscle in vitro that has structural and functional characteristics, such as cardiac muscle cell (cardiomyocyte) alignment, net contractile force, action potential propagation speed, and Ca^{2+} handling, comparable to those of the native myocardium[20] – the contractile part of the heart located in the heart wall and comprised of cardiomyocytes. These engineered tissues could potentially be implanted into the infarcted area and incorporated with the native myocardium to restore the contractile capabilities of the heart.[21] The main advantage of the tissue engineering approach compared to cell therapies is that the implanted tissue is already comprised of functional cardiomyocytes electrically coupled with each other as opposed to stem cells that need to differentiate into cardiomyocytes and organize themselves into muscle tissue.

Additionally, engineered tissues can be used in drug testing and development of heart-on-chip devices. Development of a single drug costs billions of dollars and takes around 12 years.[22] One of the main issues in this process is high failure rate at the human testing phase – only 12% of drugs entering that phase make it to the market place.[23] The main reason for such a high failure rate is that the positive results of pre-clinical tests usually performed on animals often don’t translate to the human studies – drugs often show higher toxicity in humans compared to animals. However, studies based on engineered human tissues can solve this problem when used in combination with and in addition to animal studies. This can dramatically decrease the time and the cost of drug development, facilitating the production of cheaper and more effective medication. Finally, engineered myocardium can be used in heart-on-a-chip devices aimed to model the native myocardium. These devices can be used to study and compare developmental and physiological processes occurring in a healthy and a diseased heart in order to better understand the nature of various heart conditions.

1.4 Native Heart ECM as a Scaffold for Engineering Tissue Organization

One of the most intuitive approaches to build cardiac muscle from individual cells is to let the natural heart environment guide the organization of these cells. This has been previously done by reseeding decellularized hearts with various types of cells, such as ESCs,[24,25] cardiac progenitor cells,[25,26] and cardiomyocytes[27] (fig. 1.3). However, cardiac tissue built using this approach can only achieve a small fraction of the contractile force of the native heart due to the low cell repopulation rate, mostly due to the difficulties in delivering the cells throughout the decellularized tissue. Cell delivery is usually done via pumping the cell solution through the heart’s coronary vasculature, but cells fail to migrate away from the blood vessels inside the tissue. Therefore, it is clear that a more informed bottom-up approach based on the understanding of the complex hierarchical structure of myocardium is required to build heart muscle from individual cells.

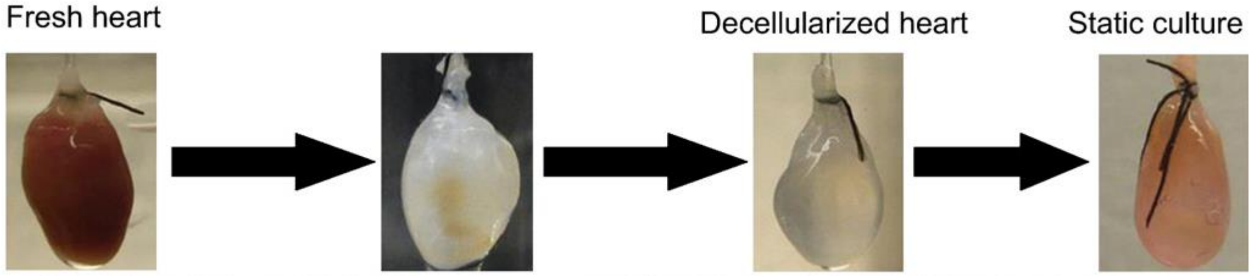

Figure 1.3. Decellularization of a mouse heart with SDS and its repopulation with ESCs. Adapted from Ng, S. L. J., K. Narayanan, et al. (2011). Biomaterials **32**(30): 7571-7580.

1.5 Techniques for Inducing Cardiac Muscle Tissue Alignment

Cardiomyocytes in the myocardium are organized into laminar sheets wrapped around one another and form the middle part of the heart wall. Within each sheet cardiomyocytes are uniaxially aligned and mechanically and electrically coupled allowing for synchronous contraction and maximization of the contractile force. Therefore, achieving high cardiomyocyte alignment in the engineered tissues is one of the key goals of cardiac tissue engineering. Most of existing techniques to guide cell alignment are aimed at controlling various characteristics of the cell environment, such as composition,[28] substrate topography (fig. 1.4A-C),[29-32] 3D micro-structure (fig. 2B,G),[30,33,34] external mechanical force (fig. 1.4D),[35] and mechanical stiffness (fig. 1.4 E,F).[32,36,37] Promising results in developing functional cardiac tissue in 2D and 3D have been achieved using these techniques. However, these engineered tissues do not achieve the level of structural organization and physiological function of the native heart.

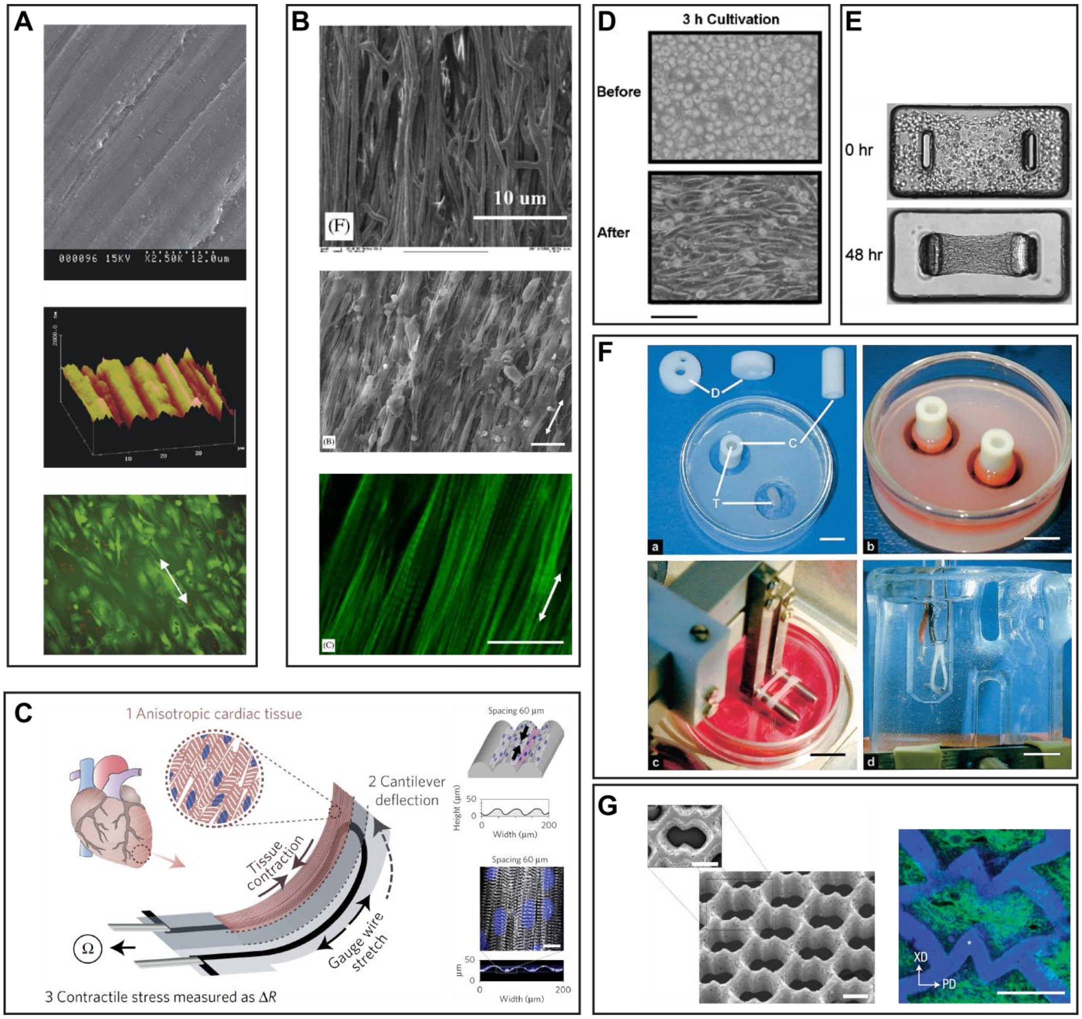

Figure 1.4. Overview of the methods for stimulating engineering cardiac tissue alignment. A: Cardiomyocyte alignment stimulated by micro-groves on a substrate abraded by lapping paper. **B:** Cardiomyocyte alignment stimulated by an aligned electrospun poly(L-lactide) (PLLA) scaffold. **C:** 3D-printed micro-groves with 60 µm spacing stimulate cardiac tissue alignment in a microphysiological device. **D**: Cardiomyocyte alignment stimulated by mechanical stretching of the substrate. **E:** 3D cardiac micro-tissue is formed by gel compaction around the posts. Cardiomyocytes align between the posts by the mechanical anisotropy of the environment. **F:** Cardiac tissue formed by gel compaction around a silicone cylinder. Cardiomyocytes align along the ring due to mechanical anisotropy the environment. **G:** Honeycomb-like scaffold for cardiac tissue engineering. Cardiomyocyte alignment is stimulated via anisotropy of the mechanical stiffness of the scaffold.

Figure sources: A: Au, H. T. H., et al. (2007). Biomaterials 28(29): 4277-4293. **B:** Zong, X., et al. (2005). Biomaterials 26(26): 5330-5338. **C:** Lind, J. U., et al. (2017). Nat Mater 16(3): 303-308. **D:** Matsuda, et al. (2004). Biochemical and Biophysical Research Communications 326(1): 228-232. **E:**

West, A. R., et al. (2013). American Journal of Physiology-Lung Cellular and Molecular Physiology 304(1): L4-L16. **F:** Zimmermann, W. H. (2001). Circulation Research 90(2): 223-230. **G:** Engelmayr, G. C., et al. (2008). Nature Materials 7(12): 1003-1010.

1.6 Cardiac ECM Composition

One of the ways to improve currently existing cardiac tissue engineering techniques lies within mimicking the environment of the developing heart. This approach starts with determining the critical parameters of the cell environment required for the formation of the aligned myocardium, as the complete recapitulation of the *in vivo* state is impossible. The cardiomyocyte environment in myocardium is comprised of ECM, which consists of a vast variety of proteins, polysaccharides, proteoglycans, and other components, and has a broad spectrum of functions, such as providing mechanical and structural support, regulating cell signaling pathways, controlling nutrient and gas exchange, etc. A group of ECM proteins, such as laminin, fibronectin, and collagen type I, serve as adhesion molecules that provide attachment sites for cells.[38] A study by Williams et al revealed that the composition of cardiac ECM changes during heart development.[39] Interestingly, it was found that fibronectin is the most abundant protein in embryonic heart ECM, while in the adult heart it’s substituted by collagen type I and laminin (fig. 1.5). Additionally, transient fibronectin expression can be observed in adult hearts undergoing hypertrophic growth indicating its crucial role in heart growth and development, which makes fibronectin a promising material for cardiac tissue engineering.[40]

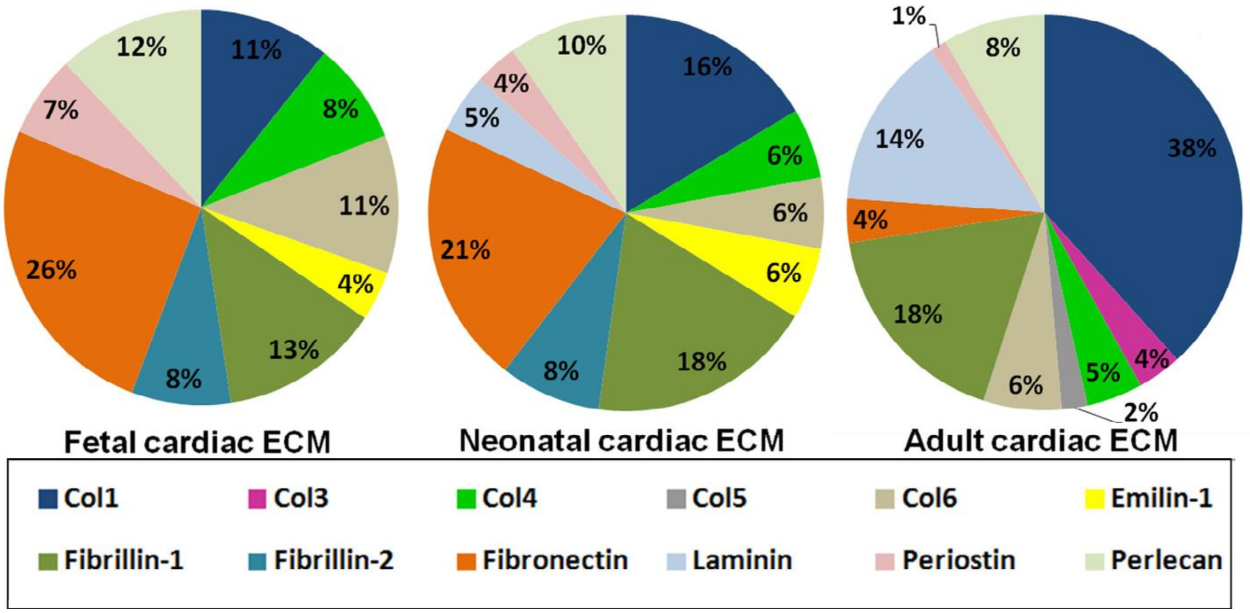

Figure 1.5. Cardiac ECM composition represented as a percentage of the 15 most abundant proteins detected by LC-MS/MS. Adapted from: Williams, C., K. P. Quinn, et al. (2014) Acta Biomater **10**(1): 194-204.

1.7 The Role of Fibronectin in Tissue Function and Development

Upregulation of fibronectin during cardiac development and remodeling indicates its importance for these processes. Although the exact mechanism in which fibronectin promotes cardiac development is still unknown, significant progress has been achieved in elucidating the structure and function of fibronectin in ECM.

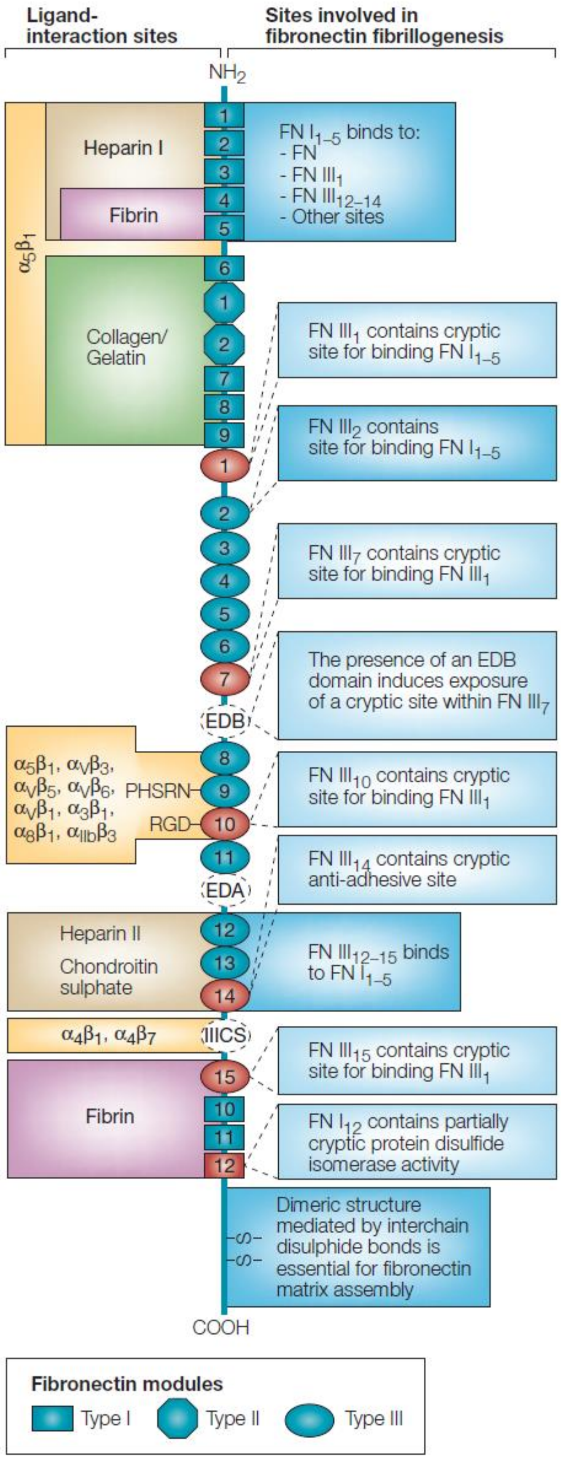
Ligand-interaction sites
Sites involved in fibronectin fibrillogenesis
NH_2
Heparin I
Fibrin
$\alpha_5\beta_1$
Collagen/ Gelatin
FN I_{1-5} binds to:
- FN
- FN III_1
- FN III_{12-14}
- Other sites
FN III_1 contains cryptic site for binding FN I_{1-5}
FN III_2 contains site for binding FN I_{1-5}
FN III_7 contains cryptic site for binding FN III_1
The presence of an EDB domain induces exposure of a cryptic site within FN III_7
EDB
$\alpha_5\beta_1$, $\alpha_V\beta_3$, $\alpha_V\beta_5$, $\alpha_V\beta_6$, $\alpha_V\beta_1$, $\alpha_3\beta_1$, $\alpha_8\beta_1$, $\alpha_{IIb}\beta_3$
PHSRN
RGD
FN III_{10} contains cryptic site for binding FN III_1
FN III_{14} contains cryptic anti-adhesive site
EDA
Heparin II
Chondroitin sulphate
FN III_{12-15} binds to FN I_{1-5}
$\alpha_4\beta_1$, $\alpha_4\beta_7$
IIICS
FN III_{15} contains cryptic site for binding FN III_1
Fibrin
FN I_{12} contains partially cryptic protein disulfide isomerase activity
–S–
–S–
Dimeric structure mediated by interchain disulphide bonds is essential for fibronectin matrix assembly
COOH
1
2
3
4
5
6
1
2
7
8
9
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
10
11
12
Fibronectin modules
Type I
Type II
Type III

Figure 1.6. Structure and function of fibronectin modules. Adapted from Geiger, B., A. Bershadsky, et al. (2001). Nat Rev Mol Cell Biol **2**(11): 793-805.

The fibronectin network structure and the process of its assemly have been extensively studied in the past. Recent advances in this area have been reviewed by Singh et al.[41] Fibronectin is primarily generated by fibroblasts, it consists of two subutins with molecular weight between 230-270 kDa depending on alternative splicing. Fibronectin subunit has modular scructure with modules organized into binding sites for collagen, integrins, heparin, fibronectin, and other molecules. After synthesis fibronectin molecules are assembled into insoluble fibrils. This process creates a fibrillar network around cells and is crucial for fibronectin function. It starts with soluble fibronectin dimer attaching to cell’s integrins, which triggers cytoskeleton remodeling leading to “unrolling” the fibronectin by cell-generated force. This, in turn, exposes fibronectin-binding sites that allow of the attachment of another fibronectin molecule. This process repeats over and over again leading to the formation of a fibronectin fibril. Although only α5β1 integrins are capable of binding soluble fibronectin and triggring fibrillogenesis, fibrillar fibronectin can attach to a significantly wider range of integrins, such as α4β1, αvβ1, αvβ3, and αIIbβ3. Interestingly, as α5β1 integrins require fibronectin's synergy site to attach, applying sufficient amount of stress to the fibronectin molecule may render it incapable of binding to this type of integrins due to the tension-induced deactivation of the synergy site. Adhesion to fibronectin can trigger various responses in the cell affecting its gene expression levels, cytoskeleton restructuring, contractility, migration, proliferation, etc.[42] Further, depending on the type of the integrins involved in the formation of focal adhesion and the mechanical environment these responses can be significantly different.[43]

Aside from being responsible for cell adhesion, fibronectin is crucial for building other ECM components, such as collagens, fibrillin, fibulin, latent TGF-β binding protein, and tenascin-C.[41] Therefore, the processes of ECM remodeling often have to start with the remodeling of fibronectin. During the embryonic heart development such remodeling occurs constantly as the heart muscle undergoes alignment, thickening, and vascularization. Particularly, fibronectin network is known to serve as a guide for the formation of coronary vasculature, as new blood vessels form along the existing fibronectin fibrils.[44] Fibronectin has also been shown to play an important role in regulating the flow of various diseases, including tumor development and metastasis.[45,46] Finally, previous studies reported that fibronectin can bind a wide range of hormones, growth factors, and other signaling molecules mediating their activity and presentation to the cells and therefore regulating cell response to them.[47-49]

1.8 Using Fibronectin Micropatterns to Drive Cardaic Tissue Formation

It has been previously shown that fibronectin, deposited onto a substrate in an anisotropic pattern, can guide cells to align along one direction.[50] Particularly, the pattern consisting of 20 µm wide lines of alternating high and low density fibronectin (20x20) has been shown to stimulate cells to form an anisotropic cardiac monolayers capable of synchronous contractions, with peak systolic stress an order of magnitude higher than that of the isotropic monolayer (fig. 1.7). Although this study wasn't aimed at mimicking the *in vivo* environment, it showed that ECM proteins can drive cell alignment, which suggests a possibility of a similar process occurring in a developing heart. In this dissertation research, we aimed to modify the pattern used in this technique to mimic the structure of native fibronectin in a developing heart**. We hypothesized that recapitulating the structure of fibronectin in a developing heart on a 2D substrate would produce a higher degree of cellular alignment, higher action potential propagation speed along the direction of alignment, and higher contractile stress of the engineered tissues compared to the 20x20 pattern**. To achieve this goal, we looked at fibronectin in chick embryonic myocardium, as it is during this developmental stage when most heart growth and organization occurs. Therefore, mimicking the cell environment in embryonic heart in vitro can potentially direct cardiomyocytes to organize in a tissue that mimics the function of native myocardium. Another reason the embryonic heart environment is preferred compared to the environment of the adult heart is that cells used in tissue engineering applications typically have an immature, embryonic-like phenotype and thus, we believe, require an embryonic-like environment.

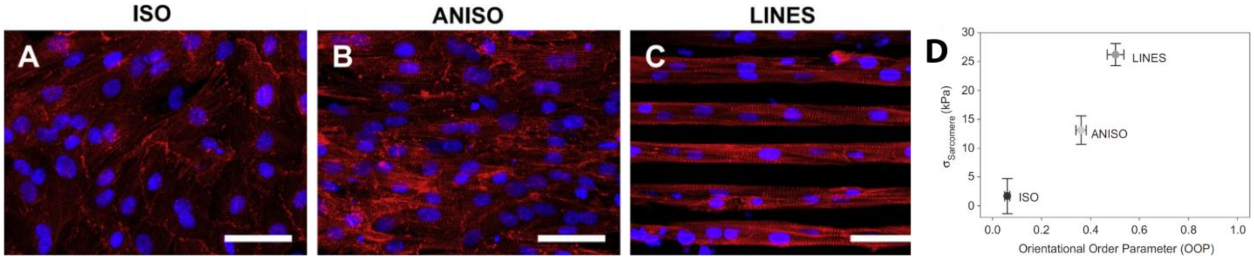
ISO
ANISO
LINES
A
B
C
D
30
25
20
15
10
5
0
$\sigma_{Sarcomere}$ (kPa)
LINES
ANISO
ISO
0.0
0.2
0.4
0.6
0.8
1.0
Orientational Order Parameter (OOP)

Figure 1.7. Cardiac tissues on fibronectin-patterned substrates. A: Isotropic cardiac tissue formed on isotropic fibronectin coating. **B:** Anisotropic cardiac tissue formed on the fibronectin pattern of 20 µm wide lines of alternating high and low density fibronectin. **C:** Aligned strands of cardiomyocytes formed on 20 µm wide fibronectin lines with 20 µm spacing. **D:** Systolic peak stress generated by cardiac tissues on fibronectin patterns as a function of the OOP of sarcomeric z-lines of cardiomyocytes. Adapted from: Feinberg, A. W., P. W. Alford, et al. (2012). Biomaterials **33**(23): 5732-5741.

Imaging fluorescently stained fibronectin in chick embryonic myocardium revealed that it has fibrillar structure with characteristic fibril diameters between 0.2 and 2 µm, which is of significantly smaller scale than the 20x20 µm pattern used in the previously described technique. In order to generate the biomimetic pattern, we took 3D confocal images of fibronectin in the chick embryonic myocardium and converted them into a combined 2D pattern with 2 µm resolution. Although this pattern has several limitations in the way in resembles the native fibronectin structure, such as lower resolution, lack of 3D structure, and lack of fibrillary structure, it is vastly different from any line pattern that was used before, and thus can provide an important insight into the way cardiomyocyte alignment occurs *in vivo*. Additionally, we engineered another pattern of 2 µm wide lines with 2 µm spacing (2x2) as another control that has the same resolution as the biomimetic pattern, but doesn’t mimic the structure of native fibronectin.

1.9 Dissertation Organization

In chapter 2 of this thesis we discuss the process of the biomimetic pattern derivation from confocal images of heart ECM and verify the protein patterning quality using fluorescently labeled fibronectin. Then we compare the alignment of isolated cardiomyocytes and cardiac monolayers on the biomimetic and line patterns. We discover and study the unique cell density-dependent response of chick cardiomyocytes to the biomimetic pattern, which allows us to determine the role of cell-cell and cell-substrate interactions in the formation of the aligned cardiac tissue.

In chapter 3 we differentiate human iPSCs and ESCs into cardiomyocytes by adapting a previously reported monolayer-based differentiation technique and then use these cardiomyocytes to engineer human cardiac tissues that are more relavant to human therapies than the chick-derived tissues. By measuring iPSC-CM alignment at low and high densities, we compared the response of these cells to the fibronectin patterns to the response of chick cardiomyocytes. Further, we developed a technique for measuring cardiac tissue contraction propagation speed using calcium imaging in a line scan mode. Additionally, we studied the influence of human CFBs and CFB-generated ECM on cardiac tissue alignment. Finally, we analyzed the effect of the tri-iodo-L-thyronine (T3) based maturation of iPSC-CMs on their ability to align on the fibronectin patterns. This study allowed us to determine the main factors responsible for the organization of iPSC-CMs into a functional tissue, which lays a foundation for developing new techniques for engineering more mature biomimetic cardiac tissues that closer resemble the structure and function of native myocardium.

In chapter 5 we aimed at overcoming one of the main limitations of our biomimetic pattern by developing a new biomimetic pattern with sub-micron resolution. To make master molds for microcontact printing at this resolution, we used a 3D nano-printing tool Nanoscribe that is capable of printing features as small as 0.2-0.4 µm and a 0.7 µm resolution photomask. Although the work on this project is still in progress, significant progress has been made in optimizing the process of mold fabrication. The results of this work will reveal the role of the finer fibronectin features in stimulating cardiac tissue alignment.

Chapter 2: Engineering chick cardiac muscle based on chick embryonic heart ECM

2.1 Abstract

Cardiac tissue engineering offers a potential approach to repairing post-infarct myocardium and provides a convenient platform for testing and developing drugs. Understanding the processes guiding cardiomyocyte organization into a functional muscle is the key to developing more advanced cardiac tissue engineering techniques. During embryonic development, the heart's ECM is one of the crucial factors guiding myocardial organization into laminar sheets of uniaxially aligned, electrically and mechanically coupled cardiomyocytes wrapped around the heart. We hypothesized that recapitulating the ECM structure of the developing myocardium *in vitro* could guide cardiomyocytes to form a tissue mimicking the structure and function of the native myocardium. To test this hypothesis, we modified a previously reported technique of controlling cardiac alignment by patterning ECM proteins on a substrate by generating a biomimetic pattern based on the structure of the embryonic heart's most abundant ECM protein, fibronectin, in order to induce the formation of an aligned functional cardiac monolayer. We found that the chick cardiomyocyte alignment on the biomimetic pattern, unlike commonly used line patterns, is cell-density-dependent, which suggests a role for cell-cell interactions in stimulating cardiac alignment. By looking at local cell-ECM interactions, and using blocking antibodies to inhibit N-cadherin-based cell-cell junctions, we identified that while cell-ECM interactions determine chick cardiomyocyte alignment at low densities, at high densities it is primarily determined by cell-cell interactions. Additionally, we identified the role of CFB impurities on chick cardiomyocyte alignment. The

results of this study suggest that cell-cell interactions rather than cell-ECM interactions drive the alignment of cardiomyocytes in developing myocardium.

2.2 Introduction

Cardiac tissue engineering aims at recapitulating the structure and function of native myocardium *in vitro*. While significant progress has been made in the past few years, current techniques are still not capable of achieving cellular organization and physiological parameters comparable to those of the native heart. Therefore, new improved techniques are needed to engineer more mature cardiac muscle. During embryonic development, cardiomyocytes organize into highly aligned contractile sheets wrapped around the heart. Thus, recapitulating the embryonic heart niche in vitro could guide cardiomyocytes to form cardiac tissue recapitulating structural and functional characteristics of native myocardium. However, as it is impossible to fully recreate the heart's environment, crucial factors responsible for organizing cells into a functional muscle need to be determined. In this chapter we looked at the structure of fibronectin in a chick embryonic heart. Fibronectin is the most abundant protein in the embryonic heart ECM[39] and it has been shown to be crucial for myocardium regeneration,[51] which makes it a promising material for building cardiac tissue.

The development of the chick embryonic heart is similar to that of the human heart, although the development occurs much faster. The stages of chick embryonic development were described by Hamburger and Hamilton in 1951.[52] Cardiac muscle is derived from splanchnic mesoderm that undergoes thickening and folding into a primitive heart tube during stages HH 9-10 corresponding to 29-38 hours of incubation. After that, the heart tube forms a loop that undergoes further morphogenesis to form 4 chambers, the myocardium thickens and cardiomyocytes align in laminar sheets wrapped around the heart (fig. 2.1).[53] During this process coronary vasculature starts developing to supply thicker myocardium with sufficient amount of oxygen and nutrition. Tomanek et al summarized the current knowledge about the process of coronary vasculature development.[44] Cells that give rise to the coronary vasculature, including endothelial cells, smooth muscle cells, and fibroblasts, originate in the epicardium. First, epicardial cells differentiatie into angioblasts via epithelial-mesenchymal transformation and migrate deep into the myocardium, where they further differentiate into endothelial cells and form vascular tubes along the fibronectin network – process known as vasculogenesis. Vascular tubes undergo further growth, fusion, and branching to form an interconnected network. Later coronary arteries are formed by the ingrowth of the capillary plexus and penetration of the aorta at the left and right cusps, while smooth muscle cells migrate towards them to form the arterial wall. Veins are formed from the vascular channels located at the sinus venosus. Further vasculature organization involving branching of the arteries and growth of new arterioles and capillaries keeps occurring throughout the embryonic and even during the early post-natal stage of development.

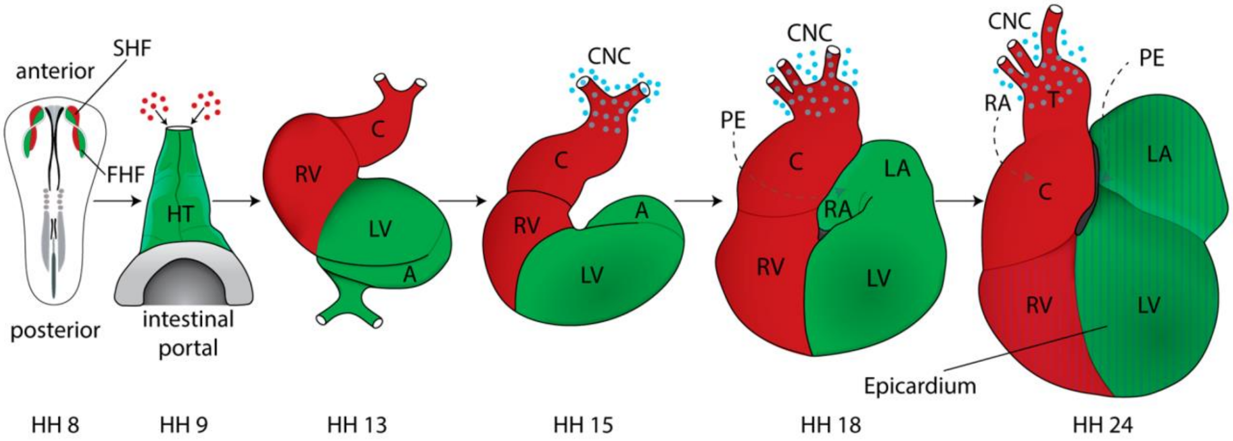

Figure 2.1. Chick embryonic heart development schematic between stages HH 9 and HH 24. At the stage HH 8 cells of splanchnic mesoderm move together to form the heart tube (HT) at the stage 9. After that heart tube starts thickening and looping to form the heart. Adapted from Wittig, J. and A. Münsterberg (2016). Journal of Cardiovascular Development and Disease **3**(2): 12.

We stained and imaged 6-day (stage HH 28-30) embryonic heart fibronectin in the aligned part of ventricular myocardium to create a template for engineering cardiac muscle in vitro. At this stage, all 4 heart chambers are formed and the myocardium contains several layers of highly aligned cardiomyocytes on the outside with less well-aligned trabeculated inner and middle parts. Trabeculated structure is important at this stage of development due to the lack of sufficient vasculature. Low vascularization makes this stage optimal for imaging myocardial fibronectin as forming blood vessels at later stages generate large amounts of high-intensity fibronectin signal within the myocardium, making it harder to filter out the fibronectin ECM corresponding to the cardiomyocytes within the cardiac muscle.

Using confocal z-stacks of fibronectin in chick myocardium, we derived a 2D biomimetic pattern with 2 µm resolution that can be microcontact printed onto a substrate. It has been previously shown that fibronectin micropatterned onto the substrate in 20 µm wide lines of alternating high and low density protein can guide the formation of an aligned contiguous monolayer capable of producing an order of magnitude higher force compared to isotropic tissues.[50] Confocal images of fibronectin in embryonic heart showed that fibronectin has fibrillar structure with fibril diameter varying between 0.2 and 2 µm, which is an order on magnitude smaller compared to the line width of the 20x20 pattern used in this technique.[54] Therefore, the biomimetic pattern we developed resembles the native fibronectin structure significantly closer despite its limitations in terms of resolution.

We hypothesized that cardiomyocytes seeded on the biomimetic pattern would form a cardiac monolayer that more closely recapitulates the structural and functional characteristics of the native myocardium, such as cell alignment, action potential propagation speed, and contractile force, compared to the tissues grown on the 20x20 pattern. Additionally, we introduced a pattern consisting of 2 µm wide lines with 2 µm spacing (2x2), which served as another control pattern that has the same resolution as the biomimetic pattern, but does not resemble the structure of the native fibronectin. To test our hypothesis, we compared the response of chick embryonic cardiomyocytes to the biomimetic pattern and control line patterns, which revealed that despite significant differences between the patterns, they produce cardiac monolayers with similar cardiomyocyte alignment. However, by seeding cardiomyocytes on the patterns at low density, we observed the unique density dependent effect of the biomimetic pattern, but not the line patterns, on cell alignment. To explain this effect, we looked at local effects of the biomimetic and line patterns on cardiomyocyte attachment, orientation, and alignment. This allowed us to determine the role of cell-cell and cell-ECM interactions on cardiomyocyte alignment as a function of cell density. We later confirmed our findings by determining the effect of inhibiting N-cadherin-based cell-cell interactions on cardiomyocyte alignment.

2.3 Materials and Methods

2.3.1 Fabrication of PDMS Stamps

Poly-(dimethylsiloxane) (PDMS) stamps with three different micropatterns (20x20, 2x2, and biomimetic) were used to pattern fibronectin onto substrates for cells. The process of PDMS stamp fabrication is based on the previously described technique (fig. 2.2(i)-(iv)).[55] Briefly, a glass wafer was spin-coated with the photoresist SPR 220.3 at 5000 RPM, then put tightly under the photomask containing the desired pattern on a horizontal surface, and exposed to UV-light through the mask. The exposed parts of the photoresist were washed away using developer MF-319, the cover glass was washed in distilled water and dried with a nitrogen gun. To decrease photoresist adhesion to PDMS, it was silanized by incubating it next to an open container of 2% dimethyldichlorosilane solution (PlusOne Repel-Silane ES, GE Healthcare) for 24 hours in a vacuum desiccator. Then PDMS (Sylgard 184 base mixed 10:1 with the curing agent) was cast on top of the mold, cured at 65 ^{O}C for 24 hours, and stamps were cut out of the PDMS layer.

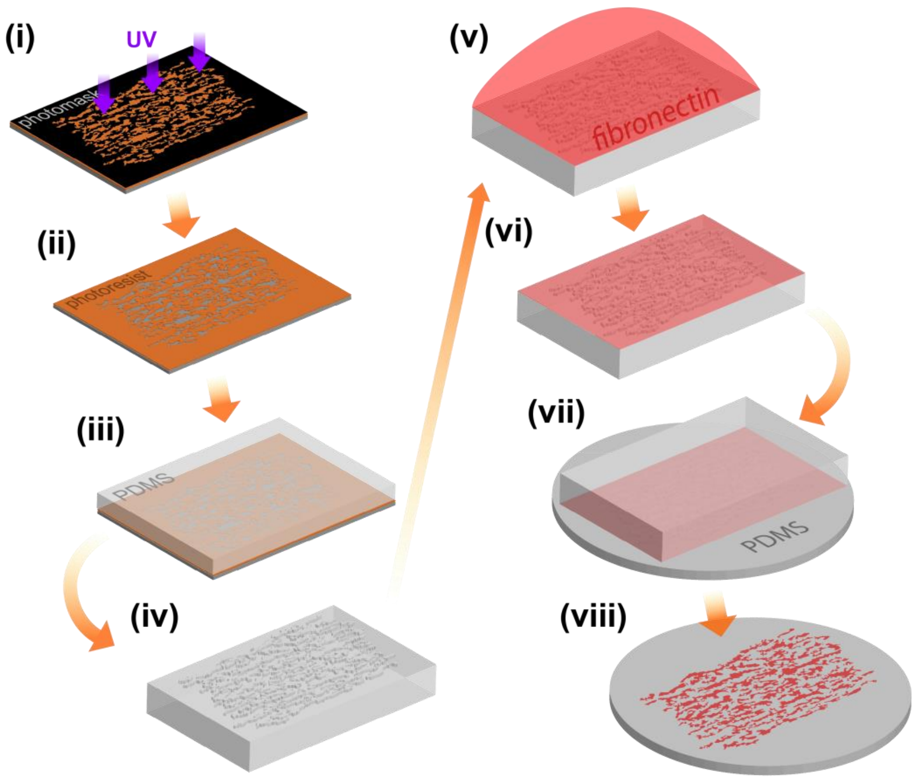

Figure 2.2. Schematic of PDMS stamp fabrication via photolithography and microcontact printing. **(i)** – a layer of photoresist is exposed to UV light through a photomask; **(ii)** – master mold made after the UV-exposed parts of photoresist are washed off; **(iii)** – PDMS is cast and cured on top of the master mold; **(iv)** – PDMS stamp is peeled off the mold; **(v)** PDMS stamp is coated with fibronectin solution; **(vi)** – PDMS stamp with a layer of adsorbed fibronectin; **(vii)** – PDMS stamp is brought in contact with the UV-ozone pre-treated substrate; **(viii)** – substrate with the printed fibronectin pattern.

2.3.2 Substrate Preparation

Fibronectin-patterned PDMS-coated glass coverslips were used as substrates for cells. PDMS-coated coverslips were prepared using Sylgard 184 silicone elastomer according to the previously described procedure.[56] Briefly, Sylgard 184 base and curing agent were mixed at the mass ratio of 10:1 followed by the mixture and defoaming in a “Thinky conditioning mixer”. Spin-coating at 4000 RPM (Table 1) was used to coat coverslips with a thin PDMS layer. After spin-coating, coverslips were put in an oven at 65 ^{0}C for 24 hours in order to cure PDMS.

Table 1. PDMS spin-coating procedure

Step	Action
1	Accelerate to 500 RPM in 5 seconds, dwell for 5 seconds
2	Accelerate to 1000 RPM in 5 seconds, dwell for 5 seconds
3	Accelerate to 3000 RPM in 10 seconds, dwell for 10 seconds
4	Accelerate to 4000 RPM in 10 seconds, dwell for 60 seconds
5	Decelerate to 2000 RPM in 10 seconds, dwell for 15 seconds
6	Decelerate to 1000 RPM in 10 seconds, dwell for 10 seconds
7	Decelerate to 500 RPM in 5 seconds, dwell for 5 seconds

Fibronectin patterns were microcontact printed onto the PDMS-coated coverslips according to a previously described technique with minor modifications (fig. 2.2(v)-(vii)).[57] Briefly, PDMS stamps were cleaned by sonication in 50% ethanol for 45-60 minutes and dried using pressurized nitrogen. Then the patterned side of each stamp was incubated in 50 µg/mL solution of human plasma fibronectin (unlabeled or labeled with Alexa Fluor 546 Maleimide fluorophore) for 60 minutes, washed in sterile water, and dried with a nitrogen gun. To transfer fibronectin from the stamp onto the coverslip, the coverslip was UV-Ozone treated for 15 minutes, and then the patterned side of the stamp was brought in contact with the coverslip for 5 minutes. The pattern transfer was verified for each stamp by confocal microscopy using fluorescently labeled fibronectin. For the 20x20 pattern an additional low-density coating was applied after patterning by incubating the substrates in 5 µg/mL fibronectin solution for 20 minutes. This was done to improve the spreading of chick cardiomyocytes on this pattern so that the confluent monolayer can be formed. The 2x2 and the biomimetic patterns were incubated in 1% w/v solution of Pluronic F127 to reduce cell adhesion in between fibronectin features and improve cardiomyocyte alignment.

2.3.3 Embryonic Chick Cardiomyocyte Isolation

All chick cardiomyocytes used in experiments were isolated from 8 day old chick embryos as previously described with minor modifications.[58] First, the eggshells were opened, embryos removed, hearts cut out of the embryos and the atria removed leaving only ventricles. Then each ventricle was cut in 10-20 pieces and incubated in 1X TrypLE Express (Thermo Fisher) solution for 7 minutes at 37 ^{0}C. The supernatant was then removed and mixed with seeding medium (M199, 1% penicillin/streptomycin, 10% heat inactivated fetal bovine serum (HI-FBS)) for enzyme deactivation. New TrypLE Express solution was added to the minced hearts and the same procedure was repeated 4-6 times. After that, cell solution was centrifuged, resuspended in seeding medium, and pre-plated in T75 flasks 2 times for 45 minutes to remove fibroblasts from the solution. After pre-plating, cells were centrifuged and resuspended in seeding medium, cell density was counted and cells were seeded onto the substrates fabricated above. After seeding, cells were kept in seeding medium for 24 hours, then the medium was changed to maintenance medium (M199, 1% penicillin/streptomycin, 2% HI-FBS) in order to slow down fibroblast proliferation.

2.3.4 Fixation, Staining, and Fluorescent Microscopy

Cells were fixed and permeabilized with 4% formaldehyde and 0.1% Triton-X 100, and stained with 0.5% DAPI (to stain nuclei), 1.5% phalloidin (to visualize the actin cytoskeleton), and 0.5% of corresponding primary antibodies – for other proteins, such as α-actinin (Sigma A7811), N-cadherin (Sigma C2542), and fibronectin (Sigma F3648). Samples were incubated with the dyes and primary antibodies for 60 minutes, washed in 1X PBS, incubated in the solution of secondary antibodies corresponding to the primary antibodies for 60 minutes, and washed in PBS again. After that coverslips were mounted for imaging onto glass slides with

ProLong Gold Anti-Fade preservative. Confocal laser scanning microscopy (LSM Zeiss 700) was used to obtain fluorescent images.

2.3.5 Cell Alignment Analysis

Cell actin alignment was measured using a custom MATLAB script based on local actin filament orientation analysis (Appendix 1). First, confocal images of cardiac tissues (fig. 2.3 A) stained for nuclei (fig. 2.3B), actin (fig. 2.3C), and α-actinin (fig. 2.3D) were taken. Local orientations of actin filaments were detected (fig. 2.3E), and based on the provided threshold (varying from 0 to 1) associated with the filament prominence, a filament mask was created determining which orientations to take into analysis (fig. 2.3F). Actin and α-actinin channels were processed to produce binary masks of cell location (fig. 2.3G) and cardiomyocyte location (fig. 2.3H), respectively. Using combinations of these masks, each orientation was assigned to a cell type (fig. 2.3I). Angular distribution of actin filaments (fig. 2.3J) was then used for each cell type to calculate the orientational order parameter (OOP) – a measure of alignment varying between 0 and 1, where 0 corresponds to isotropic distribution and 1 – to perfectly co-aligned filaments (fig. 2.3K). Heat maps and histograms were created by detecting location of the actin filaments relative to the fluorescently labeled fibronectin pattern and calculating alignment data for each location separately. Actin and α-actinin-based masks were also used to determine the cardiomyocyte purity and the overall cell surface coverage.

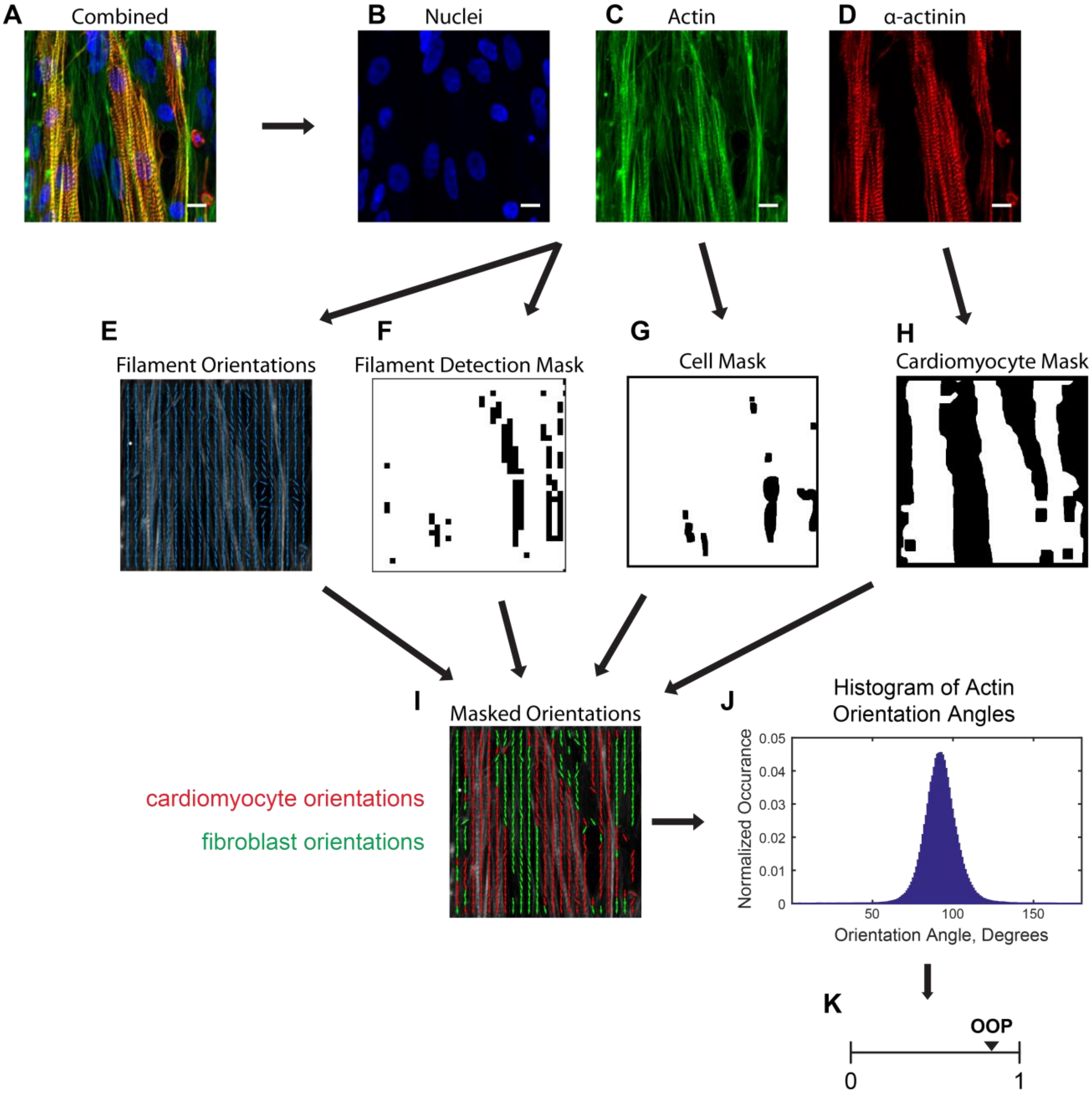

Figure 2.3. Schematic of actin filament alignment analysis. A-D: Cardiac tissue stained for nuclei, actin, and α-actinin. **E:** Example of orientation detection of actin filaments represented by blue arrows. **F:** Mask based on the minimal gradient of brightness across the filament. **G:** Mask based on actin channel indicates the location of cells and is used to calculate cell surface coverage. **H:** Mask based on the α-actinin channel indicates the location of cardiomyocytes. **I:** Filtered orientation map removes orientations maksed out by the mask **(F)**, and assignes cell type to the rest of orientations based on the masks **(G**) and **(H**). **J:** histogram of cardiomyocyte actin orientation angles that is used to calculate OOP **(K)**.

2.4 Results

2.4.1 Generation of the Biomimetic Pattern

In order to engineer a pattern that mimics the structure of fibronectin in an embryonic heart, we stained 6-day embryonic chick hearts for fibronectin (fig. 2.4A). 6-day embryos were chosen because at this stage aligned myocardium can already be observed, while coronary vasculature characterized by high fibronectin expression around forming blood vessels is still mostly undeveloped. The majority of myocardium at that stage is trabeculated with only a few aligned layers of cardiomyocytes on the outer, epicardial side. We used confocal microscopy to obtain 3D z-stacks of these aligned areas (fig. 2.4B). The acquired z-stacks were processed in Imaris software to remove fibronectin corresponding to forming blood vessels and the adjacent epicardium (fig. 2.4C). Additionally, in order to meet the photomask resolution limit, we removed features smaller than ~1 µm in size using Imaris's 3D-segmentation tool (fig. 2.4D), then created maximum intensity projections of the images along z-axis and converted them into a binary format by applying a threshold (fig. 2.4E). After that the images were further processed to satisfy the photomask resolution limit of 2 µm and fix fibronectin alignment deviations. Due to the small size of the 6-day old chick embryonic heart (~2-3 mm in diameter), its natural curvature doesn't allow imaging large areas of aligned myocardium. Therefore, in order to generate a larger pattern, we had to manually combine fibronectin features from four separate images into one bigger pattern (fig. 2.4F). This combined pattern was designed in such a way that if replicated into a 2D array, the borders of the adjacent repeats would match with each other. This allowed us to create a larger scale 5x5 mm array that can be used to grow cardiac tissues (fig. 2.4G). This final array was printed onto a photomask along with the 20x20 and the 2x2 patterns and later used to produce PDMS stamps for microcontact printing.

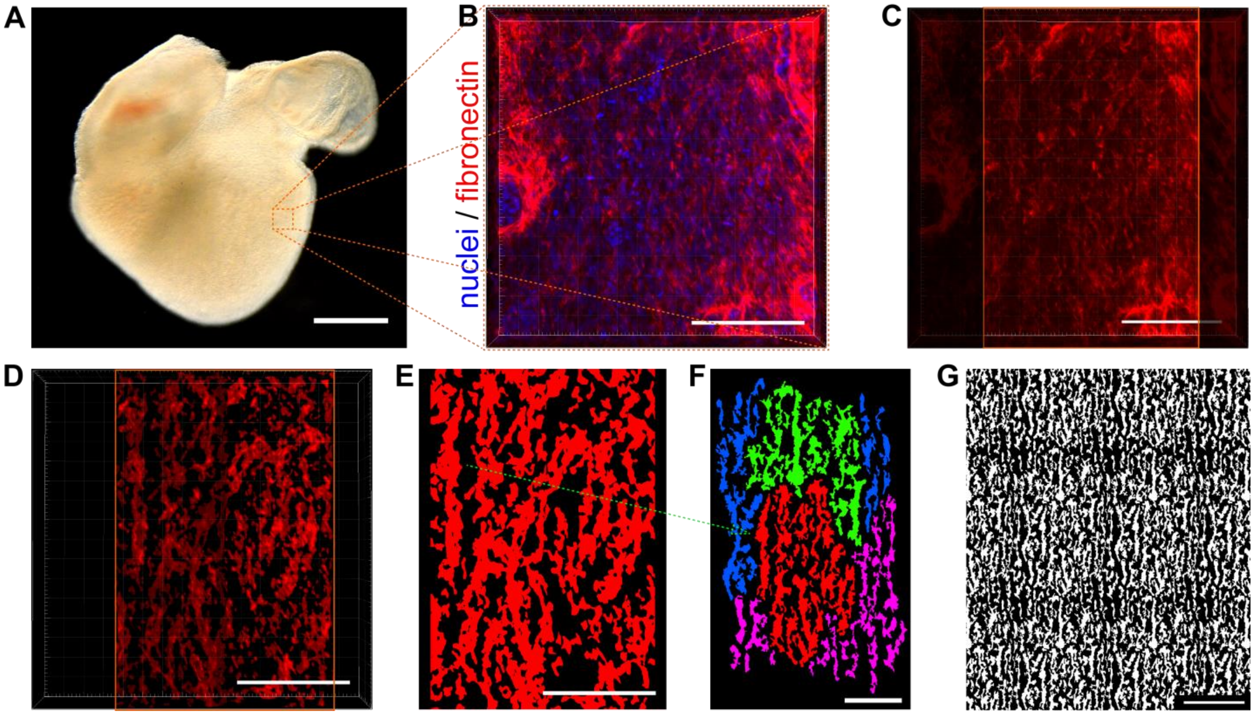

Figure 2.4. Schematic of the biomimetic pattern derivation. A: Embryonic chicken heart. **B:** Representative confocal 3D image of embryonic chick myocardium stained for nuclei and fibronectin. **C:** Fibronectin in chick myocardium. The darkened areas correspond to forming blood vessels. The highlighted area was used for the pattern generation. **D:** Filtered image of fibronectin in chick myocardium. Any feature smaller than 1 µm was removed. **E:** Chick heart fibronectin projected along z-axis and converted into a binary image. **F:** Final pattern combined from 4 images. Each color represents a separate image it came from. Green arrow shows an example of a feature transferred from image E. **G:** 2D-array of the combined pattern shown in F that was printed onto a photomask. Scale bars: A: 500 µm, B-F: 50 µm, G: 100 µm.

The quality of fibronectin pattern transfer during microcontact printing was tested for each PDMS stamp using fluorescently labeled fibronectin and confocal microscopy to ensure reliable patterning across the whole area of the stamp. During the optimization of the master mold fabrication process, we determined that ensuring good contact between photoresist and the mask and precisely controlling the time of the development are the most critical factors for producing high quality molds. Gaps between the photomask and the photoresist layer can cause resolution loss (fig. 2.5A) and overdevelopment of the UV-exposed photoresist can cause its detachment from the substrate (fig. 2.5B). By optimizing fabrication parameters, we achieved consistent transfer of fibronectin for all three patterns without loss of resolution (fig. 2.6).

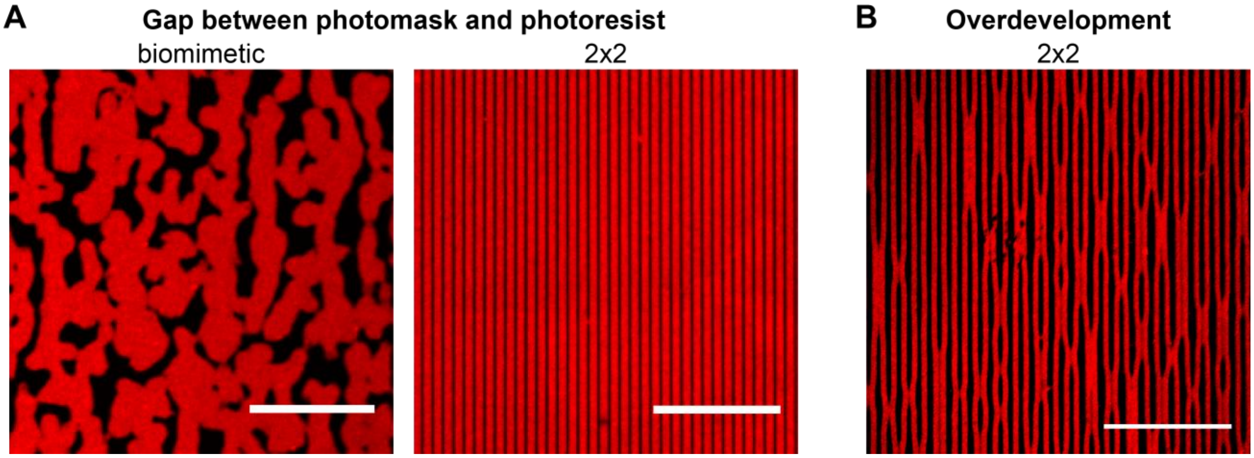

Figure 2.5. Common defects observed during the optimization of the master mold fabrication. A: Gap between photoresist and photomask during UV-exposure causes blurring of the features for the biomimetic pattern and widening of the lines in the 2x2 pattern. **B:** Too long incubation in the developer causes photoresist to detach from the wafer making the lines merge together.

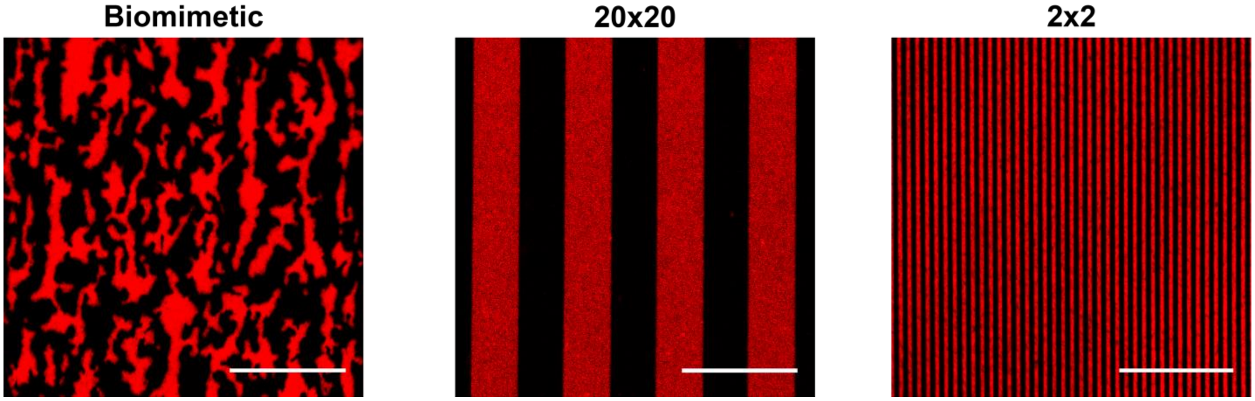

Figure 2.6. Microcontact printed fluorescently labeled fibronectin on PDMS-coated coverslips confirms the fidelity of the protein transfer. Scale bars: 50 µm.

Before testing cardiomyocyte response to the patterns, it is important to quantify the difference in structure between them. Fibronectin OOP can serve as a convenient measure of pattern alignment. For the line patterns OOP always equals 1 as all lines are strictly parallel to each other. To quantify the OOP of the biomimetic pattern (fig. 2.7A), it had to be converted to a ridge-like stucture. Firstly, we measured the OOP of the pattern's edge (fig. 2.7B) and found it to be relatively low (0.38) due to the complex structure of the shapes comprising the pattern. Then we skeletonized the pattern unsing a MATLAB script (fig. 2.7C), which resulted in OOP = 0.35. This low OOP was mostly due to the artifacts of the resulting skeleton that was heavily branched. To remove the influence of there artifacts, we used a MALTAB sctript to remove any side braches in each shape, resulting in a significantly "cleaner" skeleton with the OOP = 0.54 (fig. 2.7D). This OOP represents the overall anisotropy of fibronectin shapes in the biomimetic pattern, while the OOP of the pattern's edge serves as an indication of the heterogeneity of the pattern at the micrometer scale.

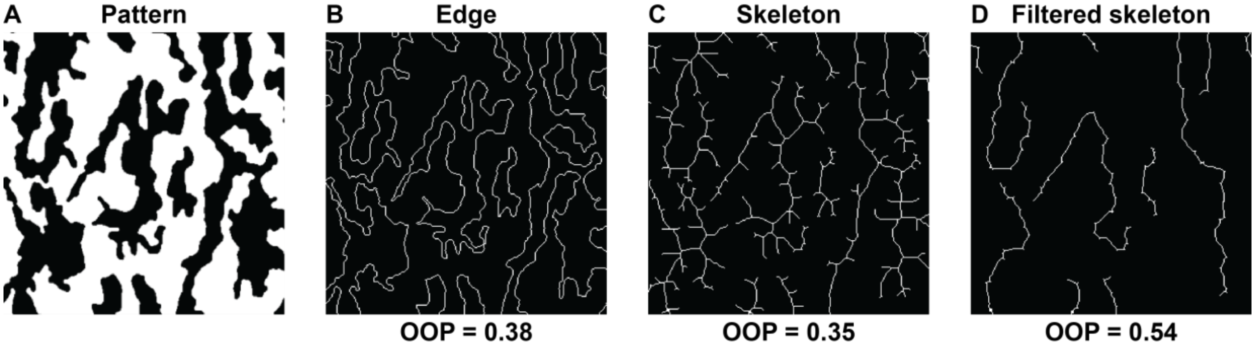

Figure 2.7. Alignment analysis of the biomimetic pattern. A: Cropped area of the biomimetic pattern. **B:** Biomimetic pattern's edge alignment analysis showed OOP = 0.38. **C**: Skeletonized biomimetic pattern's alignment analysis showed OOP = 0.35. **D:** Skeletonized pattern was filtered in MATLAB to remove all side branches, which resulted in OOP = 0.54.

Another important charachetistic of our fibronectin patterns is the density of the fibronectin/no fibronectin interface. It's been shown that cells tend to form most focal adhesions along such interfaces, making them important for defining cell shape and alignment. The interface density (ρ_{int}) for the 20x20 and the 2x2 patterns can be derived empirically using simple calculations. We calculated the ρ_{int} = 50 mm^{-1}for the 20x20 pattern and ρ_{int} = 500 mm^{-1} for the 2x2 pattern. To calculate ρ_{int} for the biomimetic pattern, we used a MATLAB script, which resulted in ρ_{int} = 398 mm^{-1}. This makes the biomimetic interface density to be intermediate between the 20x20 and the 2x2 patterns.

Confocal z-stacks of fibronectin in chick myocardium were also analyzed for alignment. We found that the measured alignment depends on the quality of fibronectin staining and varies between 0.32 and 0.39, which is similar to the alignment of our biomimetic pattern's edge and its unfiltered skeleton. The actin alignment of cardiomyocytes in chick myocardium could not be reliably analyzed due to the insufficient quality of the actin staining resulting in lower measured alignment (OOP = 0.43) than can be expected from the observed myocardium structure. Higher quality stainning and imaging are required to more reliably detect cardiomyocyte alignment in chick myocardium.

2.4.2 Chick Cardiomyocyte Alignment on Fibronectin Patterns

In order to determine cardiomyocyte response to the fibronectin patterns, chick cardiomyocytes harvested from 8-day embryos were seeded onto the patterns at low (60,000 cells/cm^2) and high (450,000 cells/cm^2) densities. The low density was chosen to achieve a culture of isolated cells that have minimal interaction with each other and the high density was chosen to generate a 100% confluent monolayer. During optimization of the seeding density for the confluent monolayer we found that if the seeding density is slightly lower than 450,000 cells/cm^2, confluent monolayers can still form, but the cardiomyocyte purity would be lower due to the larger initial unoccupied area for fibroblasts to spread into. If seeding density is much higher than 450,000 cells/cm^2, the resulting monolayers become more isotropic due to the lack of space for cells to spread onto the micropatterned surface. After 3 days of culture cells were fixed, stained for nuclei, actin, and α-actinin, and imaged using confocal microscopy (fig. 2.8). The acquired confocal images were analyzed using a custom MATLAB script to measure the alignment of actin filaments by calculating OOP (fig. 2.9). Results showed that all patterns, despite being significantly different from each other, stimulate the formation of highly aligned monolayers with similar OOP values. The high density alignment was measured for 15 to 18 tissues per pattern, which came from 5 separate batches of embryos to account for variation between the batches. This allowed us to detect statistically significant differences between cardiac alignments on all three patterns despite the fact that the patterns showed similar OOPs. At low density, cells on the biomimetic pattern showed significantly lower alignment, while for the line patterns the OOP values stayed the same as in the high-density case. Due to larger differences between OOP values for the biomimetic and the line patterns at low cell density, 6 tissues per pattern was sufficient to detect statistically significant differences.

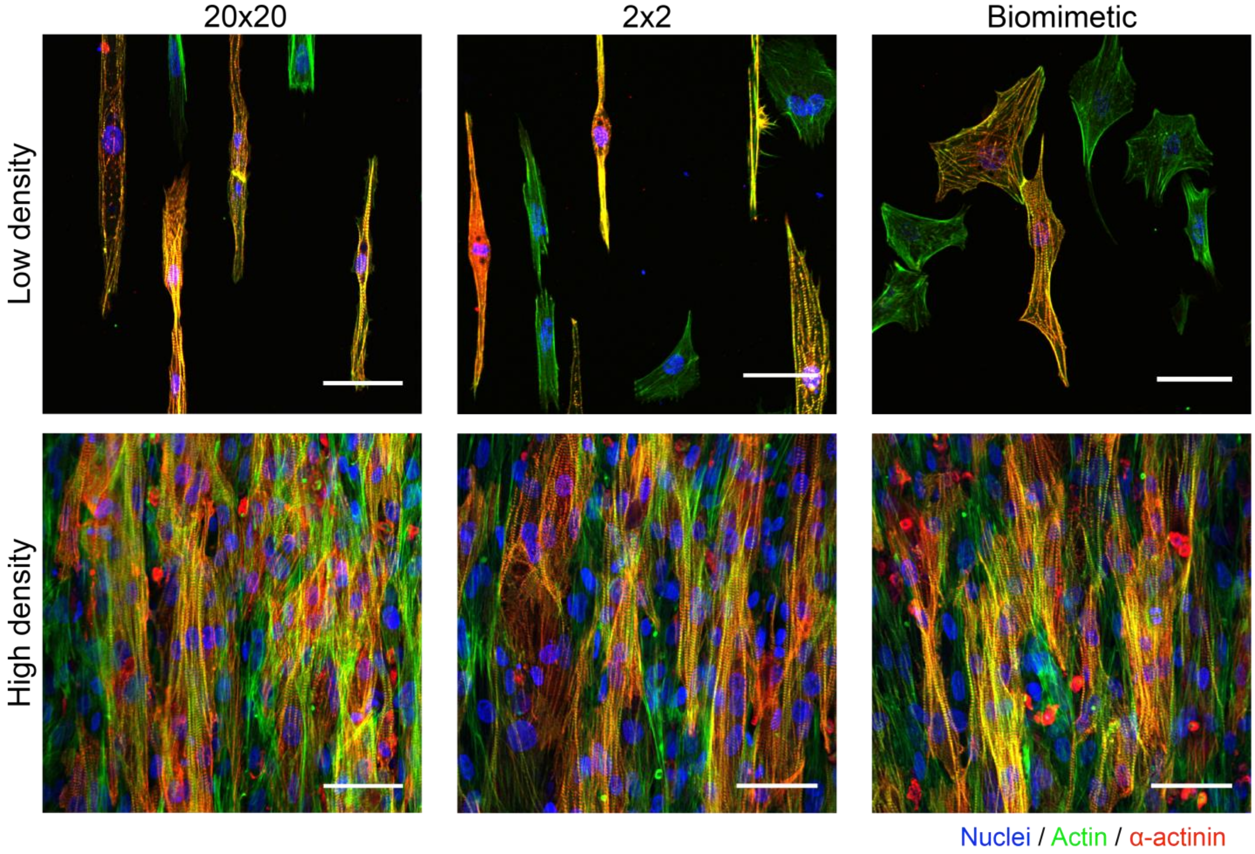

Figure 2.8. Representative confocal images of chick cardiomyocytes on fibronectin patterns at low (isolated cells) and high (monolayer) densities. While cardiomycoytes at low density show pattern-dependent alignment, at high density tissue alignment on different patterns is similar. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 50 µm.

In order to better understand how tissue alignment induced by the biomimetic pattern depends on cell density, we measured cardiomyocyte OOP as a function of cell density. To do so, we seeded chick cardiomyocytes on the biomimetic pattern at various densities ranging from 60,000 cells/cm^2 to 450,000 cells/cm^2, obtained and analyzed confocal images of stained cells for actin OOP as described previously (fig. 2.10). Additionally, for each image we measured nuclei density and the fraction of surface area occupied by cells using the actin channel. This was done to account for cell death and fibroblast proliferation during culture that affects the final amount of cells on the substrate. We found the relationship between these two parameters to be linear until cells reach 100% confluence (the occupied area fraction expressed in percentage points) around 900 nuclei/mm^2, after which the increase of nuclei density doesn't have an effect on the occupied area (fig. 2.11A). This suggests that the average contact area between a cell and the substrate stays constant until the confluence reaches 100%, after which it starts decreasing. The graph of average cell area as a function of nuclei density supports this conclusion (fig. 2.11D). Alignment analysis showed linear increase of OOP with the occupied area fraction (fig. 2.11B) and nuclei density (fig. 2.11C) until the saturation density of ~900 nuclei/mm^2, after which OOP remains constant. This unique density-dependent change of alignment observed on the biomimetic pattern suggests that cell-cell interactions influence tissue alignment at higher densities. Further, the similarity of OOP values for all three patterns at high density indicates that cell-substrate interactions are not a dominant force that stimulates alignment in cardiac monolayers. However, further evidence is needed to support these conclusions and elucidate the mechanism behind the differences of cell response between the biomimetic and the line fibronectin patterns.

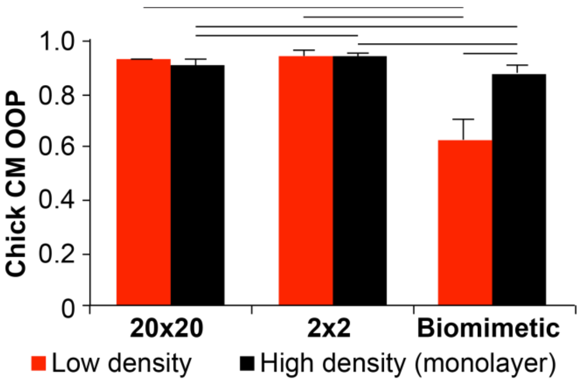

Figure 2.9. Chick cardiomyocyte OOP analysis at low and high cell densities (N ≥ 15) shows that at high density cardiomyocytes achieve high OOPs on all patterns despite of significant differences in pattern structure. Additionally, density-dependent alignment on the biomimetic pattern, but not the line patterns, was observed, indicating the role of cell-cell interactions in stimulating cardiomyocyte alignment

at high density. Horizontal lines indicate statistically significant difference ($p<0.05$, two-way ANOVA, N ≥ 6). Error bars represent standard deviations.

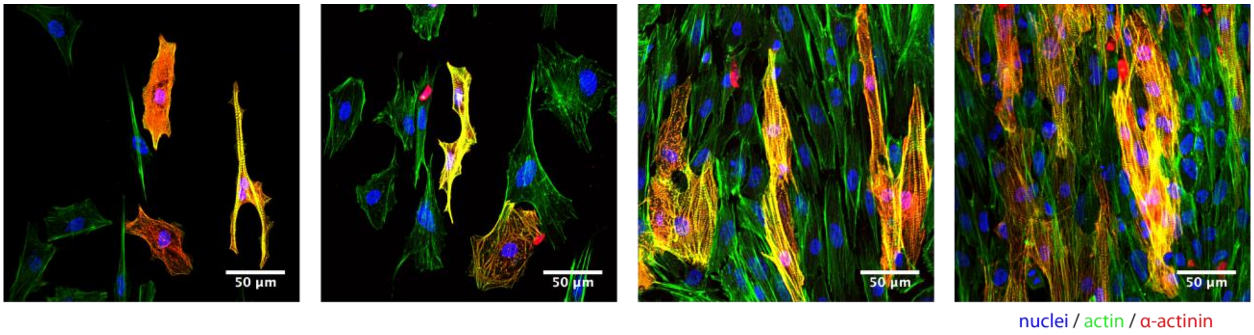

Figure 2.10. Representative images of chick cardiomyocytes on the biomimetic pattern at various cell densities starting at low-density isolated culture to a high-density confluent monolayer. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 50 µm.

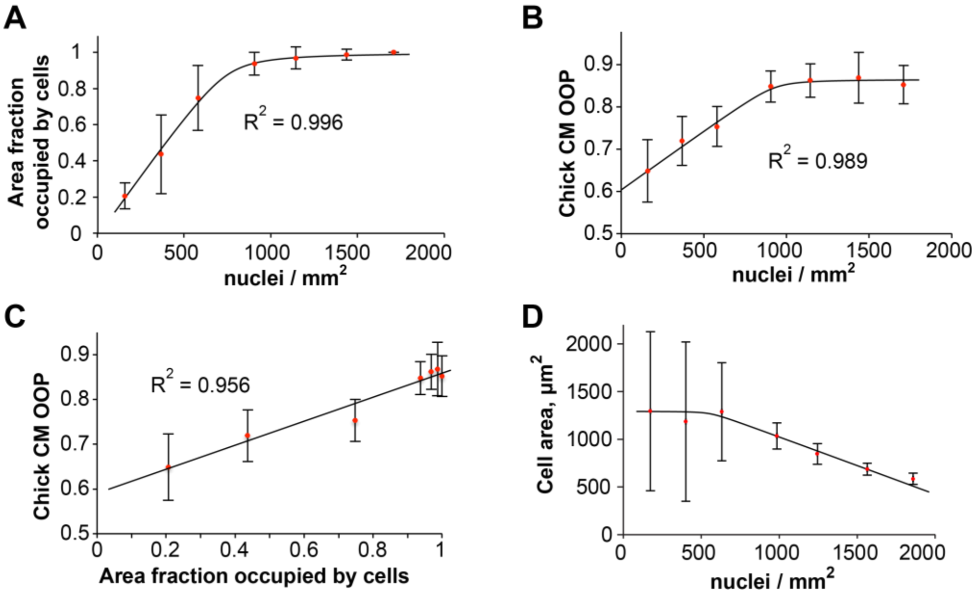

Figure 2.11. Chick cardiomyocyte alignment on the biomimetic pattern as a function of cell density. A: Surface area fraction occupied by cells as a function of nuclei density shows linear behavior below 900 nuclei/mm^2 reaching saturation after 100% confluency is achieved. This indicates that cells tend to maintain constant contact area with the substrate in a sub-confluent culture. **B-C:** Chick cardiomyocyte OOP as a function of nuclei density (**B**) and occupied area fraction (**C**) show linear increase of OOP with cell density in sub-confluent cultures. When cells reach 100% confluence, OOP becomes independent of nuclei density. **D:** Average cell area as a function of nuclei density. Error bars represent standard deviations.

2.4.1 Single Cell Analysis

Analyzing actin OOP as a function of cell density gave us an important insight into the mechanism of tissue alignment on fibronectin patterns. However, this technique does not explain the behavior of individual cells as no distinction is made between actin filaments from different cells. While it is extremely difficult to distinguish between cells in a high density culture, in a low density culture the majority of cells are either isolated from one another or interact with only a small amount of other cells. This allows us to develop a MATLAB script that detects individual cells based on the actin channel, determines cell type using the α-actinin channel, and measures several characteristics of single cells, such as surface contact area, aspect ratio, and ellipticity, on fibronectin patterns (Appendix 2). Aspect ratio of a cell was defined as an aspect ratio of main axes of an ellipse that has the same surface area and second moments as the cell in an image. Ellipticity was calculated based on the axes lengths of the same ellipse. Using this script, we found that average cardiomyocyte area is larger than average fibroblast area on all three patterns, as well as the uniformly coated substrate (fig. 2.12A). Additionally, cardiomyocyte area was found to be dependent on the pattern as well, although further studies with larger sample size are required to determine whether this effect is statistically significant. Analyzing cell shape characteristics like aspect ratio and ellipticity, we showed that although they generally correlate with the OOP values for the 20x20 and 2x2 patterns, cells on the biomimetic pattern have a similar shape to that on the uniformly coated substrates (fig. 2.12 B, C). However, cardiomyocyte OOP on the biomimetic pattern is significantly higher than that on the isotropic pattern, which indicates that on the biomimetic pattern cells are co-oriented, while on the isotropic pattern their orientations are random.

While studying characteristics of isolated cells helps characterize the influence of cell-substrate interactions on cell shape and alignment, analyzing small clusters of touching cells in a low-density culture can serve as an intermediate model between isolated cells and a high-density monolayer, as these touching cells have are influenced by both cell-cell and cell-substrate interactions. To do so, we created another MATLAB script that detects cell-cell interactions and sorts cells based on the type of the cells participating in such interactions (Appendix 3). However, only interactions between cell clusters comprised of two cells were analyzed while ignoring any larger clusters due to the increasing difficulty of their characterization. By applying this analysis on the 2x2 pattern, we found that isolated fibroblasts and fibroblast-fibroblast couples constitute the majority of the cells in a low-density culture (fig. 2.12D). This can be attributed to the combination of fibroblast proliferation and high unoccupied substrate area that allows fibroblasts to spread and proliferate faster. Further, cell aspect ratio analysis revealed that while cardiomyocyte-fibroblast interactions don't influence cardiomyocyte aspect ratio, cardiomyocyte-cardiomyocyte interactions result in the decrease of aspect ratio (fig. 2.12E). This effect is particularly interesting as for the 2x2 pattern in the high-density monolayer cardiomyocyte-cardiomyocyte interactions are plentiful, but tissue alignment is similar to the low density case, which means that at higher densities the observed decrease of alignment is negated. Finally, analyzing the effect of cell-cell interactions on cell area, we found that cardiomyocyte-fibroblast interactions tend to slightly increase cardiomyocyte area, while cardiomyocyte-cardiomyocyte interactions decrease it (fig. 2.12F). However, further research is required to determine whether these differences are statistically significant and what mechanism is responsible for this behavior.

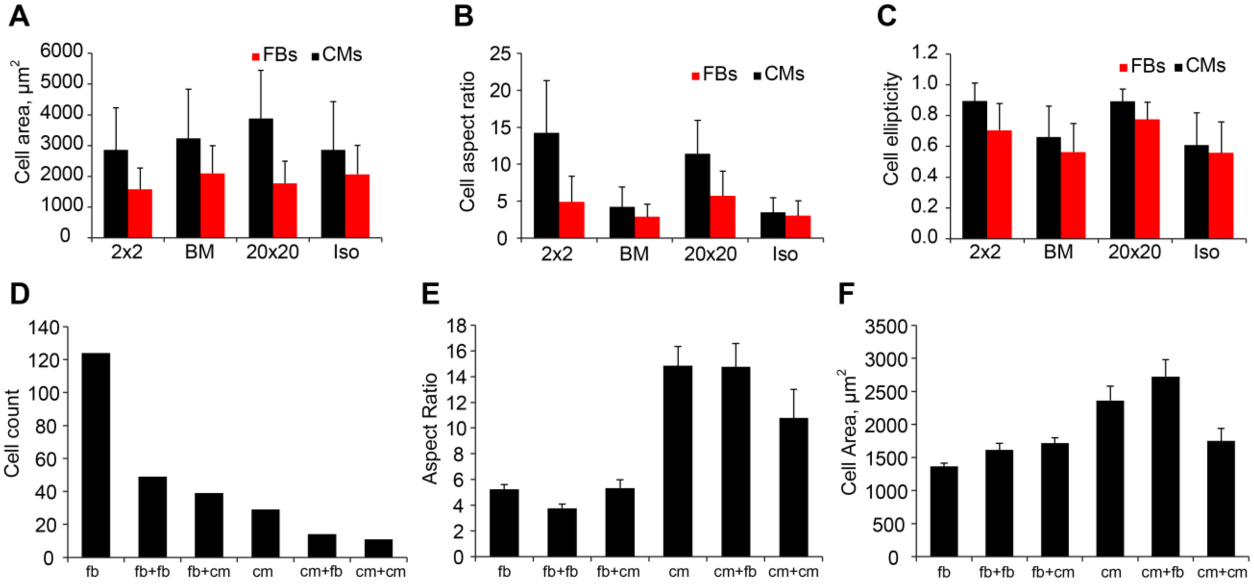

Figure 2.12. Single cell analysis in a low density culture on fibronectin patterns. A: Average cell area. Cardiomyocytes were found to have a higher surface area than fibroblasts. **B:** Average cell aspect ratio. Interestingly, cells on the biomimietic pattern show similar aspect ratio as the isotropic pattern, although OOP on the biomimetic pattern is significantly higher than OOP on the isotropic substrate. **C:** Cell ellipticity on fibronectin patterns. **D-F** shows analysis of single and interacting cells for the 2x2 pattern. **D:** Analysis of isolated and interacting cell count on the 2x2 pattern. Isolated fibroblasts were found to be the most abundant cell type in a low-density culture due to their proliferation. **E:** Aspect ratio of isolated and interacting cells on the 2x2 pattern. **F:** Average area of isolated and interacting cells.

2.4.2 Role of Fibroblasts in Chick Cardiac Tissue Alignment

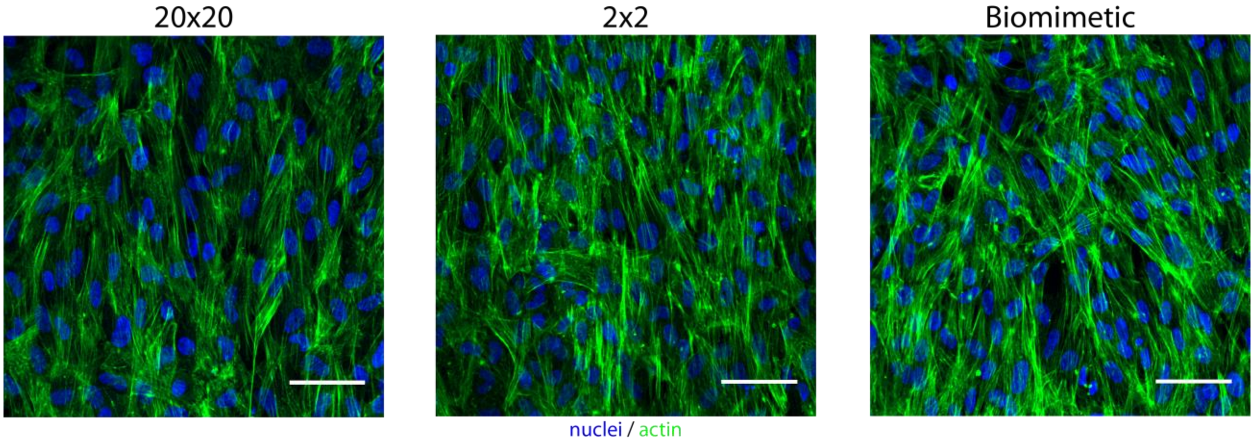

Figure 2.13. Representative immunostained images of chick fibroblasts on fibronectin patterns show lower alignment compared to chick cardiomyocytes. Nuclei are shown in blue, actin – green. Scale bars: 50 µm.

Another important factor that may potentially influence the alignment of cardiac monolayers is impurities in cardiomyocyte population. Although the cardiomyocyte harvesting protocol includes purification via pre-plating, the final cell population contains a small amount of non-cardiomyocytes primarily consisting of fibroblasts with small amounts of endothelial and smooth muscle cells. These non-cardiomyocytes, especially fibroblasts, proliferate over time, lowering cardiomyocyte purity and potentially influencing OOP. Using α-actinin staining to detect cardiomyocytes, we were able to filter them out and analyze only non-cardiomyocytes in cardiac tissues. Additionally, to measure the intrinsic response of fibroblasts to the fibronectin patterns, we engineered cell monolayers with the fibroblast-rich population that is normally discarded during cardiomyocyte purification on fibronectin patterns (fig. 2.13). We found that while the alignment of a fibroblast monolayer is lower than that of a cardiac-rich monolayer, in a cardiac-rich monolayer the alignment of both fibroblasts and cardiomyocytes is equal (fig. 2.14A). This effect can be attributed to the large amount of cell-cell interactions between cardiomyocytes and fibroblasts in a cell-dense monolayer. In contrast, in a low-density culture there is almost no interaction between cells, and therefore cardiomyocyte and fibroblast alignments become decoupled. Our analysis showed that cardiomyocyte alignment in this case is higher than that of the fibroblasts (fig. 2.14B). Additionally, by analyzing fibroblast alignment in cardiomyocyte-rich cultures at various cell densities, we found that fibroblast OOP stays lower than the cardiomyocyte OOP in sub-confluent cultures, while converging to the cardiomyocyte OOP when cell confluence approaches 100% (fig. 2.14C). These findings indicate an overall lower tendency of non-cardiomyocytes to align on fibronectin patterns, which can decrease the cardiomyocyte OOP in high density monolayers.

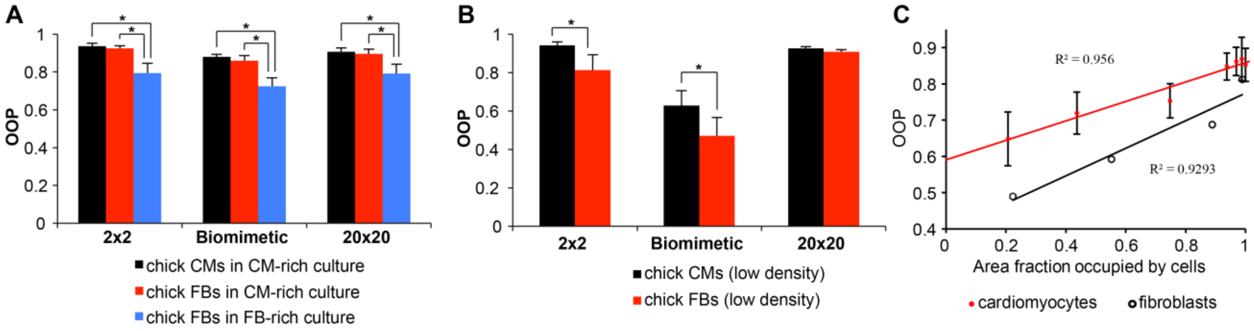

Figure 2.14. Chick fibroblast alignment on fibronectin patterns and their effect on cardiomyocyte alignment in chick cardiac tissues. A: Comparison of chick cardiomyocyte OOP to fibroblast OOP in cardiac-rich and fibroblast-rich monolayers indicates that although fibroblasts show lower intrinsic alignment, in a mixed high-density culture the alignment of both cell types is equal. **B:** Chick cardiomyocyte and fibroblast OOP analysis in a low-density cardiomyocyte-rich culture reveal that individual fibroblasts show lower alignment compared to cardiomyocytes. Due to the low density of the culture, the alignment of each individual cell can be considered independent of other cells. (*) represents statistically significant difference ($p < 0.05$, one-way ANOVA).

2.4.3 Heat Maps of Local Cardiomyocyte Alignment on the Biomimetic and 20x20 Patterns

Confocal images of cardiomyocytes on fibronectin patterns show drastically different cardiomyocyte response to the same local fibronectin features at low and high density (fig. 2.15). Specifically, cells tend to follow the shape of fibronectin features at low density, while at high density they seem to completely ignore it. We quantified this difference by creating heat maps showing local effects of the biomimetic and the 20x20 pattern on various cell characteristics, which allowed us to determine how the role of cell-substrate interactions on tissue alignment changes with cell density. Particularly, we patterned fluorescently labeled fibronectin onto substrates, seeded 8 day chick cardiomyocytes on them at low (60,000 cells/cm^2) and high (450,000 cells/cm^2) densities, cultured them for 3 days, then fixed, stained, and imaged the tissues with the underlying pattern. To create heat maps, we developed a Matlab script (Appendix 4) that detects the pattern's unit cell position and orientation (fig. 2.16A), then attaches a reference frame with the origin in a pre-defined location in the unit cell and the axes parallel to the translocation vectors of the unit cell array (fig. 2.16B). Then for each detected actin filament we calculated the coordinates of that filament in that reference system (fig. 2.16C), which allowed us to combine the location-specific orientation data from multiple images. We split the biomimetic pattern unit cell (155x250 µm) and the 20x20 unit cell (40x1 µm) into 1x1 µm bins and for each bin we calculated the amount of detected actin filaments, mean, median, and the most probable (mode) orientation angles, standard deviation of orientation angle, and OOP.

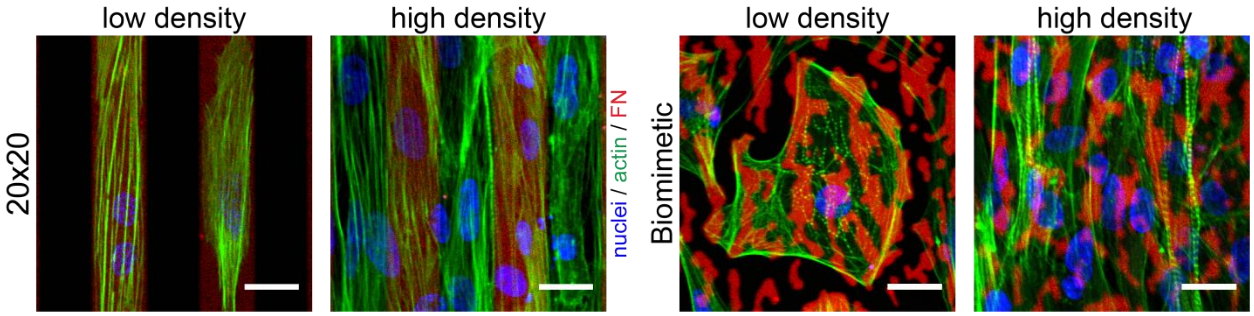

Figure 2.15. Cardiomyocyte response to the same pattern features at low and high cell densities on the 20x20 and the biomimetic patterns show significant differences. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 20 µm.

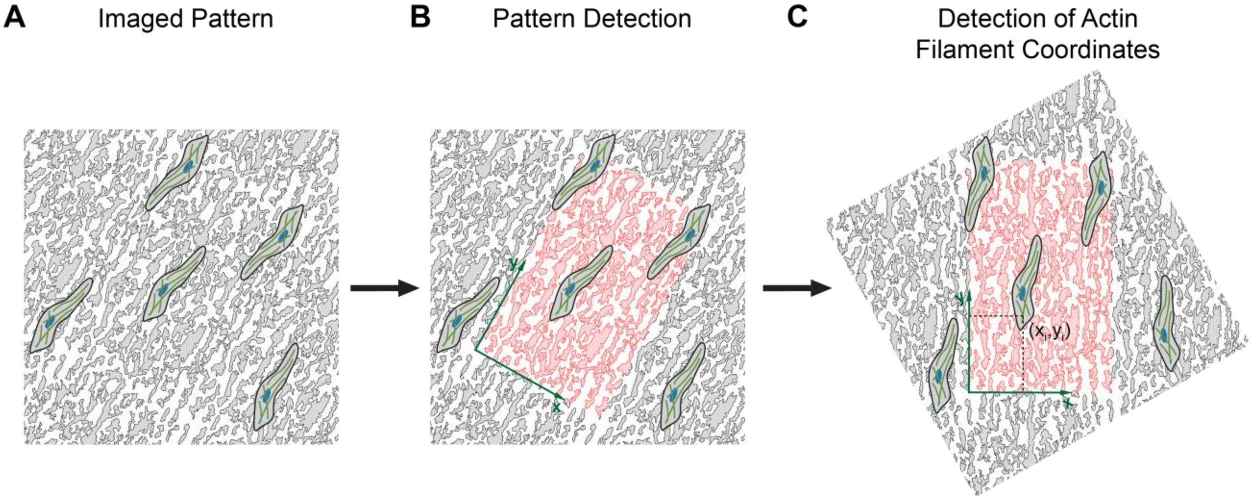

Figure 2.16. Schematic of actin filament orientation data acquisition for heat maps. A: Schematic of cells on the pattern. **B:** Detection of the pattern's unit cell location and orientation. The reference system relative to which all orientations will be calculated is shown. The red area highlights one of the pattern's unit cells. **C:** Detection of actin filament coordiantes relative to the introduced reference frame. Orientation data of all filaments is split into bins based on the detected coordinates.

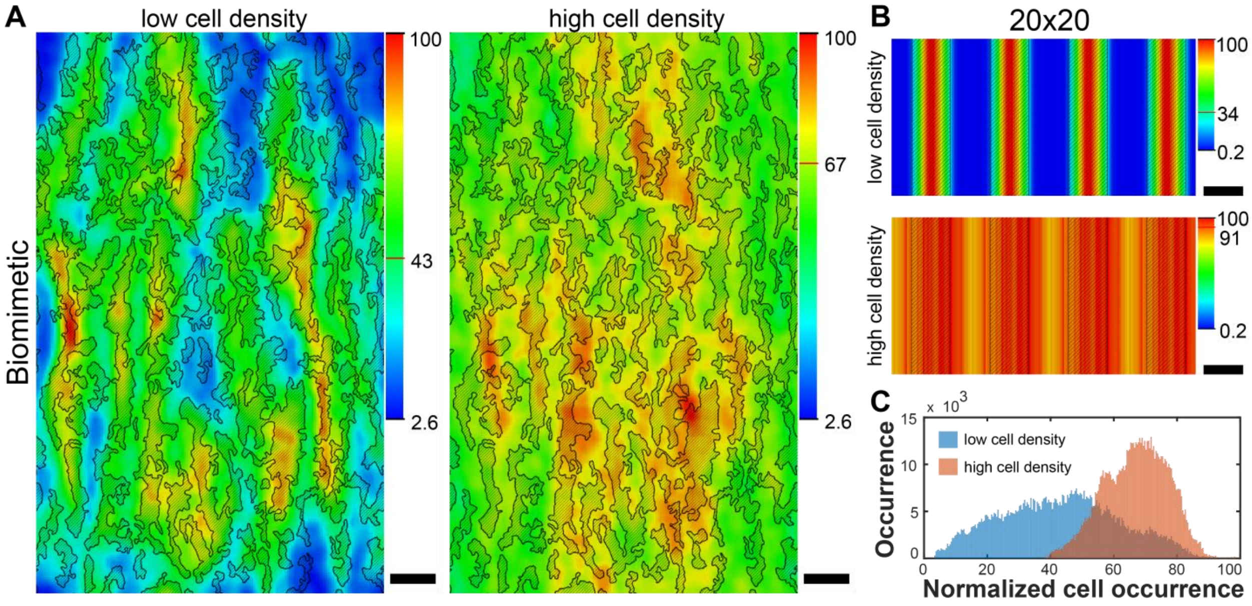

Figure 2.17. Distribution of normalized cell actin occurrence on fibronectin patterns. A, B: Heat maps of normalized cell occurrence on the biomimetic **(A)** and the 20x20 **(B)** patterns al low and high cell densities. Low density maps are highly heterogeneous with high occurrence areas correlating with the location of fibronectin features. High density maps are significantly more homogeneous and high-occurrence areas don not correlate with fibronectin features. Average value is indicated on the scale of each map with the red mark. **C:** Histogram of normalized cell occurrence distribution on the biomimetic pattern show significantly higher variance at low density compared to high density. Scale bars: 20 µm.

The biomimetic heat maps for normalized cell occurrence (number of detected actin filaments normalized to a 100) reveal that at low density cardiomyocytes show high heterogeneity in their attachment rate on the pattern with higher attachment rates observed around large fibronectin features and lower – around areas with lower fibronectin density (fig. 2.17A). At high density, however, cell attachment becomes significantly more uniform and independent on the underlying fibronectin pattern. Histograms of normalized cell occurrence derived from the data shown in the heat maps further confirms these findings – the ratio of cell occurrence in the most and the least populated 1x1 µm bins is 38:1 at low density and 2.6:1 at high density (fig. 2.17C). These results indicate that cell attachment and spreading is much more susceptive to cell-substrate interactions at low density, when cell-cell interactions are rare. The same relationship can be observed on the 20x20 heat maps (fig. 2.17B); however, due to the larger gap between fibronectin features compared to the biomimetic pattern, cell attachment at low density is almost zero in between the lines.

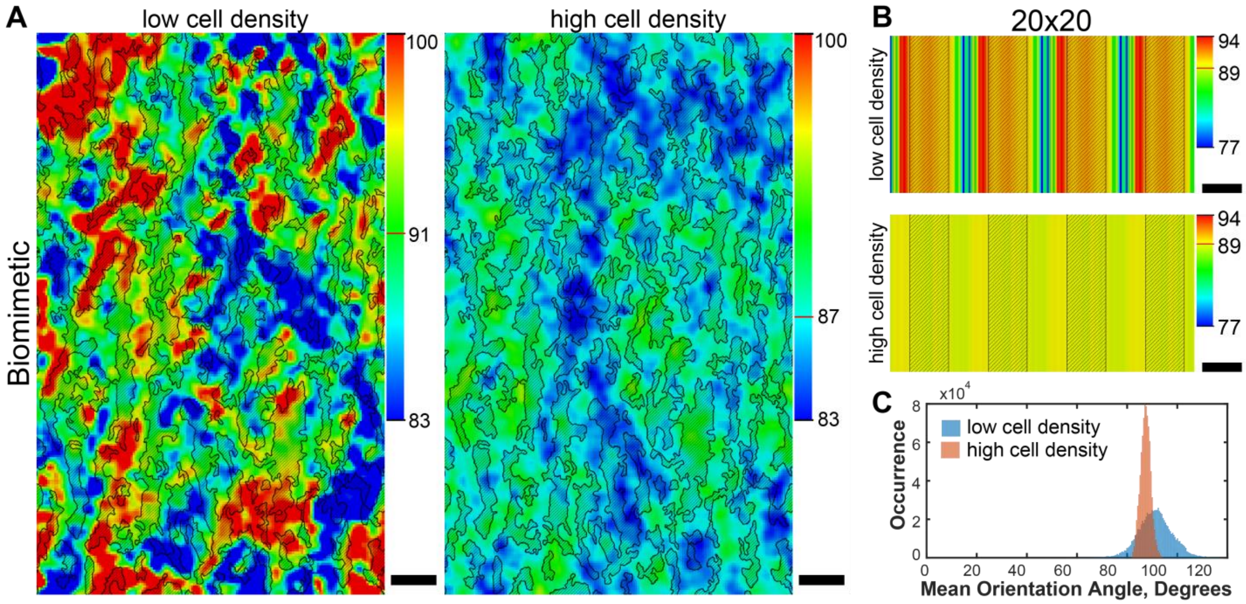

Figure 2.18. Distribution of mean actin filament orientation angle on fibronectin patterns. A, B: Heat maps of mean orientation angle on the biomimetic **(A)** and the 20x20 **(B)** patterns al low and high cell densities. Low density maps show significantly higher heterogeneity compared to the high density ones. Average value is indicated on the scale of each map with the red mark. **C:** Histogram of mean orientation angle distribution on the biomimetic pattern show significantly higher variance at low density compared to high density. Scale bars: 20 µm.

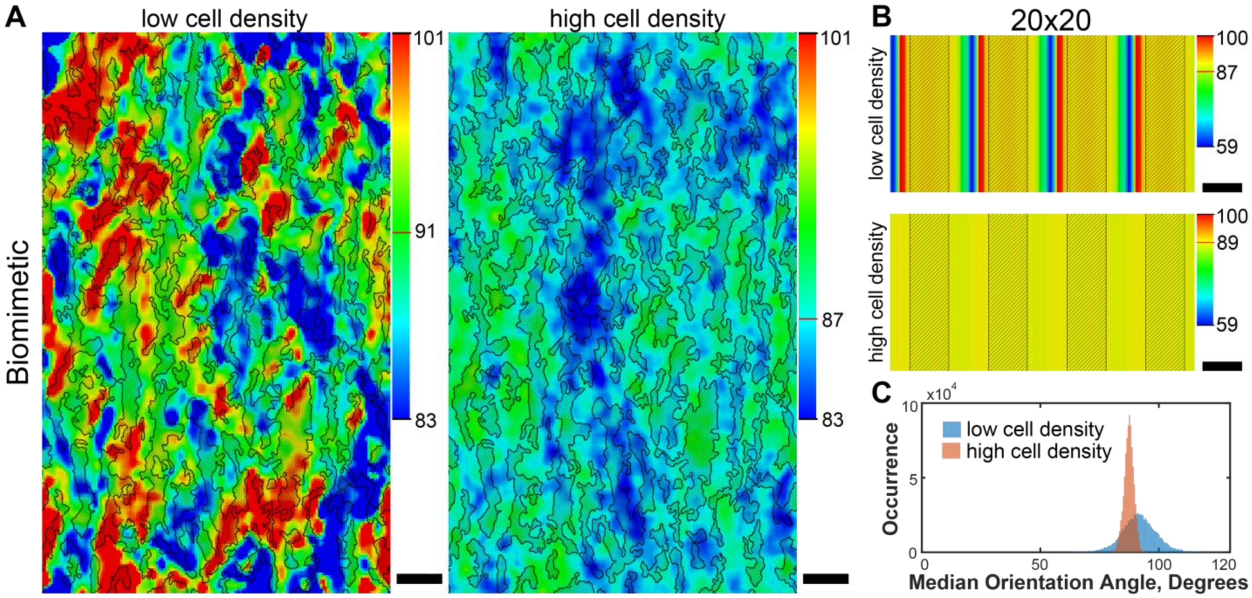

Figure 2.19. Distribution of median actin filament orientation angle on fibronectin patterns. A, B: Heat maps of median orientation angle on the biomimetic **(A)** and the 20x20 **(B)** patterns al low and high cell densities. Low density maps show significantly higher heterogeneity compared to the high density ones. Average value is indicated on the scale of each map with the red mark. **C:** Histogram of median orientation angle distribution on the biomimetic pattern show significantly higher variance at low density compared to high density. Scale bars: 20 µm.

Heat maps of mean and median orientation angles for the biomimetic pattern (fig. 2.18A, 2.19A), similarly to the heat maps of normalized cell occurrence, show significantly higher heterogeneity at low density compared to the high density. This stays in agreement with the previously observed tendency of cardiomyocytes to follow local fibronectin features at low density, but not the high density (fig. 2.15). The 20x20 pattern exhibits similar behavior (fig. 2.18B, 2.19B), although the orientation angle of filaments located between high-density fibronectin lines appears to be more variable due to the fact that the amount of cells detected in those areas is extremely low and the cells that occupy these areas normally bridge across two adjacent lines and thus don't have a consistent orientation. Histograms of mean and median orientation angles for the biomimetic pattern (fig. 2.18C, 2.19C) stay in agreement with the conclusions from the corresponding heat maps, but also show that the peaks of angle distributions at low density are shifted relative to the corresponding peaks at high density. For example, the peak on the mean orientation angle histogram shifts from ~92^{o} at low cell density to ~87^{o} at high cell density. The phenomenon of cell orientation being not parallel to the direction of fibronectin alignment has been observed in the earlier studies with C2C12 myotubes.[59] However, there is currently no explanation of this effect and additional studies need to be performed to determine the reason of such unique cell behavior.

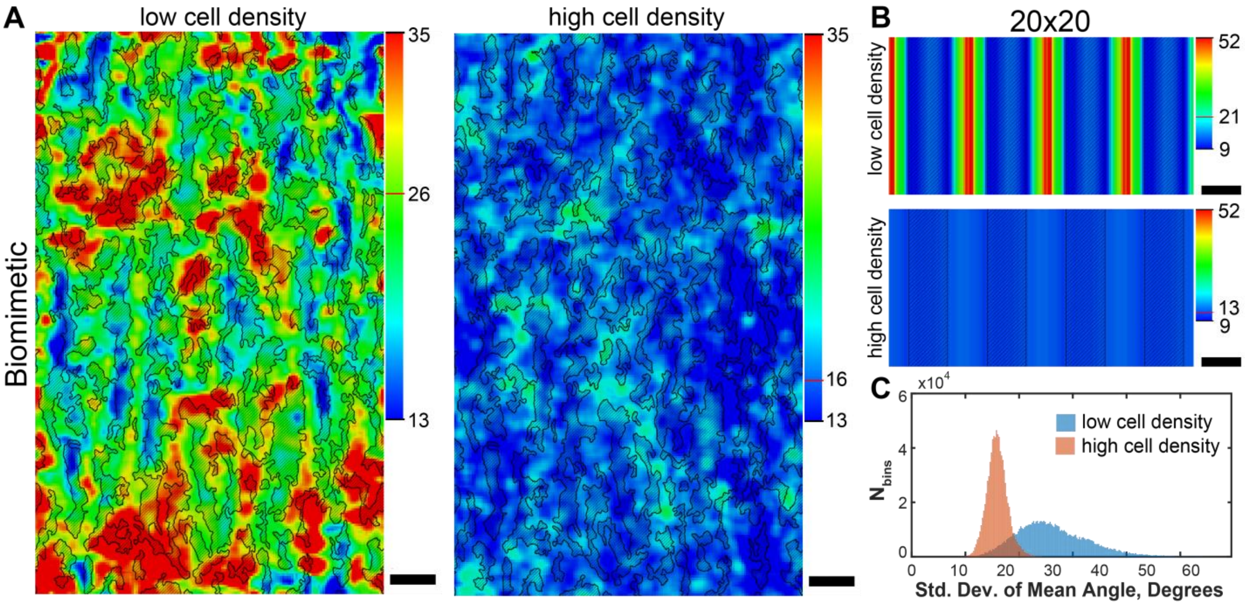

Figure 2.20. Distribution of standard deviation of orientation angle on fibronectin patterns. A, B: Heat maps of standard deviation of orientation angle on the biomimetic **(A)** and the 20x20 **(B)** patterns al low and high cell densities. Low density maps show significantly higher heterogeneity and higher average values corresponding to lower alignment compared to the high density ones. Average value is indicated on the scale of each map with the red mark. **C:** Histogram of standard deviation of orientation angle distribution on the biomimetic pattern show significantly higher variance and a higher peak value corresponding to lower alignment at low density compared to high density. Scale bars: 20 µm.

Heat maps of the standard deviation of orientation angle and OOP for the biomimetic pattern (fig. 2.20A, 2.21A) show that cell alignment is highly non-uniform at low density and significantly more uniform at high density, which is similar to the behavior observed for the previously discussed heat maps. Additionally, at low density the high alignment areas follow more elongated fibronectin features, indicating that cells align along these features. The histograms corresponding to these heat maps (fig. 2.20C, 2.21C) show the same relationship between the low and high density cultures. The 20x20 heat maps of the standard deviation of orientation angle and OOP reveal that at low density cardiomyocytes are highly aligned on the lines, while their alignment between the lines is significantly lower (fig. 2.20B, 2.21.B). This makes the low density OOP averaged based on the filament location significantly lower compared to the high density case, although, as reported earlier, when OOP is normalized on the number of detected filaments, it's as high as the in the high-density case. This is due to the fact that the majority of cells are attached on the fibronectin lines, where OOP is high.

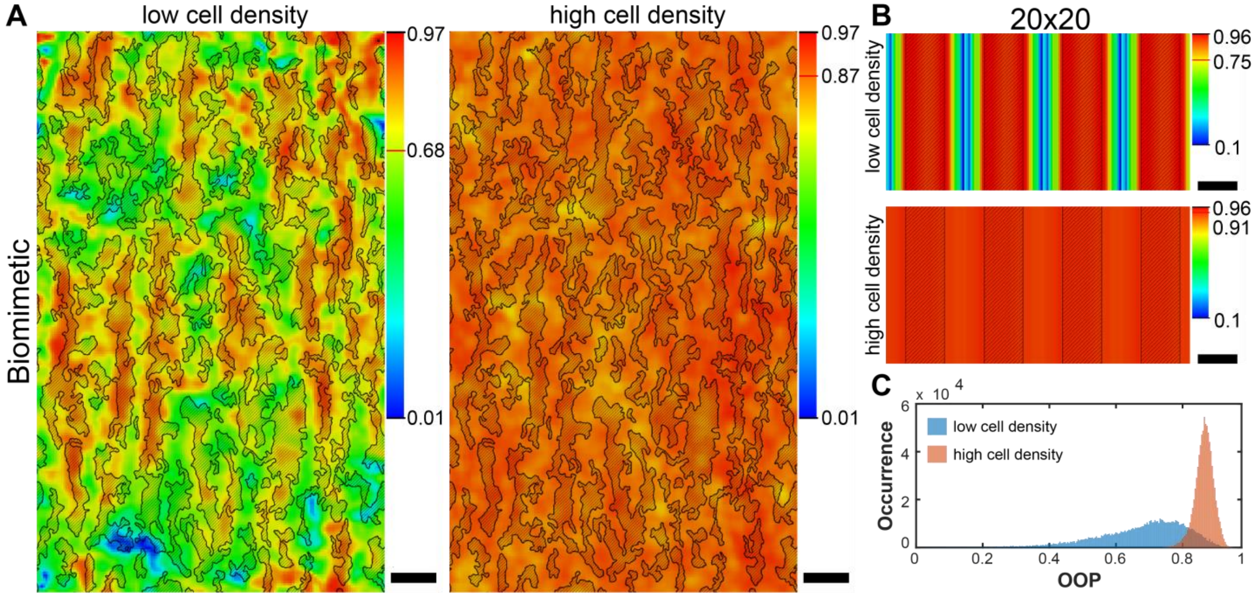

Figure 2.21. Distribution of cardiomyocyte OOP on fibronectin patterns. A, B: Heat maps of OOP on the biomimetic **(A)** and the 20x20 **(B)** patterns al low and high cell densities. Low density maps show significantly higher heterogeneity and lower average values compared to the high density ones. Average value is indicated on the scale of each map with the red mark. **C:** Histogram of OOP distribution on the biomimetic pattern show significantly higher variance and a higher peak compared to high density. Scale bars: 20 µm.

2.4.4 The Effect of N-cadherin Inhibition on Chick Cardiomyocyte Alignment on the Biomimetic Pattern.

Although heat maps do not reveal the primary driving force behind cardiomyocyte alignment in a high density monolayer, we hypothesized that this force is cell-cell interactions. To test this, we studied the effect of the inhibition of cell-cell interactions on cardiomyocyte alignment. Particularly, we chose to use blocking antibodies (Sigma C2542) to inhibit N-cadherin – a trans-membrane protein involved in the formation of one of the most abundant cell-cell junctions between cardiomyocytes. Additionally, N-cadherin junctions, similarly to focal adhesions, serve as attachment sites for actin filaments and thus can directly affect actin alignment. Studies have shown that introducing blocking antibodies reduces the amount of N-cadherin junctions.[60,61] However, the complete blockage of cell-cell interactions does not occur due to the existence of other interactions via desmosomes and gap junctions as well as physical interaction.[62]

Firstly, to determine the optimal antibody concentration, we cultured cardiomyocytes with various concentrations of the blocking antibodies and looked at N-cadherin localization in cells (fig. 2.22). The antibodies were added to the cardiomyocyte maintenance medium (97% M-199 medium, 2% HI-FBS, 1% penstrep) that was introduced to the cells 8 hours after seeding to ensure that cells attach to the substrate, but not spread enough to reach other cells and form N-cadherin junctions. The results showed that without the blocking antibodies N-cadherin is primarily expressed on the cell membrane along cell-cell interface. With the increase of the blocking antibodies N-cadherin becomes internalized indicating that it is no longer involved in the formation of cell-cell junctions. The antibody concentration of 15.2 µg/mL was chosen to analyze the effect of blocking on cardiomyocyte alignment as almost no N-cadherin along cell-cell interface was observed at this concentration. We incubated chick cardiomyocytes on the biomimetic pattern at various cell densities ranging from 60,000 cells/cm^2 to 450,000 cells/cm^2 with and without N-cadherin antibodies for 3 days and measured their alignment along with the final confluence that was estimated based on the actin staining. The results showed that the effect of N-cadherin blocking is dependent on cell density (fig. 2.23D). At the lowest density (fig. 2.23A) the majority of cells are isolated from one another, therefore blocking has no effect on alignment, as cell-cell contacts are rare. With the increase of density (fig. 2.23B) the cardiomyocyte OOP in the control group starts increasing, while the OOP of cells with blocking antibodies remains the same as the OOP of isolated cells. This shows that N-cadherin-based cell-cell junctions do influence cell alignment and are at least partially responsible for the observed increase of cardiomyocyte OOP. Finally, at high cell densities approaching a confluent monolayer (fig. 2.23C), OOP of cells with blocking antibodies increases and converges with the OOP of the control group. We attribute this to the increased influence of the other types of interaction that we did not inhibit, which eventually renders N-cadherin blocking ineffective. To perform statistical analysis on this data, we split the OOP points into 7 groups based on the area fraction occupied by cells and then performed a two-way ANOVA with a pair-wise post-comparison to compare the OOP of the cells with and without the blocking antibodies at different densities.

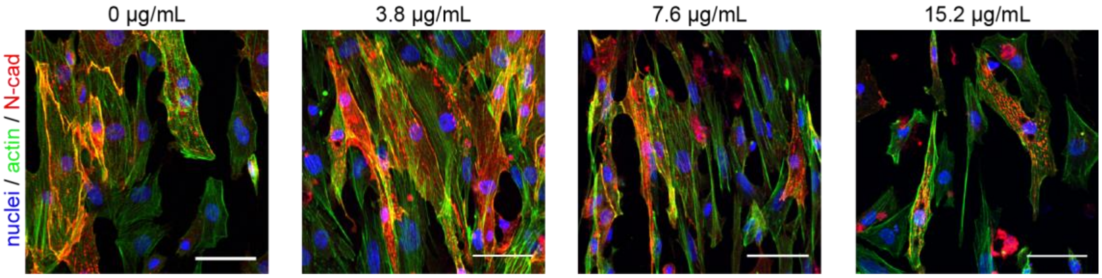

Figure 2.22. N-cadherin localization in chick cardiomyocytes cultured with different concentrations of N-cadherin blocking antibodies. As the concentration of antibodies increases, N-cadherin moves from the membrane inside the cell, where it cannot participate in the formation of cell-cell junctions. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 20 µm.

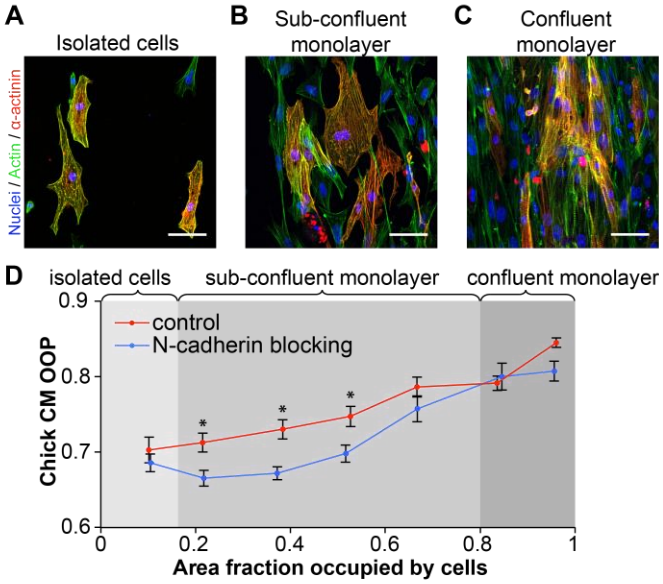

Figure 2.23. The effect of N-cadherin blocking antibodies on chick cardiomyocyte alignment on the biomimetic pattern. A-C: Chick cardiomyocytes at low, intermediate, and high densities on the biomimetic pattern. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 50 µm. **D:** The effect of N-cadherin inhibition using blocking antibodies on cardiomyocyte OOP as a function of cell density. In a low-density culture corresponding to isolated cells, blocking antibodies have no effect on cardiomyocyte OOP because cell-cell interactions are rare. At intermediate cell densities corresponding to 20%-60% confluent cultures, the increase in OOP was observed for the control tissues while the tissues with inhibited N-cadherin interactions maintained significantly lower OOP corresponding to lower-density cultures. At high cell densities the effect of N-cadherin inhibition has no significant impact on cardiomyocyte OOP, which we attribute to the role of other cell-cell interactions, including physical interactions that were not inhibited. Error bars represent standard errors. “*” indicates statistically significant difference in OOP samples with and without blocking antibodies (N≥18, $p < 0.05$, two-way ANOVA performed on data points combined into groups based on the area fraction occupied by cells).

2.4.5 Vinculin Staining of Chick Cardiomyocytes and Fibroblasts

Heat maps of cardiomyocyte attachment and alignment showed that the role of cell-ECM interactions in cardiomyocyte organization of fibronectin patterns changes with cell density. Cell-ECM interactions occur through the complex of various trans-membrane and internal proteins known as focal adhesions. Focal adhesions connect fibronectin molecules outside the cell to actin filaments inside it and play a key role in translating mechanical stresses across cell membrane. Therefore, we aimed to look at how the orientation of focal adhesions correlates with the change of cardiomyocyte OOP between different fibronectin patterns and as a function of cell density on the biomimetic pattern. To do so, we stained chick cardiomyocytes and fibroblasts for a focal adhesion protein, vinculin, using antibodies (Sigma V4139). We found that although the staining works well for the fibroblasts, vinculin signal in cardiomyocytes is extremely low and cannot be used to reliably analyze the orientation of focal adhesions (fig. 2.24). Therefore, further studies are required to determine a more reliable way to detect cardiomyocyte focal adhesions and analyze the effect of the cell density and fibronectin pattern on focal adhesion orientation.

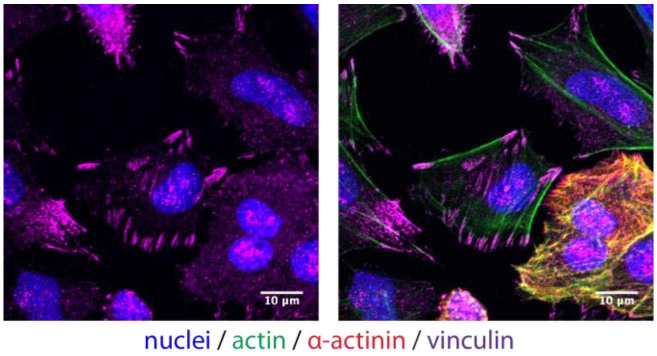

Figure 2.24. Chick cardiomyocytes and fibroblasts stained for vinculin shows high vinculin signal in fibroblasts and low signal in cardiomyocytes. Left image shows only nuclei and vinculin, right image shows nuclei, vinculin, actin, and α-actinin. Nuclei are shown in blue, actin – green, α-actinin – red, vinculin - purple. Scale bars: 10 µm.

2.5 Discussion and Conclusions

The biomimetic pattern we derived from the images of fibronectin in chick embryonic myocardium represents an important step towards developing a biomimetic approach in cardiac tissue engineering. However, it is also important to understand the limitations of the way the designed pattern mimics the embryonic cardiac environment. Firstly, our biomimetic pattern lacks 3D structure, which exists in the native heart. Secondly, the 2 µm resolution limit of the photomask makes it incapable of recreating the finer structure of fibronectin fibrils present in myocardium. Thirdly, due to the differences in the way fibronectin molecules assemble in fibrils, micropatterned fibronectin does not resemble the supramolecular structure of the native fibrils, which may affect the type of integrins involved in cell-fibronectin interactions. And finally, our biomimetic pattern represents a combination of fibronectin features from several different locations in the heart, and for each of those locations the pattern features were modified to fit together. These limitations inevitably introduce differences in the way cells react to the biomimetic pattern and the native fibronectin structure. However, the biomimetic pattern is still significantly closer to the structure of embryonic ECM than the previously used line patterns and in this study we were able to show significant differences in the way cardiomyocytes respond to this pattern and line patterns.

Engineering aligned chick cardiac tissues on various fibronectin patterns revealed that although the underlying pattern determined the overall direction of cardiomyocyte alignment in a high-density monolayer, it has little influence on the degree of final tissue alignment represented as OOP. Lowering cell density reduces the amount of cell-cell interactions and thus inevitably increases the influence of other types of interactions present in the system, particularly cell-ECM interactions. Observing cardiomyocytes on fibronectin patterns at low density showed that the cell-ECM interactions do result in different OOP values depending on the pattern. This indicates that cell-ECM interactions are not the main driving force in the formation of an aligned cardiac monolayer. Heat maps further confirmed this conclusion by showing that at low-density cell attachment, orientation, and alignment highly depend on local fibronectin features, while at high density these parameters become uniformly distributed throughout the pattern. N-cadherin blocking experiment showed that cell-cell interactions play an important role in stimulating high cardiomyocyte alignment in a high-density monolayer, although we were only able to observe their effect in a sub-confluent culture due to the presence of other types of cell-cell interactions we did not inhibit. Importantly, as the biomimetic pattern was based on the fibronectin structure of embryonic heart, our results suggest that this fibronectin structure cannot be directly responsible for stimulating myocardial alignment, although fibronectin itself is still crucial for heart development.

Comparing heat maps for the biomimetic and the 20x20 pattern also revealed an important difference in the way the biomimetic environment and artificial environment of line patterns stimulate tissue alignment. The line patterns were designed and extensively used in the past to stimulate the alignment of various types of cells.[50] The usual rationale behind using those patterns is based on using cell-ECM interactions to force cells to follow the features of a highly anisotropic line pattern and form an aligned monolayer. However, in case of chick cardiac tissues, we have shown that a highly aligned line pattern is not required to create the same level of a high-density tissue alignment. In fact, one of the main problems scientists face when using line patterns is cell connectivity in a transverse direction that prevents effective propagation of action potential in the direction perpendicular to the direction of cell alignment. Relying on cell-cell interactions to stimulate alignment may solve this problem and create cardiac monolayers with a higher degree of electrical coupling. However inconsistencies in cardiomyocyte purity in chick tissues make it extremely difficult to compare functional characteristics of these tissues and a more reliable system is required to make this analysis.

The inevitable presence of fibroblasts in the cardiomyocyte population isolated from chick embryos has the potential to influence the alignment of cardiac tissues and, therefore, their effect on cell alignment needs to be assessed. We found that chick fibroblasts, when seeded on fibronectin patterns, showed lower alignment compared to the cardiomyocyte-rich population at the same density, suggesting that the presence of fibroblasts lowers the overall alignment of the chick cardiac tissues. Additionally, using an α-actinin-based mask we were able to measure cardiomyocyte and fibroblast alignment in the same culture independently revealing that in a cardiomyocyte-rich high-density monolayer, the alignment of cardiomyocytes and fibroblasts is the same, while in the low-density culture the fibroblast alignment is lower than that of the cardiomyocytes. This indicates that interactions between cardiomyocytes and fibroblasts influence each other's alignment and serves as another indication of the important role of cell-cell interactions in the formation of a cell monolayer.

In conclusion, recreating the structure of fibronectin in embryonic myocardium *in vitro* using a chick model allowed us to determine the mechanism of cardiac tissue alignment and better understand the role of fibronectin in this process, which can help develop improved techniques for engineering functional cardiac tissues that resembles native myocardium. However, the limitations of this system pose significant challenges in studying functional characteristics of the engineered tissues and translating the results of this study to clinically relevant human therapies. Thus, a more appropriate system is required to overcome these limitations. Recent advances in generation of large amounts of high-purity patient-specific human iPSC-CMs make them a promising cell source for engineering highly consistent human cardiac tissues. Translating the results of this study to engineering aligned iPSC-CM monolayers presents the next step towards developing the biomimetic approach in cardiac tissue engineering.

Chapter 3: Human Pluripotent Stem Cell-Derived Cardiac Tissues on Fibronectin Patterns.

3.1 Abstract

While the chick model provides a great platform to study the effect of the embryonic heart ECM structure on cardiac tissue alignment, only human tissues can be used to understand and treat cardiac disease in humans. Recent advances in cardiomyocyte differentiation from human ESCs and iPSCs have opened new possibilities to develop clinically relevant tissues with patient-specific genotype. However, direct injection of these cells into the heart has proved ineffective due to the low survival rate caused by the hostile environment of the infarcted myocardium. Therefore, cardiomyocytes need to be organized into a functional tissue prior to implantation in order to increase the chance of survival and successful integration with the host tissue. In this study we seeded iPSC-CMs onto previously developed biomimetic and line fibronectin patterns to generate anisotropic cardiac monolayers. We found that the iPSC-CM response to these patterns is significantly different to that of chick embryonic cardiomyocytes with overall lower alignment, which we attribute to the immature phenotype of iPSC-CMs. Further, we showed that, unlike with chick cardiomyocytes, the addition of primary CFBs increases iPSC-CM alignment, suggesting their potential in engineering better cardiac tissues. Finally, we revealed that using iPSC-CMs matured using T3 hormone significantly increases the alignment of cardiac tissues, indicating the importance of developing new techniques for iPSC-CM maturation.

3.2 Introduction

One of the main challenges of using engineered tissues for human cardiac therapies is the response of the host to the implanted tissue.[63] Specifically, when the cells used to generate the tissue are allogeneic or xenogeneic, this triggers an immune response against the foreign body. To minimize these problems, the ideal cell source for tissue engineering should be autologous, i.e. derived from the same patient who needs the transplant. For a long time there were no reliable cardiomyocyte sources that could satisfy these criteria and produce cells in large amounts. Terminally differentiated cardiomyocytes do not normally proliferate and adult stem cells have limited proliferative capacity. ESCs have shown promising results in their ability to differentiate into functional cardiomyocytes, but are not patient-specific. However, in 2007 the discovery of iPSCs and iPSC differentiation into cardiomyocytes was reported.[64] The techniques of inducing pluripotency from primary cells have been significantly improved and today the generation of patient-specific iPSCs can be performed easily, reliably, and at a significantly lower cost compared to the early methods.

The techniques for differentiating iPSCs into cardiomyocytes have been significantly improved as well.[65] The first techniques were developed for ESC differentiation and they relied on the paracrine signaling provided by endoderm-like (END-2) stromal cells that the ESCs were co-cultured with.[66] This technique suffered from low and unreliable efficiency (usually around 1% cardiomyocytes in the final culture) and lack of understanding of the differentiation mechanism due to the large amount of unknown components in the system. Later this technique was replaced by a more controlled embryoid body differentiation, where ESCs were cultured in 3D-aggregates called embryoid bodies (EBs) due to their similarity with the embryo, and provided with growth factors and other regulatory molecules to direct their differentiation towards the cardiac pathway.[67] Early EB differentiation protocols suffered from the same drawbacks as the differentiation via co-culture with END-2 stromal cells, but over the years they have been significantly improved by moving towards chemically defined environment and controlling the process of EB formation more precisely.[68,69] First iPSC differentiation protocols were based on the EB protocols for ESCs. However, the later invention of monolayer-based protocols in 2007 revolutionized the area. The first monolayer-based protocol reported ~30% efficiency, and although it had reproducibility issues, its efficiency was significantly higher than any EB-based protocols existed at that time.[70] Later in 2012 the protocol was improved to achieve ~90% pure cardiomyocyte differentiation with significant improvements in batch-to-batch and line-to-line reproducibility.[71] In 2014 another group developed a simplified version of this protocol, which removed any unnecessary components from the media and optimized the cost of differentiation without reducing the efficiency.[72] To this date, monolayer-based differentiation of iPSCs towards iPSC-CMs is the easiest and the most efficient way to produce large amounts of patient-specific cardiomyocytes (fig. 3.1).

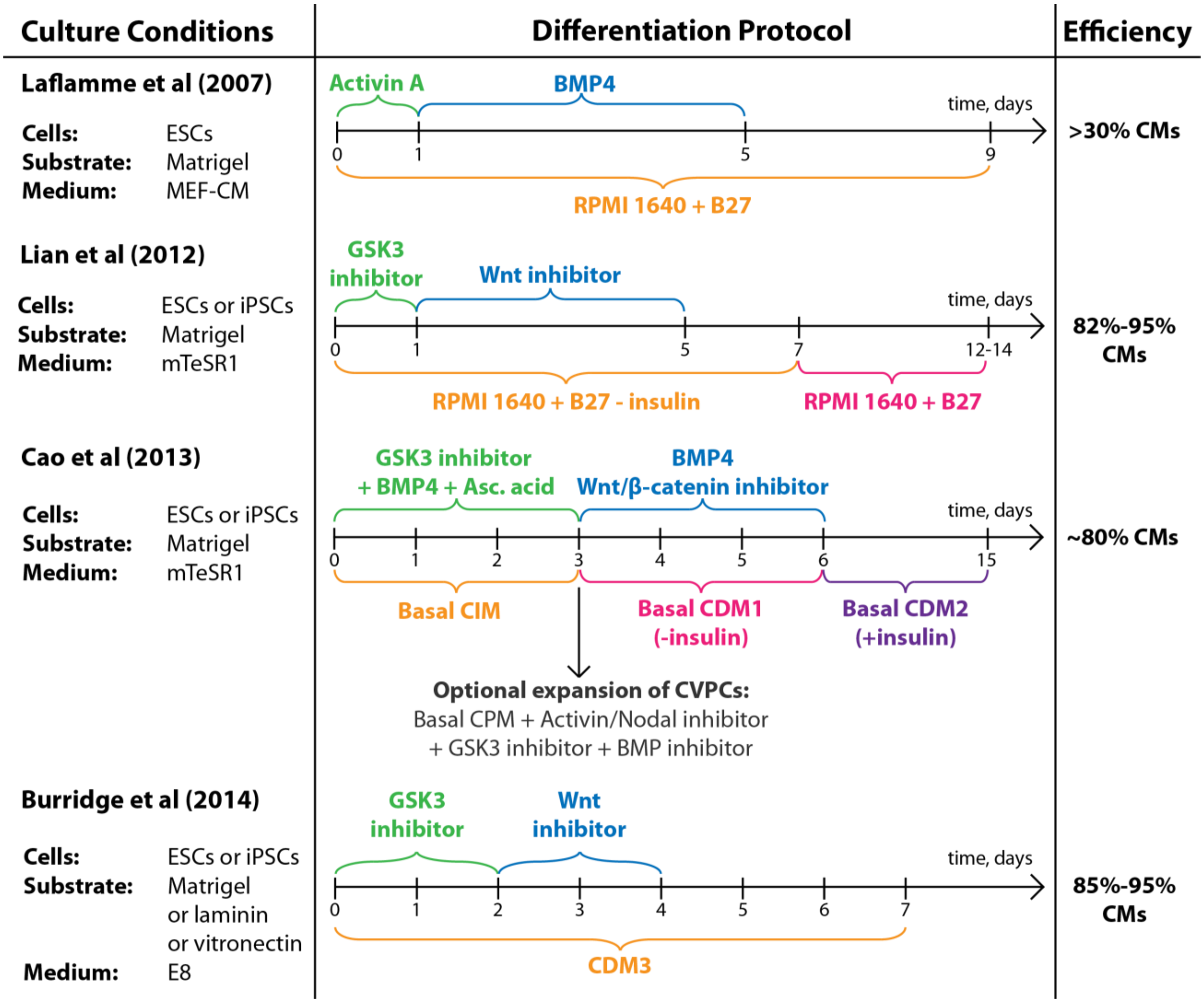

Figure 3.1. Timelines of PSC monolayer-based differentiation methods. Culture conditions are shown in the left column, differentiation efficiency – in the right column. Timelines show information about media used for differentiation (below time axis) and the additional components added to it (above time axis). Information about inhibitors used in each method can be found in the Table 1. The last time mark indicates the beginning of spontaneous contractions of the derived cardiomyocytes. Adapted from: Batalov, I. and A. W. Feinberg (2015). Biomarker insights **10**(Suppl 1): 71.

However, there are still problems with using these cardiomyocytes for tissue engineering applications with the main challenge being the immature phenotype of iPSC-CMs characterized by cardiomyocyte proliferation, low amount of sarcomeres, low cell aspect ratio, low contractile force, embryonic-like action potential profile, lack of T-tubules, low expression levels of ion channels upregulated in adult cells (Ca^{2+} channel, Na^{+}-Ca^{2+} exchanger, $K_{ir}2.1$ inward K^{+} rectifier, and Kv1.4 channel),high expression levels of early stage ion channels that are downregulated in adult cells (HCN_1, HERG1b channels), and different responses to regulatory molecules and drugs.[73] There are several existing techniques for iPSC-CM maturation, such as prolonged culture (up to 1 year), mechanical stimulation,[74,75] electrical stimulation,[76-78] and culture with the T3 hormone,[79] and while some of them work better than others, there is still no known way to convert early iPSC-CMs into adult-like cardiomyocytes.

In the previous chapter we determined the response of chick cardiomyocytes to the biomimetic pattern derived from the images of fibronectin in chick embryonic myocardium and elucidated the role of cell-cell and cell-ECM interactions in cardiomyocyte alignment. In this chapter we used a human ESC line HUES9 and an iPSC line 13FLVNOC1 to derive cardiomyocytes and generate cardiac monolayers and described the differences in the response of these cells to the patterns. Further, we analyzed the density-dependent iPSC-CM alignment of fibronectin patterns and compared it to the alignment of chick cardiomyocytes. Additionally, we determined the role of iPSC-CM interactions with fibroblasts and fibroblast-generated ECM on the alignment of iPSC-CM tissues using human primary CFBs. Finally, we used the T3 hormone to mature human cardiomyocytes and determined the effect of such maturation on iPSC-CM alignment.

3.3 Materials and Methods

3.3.1 Fabrication of PDMS Stamps

PDMS stamps with three micropatterns (20x20, 2x2, and biomimetic) were used to pattern fibronectin onto substrates for cells. The process of PDMS stamp fabrication is based on the previously described technique (fig. 2.2(i)-(iv)).[55] Briefly, a glass wafer was spin-coated with the photoresist SPR 220.3 at 5000 RPM, then put tightly under the photomask containing the desired pattern on a horizontal surface, and exposed to UV-light through the mask. The exposed parts of the photoresist were washed away using developer MF-319, the cover glass was washed in distilled water and dried with a nitrogen gun. To decrease photoresist adhesion to PDMS, it was silanized by incubating it next to an open container of 2% dimethyldichlorosilane solution (PlusOne Repel-Silane ES, GE Healthcare) for 24 hours in a desiccator. Then PDMS (Sylgard 184 base mixed 10:1 with the curing agent) was cast on top of the photoresist, cured at 65 ^{O}C for 24 hours, and stamps were cut out of the PDMS layer.

3.3.2 Substrate Preparation

Fibronectin-patterned PDMS-coated glass coverslips were used as substrates for cells. PDMS-coated coverslips were prepared using Sylgard 184 silicone elastomer according to the previously described procedure.[56] Briefly, Sylgard 184 base and curing agent were mixed at the mass ratio of 10:1 followed by the mixture and defoaming in a “Thinky conditioning mixer”. Spin-coating at 4000 RPM (Table 1) was used to coat coverslips with a thin PDMS layer. After spin-coating, coverslips were put in an oven at 65 ^{0}C for 24 hours in order to cure PDMS.

Fibronectin patterns were microcontact printed onto the PDMS-coated coverslips according to a previously described technique with minor modifications (fig. 2.2(v)-(vii)).[57]

Briefly, PDMS stamps were cleaned by sonication in 50% ethanol for 45-60 minutes and dried using pressurized nitrogen. Then the patterned side of each stamp was incubated in 50 µg/mL solution of human plasma fibronectin (unlabeled or labeled with Alexa Fluor 546 Maleimide fluorophore) for 60 minutes, washed in sterile water, and dried with a nitrogen gun. To transfer fibronectin from the stamp onto the coverslip, it was UV-Ozone treated for 15 minutes, and then the patterned side of the stamp was brought in contact with the coverslip for 5 minutes. The pattern transfer was verified for each stamp by confocal microscopy using fluorescently labeled fibronectin. Before cell seeding, all patterns were incubated in 1% w/v solution of Pluronic F127 to reduce cell adhesion in between fibronectin features and improve cardiomyocyte alignment.

3.3.3 Fixation, Staining, and Fluorescent Microscopy

Cells were fixed and permeabilized with 4% formaldehyde and 0.1% Triton-X 100, and stained with 0.5% DAPI (to stain nuclei), 1.5% phalloidin (to visualize the actin cytoskeleton), and 0.5% of corresponding primary antibodies – for other proteins, such as α-actinin (Sigma A7811), N-cadherin (Sigma C2542), and fibronectin (Sigma F3648). Samples were incubated with the dyes and primary antibodies for 60 minutes, washed in 1X PBS, incubated in the solution of secondary antibodies corresponding to the primary antibodies for 60 minutes, and washed in PBS again. After that coverslips were mounted for imaging onto glass slides with ProLong Gold Anti-Fade preservative. Confocal laser scanning microscopy (LSM Zeiss 700) was used to obtain fluorescent images.

3.3.4 Cell Alignment Analysis

Cell actin alignment was measured using a custom MATLAB script based on local actin filament orientation analysis (Appendix 1). First, confocal images of cardiac tissues (fig. 2.3 A) stained for nuclei (fig. 2.3B), actin (fig. 2.3C), and α-actinin (fig. 2.3D) were taken. Local orientations of actin filaments were detected (fig. 2.3E), and based on the provided threshold (varying from 0 to 1) associated with the filament prominence, a filament mask was created determining which orientations to take into analysis (fig. 2.3F). Actin and α-actinin channels were processed to produce binary masks of cell location (fig. 2.3G) and cardiomyocyte location (fig. 2.3H) respectively. Using combinations of these masks, each orientation was assigned to a cell type (fig. 2.3I). Angular distribution of actin filaments (fig. 2.3C) was then used for each cell type to calculate the orientational order parameter (OOP) – a measure of alignment varying between 0 and 1, where 0 corresponds to isotropic distribution and 1 – to perfectly co-aligned filaments. Heat maps and histograms were created by detecting location of the actin filaments relative to the fluorescently labeled fibronectin pattern and calculating alignment data for each location separately. Actin and α-actinin-based masks were also used to determine the cardiomyocyte purity and the overall cell surface coverage.

3.3.5 iPSC and ESC Culture and Differentiation

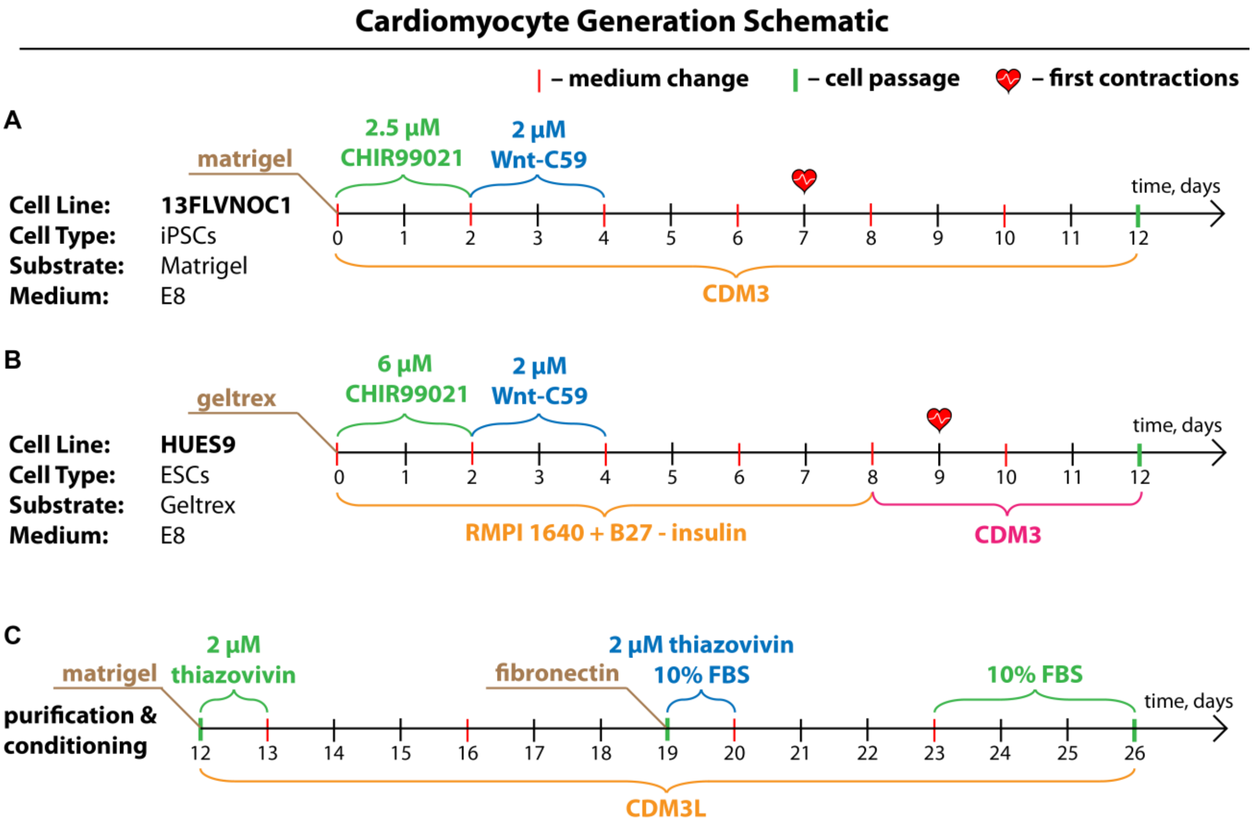

Figure 3.2. Schematic of cardiomyocyte generation from the ESC line HUES9 and the iPSC line 13FLVNOC1. A: Timeline of iPSC (13FLVNOC1 line) differentiation into cardiomyocytes. **B:** Timeline of HUES9 differentiation into cardiomyocytes. **C:** Timeline of post-differentiation metabolic purification and fibronectin conditioning of iPSC-CMs and ESC-CMs.CHIR99021 is a glycogen synthase kinase 3 (GSK3) inhibitor, Wnt-C59 is a Wnt inhibitor, FBS is fetal bovine serum, thiazovivin is a ROCK inhibitor.

Human iPSC line 13FLVNOC1 was provided by Prof. Wu's lab at Stanford University, USA. iPSCs were cultured in Matrigel-coated 6-well plates in E8 medium and passaged using EDTA. iPSC-CM passaging was done by incubating the cells in 1 mg/mL collagenase type II for 40 min (1st passage after differentiation), 20 min (2nd passage after differentiation), or 15 min(3rd and higher passage after differentiation), after which collagenase was removed from the wells and replaced with TrypLE Express for 5 min. Then cells were detached using a 1 ml pipette, diluted with DMEM/F12, centrifuged at 200 G for 7 min, and resuspended in the appropriate medium.

The monolayer-based iPSC differentiation technique was based on the previously described one developed in Prof. Wu's group with minor changes (fig. 3.2A). Shortly, iPSC-CMs were seeded on Matrigel-coated 6-well plates at 125,000 cell/well, cultured in the E8 medium for 4 days until they reach ~85% confluency, then the medium was changed to CDM3 (DMEM/F12, 500 µg/mL human recombinant albumin, 213 µg/mL ascorbic acid-2-phosphate) supplemented with 2.5 µM of the GSK3 inhibitor CHIR99021 for the first 2 days, then switched to CDM3 with 2µM of the Wnt inhibitor Wnt-C59 for the next 2 days, and then cells were cultured in CDM3 for the next 8 days with the medium changed every other day. Spontaneous cardiomyocyte beating was normally observed at day 7. Cardiomyocyte purity of >85% was consistently achieved using this technique.

Human ESC line HUES9 was provided by Prof. van der Meer's group at University of Groningen, Netherlands. ESCs were cultured in Geltrex-coated 6-well plates in E8 medium and passaged using TrypLE Express. ESC-CM passaging was done by incubating cell in TrypLE Express for 15 minutes, then detaching cells using a 1 mL pipette, diluting them with DMEM/F12, centrifuging at 200 G for 7 minutes, and resuspending in the appropriate medium.

The ESC monolayer-based differentiation technique was adapted from Prof. van der Meer's group (fig. 3.2B). Shortly, ESCs were seeded on Geltrex-coated 6-well plates at 125,000 cell/well, cultured in the E8 medium for 4 days until they reach ~85% confluency (alternatively, they can be seeded at 320,000 cells/well and cultured for 3 days), then the medium was changed to RPMI+B27 w/o insulin supplemented with 6 µM CHIR99021 for the first 2 days, then switched to RPMI+B27 w/o insulin with 2µM Wnt-C59 for the next 2 days, and then cells were cultured in RPMI+B27 w/o insulin for 4 days and CDM3 for the next 4 days with the medium changed every other day. Spontaneous cardiomyocyte beating was normally observed at day 9. Cardiomyocyte purity of >80% was consistently achieved using this technique.

On day 12 of differentiation, iPSC-CMs and ESC-CMs were passaged into Matrigel-coated wells and purified in CDM3L medium (CDM3 medium without glucose, supplemented with 5 mM sodium DL-lactate), according to the previously described technique for 7 days (fig. 3.2C).[72] After that cells were passaged into fibronectin-coated 6-well plates and cultured for 7 days in CDM3L supplemented for the first 24 hours with 10% FBS and 2 µM thiazovivin and for the last 3 days with 10% FBS (fig. 3.2C). Then cells were detached from the substrate and seeded at the appropriate density onto micropatterned PDMS-coated coverslips in CDM3 medium supplemented with 20% FBS and thiazovivin for the first 24 hours, after which medium was switched to CDM3 with 10% FBS.

For the experiments with matured cardiomyocytes, during the fibronectin conditioning stage iPSC-CMs were cultured in CDM3 media supplemented with 20 ng/mL 3-iodo-L-thyronine according to the previously reported technique.[79]

3.4 Results

3.4.1 Cardiomyocyte Conditioning

To generate cardiomyocytes for engineering human tissues, we differentiated the iPSC line, 13FLVNOC1, and the ESC line, HUES9, using a monolayer-based technique that provides >80% pure cardiomyocyte population (fig. 3.2A, B). The iPSCs were generously provided by Prof. Wu's group at Stanford University, and the ESCs – by Prof. van der Meer's group at the University of Groningen, Netherlands. After differentiation, we metabolically purified these cardiomyocytes to >95% by substituting glucose with sodium-L-lactate. However, seeding these purified cardiomyocytes on fibronectin patterns resulted in relatively low cell spreading. Additionally, cells did not follow the fibronectin patterns, but rather maintained a round shape and showed the tendency to cluster together in dense beating aggregates. We attributed this behavior to low expression levels of fibronectin-specific integrins $\alpha_5\beta_1$ and $a_v\beta_3$, although further studies are required to prove that. In order to improve cardiomyocyte spreading on fibronectin patterns, we cultured them on uniformly coated fibronectin for 7 days with the addition of 10% fetal bovine serum (FBS) to the media for the last three days (fig. 3.2C). After such conditioning, human cardiomyocytes showed significantly increased attachment and spreading rates on fibronectin patterns.

3.4.2 HUES9-derived Cardiomyocytes on Fibronectin Patterns

To form aligned confluent monolayers, we seeded conditioned HUES9-derived cardiomyocytes (HUES9-CMs) on the 20x20 fibronectin pattern at a density of 400,000 cells/cm^2. We found that even conditioned HUES9-CMs showed lower spreading compared to chick cardiomyocytes and formed thicker multi-layered tissues (fig. 3.3A). Imaging these tissues using confocal microscopy revealed that the bottom layer of the tissue was aligned along the fibronectin features (fig. 3.3B), but the alignment became worse in the middle of the tissue (fig. 3.3C) and turns into an almost completely isotropic distribution of sarcomeres near the top of the tissue (fig. 3.3D). This indicated that although the 20x20 pattern stimulates HUES9-CMs alignment, its effect decreases with the distance between actin filaments and the substrate.

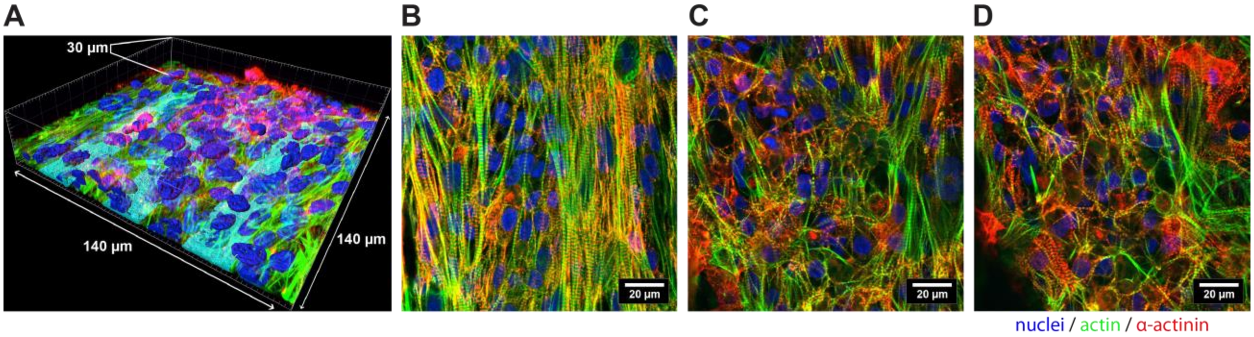

Figure 3.3. HUES9-CMCFB tissues on the 20x20 pattern. A: 3D confocal image showing that the tissue is multi-layered. Fibronectin pattern and nuclei were segmented for better visualization using Imaris software. **B:** A slice from the z-stack of the bottom layer of the tissue near the substrate shows high sarcomere alignment. **C:** A slice of the middle of the tissue shows significantly lower sarcomere alignment compared to the bottom layer. **D:** A slice of the top of the tissue furthest from the substrate shows almost no alignment. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 20 µm.

The same results were observed for the 2x2 and the biomimetic patterns. Confocal images of the bottom layer of HUES9-CMs on all three patterns showed sarcomere alignment (fig. 3.4A, B). However, in the middle and the top parts of the tissue the alignment was decreased, resulting in low overall alignment. We attribute this behavior to the relative low fibronectin-specific integrin levels of these cells compared to the expression of cell-cell adhesion molecules, such as N-cadherin, which results in a higher affinity of HUES9-CMs to each other than to the substrate. However, further studies are necessary to verify this hypothesis. The inability of HUES9-CMs to form a confluent layer of single cells complicates their use for engineering aligned cardiac tissues via patterning fibronectin features on the substrate.

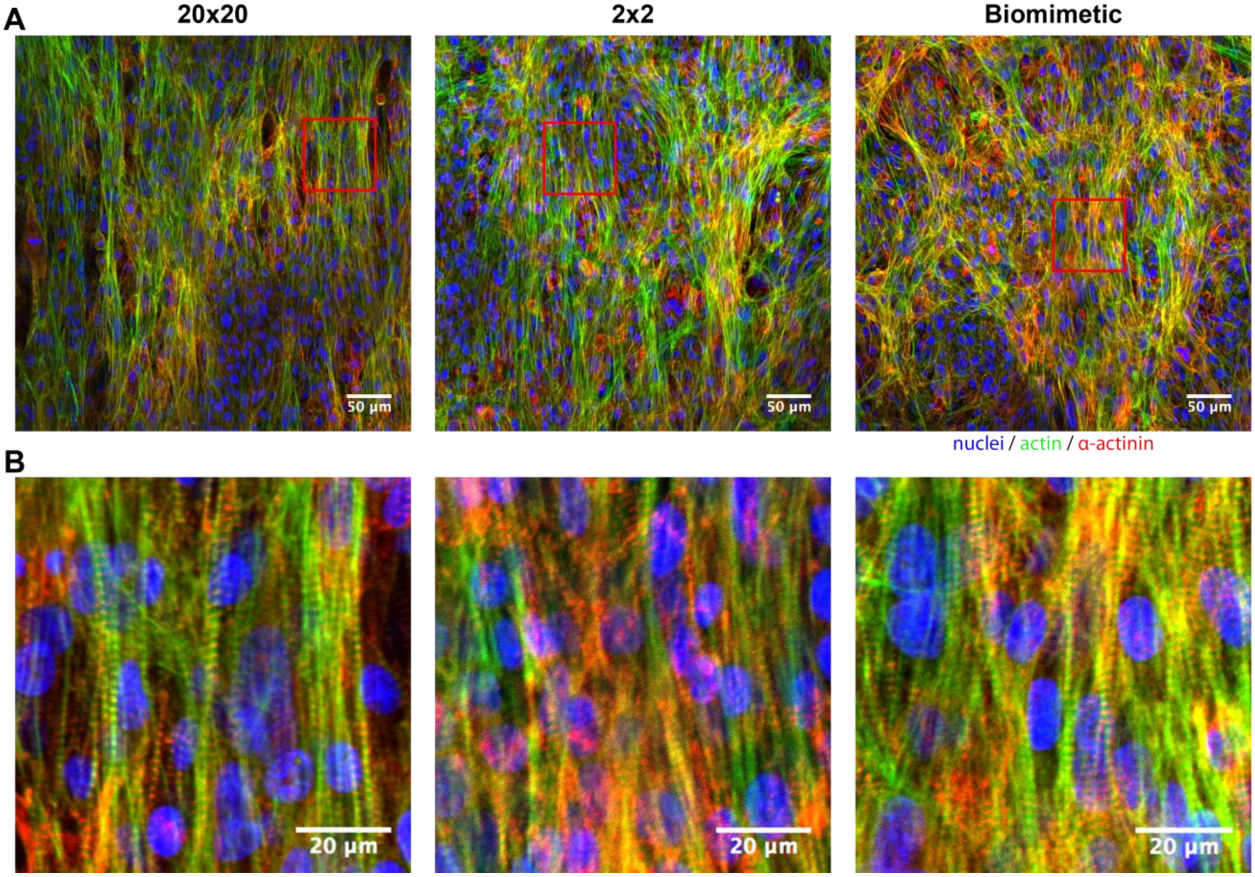

Figure 3.4. The bottom actin layer of HUES9-CMs on fibronectin patterns. A: The bottom layer of actin filaments of HUES9-CMs on fibronectin patterns. Scale bars: 50 µm. **B:** zoomed in areas of the areas in (A) marked with the red square. Scale bars: 20 µm. Nuclei are shown in blue, actin – green, α-actinin – red.

3.4.3 iPSC-CMs on Fibronectin Patterns

By seeding iPSC-CMs on fibronectin patterns, we found that they spread more readily on the patterns and therefore, a lower density is required to form a confluent monolayer. This makes the cardiomyocytes derived from the iPSC line 13FLVNOC1 more suitable for engineering aligned monolayers on fibronectin patterns compared to HUES9-CMs. We compared iPSC-CM response to all three patterns to the response of chick cardiomyocytes at high and low densities. We seeded iPSC-CMs on fibronectin patterns at low (60,000 cells/cm^2) and high (250,000 cells/cm^2) densities. After three days of incubation samples were stained and imaged using confocal microscopy, which revealed that iPSC-CMs have a more circular, less elongated shape compared to chick cardiomyocytes (fig. 3.5A, B). Actin OOP analysis (fig. 3.6) revealed overall lower alignment of iPSC-CMs compared to chick CMs, which can be attributed to the relative iPSC-CM immaturity, which is common to PSC-derived cells. Further, iPSC-CM OOP was found to be much more variable when comparing cells from different batches, which is an indication of inconsistencies in cell phenotype between differentiations with respect to their response to the patterns. Additionally, we found that cardiomyocyte OOP is higher at low density for the line patterns (although only the 2x2 pattern showed statistically significant difference) and the same as the high density OOP for the biomimetic pattern indicating that the influence of cell-cell and cell-substrate interactions on iPSC-CM alignment is different compared to chick cardiomyocytes.

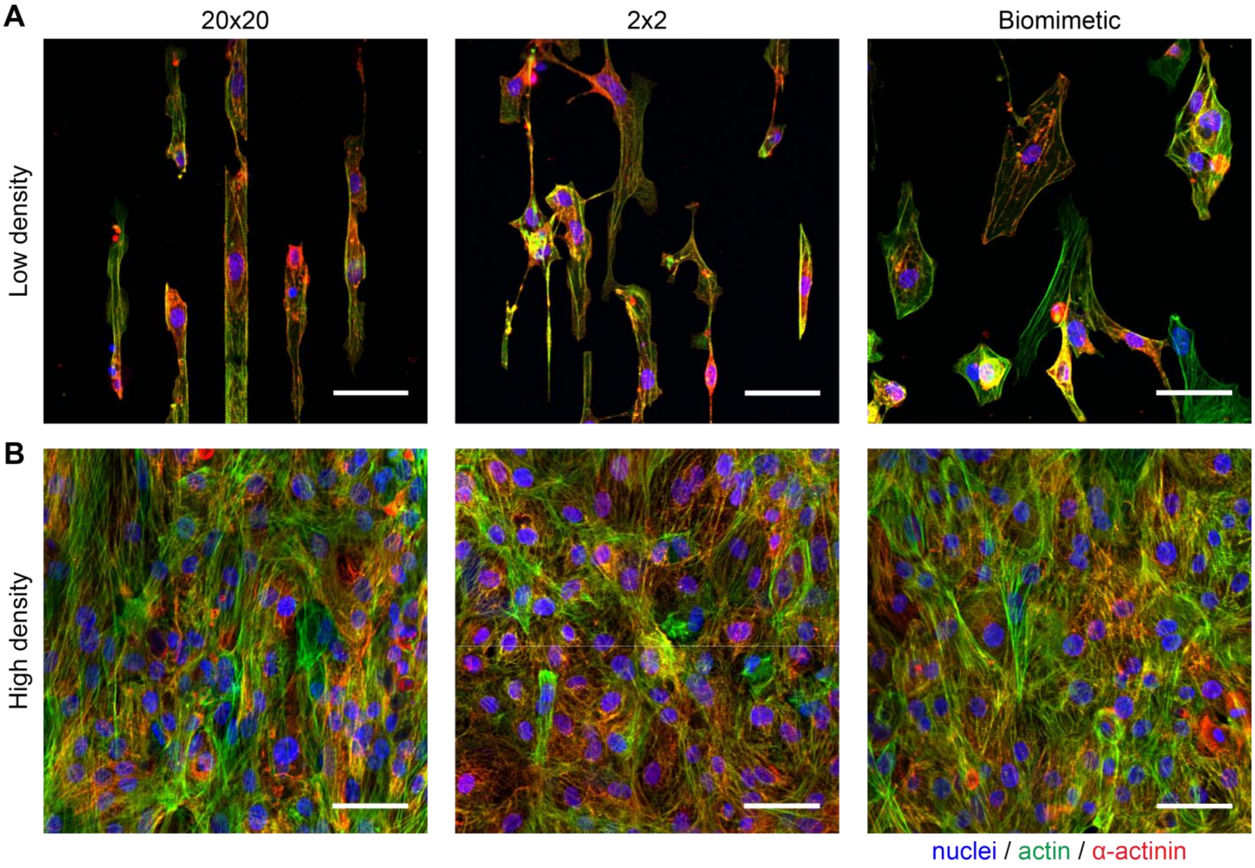

Figure 3.5. Representative images of immunostained iPSC-CMs at low (A) and high (B) cell densities on fibronectin patterns show overall poor sarcomere organization and more circular shape of iPSC-CMs compared to chick cardiomyocytes. Nuclei are shown in blue, actin – green, α-actinin – red. Scale bars: 50 µm.

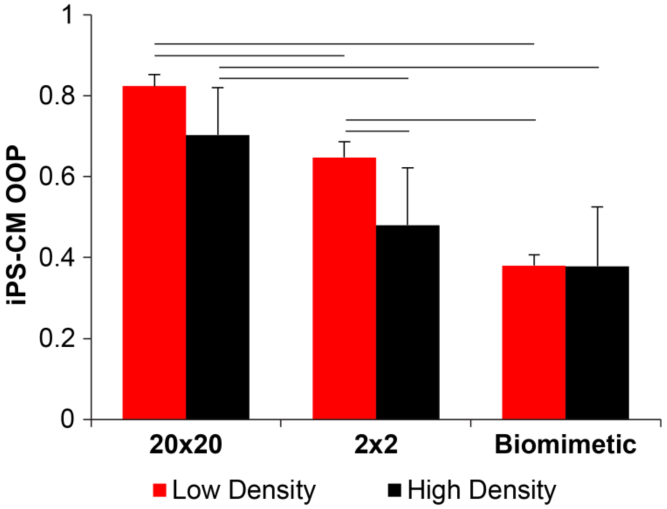

Figure 3.6. iPSC-CM OOP analysis at low and high cell densities. iPSC-CMs show lower overall alignment compared to chick cardiomyocytes as well as significant differences in the effect of cell density on OOP. This can be attributed to the immaturity of iPSC-CMs, species-specific differences in cell phenotype, and the fact that chick cardiomyocytes were harvested from embryonic heart, while iPSC-CMs were cultured in vitro prior to the experiments. Horizontal lines correspond to statistically significantly different values (two-way ANOVA, $p<0.05$, $N \geq 3$). Error bas represent standard deviations.

3.4.4 Ca^{2+} Imaging and Contraction Wave Propagation Measurement of iPSC-CM Monolayers

High purity of cardiomyocytes generated from iPSCs eliminates the problem that prevented reliable measurement of contraction propagation speed in chick tissues. Contraction cycle in cardiomyocytes involves release of calcium from sarcoplasmic reticulum into the cytoplasm, where it can interact with troponin to reveal the myosin binding site on actin filaments and initiate contraction. During relaxation calcium is being pumped back into sarcoplasmic reticulum so it can be used for the next contraction cycle. We used calcium-sensitive fluorophore to visualize this dynamic and measure the propagation speed of the contraction wave in iPSC-CM tissues. For electrical stimulation, one needle-shaped platinum electrode was placed about 2 mm above the tissue and another electrode of an arc shape was placed around the circumference of the plate. A 50 µs wide square-shaped signal was sent every 1 second to the electrodes to initiate tissue contraction under the needle electrode. Due to the high propagation speed of the contraction wave, a high frame rate of imaging was required to provide sufficient resolution for an accurate speed measurement. We used the line scan mode of a Leica confocal microscope to achieve imaging at 500 FPS, which corresponds to 2 µs per frame. The scanned line was chosen to be oriented in parallel with the direction of the contraction propagation to ensure that the measured speed is the true wave front propagation speed. A typical line scan video can be represented as a 2D image with the horizontal axis corresponding to the coordinate along the scanned line and the vertical axis – to the time (fig. 3.7A). Due to the high noise of the signal, it had to smoothed before analysis. This was done by applying a 2D Gaussian filter in the space and time domains to achieve the best quality of the signal (fig. 3.7B). The wave fronts were detected by finding maxima of the derivative of pixel brightness over time (fig. 3.7C). To gather sufficient amount of data for accurate analysis, each line scan video contained 10-20 contractions. For each contraction the location of the wave front was detected as a function of time and centered around 0 ms, after which the data from different contractions was combined into one set (fig. 3.7D). To extract the wave front propagation speed, we applied linear regression to the wave front trajectory data and measured the slope of the regression line. Reliable results can only be achieved if the wave propagation is linear, like in fig. 3.7D. However, due to irregularities in the tissues, contractions can propagate in various non-linear directions and even change trajectory from contraction to contraction, like in the fig.

3.7E, in which case the final plot contains data about two different wave front propagation trajectories (fig. 3.7F, G). These non-uniformities make the wave front propagation speed measurements inaccurate as the speed does not stay constant. This problem can be resolved by optimizing the cell seeding density, ensuring uniform distribution of cells, and taking line scan videos over a larger distance so that the observed irregularities can be averaged.

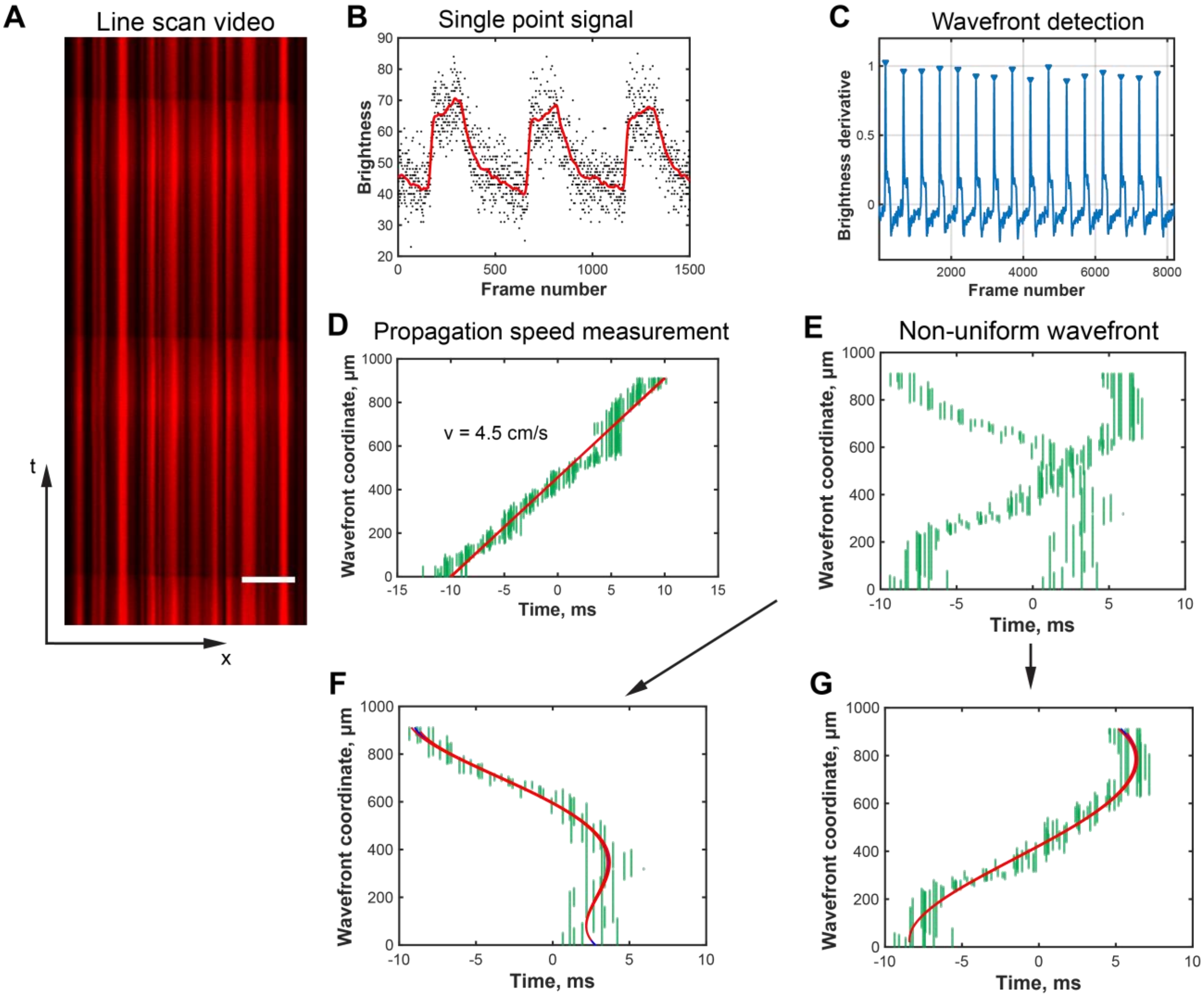

Figure 3.7. Contraction propagation speed measurement using line scan videos. A: Representative line scan video of calcium fluorophore in iPSC-CMs paced at 1 Hz. Scale bar: 200 μm. **B:** Representative plot of the brightness on one pixel over time. Red line shows the average signal. **C:** Wavefront detection by identifying maxima of the derivative of pixel brightness. **D:** Typical plot of wavefront propagation along the scanned line. Shown data is a combination of the data from 16 contraction cycles centered on 0 ms. Propagation speed is the slope of the fitted line calculated by linear regression of the data. **E:** An example of non-uniform propagation with a non-linear profile, where different contractions have different

trajectories. **F,G:** The trajectories shown in **(E)** separated to analyze the propagation speed of each trajectory. However, due to the non-linear nature of the trajectories, the propagation speed is not constant and cannot be measured reliably.

3.4.5 The role of CFBs on iPSC-CM alignment on fibronectin patterns

When comparing engineered tissues produced with chick embryonic cardiomyocytes and human iPSC-CMs, cell purity is an important factor that can potentially influence alignment. As we indicated earlier, tissues produced using chick embryonic cardiomyocytes contain ~30% non-cardiomyocytes, majority of which are fibroblasts, which decreases the cardiomyocyte OOP in a monolayer. On the other hand, due to high efficiency of the differentiation protocol, iPSC-CMs are obtained at a very high purity that cannot be achieved for chick CMs. In order to make the comparison between embryonic chick and human iPS-derived cardiac tissues more adequate, we added primary human CFBs to the iPSC-CM population. This also allowed us to determine the effect of CFBs on the alignment of iPSC-CMs in monolayer.

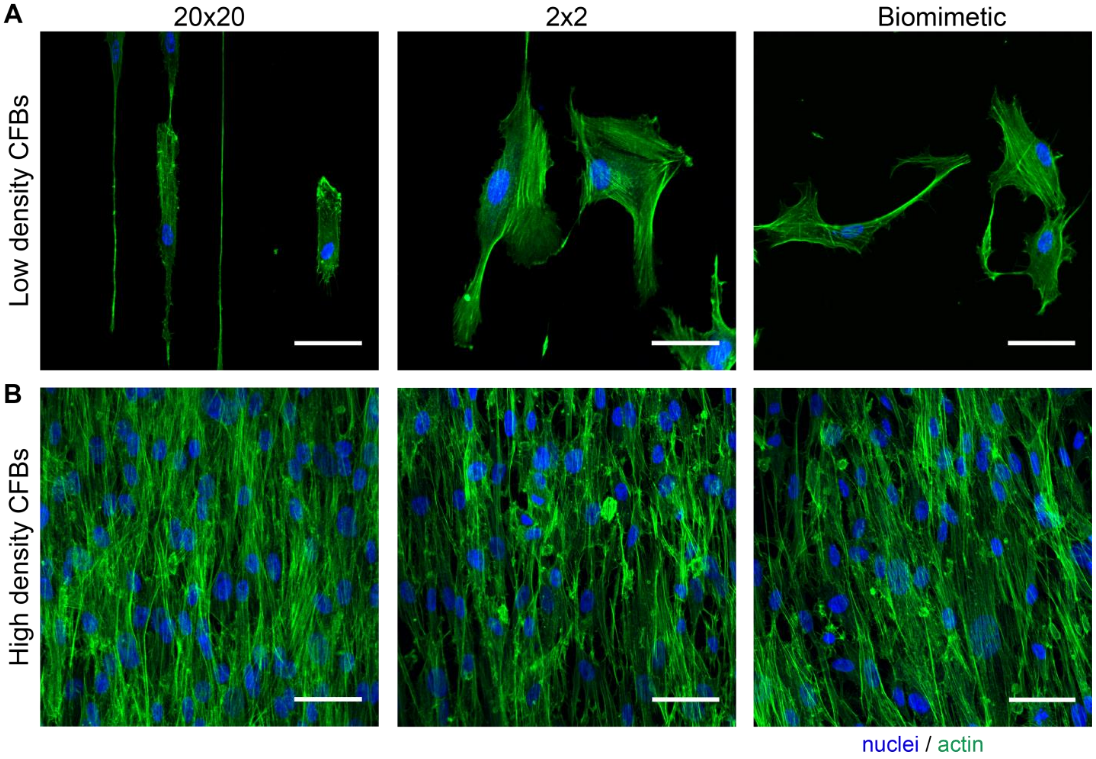

Figure 3.8. CFBs on fibronectin patterns A: at low density show pattern-dependent alignment, **B:** at high density show high alignment for all three patterns. Nuclei are shown in blue, actin – green. Scale bars: 50 µm.

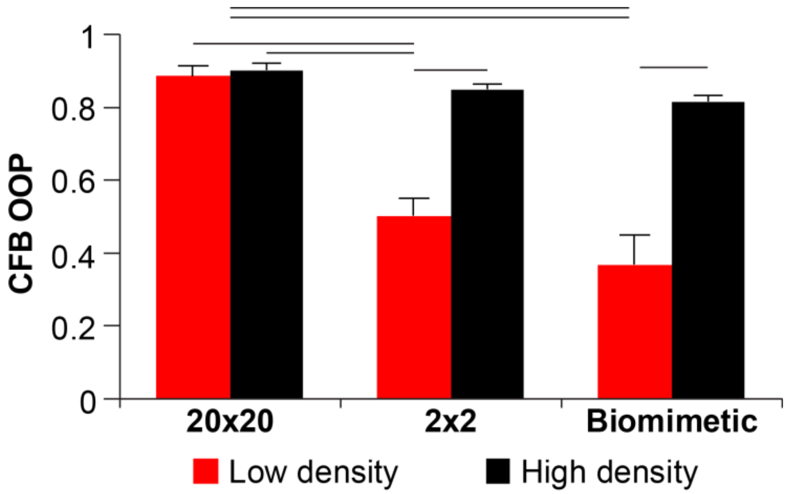

Figure 3.9. Alignment analysis of CFBs on fibronectin patterns shows that while at low density CFB OOP is pattern-dependent, at high density it's high for all three patterns. This suggests the potential of CFBs to improve iPSC-CM alignment. Error bars represent standard deviations, horizontal lines indicate statistically significant difference (2-way ANOVA, $p < 0.05$)

First, we analyzed the CFB response to the fibronectin patterns at low (30,000 cells/cm^2, fig. 3.8A) and high (250,000 cells/cm^2, fig. 3.8B) densities. OOP analysis showed that CFBs have low alignment at low density and significantly higher alignment at high density on all patterns except the 20x20, which showed high alignment at both densities (fig. 3.9). This density-dependent response is similar to that of chick fibroblasts, however, the fact that the high density CFB OOP is higher than that of iPSC-CMs indicates the potential of CFBs to improve iPSC-CM alignment, which is the opposite of what was observed for chick cardiomyocytes.

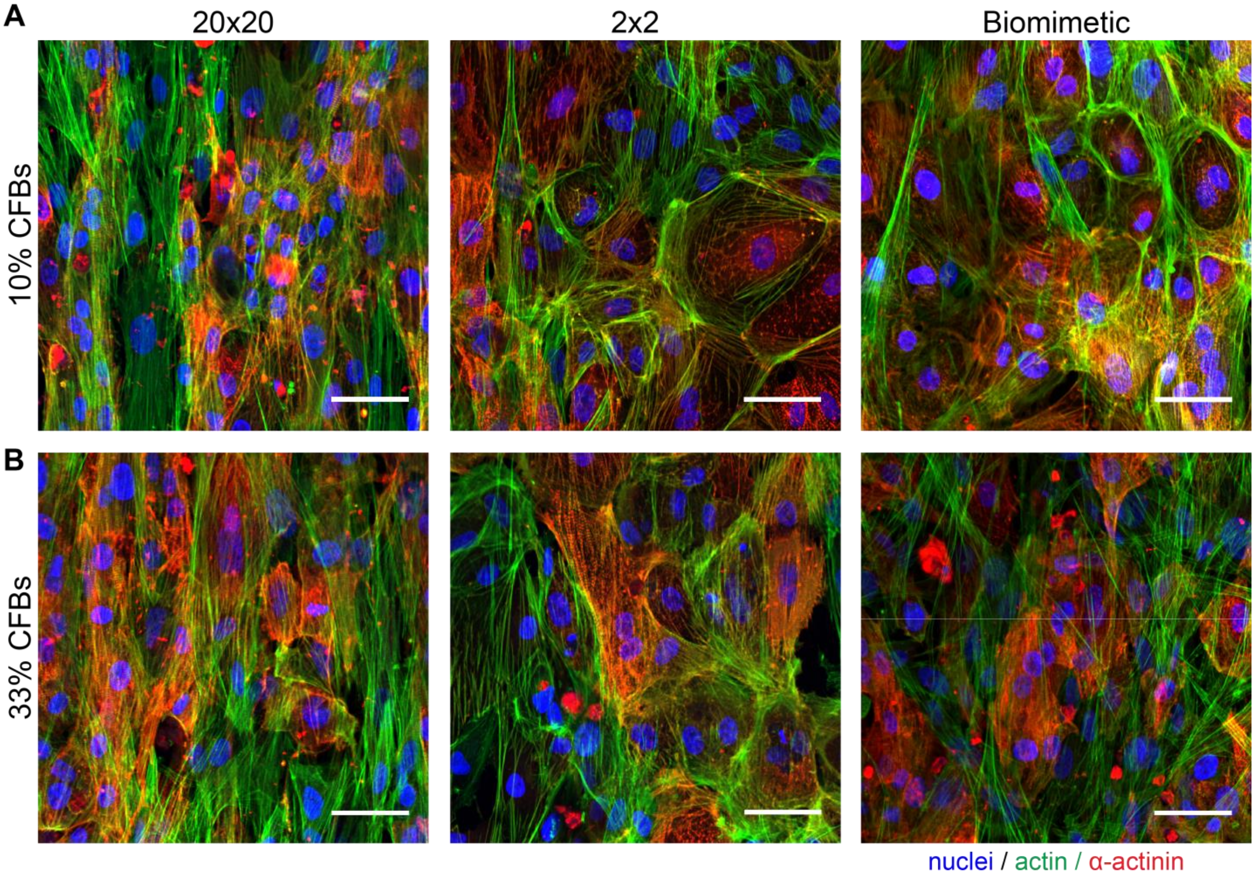

Figure 3.10. Representative images of iPSC-CMs seeded with 10% (A) and 33% (B) CFBs on fibronectin patterns. Fibroblasts can be distinguished from cardiomyocytes by the lack of α-actinin. Scale bars: 50 µm.

To determine the CFB influence on iPSC-CM alignment, we added 10% (fig. 3.10A) and 33% (fig. 3.10B) of CFBs to iPSC-CMs and seeded them on the fibronectin patterns. After 3 days of incubation the amount of CFBs increased due to their proliferation to about 25% and 55% respectively, which was assessed using α-actinin staining analysis in MATLAB (fig. 3.11A). It is worth noting that the final cardiomyocyte purity was dependent on the pattern, which can be attributed to the differences in cardiomyocyte adhesion and spreading rates leading to different amount of free area for fibroblasts to proliferate and spread into. Actin alignment analysis showed that the increase of CFB fraction increased iPSC-CM OOP for all three patterns, although the statistical significance was achieved only by the 20x20 and the 2x2 pattern at 33% CFBs (fig. 3.11B). This increase is most likely caused by cell-cell interactions between CFBs and iPSC-CMs that result in achieving an alignment that lies in between the OOPs of the pure populations of each cell type. However, it is also possible that this effect was caused by CFB-generated ECM.

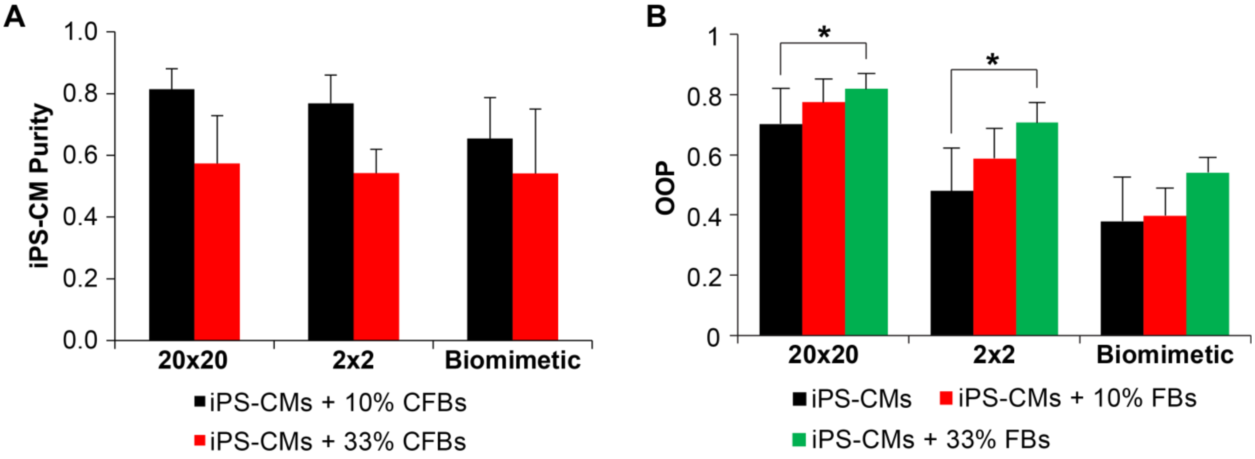

Figure 3.11. Analysis of iPSC-CMs mixed with CFBs on fibronectin patterns. A: Final iPSC-CM purity of cardiac tissues containing 10% and 33% CFBs upon seeding after three days of culture. The final purity is lower than the initial purity due to CFB proliferation. **B:** OOP of iPSC-CMs with CFBs on fibronectin patterns shows increase of tissue alignment with the addition of CFBs for all patterns. “*” indicates statistically significant difference ($p < 0.05$, one-way ANOVA).

To assess the potential influence of CFB-generated ECM on cardiomyocyte alignment, we cultured monolayers of CFBs on fibronectin patterns for 3 days, during which cells generated significant amounts of ECM (fig. 3.12B). We were able to separate patterned fibronectin (fig. 3.12A) and CFB-generated one (fig. 3.12B) during imaging by using pre-labeled fibronectin for patterning. Interestingly, despite of the apparent alignment of CFB-generated fibronectin fibrils, the direction of alignment did not strictly correspond to the direction of patterned fibronectin alignment, but rather followed the orientation of CFBs.

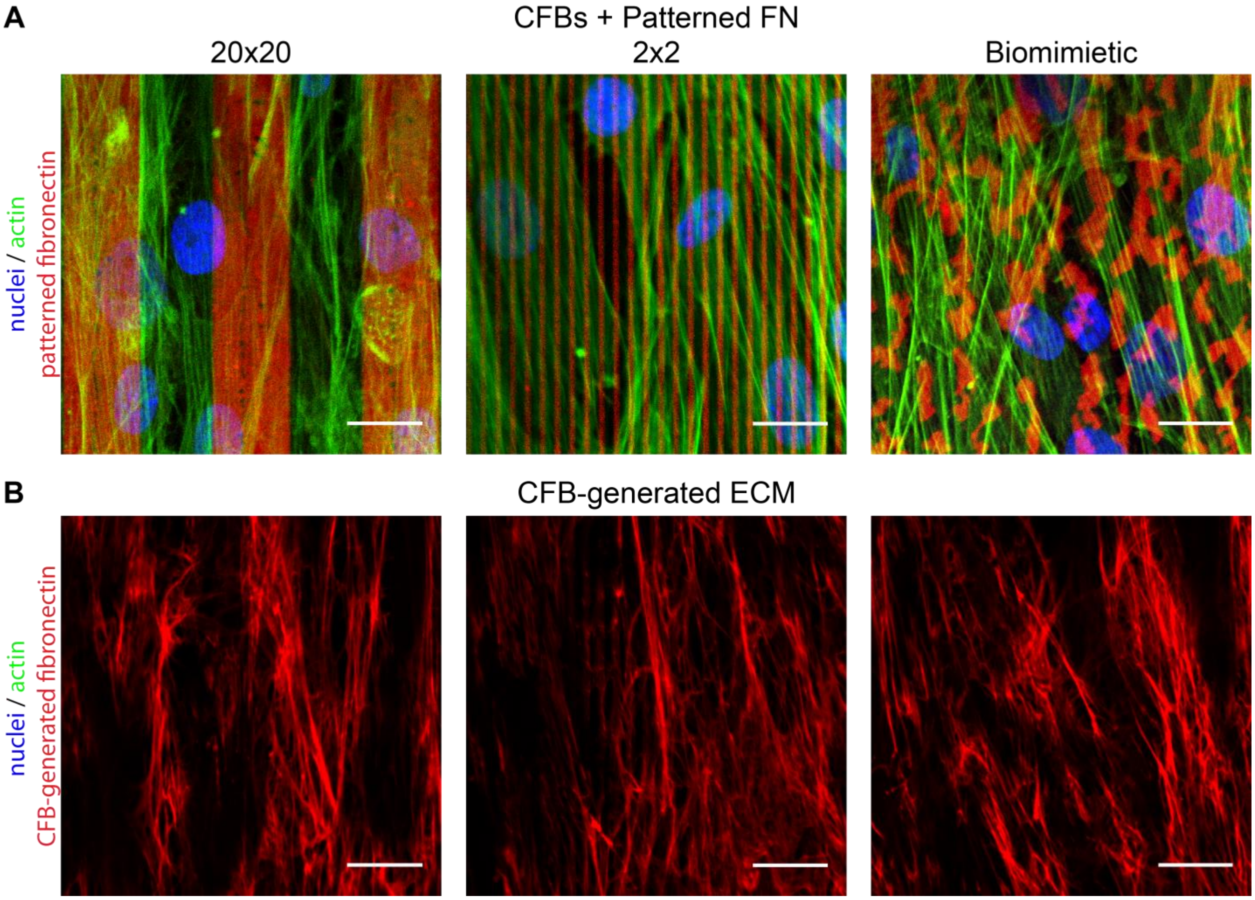

Figure 3.12. CFBs and CFB-generated ECM on fibronectin patterns. A: CFBs on fibronectin patterns at high density. **B:** Fibronectin generated by CFBs corresponding to the images shown in **(A)** show significant amount of generated fibronectin for all three patterns. Scale bars: 20 µm.

To remove CFBs from the substrate without significantly damaging the ECM, they were lysed by the incubation in 2M Urea solution for 15 minutes. Stained images of the resulting substrates (fig. 3.13A) showed that this process effectively removes cell components while preserving significant amounts of ECM, although some amount of fibronectin loosely attached to the substrate was removed during pipetting. Next, iPSC-CMs were seeded on these substrates at 250,000 cells/cm^2 to form confluent monolayers the same way it was previously done for the regular patterned substrates (fig. 3.13B). As the ECM remaining after CFB removal had fibrillary structure, we were able to analyze its alignment and found it to be overall lower than the alignment of CFBs that generated it with the exception of the 20x20 pattern that had lower amount of CFB-generated fibronectin, potentially due to the weaker attachment to the substrate (fig. 3.14A). Alignment analysis revealed that with the exception of the 20x20 pattern that showed no difference, iPSC-CM OOP was significantly lower on the substrates with CFB-generated ECM (fig. 3.14B). This result serves as the evidence of CFB-generated ECM not being responsible for the increase of OOP upon the addition of CFBs to iPSC-CMs in cardiac monolayers, and therefore this increase is most likely due to fibroblast-cardiomyocyte interactions.

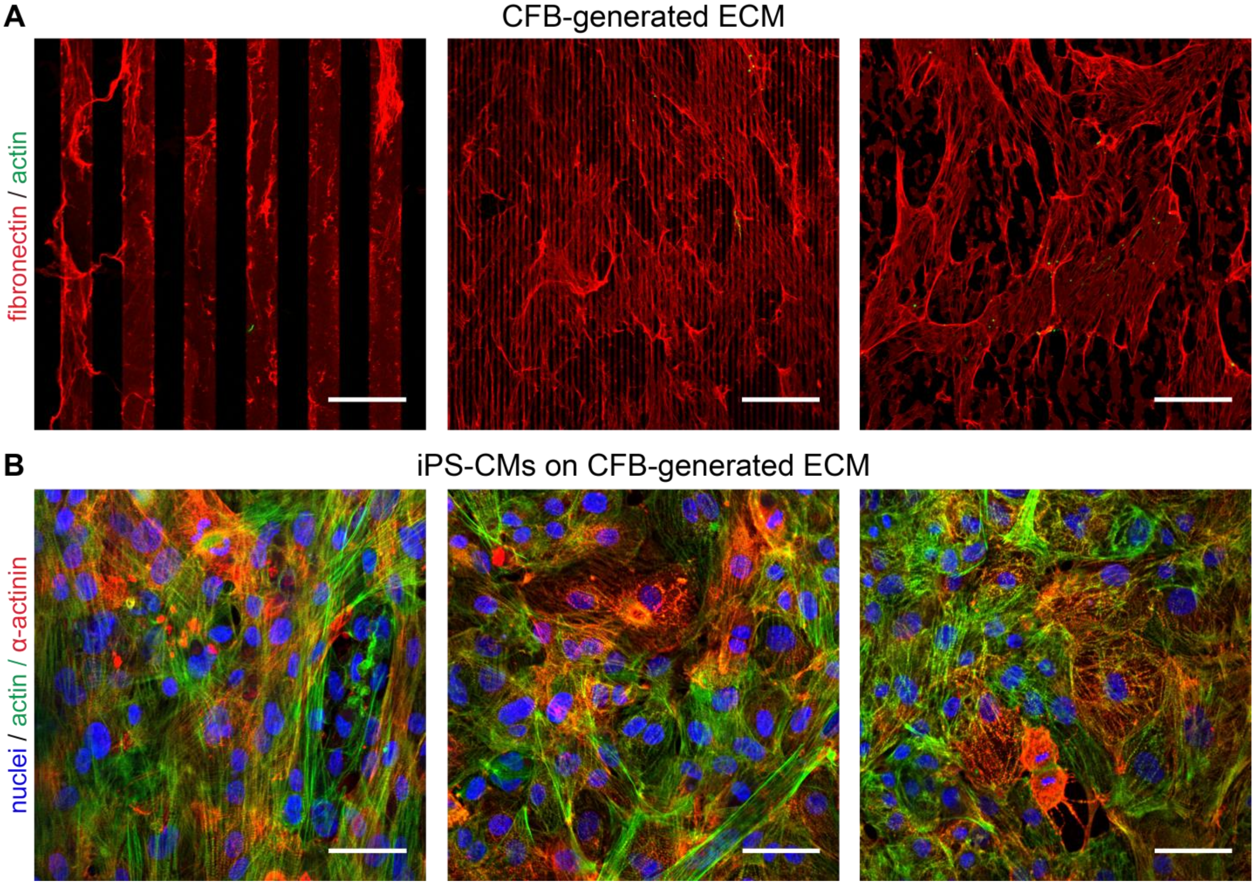

Figure 3.13. CFB-generated ECM as a substrate for iPSC-CM monolayers. A: Fibronectin left on the patterned substrates after CFB removal. 20x20 pattern has significantly less CFB-generated fibronectin, likely due to the weaker attachment to the original fibronectin pattern. **B:** iPSC-CMs on CFB-generated ECM. Scale bars: 50 µm.

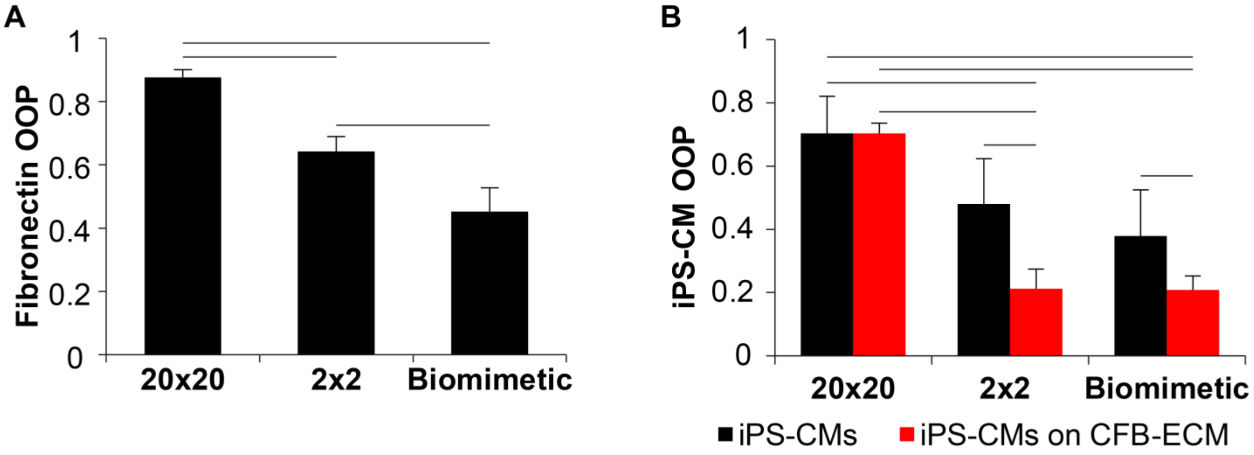

Figure 3.14. Alignmnet analysis of CFB-generated fibronectin and iPSC-CMs on CFB-generated ECM. A: OOP analysis of fibronectin left on patterned substrates after CFB removal shows relatively low alignment compared to CFB actin alignment. **B:** OOP analysis of iPSC-CMs on CFB-generated ECM

shows that CFB-ECM decreases cardiac tissue alignment compared to the tissues formed on patterned fibronectin with the exception of the 20x20 pattern, which showed no significant difference, likely due to the low overall amount of CFB-ECM left after CFB removal. Scale bars: 50 µm. “*” indicates statistically significant difference ($p < 0.05$, one-way **(A)** and two-way **(B)** ANOVA).

3.4.6 T3-matured iPSC-CMs on fibronectin patterns

One of the major problems of using iPSC-CMs for tissue engineering is the immaturity of these cells characterized by many functional and structural characteristics, among which there are low cell aspect ratio, low amount of sarcomeres, poor sarcomere structure, and low contractile force. These characteristics are directly responsible for poor actin alignment of iPSC-CMs on fibronectin patterns compared to chick embryonic cardiomyocytes. Although there is currently no known protocol to achieve significant maturation without long-term electrical and mechanical stimulation, studies have shown that incubation with the T3 hormone (tri-iodo-L-thyronine) is an easy way to increase the amount of sarcomeres and the contractile force of cardiomyocytes. We sought to elucidate the effect of such maturation on iPSC-CM alignment on fibronectin patterns. To mature cardiomyocytes, we cultured them in CDM3 medium supplemented with 20 ng/mL of T3 for 7 days during the fibronectin conditioning stage. T3-matured iPSC-CMs were seeded on fibronectin patterns at high density (312,000 cells/cm^2) to form a monolayer (fig. 3.15A) and incubated for 3 days. A higher density compared to the non-matured iPSC-CMs was used due to the lower spreading that T3-matured iPSC-CMs showed. Actin alignment analysis of the produced tissues showed significant increase in OOP for the 2x2 and the biomimetic pattern compared to non-matured cardiomyocytes (fig. 3.16B). This supports our previous statement regarding iPSC-CM immaturity being responsible for low OOP of cardiac monolayers and we expect that further maturation of these cells would produce even larger increase of tissue alignment.

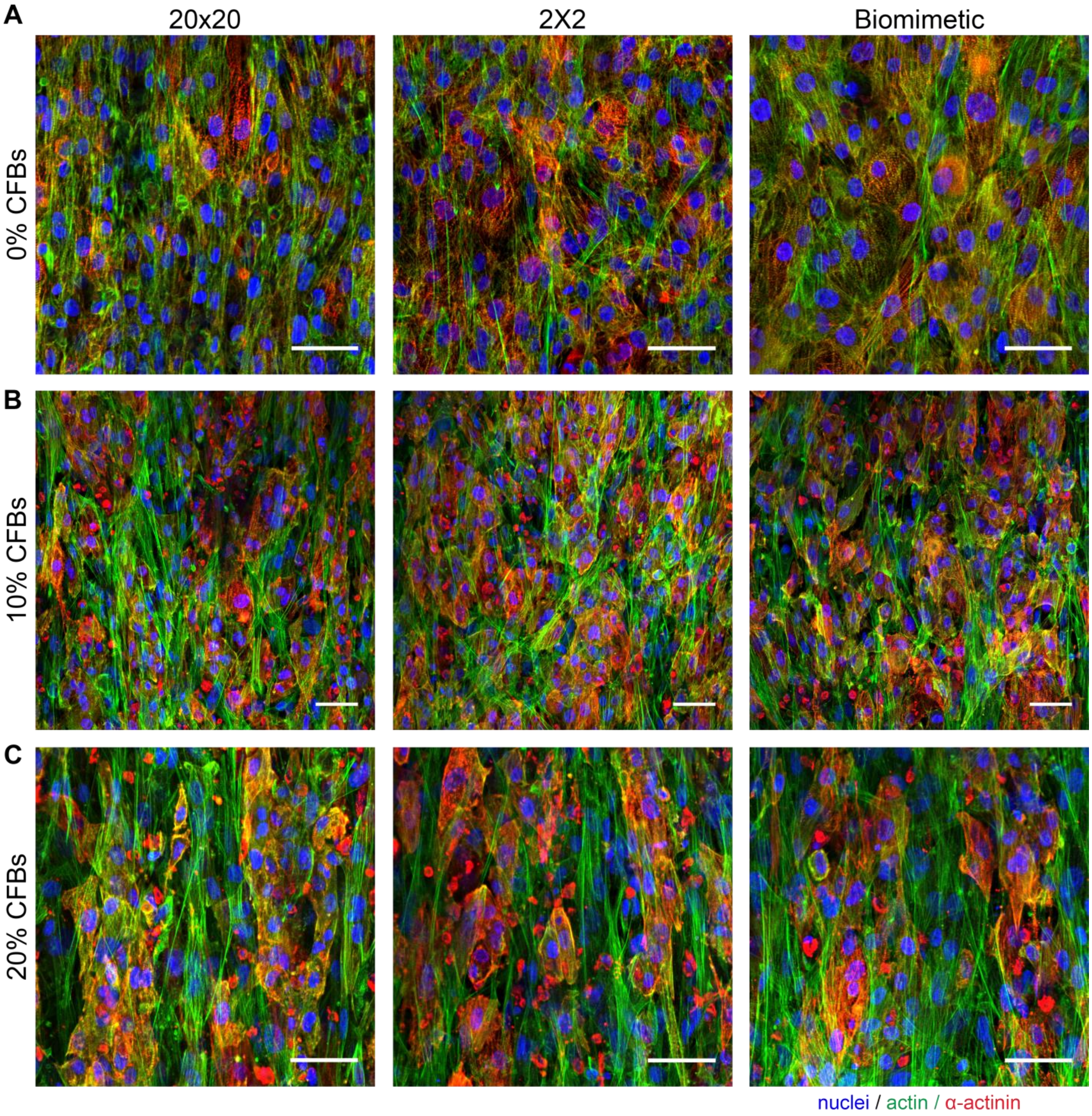

Figure 3.15. iPSC-CMs matured with T3 hormone with 0% (A), 10% (B), and 20% (C) of human CFBs on fibronectin patterns. T3-matured iPSC-CMs show higher sarcomere organization and more pronounced contractions in culture compared to non-matured cardiomyocytes.

Further, to test the effect of CFBs on T3-matured cells, we added 10% (fig. 3.15B) and 20% (fig. 3.15C) of CFBs to T3-matured iPSC-CMs, which resulted in about 25% and 40% CFBs correspondingly after 3 days of culture (fig. 3.16A). Actin alignment analysis showed that the addition of CFBs didn’t produce a statistically significant change of the alignment compared to pure T3-matured iPSC-CMs (fig. 3.16B). We attribute this result to the fact that OOP values of CFB-monolayers and T3-matured iPSC-CM monolayers are much closer to each other, so the effect of mixing these cells together on OOP would be smaller and harder to detect.

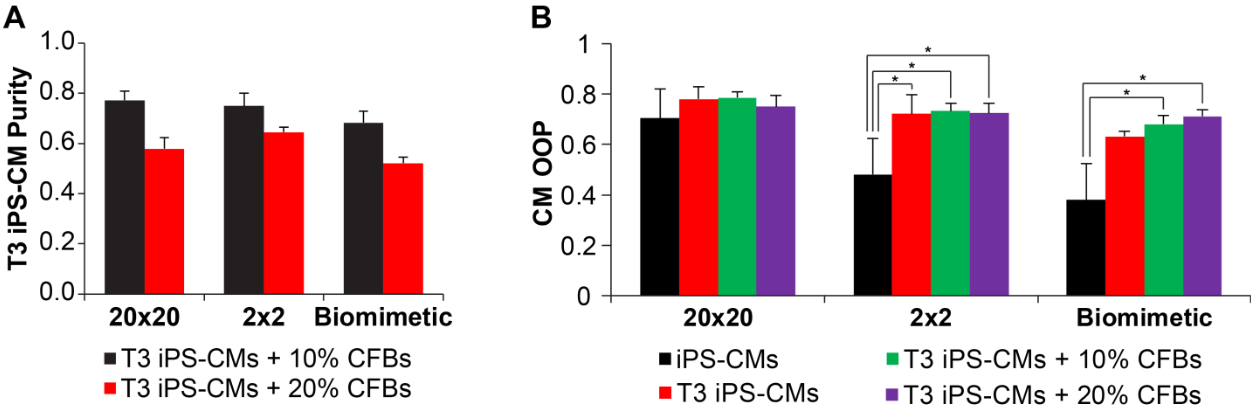

Figure 3.16. Analysis of T3-matured iPSC-CMs on fibronectin patterns. A: Final T3-matured iPSC-CM purity after 3 days of culture with 10% and 20% initial concentrations of CFBs. The final purity is lower than the initial purity due to CFB proliferation. **B:** OOP analysis of T3-matured iPSC-CMs showed that T3-maturation significantly increases cardiac tissue alignment for the 2x2 and the biomimetic pattern, while the addition of CFBs to T3-matured tissues has no significant effect on their alignment. “*” indicates statistically significant differences ($p < 0.05$, one-way ANOVA for each pattern seperetely)

3.4.7 Thick T3-matured iPSC-CM tissues

As we showed earlier, cardiomyocytes derived from different pluripotent stem cell lines can have significantly different response to the fibronectin patterns. Particularly, HUES9-CMs showed lower degree of spreading forming thicker multi-layered tissues with lower overall alignment due to the low alignment of the top and the middle parts of the tissue. In contract, iPSC-CMs showed significantly higher spreading and were capable of forming 100% confluent monolayers on all three patterns, which made them the preferred cell type for our studies. However, during the optimization of the seeding density, we found that the alignment of confluent iPSC-CM monolayers was highly susceptible to the cell density. Particularly, the highest alignment was achieved when using the lowest possible density that can still lead to a confluent monolayer, and if cell density was higher than the optimal, tissue alignment would decrease.

We sought to elucidate the reason behind such high susceptibility of tissue alignment to cell density in a monolayer as it was not observed in the chick tissues. To do so, we seeded T3-matured iPSC-CMs on fibronectin patterns at 400,000 cells/cm^2 that after 3 days of incubation formed thick multi-layered tissues. Using confocal microscopy, we acquired z-stacks of these tissues, which showed a similar decrease of sarcomere alignment as a function of the distance from the substrate (fig. 3.17A). Using a modified MATLAB script, we analyzed the alignment of these tissues slice-by-slice. The slice corresponding to the bottom of the tissue was chosen based on the actin coverage of the slice in order to ensure that when combining orientation data from different locations in the sample, the distance from the substrate stays consistent (fig. 3.17B). Slice-by-slice actin alignment analysis showed that tissue OOP decreases as a function of the distance from the substrate for all three patterns (fig. 3.17C). However, this decrease happens differently depending on the pattern. Particularly, the OOP on the 20x20 pattern stays constant within 2 µm from the substrate, while for the 2x2 pattern it starts to go down immediately. These results, similarly to the results observed for the HUES9-CM tissues, indicate that interactions between cells and the pattern stimulate actin alignment near the substrate, while their influence goes down with the distance between actin filaments and the substrate, where cell-cell interactions dictate tissue structure.

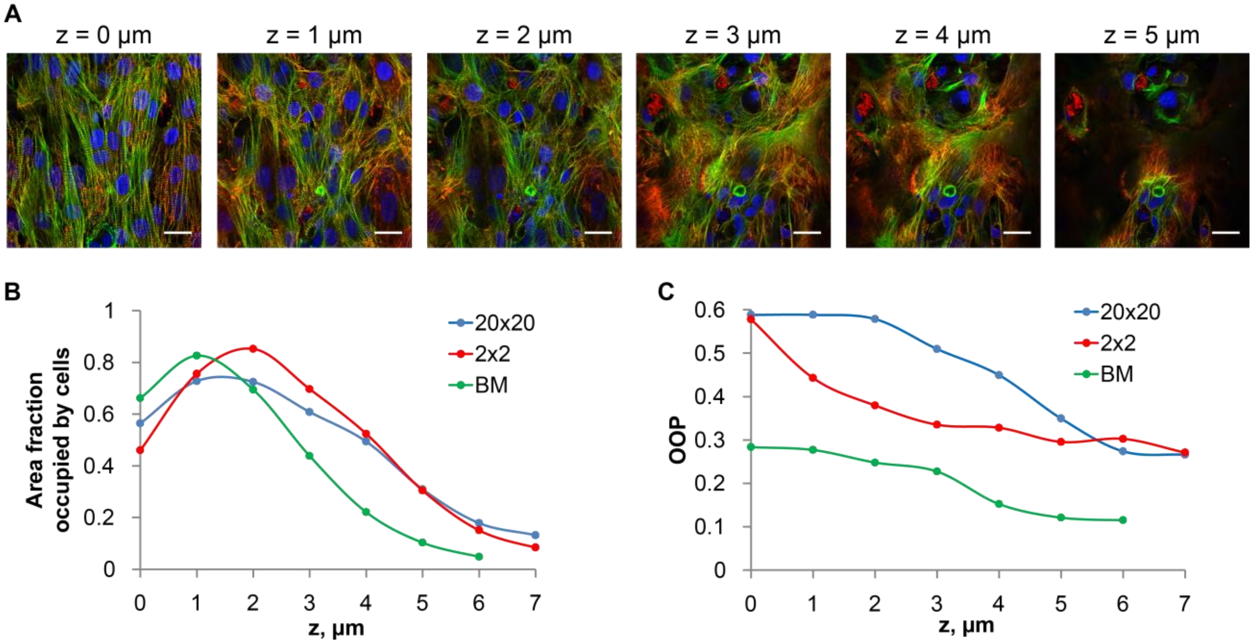

Figure 3.17. Alignment of multi-layered T3-matured iPSC-CMs on fibronectin patterns. A: Z-stack of multi-layered tissue shows decrease of sarcomere alignment with the increase of the distance from the substrate. Similar effect was observed for HUES9-CMs. Scale bar: 20 µm. **B:** Area fraction occupied by actin filaments as a function of the distance from the substrate. **C:** Actin OOP as a function of the distance from the substrate corroborated the alignment decrease observed in **(A)**. Interestingly, although the OOP at the substrate level is equal for the 20x20 and the 2x2 patterns, the 20x20 pattern maintains high alignment throughout much higher distance than the 2x2 pattern.

3.5 Discussion and conclusions

Using cardiomyocytes derived from human ESCs and iPSCs to engineer anisotropic 2D cardiac tissues, we found that there is significant variability between cardiomyocytes produced from the ESC line HUES9 and the iPSC line 13FLVNOC1. Specifically, iPSC-CMs showed higher degree of spreading on fibronectin patterns, which made them the preferred cell line for 2D cardiac tissue engineering. Other studies have also reported phenotype variability between different ESC lines and different iPSC lines.[65] Possible causes of these differences include different iPSC generation techniques, different culture conditions, different differentiation protocols, and different cell genotypes. Therefore, to minimize these differences, it is crucial to standardize iPSC derivation, culture, and differentiation protocols, and understand the factors responsible for the observed phenotype differences in more detail, so that experimental results acquired for one cell line can be applied to others.

Comparing iPSC-CM and chick cardiomyocyte response to the fibronectin, we found that newly differentiated iPSC-CMs required conditioning on fibronectin-coated substrates to show sufficient spreading for forming confluent monolayers, while chick cardiomyocytes didn’t. We attribute this difference to the fact that cardiomyocytes harvested from chick embryos are taken from fibronectin-rich environment of heart ECM, while iPSC-CMs are cultured on Matrigel, in which laminin is the most abundant cell adhesion protein.[80] Therefore, chick cardiomyocytes are more likely to have developed sufficient amounts of fibronectin-specific integrins to show adequate spreading compared to iPSC-CMs. However, further studies are needed to test this hypothesis.

Using cardiomyocytes derived from human iPSCs, we successfully engineered clinically relevant aligned cardiac tissues, although the immature iPSC-CM phenotype resulted in more circular cell shape with lower aspect ratio, and thus lower OOP. Additionally, large variations of iPSC-CM alignment were observed from differentiation to differentiation indicating inconsistent cell phenotype. Comparing low and high density OOP of iPSC-CMs revealed that the relationship between the alignment driven by cell-cell and cell-substrate interactions is different from that of chick cardiomyocytes. Particularly, the 20x20 and the 2x2 patterns showed higher OOP at low density indicating that $OOP_{cell\text{-}cell} < OOP_{cell\text{-}ECM}$ for these patterns, while the biomimetic pattern showed no density dependence of alignment indicating that $OOP_{cell\text{-}cell} \approx OOP_{cell\text{-}ECM}$.

The addition of human CFBs was found to increase iPSC-CM alignment, although the increase was significant only at high CFB fractions (30-50%), when iPSC-CM connectivity becomes insufficient for synchronous tissue contraction. However, this result shows the possibility of improving the alignment of immature cardiomyocytes by interaction with fibroblasts and suggests that cardiomyocyte-fibroblast interaction may play a role in myocardial alignment during the embryonic development. This is also supported by other studies that have shown that in 3D culture fibroblasts can facilitate the formation of functional tissue by remodeling ECM and creating favorable environment for promoting cardiomyocyte maturation and organization.[81-83] However, in our study CFB-generated ECM was not found to promote iPSC-CM alignment.

Finally, the increase of alignment we observed when using T3-matured iPSC-CMs compared to regular iPSC-CMs along with the fact that more mature chick cardiomyocytes show higher alignment on fibronectin patterns indicates that the immature phenotype is one of the main factors responsible for low alignment of iPSC-CMs. Considering the fact that T3-maturation results in a relatively modest improvement of structural and functional cell characteristics, such as cell area, cell anisotropy, twitch force, and contraction dynamics, we expect that using more advanced maturation procedures would further improve cell alignment and function of engineered cardiac monolayers. However, as no currently available technique can produce adult-like cardiomyocytes, determining new methods to mature iPSC-CMs is the key to engineering more functionally and structurally physiological tissues.

Chapter 4: Biomimetic Fibronectin Pattern with Sub-Micron Resolution

4.1 Abstract

Previously we showed that the interaction between chick cardiomyocytes and the biomimetic fibronectin pattern is significantly different compared to the line patterns commonly used to stimulate the formation of an aligned tissue. However, one of the main limitations of the biomimetic pattern we used was 2 µm resolution of the photomask, which makes it unable to recapitulate finer fibronectin structure in embryonic myocardium. This finer structure can potentially provide important cues for cardiomyocyte organization in a developing heart and, subsequently, in an engineered tissue. In this chapter we aimed to create a higher resolution biomimetic pattern from a single high-quality image of fibronectin in chick myocardium and transfer it with the sub-micron precision onto substrate. To achieve this resolution, we used two alternative approaches – a 3D-nano printing tool Nanoscribe to print a master-mold and a higher-resolution photomask to create the master mold via photolithography. Although significant progress has been made, further research is required to optimize the process of generating PDMS stamps of consistently high quality using either of these techniques, so that chick cardiomyocyte response to the developed biomimetic pattern can be tested.

4.2 Introduction

Fibronectin is the most abundant ECM protein in embryonic heart and plays an important role in myocardial development. However, it is still unknown if its role is limited to providing specific biochemical signals regulating cell behavior or whether fibronectin structure is an important part of the mechanical environment of the cells guiding cardiomyocyte organization and alignment. In the chapter 2 we studied chick cardiomyocyte response to the fibronectin pattern derived from the fibronectin structure in a chick embryonic myocardium to answer this question. The results revealed the mechanism of cardiomyocyte alignment on fibronectin patterns and specifically, the role of cell-cell and cell-ECM interactions in it. However, our biomimetic pattern had several limitations making it unable to recapitulate certain aspects of the heart ECM. One of the main limitations was the photomask resolution that prevented recapitulating fibrils smaller than 2 µm in diameter. Confocal imaging of chick myocardium revealed that the structure of the embryonic heart fibronectin is highly heterogeneous with fibril diameter ranging from 0.2 µm to 2-3 µm. Therefore, our biomimetic pattern was incapable of recapitulating a large portion of fibronectin fibrils, which can potentially have a significant impact on cardiomyocyte behavior.

In this chapter we aimed at creating a new biomimetic pattern with increased resolution that can closer recapitulate the fibronectin structure of embryonic myocardium. To do so, we used a single high-quality image of fibronectin in chick myocardium to derive a pattern that can be replicated in a 2D array suitable for engineering confluent cardiac monolayers. To create PDMS stamps for printing this pattern, we used two techniques: a 3D nano-printing tool Nanoscribe capable of achieving 0.4 µm resolution to print a master mold and a high-resolution photomask capable of achieving 0.7 µm resolution to create a master mold using photolithography. Although Nanoscribe provides higher resolution, it was found to suffer from inconsistent print quality over larger areas. Therefore, we moved to the second method using a photomask and photolithography to create more consistent master molds. This project is still in progress as we are optimizing the photolithography parameters to achieve the best mold quality. Once the PDMS stamps are made and the quality of pattern transfer is confirmed, we will test the response of chick cardiomyocytes and compare it to the previously used biomimetic pattern.

4.3 Materials and Methods

4.3.1 Fabrication of PDMS Stamps

PDMS stamps were used to pattern fibronectin onto substrates for cells. First, the master mold with the desired pattern was silanized by incubating it next to an open container of 2% dimethyldichlorosilane solution (PlusOne Repel-Silane ES, GE Healthcare) for 24 hours in a desiccator to decrease the mold adhesion to PDMS. Stamps were made by casting PDMS (Sylgard 184 base mixed 10:1 with the curing agent) onto the mold and curing it at 65 ^{0}C for 24 hours. After PDMS curing was finished, it was peeled off the mold and cut around the patterned area to make the stamps. If the mold didn't possess sufficient stability to withstand the thicker PDMS layer removal, which was the case for the Nanoscribe-printed molds, a thin PDMS layer was used to minimize the mechanical stress exerted upon the mold. To make a thicker stamp, additional PDMS was cast and cured on top of the thin PDMS stamp.

4.3.2 Substrate Preparation

Fibronectin-patterned PDMS-coated glass coverslips were used as substrates for cells. PDMS-coated coverslips were prepared using Sylgard 184 silicone elastomer according to the previously described procedure.[56] Briefly, Sylgard 184 base and curing agent were mixed at the mass ratio of 10:1 followed by the mixture defoaming in a "Thinky conditioning mixer". Spin-coating at 4000 RPM was used to coat coverslips with a thin PDMS layer. After spin-coating, coverslips were put in an oven at 65 ^{0}C for 24 hours in order to cure PDMS. Fibronectin patterns were microcontact printed onto the PDMS-coated coverslips according to a previously described technique with minor modifications (fig. 2.2(v)-(vii)).[57] Briefly, PDMS stamps were cleaned by sonication in 50% ethanol for 45-60 minutes and dried using pressurized nitrogen. Then the patterned side of each stamp was incubated in 50 µg/mL solution of human plasma fibronectin (unlabeled or labeled with Alexa Fluor 546 Maleimide fluorophore) for 60 minutes, washed in sterile water, and dried with a nitrogen gun. To transfer fibronectin from the stamp onto the coverslip, it was UV-Ozone treated for 15 minutes, and then the patterned side of the stamp was brought in contact with the coverslip for 5 minutes. The pattern transfer was verified for each stamp by confocal microscopy using fluorescently labeled fibronectin. Before cell seeding, all patterns were incubated in 1% w/v solution of Pluronic F127 to reduce cell adhesion in between fibronectin features and improve cardiomyocyte alignment.

4.3.3 Embryonic Chick Cardiomyocyte Isolation

All chick cardiomyocytes used in experiments were isolated from 8 day old chick embryos as previously described with minor modifications.[58] First, the eggshells were open, embryos removed, hearts cut out of the embryos and the atria removed leaving only ventricles. Then each ventricle was cut in 10-20 pieces and incubated in 1X TrypLE Express (Thermo Fisher) solution for 7 minutes at 37 ^{0}C. The supernatant was then removed and mixed with seeding medium (M199, 1% penicillin/streptomycin, 10% heat inactivated fetal bovine serum (HI-FBS)) for enzyme deactivation. New TrypLE Express solution was added to the minced hearts and the same procedure was repeated 4-6 times. After that, cell solution was centrifuged, resuspended in seeding medium, and pre-plated in T75 flasks 2 times for 45 minutes to remove fibroblasts from the solution. After pre-plating, cells were centrifuged and resuspended in seeding medium, cell density was counted and cells were seeded onto the substrates fabricated above. After seeding, cells were kept in seeding medium for 24 hours, then the medium was changed to maintenance medium (M199, 1% penicillin/streptomycin, 2% HI-FBS) in order to slow down fibroblast proliferation.

4.3.4 Fixation, Staining, and Fluorescent Microscopy

Cells were fixed and permeabilized with 4% formaldehyde and 0.1% Triton-X 100, and stained with 0.5% DAPI (to stain nuclei), 1.5% phalloidin (to visualize the actin cytoskeleton), and 0.5% of corresponding primary antibodies – for other proteins, such as α-actinin (Sigma A7811), N-cadherin (Sigma C2542), and fibronectin (Sigma F3648). Samples were incubated with the dyes and primary antibodies for 60 minutes, washed in 1X PBS, incubated in the solution of secondary antibodies corresponding to the primary antibodies for 60 minutes, and washed in PBS again. After that coverslips were mounted for imaging onto glass slides with ProLong Gold Anti-Fade preservative. Confocal laser scanning microscopy (LSM Zeiss 700) was used to obtain fluorescent images.

4.4 Results

4.4.1 Sub-Micron Biomimetic Pattern Derivation

To create the new biomimetic pattern with sub-micron resolution, we used a single image of fibronectin in a 6-day old chick myocardium (fig. 4.1A). Due to the improved imaging technique that involved opening and unrolling the left ventricular wall, we were able to compensate for the heart's natural curvature and image the aligned part of chick myocardium over a larger area, which allowed us to use a single image for the pattern derivation. Therefore, we alleviated another limitation of the previous biomimetic pattern that had to be combined from several different images corresponding to different areas of the myocardium. The process of pattern derivation is similar to the one described for the previous biomimetic pattern (fig. 2.4). Shortly, it involves filtering the fibronectin image to remove noise and objects smaller than 0.2 µm in diameter (fig. 4.1B), making a maximum intensity projection of the 3D z-stack along the z-axis, and applying a threshold to convert it to a binary image. The first iteration of the pattern (fig. 4.1C) was used in Nanoscribe to 3D print a master mold. After that, due to the discovered problems with the matching of the pattern's sides, we modified it such that its edges perfectly match when stacked together in an array (fig. 4.1D). To do so, we cut out the bottom part of the pattern, stacked it above the top part with a slight overlay, and manually removed any irregularities in the overlay area, which made the bottom edge match the top one. Then we repeated this process with the left and right parts of the pattern to make the left edge match the right one.

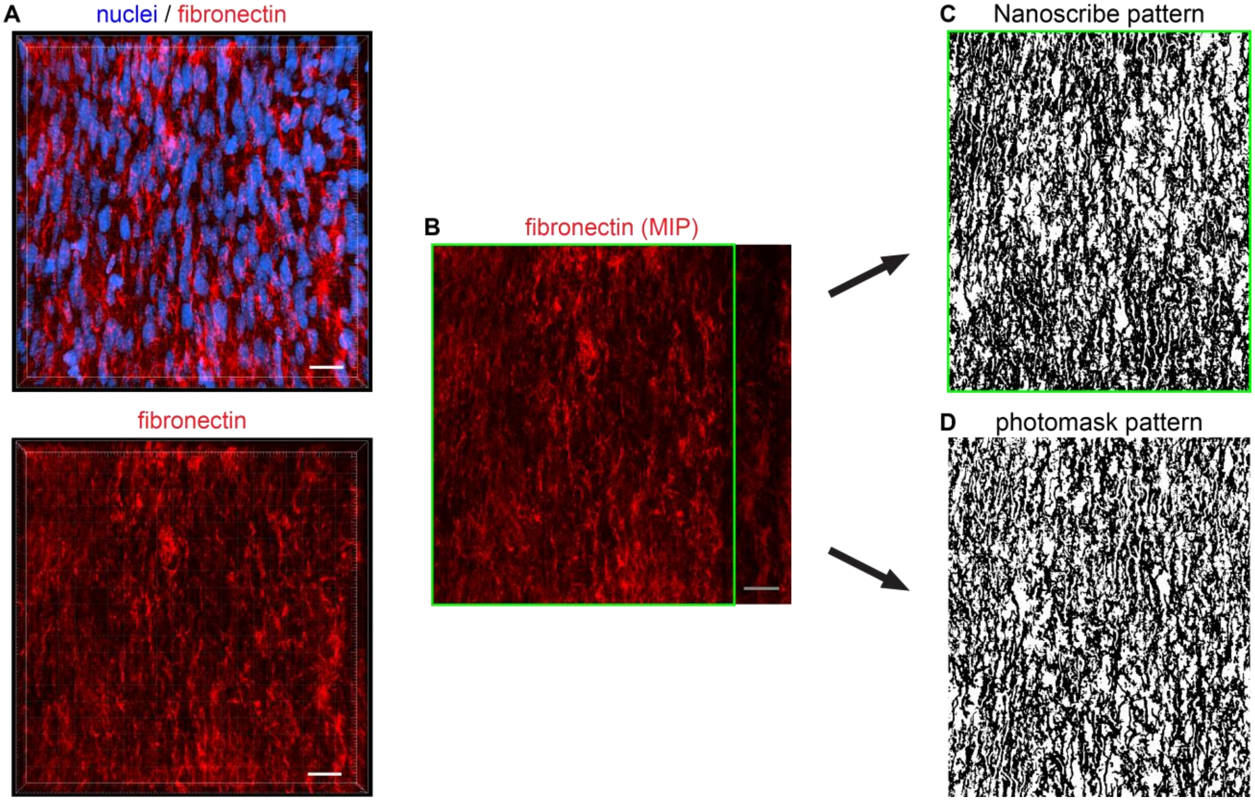

Figure 4.1. Schematic of the sub-micron biomimetic pattern derivation. A: The original images of fibronectin in 6-day old chick myocardium. **B:** Fibronectin structure inside the green frame was used for the pattern derivation. C: The sub-micron pattern derived from the fibronectin image shown in **(B)**. This pattern was used to print master molds using Nanoscribe, but due to non-uniform protein density and non-matching edges it had to be further modified to produce a new pattern **(D)** that lacks these drawbacks.

4.4.2 Using Nanoscribe to 3D Print the Biomimetic Master Mold

Nanoscribe is a 3D printing tool that uses focused laser radiation at 780 nm absorbed by liquid photoresist via two-photon absorption to induce photoresist polymerization inside a small voxel where the excitation intensity is higher than the polymerization threshold of the photoresist. As the size of the voxel highly depends on the photoresist, we used the photoresist IP-L that provides the highest possible resolution that can go down to 0.2-0.4 µm depending on the laser intensity.

First attempts to make stamps by casting PDMS onto the master mold printed with Nanoscribe revealed that the IP-L photoresist completely detaches from the glass wafer during the removal of cured PDMS. To increase the IP-L adhesion to the wafer, SU-8 photoresist was spin-coated at 5000 RPM and hard baked on the glass wafer, after which the IP-L mold was printed on top of the SU-8 layer. Additionally, the height of the mold was decreased from 2 µm to 0.6-1 µm. These changes significantly improved the stability of the mold, although large portions of the mold still detached from the substrate. To further increase mold stability, we applied post-baking at various temperatures. Although IP-L photoresist specifications indicate that it was designed to be used without post-baking, post-baking can be applied if the improved stability is needed. To optimize this process, we baked molds at various temperatures, made PDMS stamps, and assessed fibronectin patterning quality for each stamp. We found that no baking (fig. 4.2A) or baking at lower temperatures (fig. 4.2B) results in loss of features due to photoresist detachment. When baked at 90 ^{0}C (fig. 4.2C), fibronectin transfer quality is improved, although the printed features lose initial detail, possibly due to photoresist melting. Finally, baking at 80 ^{0}C for 24 hours was found to be optimal as almost no photoresist detachment was observed and the protein transfer quality was significantly improved (fig. 4.2D). To ensure maximal stamp quality and further minimize the photoresist detachment, we modified the PDMS stamp making procedure. Firstly, we cast a thin PDMS drop covering the whole pattern and cure it at 65 ^{0}C. Then we peel this thin PDMS layer off, transfer it into a separate dish, and cast additional PDMS on top of it to make the final stamp. A thinner PDMS layer applies lower stress on the mold upon removal, which results in 100% mold preservation and allows for reusing the mold several times.

Once the process of making PDMS stamps was optimized, we printed a larger 5x5 mm master mold that can be used to create cardiac monolayers. During the printing of this mold, we observed mold thickness irregularities due to the difficulties in detecting the interface between the substrate and the IP-L drop caused by the presence of the SU-8 layer. However, these irregularities do not affect the protein transfer as long as the mold height is sufficient to ensure that the features don't collapse under the stamp's weight. Another problem we observed was round gaps in the pattern 1-10 µm in size. These gaps were most likely caused by the residual silane condensate that was used to decrease PDMS adhesion to the mold. To solve this problem, the silanization time needs to be decreased to maximum 3-5 hours and the mold needs to be washed in isopropyl alcohol and dried in nitrogen after silanization. Additionally, we found that the pattern's edges didn't fit together so that when the pattern units are stacked together, a visible border can be observed. To alleviate this problem, we modified the pattern as previously described to make the edges fit together when stacked.

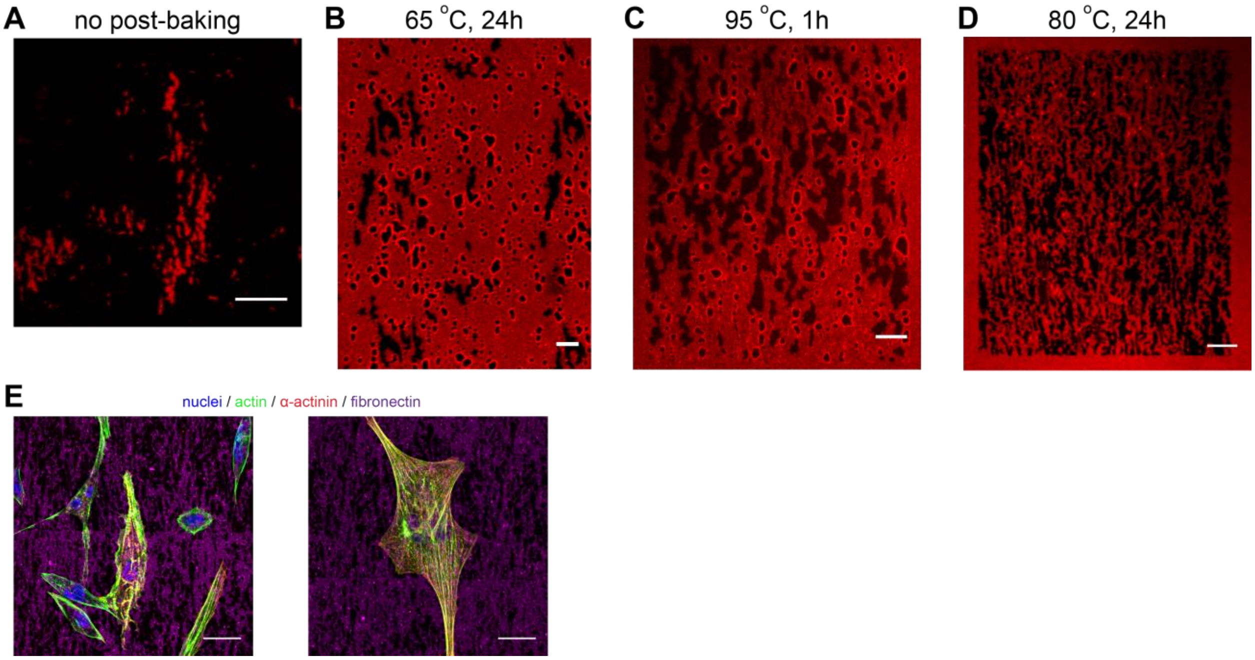

Figure 4.2. Optimization of the sub-micron biomimetic pattern quality by adjusting post-baking temperature and duration. A-D: Microcontact printed fibronectin using PDMS stamps created from

master molds that underwent different post-baking conditions. Post-baking at 80 ^{0}C for 24 hours **(D)** was found to be optimal to prevent photoresist detachment while preserving the mold structure. **E:** Chick cardiomyocytes on the sub-micron biomimetic pattern. Scale bars: **A-D:** 10 µm, **E:** 20 µm.

We seeded chick embryonic cardiomyocytes on these biomimetic pattern at high (450,000 cells/cm2) and low (60,000 cells/cm2) densities. However, the final amount of cardiomyocytes attached to the substrate turned out to be insufficient for forming a confluent monolayer (fig. 45E) and therefore the experiment needs to be repeated.

4.4.3 Using a Sub-Micron Resolution Photomask to Make the Biomimetic Master Mold via Photolithography.

Although significant progress has been achieved in creating high quality sub-micron resolution mold for the new biomimetic pattern, we decided to use the traditional photolithography approach instead due to the high cost of the Nanoscribe use. It is worth pointing out, however, that all problems we encountered during the last mold printing can be easily alleviated. To print the sub-micron biomimetic pattern onto the photomask, it had to be converted into a vector form using Adobe Illustrator's tracing tool, after which we exported it into a dwg format suitable for AutoCAD software. After that the pattern was replicated into an array and printed onto a photomask using laser beam lithography tool Heidelberg DWL with the highest possible resolution of 0.7 µm. This project is currently in progress as we are optimizing the photolithography parameters to achieve the highest master mold quality and ensure that it's stable enough for making PDMS stamps.

4.5 Discussion and conclusions

By using Nanoscribe to create master molds and optimizing printing and post-baking parameters, we were able to successfully recapitulate the fibronectin structure of a developing heart in vitro at a significantly higher resolution. However, there are still unresolved issues associated with this process, such as inconsistent feature height and micro-droplets of the silane condensate formed onto the master mold, and further research is required to eliminate them.

Additionally, we were able to produce a higher resolution photomask to be used in the photolithography process to create master molds for microcontact printing. Although the resolution that can be achieved with this technique is lower compared to the resolution of Nanoscribe-printed molds, the molds produced with photolithography are significantly more consistent in quality. As the work on this project is not yet finished, additional studies are required to optimize this technique and then measure the response of chick cardiomyocytes to this pattern.

Chapter 5: Conclusions and future directions

5.1 Conclusions and Contribution to the Field

Understanding the processes that regulate the development of embryonic myocardium is the key to creating novel techniques to engineer functional cardiac muscle tissues capable of recapitulating physiological characteristics of the native heart. In this study we used a nature-inspired approach to stimulate tissue alignment, determine the role of fibronectin structure in cardiac tissue organization, and elucidate the main factors responsible for cardiomyocyte alignment on fibronectin patterns. The results of this work lay foundation for the development of new techniques to engineer biomimetic cardiac tissues suitable for human heart muscle repair, drug development, and organ-on-chip applications.

In chapter 2 of this work we derived a biomimetic pattern based on the images of fibronectin in chick embryonic myocardium and used it to stimulate the formation of aligned cardiac monolayers. Comparing cardiac tissue alignment on this biomimetic pattern and line patterns that have been previously used in the literature to induce cell alignment, we revealed a unique density-dependent response of chick cardiomyocytes to the biomimetic pattern that was not observed for the line patterns. This response allowed us to elucidate the role of cell-cell and cell-ECM interactions in tissue alignment and show that although the fibronectin pattern affects the overall orientation of cardiomyocytes in a monolayer, it has little influence over the degree of cell alignment, which is primarily controlled by cell-cell interactions.

In chapter 3 we differentiated ESCs and iPSCs into functional human cardiomyocytes and used them to engineer clinically relevant aligned human cardiac monolayers. We found significant differences in iPSC-CM and ESC-CM response to the patterns compared to each other and to the chick cardiomyocytes. Specifically, iPSC-CMs have a more round shape, show overall lower alignment, and the relationship between cell-cell and cell-ECM driven alignment is different. Most of these differences are likely caused by the immature phenotype of stem cell-derived cardiomyocytes as we showed that even a slight improvement in cell maturity using T3 hormone significantly increased tissue alignment. Therefore, developing new techniques to promote cardiomyocyte maturation is the key to building better cardiac tissues. Further, we showed that co-culture with fibroblasts can also increase the alignment of iPSC-CMs indicating their potential role in heart development. Further studies related to understanding this role can prove instrumental in developing more advanced cardiac tissue engineering methods.

5.2 Future Directions

Although the results of this work provide an important insight into the mechanism of cardiac tissue formation, there are still many unknown components in the system that need to be taken into account and studied. Specifically, in this study we primarily focused on structural characteristics of engineered tissues, such as actin alignment. However, functional properties of the tissues, such as calcium handling, contraction propagation speed along and across the direction of alignment, and contractile force, are also crucial tissue characteristics. To do so, the calcium imaging technique we developed needs to be further improved to scan tissues over a larger area, so that local tissue irregularities don't skew the results. Contractile force of our tissues can be measured by building muscular thin films as described earlier.[84] Particularly, peak systolic stress as a function of beating frequency indicating tissue maturity needs to be analyzed. In addition to that, other structural characteristics, such as gap junctions and desmosomes, need to be studied, as they are directly responsible for propagating action potential and contractile stress across the tissue.

When engineering human cardiac tissues from iPSC-CMs, we found that without conditioning on fibronectin-coated substrates, they show extremely low spreading rate on patterned fibronectin. We hypothesized that this happens due to low expression of fibronectin-specific integrins. This hypothesis needs to be tested by measuring the expression levels of iPSC-CMs before and after conditioning and comparing them to integrin expression in chick cardiomyocytes. Additionally, fibronectin-specific integrins can be artificially overexpressed via viral infection, so that cardiomyocyte spreading rate can be measured as a function of integrin expression.

Another important limitation of our system is that the tissues are grown on a PDMS-coated substrate that's attached to a much stiffer glass coverslip. This substrate is significantly stiffer than the heart's environment and therefore provides a significantly different mechanical environment to the cells. However, the appropriate mechanical environment can be incorporated into our system by using softer, more flexible substrates that can allow for more physiologically relevant range of motion of the engineered tissues during the contraction cycle. Additionally, constraining these softer substrates along the direction of fibronectin featured would create anisotropic mechanical environment that could provide an additional stimulus for cell alignment. Soft biomaterials, such as collagen type I or fibrin hydrogels, serving as cell substrates, would also make our tissues more biocompatible, so that they can be implanted in a living tissue. Additionally, these monolayers can be stacked together to produce thicker 3D multi-layered tissues capable of generating higher contractile force.

Electrical stimulation has been previously shown to be an effective way to promote tissue organization and maturation, and thus integrating it into our system can improve the performance of engineered tissues. To do so, tissues need to be cultured in modified plates with electrodes connected to a voltage generator that can provide periodic pulse signal to trigger contractions.

Although fibronectin is the most abundant protein in a developing heart ECM, there are many other components that constitute a significant part of it. The role of some of these components in heart development and their potential use in tissue engineering applications is not fully understood. The biomimetic approach to tissue engineering we applied in this work can serve as a convenient way to study the mechanism of native cardiac tissue organization and the role of those components in it.

Finally, one of the limitations of our system is insufficient resolution of the biomimetic fibronectin pattern, which could not recapitulate features smaller than 2 µm. The work on creating a new pattern with higher resolution that can provide an important insight into the role of finer fibronectin structure on tissue formation has already started, although further studies are needed to optimize the process of protein patterning at this resolution before the response of the cells to this pattern can be determined.

References

1. Davies MJ, Thomas AC. Plaque fissuring--the cause of acute myocardial infarction, sudden ischaemic death, and crescendo angina. *British Heart Journal.* 1985;53(4):363-373.

2. Pfeffer MA, Braunwald E. Ventricular remodeling after myocardial infarction. Experimental observations and clinical implications. *Circulation.* 1990;81(4):1161-1172.

3. Rovere MTL, Bigger Jr JT, Marcus FI, Mortara A, Schwartz PJ. Baroreflex sensitivity and heart-rate variability in prediction of total cardiac mortality after myocardial infarction. *The Lancet.* 1998;351(9101):478-484.

4. Kleiger RE, Miller JP, Bigger JT, Moss AJ. Decreased heart rate variability and its association with increased mortality after acute myocardial infarction. *The American Journal of Cardiology.* 1987/02/01 1987;59(4):256-262.

5. Hellermann JP, Jacobsen SJ, Gersh BJ, Rodeheffer RJ, Reeder GS, Roger VéL. Heart failure after myocardial infarction: a review. *The American Journal of Medicine.* 2002;113(4):324-330.

6. Poss KD, Wilson LG, Keating MT. Heart Regeneration in Zebrafish. *Science.* 2002;298(5601):2188-2190.

7. Laflamme MA, Murry CE. Heart regeneration. *Nature.* 2011;473(7347):326-335.

8. Go AS, Mozaffarian D, Roger VL, et al. Heart Disease and Stroke Statistics--2014 Update: A Report From the American Heart Association. *Circulation.* 2013;129(3):e28-e292.

9. Zhang Y, Mignone J, MacLellan WR. Cardiac Regeneration and Stem Cells. *Physiological Reviews.* 2015;95(4):1189-1204.

10. Song K, Nam Y-J, Luo X, et al. Heart repair by reprogramming non-myocytes with cardiac transcription factors. *Nature.* 2012;485(7400):599-604.

11. Ieda M, Fu J-D, Delgado-Olguin P, et al. Direct Reprogramming of Fibroblasts into Functional Cardiomyocytes by Defined Factors. *Cell.* 2010;142(3):375-386.

12. Chen JX, Krane M, Deutsch MA, et al. Inefficient Reprogramming of Fibroblasts into Cardiomyocytes Using Gata4, Mef2c, and Tbx5. *Circulation Research.* 2012.

13. Qian L, Huang Y, Spencer CI, et al. In vivo reprogramming of murine cardiac fibroblasts into induced cardiomyocytes. *Nature.* 2012;485(7400):593-598.

14. Field LJ. Atrial natriuretic factor-SV40 T antigen transgenes produce tumors and cardiac arrhythmias in mice. *Science.* 1988;239(4843):1029.

15. Poolman RA, Gilchrist R, Brooks G. Cell cycle profiles and expressions of p21 CIP1 and p27 KIP1 during myocyte development. *International journal of cardiology.* 1998;67(2):133-142.

16. Eulalio A, Mano M, Dal Ferro M, et al. Functional screening identifies miRNAs inducing cardiac regeneration. *Nature.* 2012;492(7429):376-381.

17. Porrello ER, Johnson BA, Aurora AB, et al. miR-15 Family Regulates Postnatal Mitotic Arrest of CardiomyocytesNovelty and Significance. *Circulation Research.* 2011;109(6):670-679.

18. Porrello ER, Mahmoud AI, Simpson E, et al. Regulation of neonatal and adult mammalian heart regeneration by the miR-15 family. *Proceedings of the National Academy of Sciences.* 2013;110(1):187-192.

19. Ibrahim Ahmed G-E, Cheng K, Marbán E. Exosomes as Critical Agents of Cardiac Regeneration Triggered by Cell Therapy. *Stem Cell Reports.* 2014;2(5):606-619.

20. Hirt MN, Hansen A, Eschenhagen T. Cardiac Tissue Engineering: State of the Art. *Circulation Research.* 2014;114(2):354-367.

21. Jawad H, Ali NN, Lyon AR, Chen QZ, Harding SE, Boccaccini AR. Myocardial tissue engineering: a review. *Journal of Tissue Engineering and Regenerative Medicine.* 2007;1(5):327-342.

22. Lindsley CW. New Statistics on the Cost of New Drug Development and the Trouble with CNS Drugs. *ACS Chemical Neuroscience.* 2014;5(12):1142-1142.

23. Dmitrienko A, Chuang-Stein C, D'Agostino RB. *Pharmaceutical statistics using SAS : a practical guide*. Cary, N.C.: SAS Institute; 2007.

24. Ng SLJ, Narayanan K, Gao S, Wan ACA. Lineage restricted progenitors for the repopulation of decellularized heart. *Biomaterials.* 2011;32(30):7571-7580.

25. Khan AA, Vishwakarma SK, Bardia A, Venkateshwarulu J. Repopulation of decellularized whole organ scaffold using stem cells: an emerging technology for the development of neo-organ. *Journal of Artificial Organs.* 2014;17(4):291-300.

26. Lu T-Y, Lin B, Kim J, et al. Repopulation of decellularized mouse heart with human induced pluripotent stem cell-derived cardiovascular progenitor cells. *Nature Communications.* 2013;4:2307.

27. Ott HC, Matthiesen TS, Goh S-K, et al. Perfusion-decellularized matrix: using nature's platform to engineer a bioartificial heart. *Nature Medicine.* 2008;14(2):213-221.

28. Phu D, Wray LS, Warren RV, Haskell RC, Orwin EJ. Effect of Substrate Composition and Alignment on Corneal Cell Phenotype. *Tissue Engineering Part A.* 2011;17(5-6):799-807.

29. Au HTH, Cheng I, Chowdhury MF, Radisic M. Interactive effects of surface topography and pulsatile electrical field stimulation on orientation and elongation of fibroblasts and cardiomyocytes. *Biomaterials.* 2007;28(29):4277-4293.

30. Zong X, Bien H, Chung C, et al. Electrospun fine-textured scaffolds for heart tissue constructs. *Biomaterials.* 2005;26(26):5330-5338.

31. Lind JU, Busbee TA, Valentine AD, et al. Instrumented cardiac microphysiological devices via multimaterial three-dimensional printing. *Nat Mater.* 2017;16(3):303-308.

32. Wang P-Y, Yu J, Lin J-H, Tsai W-B. Modulation of alignment, elongation and contraction of cardiomyocytes through a combination of nanotopography and rigidity of substrates. *Acta Biomaterialia.* 2011;7(9):3285-3293.

33. Engelmayr GC, Cheng M, Bettinger CJ, Borenstein JT, Langer R, Freed LE. Accordion-like honeycombs for tissue engineering of cardiac anisotropy. *Nature Materials.* 2008;7(12):1003-1010.

34. Kai D, Prabhakaran MP, Jin G, Ramakrishna S. Guided orientation of cardiomyocytes on electrospun aligned nanofibers for cardiac tissue engineering. *Journal of Biomedical Materials Research Part B: Applied Biomaterials.* 2011;98B(2):379-386.

35. Matsuda T, Takahashi K, Nariai T, et al. N-cadherin-mediated cell adhesion determines the plasticity for cell alignment in response to mechanical stretch in cultured cardiomyocytes. *Biochemical and Biophysical Research Communications.* 2004;326(1):228-232.

36. West AR, Zaman N, Cole DJ, et al. Development and characterization of a 3D multicell microtissue culture model of airway smooth muscle. *American Journal of Physiology-Lung Cellular and Molecular Physiology.* 2013;304(1):L4-L16.

37. Zimmermann WH. Tissue Engineering of a Differentiated Cardiac Muscle Construct. *Circulation Research.* 2001;90(2):223-230.

38. Halper J, Kjaer M. Basic Components of Connective Tissues and Extracellular Matrix: Elastin, Fibrillin, Fibulins, Fibrinogen, Fibronectin, Laminin, Tenascins and Thrombospondins. In: Halper J, ed. *Progress in Heritable Soft Connective Tissue Diseases*. Dordrecht: Springer Netherlands; 2014:31-47.

39. Williams C, Quinn KP, Georgakoudi I, Black LD, 3rd. Young developmental age cardiac extracellular matrix promotes the expansion of neonatal cardiomyocytes in vitro. *Acta Biomater.* Jan 2014;10(1):194-204.

40. Villarreal FJ, Dillmann WH. Cardiac hypertrophy-induced changes in mRNA levels for TGF-beta 1, fibronectin, and collagen. *American Journal of Physiology - Heart and Circulatory Physiology.* 1992;262(6):H1861-H1866.

41. Singh P, Carraher C, Schwarzbauer JE. Assembly of Fibronectin Extracellular Matrix. *Annual Review of Cell and Developmental Biology.* 2010;26(1):397-419.

42. Geiger B, Bershadsky A, Pankov R, Yamada KM. Transmembrane crosstalk between the extracellular matrix and the cytoskeleton. *Nat Rev Mol Cell Biol.* 2001;2(11):793-805.

43. Larsen M, Artym VV, Green JA, Yamada KM. The matrix reorganized: extracellular matrix remodeling and integrin signaling. *Current Opinion in Cell Biology.* 2006;18(5):463-471.

44. Tomanek RJ. Formation of the coronary vasculature during development. *Angiogenesis.* 2005;8(3):273-284.

45. Bonnans C, Chou J, Werb Z. Remodelling the extracellular matrix in development and disease. *Nature Reviews Molecular Cell Biology.* 2014;15(12):786-801.

46. Barkan D, Green JE, Chambers AF. Extracellular matrix: A gatekeeper in the transition from dormancy to metastatic growth. *European Journal of Cancer.* 2010;46(7):1181-1188.

47. Kim SH, Turnbull J, Guimond S. Extracellular matrix and cell signalling: the dynamic cooperation of integrin, proteoglycan and growth factor receptor. *Journal of Endocrinology.* 2011;209(2):139-151.

48. Zhu J, Clark RAF. Fibronectin at Select Sites Binds Multiple Growth Factors and Enhances their Activity: Expansion of the Collaborative ECM-GF Paradigm. *Journal of Investigative Dermatology.* 2014;134(4):895-901.

49. Martino MM, Hubbell JA. The 12th-14th type III repeats of fibronectin function as a highly promiscuous growth factor-binding domain. *The FASEB Journal.* 2010;24(12):4711-4721.

50. Feinberg AW, Alford PW, Jin H, et al. Controlling the contractile strength of engineered cardiac muscle by hierarchal tissue architecture. *Biomaterials.* 2012;33(23):5732-5741.

51. Konstandin MH, Toko H, Gastelum GM, et al. Fibronectin is essential for reparative cardiac progenitor cell response after myocardial infarction. *Circ Res.* Jul 05 2013;113(2):115-125.

52. Hamburger V, Hamilton HL. A series of normal stages in the development of the chick embryo. *Developmental Dynamics.* 1992;195(4):231-272.

53. Wittig J, Münsterberg A. The Early Stages of Heart Development: Insights from Chicken Embryos. *Journal of Cardiovascular Development and Disease.* 2016;3(2):12.

54. Jallerat Q. *An Embrionic-inspired Approach to Engineer Functional Human Cardiac Tissue*. Pittsburgh: Biomedical Engineering, Carnegie Mellon University; 2016.

55. Wilbur JL, Kumar A, Kim E, Whitesides GM. Microfabrication by microcontact printing of self-assembled monolayers. *Advanced Materials.* 1994;6(7-8):600-604.

56. Palchesko RN, Zhang L, Sun Y, Feinberg AW. Development of polydimethylsiloxane substrates with tunable elastic modulus to study cell mechanobiology in muscle and nerve. *PloS one.* 2012;7(12):e51499.

57. Mrksich M, Dike LE, Tien J, Ingber DE, Whitesides GM. Using Microcontact Printing to Pattern the Attachment of Mammalian Cells to Self-Assembled Monolayers of Alkanethiolates on Transparent Films of Gold and Silver. *Experimental Cell Research.* 9/15/ 1997;235(2):305-313.

58. Shimizu T, Kinugawa K-i, Yao A, et al. *Platelet-derived growth factor induces cellular growth in cultured chick ventricular myocytes.* Vol 411999.

59. Sun Y, Duffy R, Lee A, Feinberg AW. Optimizing the structure and contractility of engineered skeletal muscle thin films. *Acta Biomater.* Aug 2013;9(8):7885-7894.

60. Kostetskii I, Li J, Xiong Y, et al. Induced deletion of the N-cadherin gene in the heart leads to dissolution of the intercalated disc structure. *Circ Res.* Feb 18 2005;96(3):346-354.

61. Li G, Satyamoorthy K, Herlyn M. N-Cadherin-mediated Intercellular Interactions Promote Survival and Migration of Melanoma Cells. *Cancer Research.* May 1, 2001 2001;61(9):3819-3825.

62. Matsushita T, Oyamada M, Kurata H, et al. Formation of Cell Junctions Between Grafted and Host Cardiomyocytes at the Border Zone of Rat Myocardial Infarction. *Circulation.* November 9, 1999 1999;100(suppl 2):II-262-Ii-268.

63. Babensee JE, Anderson JM, McIntire LV, Mikos AG. Host response to tissue engineered devices. *Advanced Drug Delivery Reviews.* 1998;33(1–2):111-139.

64. Takahashi K, Tanabe K, Ohnuki M, et al. Induction of Pluripotent Stem Cells from Adult Human Fibroblasts by Defined Factors. *Cell.* 2007;131(5):861-872.

65. Batalov I, Feinberg AW. Differentiation of cardiomyocytes from human pluripotent stem cells using monolayer culture. *Biomarker insights.* 2015;10(Suppl 1):71.

66. Mummery C, Ward D, Van Den Brink CE, et al. Cardiomyocyte differentiation of mouse and human embryonic stem cells*. *Journal of Anatomy.* 2002;200(3):233-242.

67. Wobus AM, Wallukat G, Hescheler J. Pluripotent mouse embryonic stem cells are able to differentiate into cardiomyocytes expressing chronotropic responses to adrenergic and cholinergic agents and Ca2+ channel blockers. *Differentiation.* 1991/12/01 1991;48(3):173-182.

68. Dahlmann J, Kensah G, Kempf H, et al. The use of agarose microwells for scalable embryoid body formation and cardiac differentiation of human and murine pluripotent stem cells. *Biomaterials.* 2013;34(10):2463-2471.

69. Mohr JC, Zhang J, Azarin SM, et al. The microwell control of embryoid body size in order to regulate cardiac differentiation of human embryonic stem cells. *Biomaterials.* 2010;31(7):1885-1893.

70. Laflamme MA, Chen KY, Naumova AV, et al. Cardiomyocytes derived from human embryonic stem cells in pro-survival factors enhance function of infarcted rat hearts. *Nature Biotechnology.* 2007;25(9):1015-1024.

71. Lian X, Zhang J, Azarin SM, et al. Directed cardiomyocyte differentiation from human pluripotent stem cells by modulating Wnt/β-catenin signaling under fully defined conditions. *Nature Protocols.* 2012;8(1):162-175.

72. Burridge PW, Matsa E, Shukla P, et al. Chemically defined generation of human cardiomyocytes. *Nat Meth.* 2014;11(8):855-860.

73. Robertson C, Tran DD, George SC. Concise Review: Maturation Phases of Human Pluripotent Stem Cell-Derived Cardiomyocytes. *Stem Cells.* 2013;31(5):829-837.

74. Kamakura T, Makiyama T, Sasaki K, et al. Ultrastructural maturation of human-induced pluripotent stem cell-derived cardiomyocytes in a long-term culture. *Circulation Journal.* 2013;77(5):1307-1314.

75. Lundy SD, Zhu W-Z, Regnier M, Laflamme MA. Structural and functional maturation of cardiomyocytes derived from human pluripotent stem cells. *Stem cells and development.* 2013;22(14):1991-2002.

76. Eng G, Lee BW, Protas L, et al. Autonomous beating rate adaptation in human stem cell-derived cardiomyocytes. *Nature Communications.* 2016;7:10312.

77. Nunes SS, Miklas JW, Liu J, et al. Biowire: a platform for maturation of human pluripotent stem cell-derived cardiomyocytes. *Nature methods.* 2013;10(8):781-787.

78. Chan Y-C, Ting S, Lee Y-K, et al. Electrical stimulation promotes maturation of cardiomyocytes derived from human embryonic stem cells. *Journal of cardiovascular translational research.* 2013;6(6):989-999.

79. Yang X, Rodriguez M, Pabon L, et al. Tri-iodo-l-thyronine promotes the maturation of human cardiomyocytes-derived from induced pluripotent stem cells. *Journal of molecular and cellular cardiology.* 2014;72:296-304.

80. Hughes CS, Postovit LM, Lajoie GA. Matrigel: A complex protein mixture required for optimal growth of cell culture. *PROTEOMICS.* 2010;10(9):1886-1890.

81. Huebsch N, Loskill P, Deveshwar N, et al. Miniaturized iPS-Cell-Derived Cardiac Muscles for Physiologically Relevant Drug Response Analyses. *Scientific Reports.* 2016;6:24726.

82. Radisic M, Park H, Martens TP, et al. Pre-treatment of synthetic elastomeric scaffolds by cardiac fibroblasts improves engineered heart tissue. *Journal of Biomedical Materials Research Part A.* 2008;86A(3):713-724.

83. Thavandiran N, Dubois N, Mikryukov A, et al. Design and formulation of functional pluripotent stem cell-derived cardiac microtissues. *Proceedings of the National Academy of Sciences.* December 3, 2013 2013;110(49):E4698-E4707.

84. Feinberg AW, Feigel A, Shevkoplyas SS, Sheehy S, Whitesides GM, Parker KK. Muscular thin films for building actuators and powering devices. *Science.* 2007;317(5843):1366-1370.

Appendices

Appendix 1. MATLAB Code for Calculating Actin OOP from Confocal Images or Z-stacks

Instructions to run the code

File to run: main script.

Required additional files

- actinDetectSlice.m (attached)
- actinDetectTest.m (attached)
- MIP.m (attached)
- OOP.m (attached)
- Fingerprint Detection Library
- bfmatlab library (https://www.openmicroscopy.org/site/support/bio-formats5.4/users/matlab/)

Required inputs during the main script run

- **First dialog box**
    - number of samples. Don't put number of images here unless you have 1 image per sample. Later you will be asked to select files for each sample separately.
    - output file title. The file with the .csv extension will be created in the same folder as the images. If the file already exists, new data will be written below the existing data.
    - actin layer number. Enter the number of the layer you want to analyze. To determine the number, you can open your image in ImageJ
    - alpha-actinin layer number. If you need to mask your orientation, put the layer number the mask should be based on. If no mask required, put "-1".
- **File selection dialog.** Select the files corresponding to the sample 1. If previously you indicated more than one sample, this dialog will appear for each sample individually.
- **Analyze fibroblasts?** This will appear in the command line if alpha-actinin layer is present. Select 'yes' if you also want to analyze cells outside of alpha-actinin mask.

- **Threshold for filament detection.** Enter a number from 0 to 1, usually within the 0.1-0.2 range. Make sure to use the same threshold for all your samples, because changing it will affect OOP.

Main script:

```
% Alignment analysis of microscopy images (2D or 3D z-stacks) of cells
stained for actin
% cytiskeleton. If the images are z-stacks, MIP is analyzed.

% INPUT: lsm image or 'any' microscopy image.
% Must contain (actin) channel, the alignment of which needs to be analyzed
% Can contain alpha-actinin channel to make cardiomyocyte-specific analysis
% If the image is a z-stack, a maximum intensity projections (MIPs) are
analyzed
% Image can contain several positions, each position will be analyzed
% separately, then all positions from all files corresponding to the same
% sample are combined for the overall alignmnet analysis

% OUTPUT: a csv file that can be open in MS Excel or a similar software
% File contains the alignment analysis results, including cell area
% coverage, mean/average/mode orientation angles, OOP

% Adapted from: Adam W. Feinberg
% Disease Biophysics Group
% School of Engineering and Applied Sciences
% Havard University, Cambridge, MA 02138

% Created by Ivan Batalov
% Regenerative Biometerials & Therapeutics Group
% Carnegie Mellon University, Pittsburgh, PA, USA

%% Request info and open image files.
inputTitles = {'number of samples:','ouput file title:','actin layer
number:','alpha-actinin layer number (-1 if none):'};
defaultInputValues = {'1','actin alignment','2','3'};

SampleInfo = inputdlg(inputTitles,'Input', 1, defaultInputValues);
SampleCount = str2num(SampleInfo{1});
actinLayerNumber = str2num(SampleInfo{3});
actininLayerNumber = str2num(SampleInfo{4});

%rootfolder = path;
if(~exist('rootfolder', 'var') || length(rootfolder) < 2)
    % rootfolder = '/Volumes/Macintosh HD 2/Drop-boxes/Dropbox (RBG)/Lab
stuff';
    rootfolder = 'F:\Images\Ivan'; % put your default folder path/name here.
The path format depends on the OS.
end

if(actininLayerNumber > 0)
    analyzeFibroblasts = input('Analyze fibroblasts? (1=yes, 0=no)');
else
    analyzeFibroblasts = 0;
```

```
end

% get files for each sample. Multiple files can be open for each sample.
FileList = cell(SampleCount,2);
for i=1:SampleCount
    message = sprintf('Select files for sample %i',i);

[file,path]=uigetfile({'*.*';'*.lsm';'*.TIF';'*.tif';'*.bmp';'*.jpg'},message
,rootfolder,'Multiselect','on');
    FileList{i,1}=path;
    rootfolder = path;
    FileList{i,2}= file;
end

xlsname = [FileList{1,1} SampleInfo{2} '.csv'];

%% Create file for results
fileID = fopen(xlsname, 'a');
if(analyzeFibroblasts)
    fprintf(fileID, 'cardiomyocytes,position,Sarcomere area fraction,Actin
area fraction,Mean,Stdev,median,Mode,OOP,,fibroblasts,position,Percent of
Actin area,Mean,Stdev,median,Mode,OOP\n');
else
    fprintf(fileID, 'cardiomyocytes,position,Sarcomere area fraction,Actin
area fraction,Mean,Stdev,median,Mode,OOP\n');
end

%% Loop for each sample
for i=1:SampleCount

    disp(strcat('Sample_', num2str(i)));
    sample = FileList{i,2};
    if ~isa(sample,'char')
        PictureCount = length(sample);
    else
        PictureCount = 1;
    end

    % Setup variables for ouput data
    All_angles = [];
    fbs_All_angles = [];
    average_PercentActinArea = 0;
    average_fbs_PercentActinArea = 0;
    average_sarcomere_area = 0;

    %% Loop over each file for sample #i
    for j = 1:PictureCount
        disp(strcat('Image_', num2str(j)));

        % Open images and load actin channel in a matrix
        if PictureCount == 1
            picture = sample;
            picturename = [FileList{i,1} sample];
        else
            picture = sample{j};
            picturename = [FileList{i,1} sample{j}];
        end
        img = bfopen(picturename); % opens microscopy images. Documentation
can be found online.

        info = img{1,4};  % Load OME metadata
        PixelSize = double(info.getPixelsPhysicalSizeX(0).value());
        Size = info.getPixelsSizeX(0).getValue();

        fprintf('Image resolution: %5.5f px/µm\n\n', 1/PixelSize);
        % Define blksze (should be 3/pixelsize based on recommendation)
        blksze = floor(3/PixelSize);
        d = floor(1.5*blksze);
        disk = strel('disk', d, 0); % used to merge sarcomeres into mask by
expanding and shrinking the image
        actin_disk = strel('disk', floor(d/2), 0); % used to merge actin into
mask

        numberOfPositions = info.getImageCount();

        %% Loop over all positions within the current file.
        for position = 1 : numberOfPositions
            disp(strcat('Position_', num2str(position)));
            numberOfChannels = info.getChannelCount(position - 1); %
numbering starts from 0, thus the shift

            if(actininLayerNumber >= 0)
                mip = MIP(img{position, 1}(:, 1), numberOfChannels,
[actinLayerNumber, actininLayerNumber]);
                actin = mip{1};
                sarcomeres = mip{2};

                threshold = (0.1*max(sarcomeres(:))+0.9*min(sarcomeres(:)));
                sarcomereMask = bwmorph(sarcomeres > threshold, 'open');
                sarcomeres_border = bwmorph(sarcomereMask, 'remove');
                % dilate and erode to fill the holes between z-lines of
                % sarcomeres
                sarcomereMask = sarcomereMask | imdilate(sarcomeres_border,
disk);
                % erode only 60%, because cells' area is higher than
sarcomere coverage
                sarcomereMask = bwmorph(sarcomereMask,'erode',floor(d*0.6));
            else
                mip = MIP(img{position, 1}(:, 1), numberOfChannels,
actinLayerNumber);
                actin = mip{1};
            end

            [imSizeX,imSizeY] = size(actin);

            maxActinArea = (imSizeX - 2*0.6*floor(d/2))*(imSizeY -
2*floor(d/2));
            maxSarcomereArea = (imSizeX - 2*0.6*d)*(imSizeY - 2*d);

            actinThreshold = (0.05*max(actin(:))+0.95*min(actin(:)));
            actinMask = bwmorph(actin > actinThreshold, 'open');
            actin_border = bwmorph(actinMask, 'remove');

            % dilate and erode to fill the holes between actin filaments
            actinMask = actinMask | imdilate(actin_border, actin_disk);
            actinMask = bwmorph(actinMask,'erode',floor(floor(d/2)*0.6));
            if(actininLayerNumber >= 0)
                actinMask = actinMask | sarcomereMask; % expand actin mask to
include sarcomere mask
            end

            %% Alignment analysis
            if ~exist('thresh','var')
                % determine the optimal threshold for filament detection
                % the same threshold should be used for all images
                thresh = actinDetectTest(actin,blksze,PixelSize);
            end

            % get filament orientation image
            nonzero_orientation =
actinDetectSlice(actin,blksze,thresh,Size,sarcomereMask);

            if(analyzeFibroblasts == 1)
                fibroblast_nonzero_orientation =
actinDetectSlice(actin,blksze,thresh,Size,(~sarcomereMask).*actinMask);
            end

            %% Calculate mode of orientation distribution
            % Convert radians to degrees
            nonzero_orientation_angles = rad2deg(nonzero_orientation);

            % hist(nonzero_orientation_angles)
            Mean = mean(nonzero_orientation_angles);
            Std = std(nonzero_orientation_angles);
            Median = median(nonzero_orientation_angles);

            % Create histogram
            [n,xout] = hist(nonzero_orientation_angles,180);
            dx = xout(2)-xout(1); % calc a single bin width
            n = n / sum( n*dx );  % normalize histogram to have area of 1

            % Find mode
            [~,I] = max(n);
            Mode = xout(I);

            %% Plot histogram of raw orientation
%             figure, bar(xout,n,'hist')              % plot normalized
histogram
%             xlim( [xout(1)-dx/2,xout(end)+dx/2] );  % make sure that the
axis is squeezed to it's limits
%             title('Histogram of Actin Orientation Angles')
%             xlabel('Degrees')
%             ylabel('Normalized Occurance')

            %% Calculate cell coverage
            PercentActinArea = nnz(actinMask)/maxActinArea;
            average_PercentActinArea = [average_PercentActinArea;
PercentActinArea];

            if(actininLayerNumber > 0)
                sarcomere_area_fraction =
nnz(sarcomereMask)/maxSarcomereArea;
                average_sarcomere_area = [average_sarcomere_area;
sarcomere_area_fraction];
            end

            %% calculate OOP
            cms_oop = OOP(nonzero_orientation);

            %% save data in excel file
            if(actininLayerNumber > 0)
                fprintf(fileID, [picture
',%3i,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,,'], position,
sarcomere_area_fraction, PercentActinArea, Mean, Std, Median, Mode, cms_oop);
            else
                fprintf(fileID, [picture
',%3i,1,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,,'], position, PercentActinArea,
Mean, Std, Median, Mode, cms_oop);
            end

            %% Gather data for the entire sample
            All_angles = [All_angles; nonzero_orientation];

            %% The same for the fibroblasts
            if(analyzeFibroblasts == 1)
                % Convert radians to degrees
                fbs_nonzero_orientation_angles =
rad2deg(fibroblast_nonzero_orientation);

                % hist(nonzero_orientation_angles)
                fbs_Mean = mean(fbs_nonzero_orientation_angles);
                fbs_Std = std(fbs_nonzero_orientation_angles);
                fbs_Median = median(fbs_nonzero_orientation_angles);

                % Create histogram
                [fbs_n,fbs_xout] = hist(fbs_nonzero_orientation_angles,180);
                fbs_dx = fbs_xout(2)-fbs_xout(1);                       % calc a
single bin width
                fbs_n = fbs_n / sum( fbs_n*fbs_dx );                    %
normalize histogram to have area of 1

                % Find mode
                [~,fbs_I] = max(fbs_n);
                fbs_Mode = xout(fbs_I);

                PercentFibroblastActinArea =
nnz(actinMask.*(~sarcomereMask))/((imSizeX)*(imSizeY));
                average_fbs_PercentActinArea = [average_fbs_PercentActinArea;
PercentFibroblastActinArea];

                fbs_oop = OOP(fibroblast_nonzero_orientation);

                % save data in excel file
                fprintf(fileID, [picture
',%3i,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f\n'], position,
PercentFibroblastActinArea, fbs_Mean, fbs_Std, fbs_Median, fbs_Mode,
fbs_oop);

                fbs_All_angles = [fbs_All_angles;
fibroblast_nonzero_orientation];
            else
```

```
                % save data in excel file
                fprintf(fileID, '\n');
            end
        end % end of positions loop
    end % end of PictureCount loop

    %% Calculate the alignment of the entire sample.
    All_angles_deg = rad2deg(All_angles);
    % Calculate stats for entire sample

    Mean = mean(All_angles_deg);
    Std = std(All_angles_deg);
    Median = median(All_angles_deg);

    % Create histogram
    [n,xout] = hist(All_angles_deg,180);
    dx = xout(2)-xout(1); % calc a single bin width
    n = n / sum( n*dx );  % normalize histogram to have area of 1

    % Find mode
    [~,I] = max(n);
    Mode = xout(I);

    %angles here should be in radians, cause cos() and sin() functions are
used
    OrientationOrderParameter = OOP(All_angles);

    %% The same for fibroblasts
    if(analyzeFibroblasts == 1)
        % Convert radians to degrees
        fbs_All_angles_deg = rad2deg(fbs_All_angles);

        % Calculate stats for entire sample
        fbs_Mean = mean(fbs_All_angles_deg);
        fbs_Std = std(fbs_All_angles_deg);
        fbs_Median = median(fbs_All_angles_deg);

        % Create histogram
        [fbs_n,fbs_xout] = hist(fbs_All_angles_deg,180);
        fbs_dx = fbs_xout(2)-fbs_xout(1); % calc a single bin width
        fbs_n = fbs_n / sum( fbs_n*fbs_dx ); % normalize histogram to have
area of 1

        % Find mode
        [~,fbs_I] = max(fbs_n);
        fbs_Mode = xout(fbs_I);

        fbs_OrientationOrderParameter = OOP(fbs_All_angles);

        %Save data
        fprintf(fileID,
'Average,all,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,,',
mean(average_sarcomere_area(:)), mean(average_PercentActinArea(:)), Mean,
Std, Median, Mode, OrientationOrderParameter);
        fprintf(fileID,
'Average_fbs,all,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f\n',
mean(average_fbs_PercentActinArea(:)), fbs_Mean, fbs_Std, fbs_Median,
fbs_Mode, fbs_OrientationOrderParameter);
```

```
    else
        if(actininLayerNumber > 0)
            fprintf(fileID,
'Average,all,%3.3f,%3.3f,%3.3f,;%3.3f,%3.3f,%3.3f,%3.3f\n',
mean(average_sarcomere_area(:)), mean(average_PercentActinArea(:)), Mean,
Std, Median, Mode, OrientationOrderParameter);
        else
            fprintf(fileID,
'Average,all,no,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f,%3.3f\n',
mean(average_PercentActinArea(:)), Mean, Std, Median, Mode,
OrientationOrderParameter);
        end
    end
end
fclose('all');
clearvars thresh;
```

actinDetectSlice.m

```
% detects non-zero orientations from an actin image
% input:
%       actin - the image to be processed
%       blsze - block size that is used to detect a single orientation
%       thresh - threshold corresponding to the dragient arcoss the fiber.
%       Should between 0 and 1.
%       Size - image size
%       sarcomereMask - a mask to be used to filter out any orientations
%       outside of it. If no mask needed, input '1'
% output: all actin orientations that passed the threshold and masking
function [ nonzero_orientation, mask ] =
actinDetectSlice(actin,blkszе,thresh,Size,sarcomereMask)
%actinDetectSlice Returns nonzero_orientation_angles for 1 slice
%   Performs actinDetect on a single Slice and returns the list of non zero
%   angles

        % Identify ridge-like regions and normalise image
        % disp('Normalizing Image and Creating Mask' )
        [normim, mask] = ridgesegment(actin, blksze, thresh);

        % Determine ridge orientations
        % disp('Calculating Ridge Orientations' )
        [orientim, reliability] = ridgeorient(normim, 1, 3, 3);
        %plotridgeorient2(orientim, 20, actin, 1000) % made figure number
1000 so it doesn't overwrite any other open figure
        %show(reliability,5)

        % Only keep orientation values with a reliability greater than 0.5
        reliability_binary = reliability>0.5;

        % Remove 10 pixel wide border where orientation values are not
accurate
        reliability_binary(:,1:1:10) = 0;
        reliability_binary(1:1:10,:) = 0;
        reliability_binary(:,Size-10:1:Size) = 0;
        reliability_binary(Size-10:1:Size,:) = 0;

        %Combine masks for cardiomyocytes and ridge blocks
        mask = mask.*sarcomereMask;

        % Multiply orientation angles by the binary mask image to remove
        % data where there are no cardiomyocytes
        newmask = mask.*reliability_binary;
        orientim = orientim.*newmask;

        % Convert 2D-array to 1D vector
        orientation = orientim(:);

        % Keep non-zero values only
        nonzero_orientation = orientation(orientation ~=0);

end
```

actinDetectTest.m

```
%   Determines 'thresh' parameter for actinDetectSlice
%   Runs actinDetect on 1 slice to determine optimal threshold for actin
%   filament detection
% input:
%       image - the image to be processed
%       blksze - block size that is used to detect a single orientation
%       PixelSize - size of 1 pixel in microns
function [ thresh ] = actinDetectTest( image ,blksze, PixelSize )

thresh = 0;

% Identify ridge-like regions and normalise image
   index = 0;
   while index < 1;
%      % Threshold of standard deviation to decide if a block is a ridge
region
       thresh = input('Enter Threshold (0.1 - 0.2): ');
       disp('Normalizing Image and Creating Mask' )
       [normim, mask] = ridgesegment(image, blksze, thresh);
       show(normim,1);
       show(mask, 2);

       % Determine if normalization and mask look good, click on image to
       % accept or press any key to enter new values
       w = input('Accept Threshold (yes = 1, no = 0): ');
       if w == 1
           disp('Image threshold accepted' )
           index = 1;
       else
           disp('Re-analyze imaging...')
       end

   end
   return

end
```

MIP.m

```
% Creates maximum intensity projection (MIP) from a z-stack
% input:
%       z_stack - z-stack to do the MIP from
%       numberOfChannels - total number of channels in the z-stack
%       channelsToAnalyze - a vector containing numbers of channels that
%       need to be present in the ouput

% output: a cell array containing the MIPs for each requested channel in
% the order they were requested
function [ max_projections ] = MIP(z_stack, numberOfChannels,
channelsToAnalyze)

    numberOfSlices = size(z_stack, 1)/numberOfChannels;
    numberOfOutputChannels = length(channelsToAnalyze);
    max_projections = cell(numberOfOutputChannels, 1);

    for output_channel = 1 : numberOfOutputChannels
        channel = channelsToAnalyze(output_channel);
        max_projections{output_channel} = zeros(size(z_stack{channel}));
    end

    for slice = 1 : numberOfSlices
        for output_channel = 1 : numberOfOutputChannels
            channel = channelsToAnalyze(output_channel);
            max_projections{output_channel} =
max(max_projections{output_channel}, double(z_stack{channel + (slice -
1)*numberOfChannels}));
        end
    end
end
```

OOP.m

```
% OOP calculate orientational order parameter of a matrix
%
% Use this function to get orientational order parameter of actin stained
% images
%
% Inputs = nonzero_orientation (array of the nonzero orientation angles)
% Outputs = Orientation_order_parameter
function [ Orientation_order_parameter ] = OOP( nonzero_orientation )

%Tensor Method for Orientational Order Parameter
%Calculate x and y components of each vector r
if length(nonzero_orientation) > 2
    Vectors(1,:) = cos(nonzero_orientation);
    Vectors(2,:) = sin(nonzero_orientation);
    %Calculate the Orientational Order Tensor for each r and the average
    %Orientational Order Tensor (OOT_Mean)
    for i=1:2
        for j=1:2
            OOT_All(i,j,:)=Vectors(i,:).*Vectors(j,:);
            OOT_Mean(i,j) = mean(OOT_All(i,j,:));
        end
    end
    %Normalize the orientational Order Tensor (OOT), this is necessary to get
the
    %order paramter in the range from 0 to 1
    OOT = 2.*OOT_Mean-eye(2);
    %Find the eigenvalues (orientational parameters) and
    %eigenvectors (directions) of the Orientational Order Tensor
    [~,orient_parameters]=eig(OOT);
    %orientational order parameters is the maximal eigenvalue, while the
    %direcotor is the corresponding eigenvector
    Orientation_order_parameter = max(max(orient_parameters));
else
    Orientation_order_parameter = 0;
end
end
```

Appendix 2. MATLAB Code for Calculating Single Cell Alignment Parameters in a Low-Density Culture

Main Script

```
%% Request data for analysis
SampleInfo = inputdlg({'Number of samples:','Excel file title:', 'nuclei
layer number', 'actin layer number:','alpha-actinin layer
number:'},'Input',1,{'1','single cell analysis','1','2','3'});
SampleCount = str2num(SampleInfo{1});

dapiLayerNumber = str2double(SampleInfo{3});
actinLayerNumber = str2double(SampleInfo{4});
actininLayerNumber = str2double(SampleInfo{5});

rootfolder = path;

%% get files for each sample
for i=1:SampleCount
    message = sprintf('Select files for sample %i',i);

[file,path]=uigetfile({'*.*';'*.lsm';'*.TIF';'*.tif';'*.bmp';'*.jpg'},message
,rootfolder,'Multiselect','on');
    FileList{i,1}=path;
    rootfolder = path;
    FileList{i,2}= file;
end

xlsname = [FileList{1,1} SampleInfo{2} '.csv'];

%% Create file for results
fileID = fopen(xlsname, 'a');
fprintf(fileID, 'picture;cell_type;cell_area µm^2;big_semi_axis
µm^2;small_semi_axis µm^2;aspect_ratio;ellipticity;orientation deg.;\n');

%% Loop for each sample
for i=1:SampleCount

    disp('Sample 1')
    sample = FileList{i,2};
    if ~isa(sample,'char')
        PictureCount = length(sample);
    else
        PictureCount = 1;
    end

    % Setup
    All_angles = [];

    %% Loop over each picture for sample #i
    for j = 1:PictureCount
        % Open images and load actin channel in a matrix
        if PictureCount == 1
```

```
            picture = sample;
            picturename = [FileList{i,1} sample];
        else
            picture = sample{j};
            picturename = [FileList{i,1} sample{j}];
        end

        img = bfopen(picturename);

        dapi = img{1,1}{dapiLayerNumber,1};
        actin = img{1,1}{actinLayerNumber,1};
        sarcomeres = img{1,1}{actininLayerNumber,1};

        info = img{1,4};  % Load OME metadata
        PixelSize = str2double(info.getPixelsPhysicalSizeX(0));
        Size = str2double(info.getPixelsSizeX(0));

        fprintf('Image resolution: %5.5f px/µm\n\n', 1/PixelSize);

        % Define blksze (should be 3/pixelsize based on recommendation)
        blksze = floor(3/PixelSize);

        % take into account only actin that is near sarcomeres

        % this part is useless now
        if ~isa(sarcomeres, 'double'),
            sarcomeres = double(sarcomeres);
            actin = double(actin);
            dapi = double(dapi);
        end

        %% making actin and sarcomere masks
        thresSarc = (0.07*max(sarcomeres(:))+0.93*min(sarcomeres(:)));
        threshAct = (0.05*max(actin(:))+0.95*min(actin(:)));
        d = floor(1.3*blksze)/blksze;

        actin = padarray(actin, [round(2*d) round(2*d)]);
        sarcomeres = padarray(sarcomeres, [round(2*d) round(2*d)]);
        dapi = padarray(dapi, [round(2*d) round(2*d)]);

        [imSizeX,imSizeY] = size(sarcomeres);

        disk = strel('disk', d, 0); % used to merge sarcomeres into mask by
expanding and shrinking the image
        actin_disk = strel('disk', floor(d/2), 0); % used to merge actin into
mask

        maxActinArea = (imSizeX - 2*0.6*floor(d/2))*(imSizeY - 2*floor(d/2));
        maxSarcomereArea = (imSizeX - 2*0.6*d)*(imSizeY - 2*d);

        sarcThreshold = (0.1*max(sarcomeres(:))+0.9*min(sarcomeres(:)));
        sarcomereMask = bwmorph(sarcomeres > sarcThreshold, 'open');
        sarcomeres_border = bwmorph(sarcomereMask, 'remove');
        % dilate and erode to fill the holes between z-lines of
        % sarcomeres
        sarcomereMask = sarcomereMask | imdilate(sarcomeres_border, disk);
        % erode only 60%, because cells' area is higher than sarcomere
coverage
        sarcomereMask = bwmorph(sarcomereMask,'erode',floor(d*0.6));

        actinThreshold = (0.05*max(actin(:))+0.95*min(actin(:)));
        actinMask = bwmorph(actin > actinThreshold, 'open');
        actin_border = bwmorph(actinMask, 'remove');

        % dilate and erode to fill the holes between actin filaments
        actinMask = actinMask | imdilate(actin_border, actin_disk);
        actinMask = bwmorph(actinMask,'erode',floor(floor(d/2)*0.6));
        actinMask = actinMask | sarcomereMask; % expand actin mask to include
sarcomere mask

        %%
        % voodoo magic to show the nice-ish image
        processedDapi = dapi./max(dapi(:));
        processedDapi = medfilt2(processedDapi, [3 3]);
        processedDapi = (processedDapi.^0.3);
        processedDapi = processedDapi.*(processedDapi > 0.1);

        dapiMask = bwmorph(processedDapi > 0, 'erode', 3);
        dapiMask = bwmorph(dapiMask > 0, 'dilate', 3);
        processedDapi = processedDapi.*dapiMask;

        initialMagnification = 60; % Depends on your monitor resolution

        %% Use polygonal tool to split cells. To end the loop, select 100%
black region (double click on the image also works).
        f = figure;
        isZeroPxSelected = 1;
        while (sum(isZeroPxSelected(:)) ~= 0);
            processedActin = (actin/max(actin(:)));
            processedActin = processedActin.*(processedActin > 0.05);
            processedActin = processedActin.^0.5;
            processedActin(processedDapi > 0) = 0;
            combinedImage = cat(3, actinMask.*(processedDapi == 0)./2,
processedActin, processedDapi);
            imshow(combinedImage,'InitialMagnification',
initialMagnification);
            actRemove = impoly();
            actRemove = createMask(actRemove);
            isZeroPxSelected = actRemove.*actinMask;
            actinMask = actinMask.*(~actRemove);
        end
        close(f);



        %% Removes cells that didn't fit the image area
        %  Aslo removes small areas caused by noise

        border = d + 3; % border from the edge of the image within wich I
don't want any cells

        cc = bwconncomp(actinMask, 4);
        imSize = [imSizeX, imSizeY];

        for index = 1 : length(cc.PixelIdxList)
            linInd = cell2mat(cc.PixelIdxList(index));
            [rows, columns] = ind2sub(imSize, linInd);
            if (min([rows; columns]) <= border) || (max([rows; columns]) >=
imSizeX - border)
                actinMask(linInd) = 0;
            end

            if numel(linInd) < (15/PixelSize)^2; % threshold area 20 µm^2
                actinMask(linInd) = 0;
            end

        end

        %% Final actin mask clean up (if necessary). To end the loop, select
100% black region (double click on the image also works).
        f = figure;
        isZeroPxSelected = 1;

        while (sum(isZeroPxSelected(:)) ~= 0);
            imshow(actinMask);
            actRemove = impoly();
            actRemove = createMask(actRemove);
            isZeroPxSelected = actRemove.*actinMask;
            actinMask = actinMask.*(~actRemove);
        end

        close(f);


        %%
        cc = bwconncomp(actinMask, 4);
        cellCount = numel(cc.PixelIdxList);
        cardioCount = 0;
        fibroCount = 0;

        clearvars cardio fibro;

        cardio(:,:,1) = zeros(imSizeX,imSizeY);
        fibro(:,:,1) = zeros(imSizeX,imSizeY);

        % determine the type of each cell and put them in a corresponding
        % array
        for index = 1 : length(cc.PixelIdxList)
            linInd = cell2mat(cc.PixelIdxList(index));
            cellActin = zeros(imSizeX, imSizeY);
            cellActin(linInd) = 1;

            actinArea = numel(linInd);
            intersection = cellActin.*sarcomereMask;
            sarcArea = nnz(intersection);

            if sarcArea/actinArea > 0.5
```

```
                cardioCount = cardioCount + 1;
                cardio(:,:,cardioCount) = cellActin;
            else
                fibroCount = fibroCount + 1;
                fibro(:,:,fibroCount) = cellActin;
            end
        end

        allCellList = cat(3, cardio, fibro);

        %% Calculate Parameters for each cell
        show(actinMask);
        hold on;
        for number = 1 : size(allCellList, 3)
            area = nnz(allCellList(:,:,number)); % mask area ~ cell area
            [rows, columns] = find(allCellList(:,:,number));

            centerRow = sum(rows)/area; % it is not necessarily integer!
            centerColumn = sum(columns)/area; % it is not necessarily
integer!

            % shifting to the center of inertia
            rows1 = rows - centerRow;
            cols1 = columns - centerColumn;

            Irr = sum(cols1.*cols1); % row = y. Irr ~ Iy
            Icc = sum(rows1.*rows1); % column = x. Icc = Ix
            Irc = -sum(rows1.*cols1); % Ixy = Iyx

            inertiaTensor = [Irr, Irc; Irc, Icc];
            [eigenVectors, diagMatrix] = eig(inertiaTensor);

            % orientation - angle between the vector and the regular
            % x-axis

            if diagMatrix(1,1) > diagMatrix(2,2)
                maxMoment = diagMatrix(1,1);
                minMoment = diagMatrix(2,2);
                mainDirection = eigenVectors(:,2);
            else
                maxMoment = diagMatrix(2,2);
                minMoment = diagMatrix(1,1);
                mainDirection = eigenVectors(:,1);
            end

            % a - big semi-axis of the fitted ellipse
            % b - small semi-axis of the fitted ellipse
            a = sqrt(area/pi*sqrt(maxMoment/minMoment));
            b = sqrt(area/pi*sqrt(minMoment/maxMoment));

            orientation = atan(-mainDirection(1)/mainDirection(2))*180/pi;
            if orientation < 0
                orientation = orientation + 180;
            end
            ellipticity = (a - b)/a;
            aspectRatio = a/b;
            color = 'g';
            % quiver(x,y,u,v);
            % red - cardiomyocytes, green - fibroblasts
            if(number <= cardioCount)
                quiver(centerColumn, centerRow, mainDirection(2)*a, 
mainDirection(1)*a, '-r', 'linewidth',2);
                color = 'r';
                cellType = 'cardiomyocyte';
            else
                quiver(centerColumn, centerRow, mainDirection(2)*a, 
mainDirection(1)*a, '-g', 'linewidth',2);
                cellType = 'fibroblast';
            end;
            ellipse(a,b,atan(mainDirection(1)/mainDirection(2)), 
centerColumn, centerRow, color);

            % converts all the relevant stuff to µm (area, inertia moments)
            micronArea = area*(PixelSize^2);
            micronA = a*PixelSize;
            mirconB = b*PixelSize;

            fprintf(fileID, [picture ';' cellType 
';%3.0f;%3.0f;%3.0f;%3.2f;%3.3f;%3.1f\n'], micronArea, micronA, mirconB, 
aspectRatio, ellipticity, orientation);
        end
        hold off;
    end
end

fclose('all');
clearvars thresh;
```

Appendix 3. MATLAB Script for Cell Cluster Analysis.

Main script

```
%% Request data for analysis
SampleInfo = inputdlg({'Number of samples:','Excel file title:', 'nuclei
layer number', 'actin layer number:','alpha-actinin layer
number:'},'Input',1,{'1','touching cell analysis','1','2','3'});
SampleCount = str2num(SampleInfo{1});

dapiLayerNumber = str2double(SampleInfo{3});
actinLayerNumber = str2double(SampleInfo{4});
actininLayerNumber = str2double(SampleInfo{5});

rootfolder = path;

%% get files for each sample
for i=1:SampleCount
    message = sprintf('Select files for sample %i',i);

[file,path]=uigetfile({'*.*';'*.lsm';'*.TIF';'*.tif';'*.bmp';'*.jpg'},message
,rootfolder,'Multiselect','on');
    FileList{i,1}=path;
    rootfolder = path;
    FileList{i,2}= file;
end

xlsname = [FileList{1,1} SampleInfo{2} '.csv'];

%% Create file for results
fileID = fopen(xlsname, 'a');
fprintf(fileID, 'picture;cell_type;cell_area µm^2;big_semi_axis
µm^2;small_semi_axis µm^2;aspect_ratio;ellipticity;orientation deg.;connected
angle;connected cell\n');

%% Loop for each sample
for i=1:SampleCount

    disp('Sample 1')
    sample = FileList{i,2};
    if ~isa(sample,'char')
        PictureCount = length(sample);
    else
        PictureCount = 1;
    end
    sample

    % Setup
    All_angles = [];

    %% Loop over each picture for sample #i
    for j = 1:PictureCount
        % Open images and load actin channel in a matrix
        if PictureCount == 1
            picture = sample;
            picturename = [FileList{i,1} sample];
        else
            picture = sample{j};
            picturename = [FileList{i,1} sample{j}];
        end

        img = bfopen(picturename);

        dapi = img{1,1}{dapiLayerNumber,1};
        actin = img{1,1}{actinLayerNumber,1};
        sarcomeres = img{1,1}{actininLayerNumber,1};

        info = img{1,4};  % Load OME metadata
        PixelSize = str2double(info.getPixelsPhysicalSizeX(0));
        Size = str2double(info.getPixelsSizeX(0));

        fprintf('Image resolution: %5.5f px/µm\n\n', 1/PixelSize);

        % Define blksze (should be 3/pixelsize based on recommendation)
        blksze = floor(3/PixelSize);

        % take into account only actin that is near sarcomeres

        % this part is useless now
        if ~isa(sarcomeres, 'double'),
            sarcomeres = double(sarcomeres);
            actin = double(actin);
            dapi = double(dapi);
        end

        thresSarc = (0.07*max(sarcomeres(:))+0.93*min(sarcomeres(:)));
        threshAct = (0.05*max(actin(:))+0.95*min(actin(:)));
        d = floor(1.3*blksze)/blksze;

        actin = padarray(actin, [round(2*d) round(2*d)]);
        sarcomeres = padarray(sarcomeres, [round(2*d) round(2*d)]);
        dapi = padarray(dapi, [round(2*d) round(2*d)]);

        [imSizeX,imSizeY] = size(sarcomeres);
        sarcomereMask = zeros(imSizeX, imSizeY);
        actinMask = zeros(imSizeX, imSizeY);

        %disp('making sarcomere mask');
        %%
        circle = zeros(2*d*blksze+1);
        for x = -d*blksze:d*blksze
            for y = -d*blksze:d*blksze
                circle(x+d*blksze+1,y+d*blksze+1) = x*x+y*y <
d*blksze*d*blksze;
            end
        end

        for x = d*blksze+1:imSizeX-d*blksze
            for y = d*blksze+1:imSizeY-d*blksze
                if sarcomeres(x,y) > thresSarc
```

```
                    sarcomereMask(x-d*blksze:x+d*blksze, y-
d*blksze:y+d*blksze) = sarcomereMask(x-d*blksze:x+d*blksze, y-
d*blksze:y+d*blksze) | circle;
                end

                if actin(x,y) > threshAct
                    actinMask(x-d*blksze:x+d*blksze, y-d*blksze:y+d*blksze) =
actinMask(x-d*blksze:x+d*blksze, y-d*blksze:y+d*blksze) | circle;
                end
            end
        end

        sarcomereMask = bwmorph(sarcomereMask,'erode',d);
        actinMask = bwmorph(actinMask,'erode',d);

        %% save the objects from the initial mask to use for cell
connectivity detection
        cc = bwconncomp(actinMask, 4);
        savedShapes = cc.PixelIdxList;



        %%
        % voodoo magic to show the nice-ish image
        processedDapi = dapi./max(dapi(:));
        processedDapi = medfilt2(processedDapi, [3 3]);
        processedDapi = (processedDapi.^0.3);
        %processedDapi = processedDapi.*(processedDapi > 0.1);

        dapiMask = bwmorph(processedDapi > 0, 'erode', 3);
        dapiMask = bwmorph(dapiMask > 0, 'dilate', 3);
        processedDapi = processedDapi.*dapiMask;

        initialMagnification = 60; % Depends on your monitor resolution

        %% Use polygonal tool to split cells. To end the loop, select 100%
black region (double click on the image also works).
        f = figure;
        isZeroPxSelected = 1;
        while (sum(isZeroPxSelected(:)) ~= 0);
            processedActin = (actin/max(actin(:)));
            processedActin = processedActin.*(processedActin > 0.05);
            processedActin = processedActin.^0.5;
            processedActin(processedDapi > 0) = 0;
            combinedImage = cat(3, actinMask.*(processedDapi == 0)./2,
processedActin, processedDapi);
            imshow(combinedImage,'InitialMagnification',
initialMagnification);
            actRemove = impoly();
            actRemove = createMask(actRemove);
            isZeroPxSelected = actRemove.*actinMask;
            actinMask = actinMask.*(~actRemove);
        end
        close(f);

        %% remove small areas caused by noise
        border = d + 3; % border from the edge of the image within wich I
don't want any cells

        cc = bwconncomp(actinMask, 4);
        imSize = [imSizeX, imSizeY];
        borderCells = zeros(imSize);

        for index = 1 : length(cc.PixelIdxList)
            linInd = cell2mat(cc.PixelIdxList(index));
            [rows, columns] = ind2sub(imSize, linInd);

            if (min([rows; columns]) <= border) || (max([rows; columns]) >=
imSizeX - border)
                borderCells(linInd) = 1;
            end
            if numel(linInd) < (15/PixelSize)^2; % threshold area 20 µm^2
                actinMask(linInd) = 0;
            end

        end

        %% Final actin mask clean up (if necessary). To end the loop, select
100% black region (double click on the image also works).
        f = figure;
        isZeroPxSelected = 1;

        while (sum(isZeroPxSelected(:)) ~= 0);
            imshow(actinMask,'InitialMagnification', initialMagnification);
            actRemove = impoly();
            actRemove = createMask(actRemove);
            isZeroPxSelected = actRemove.*actinMask;
            actinMask = actinMask.*(~actRemove);
        end

        close(f);


        %%
        cc = bwconncomp(actinMask, 4);
        cellCount = numel(cc.PixelIdxList);

        % cellInfo(row,column)
        % row = cell number
        % col 1 - cell type: 0 = firboblast, 1 = cardiomyocyte
        % col 2 - originating object (2 cells with
        % the same orig. object are touching)
        cellInfo = zeros(length(cc.PixelIdxList),2);
        cellList = cc.PixelIdxList;

        % determine the type of each cell
        for index = 1 : length(cc.PixelIdxList)
            linInd = cell2mat(cc.PixelIdxList(index));
            cellActin = zeros(imSizeX, imSizeY);
            cellActin(linInd) = 1;
            actinArea = numel(linInd);
            intersection = cellActin.*sarcomereMask;
            sarcArea = nnz(intersection);

            if sarcArea/actinArea > 0.5
                cellInfo(index,1) = 1;
            else
                cellInfo(index,1) = 0;
            end

            for obj = 1 : length(savedShapes)

                objInd = cell2mat(savedShapes(obj));
                objPixels = zeros(imSizeX, imSizeY);
                objPixels(objInd) = 1;

                intersectionObj = cellActin.*objPixels;
                if nnz(intersectionObj) > 0.5*actinArea
                    cellInfo(index,2) = obj;
                    break;
                else if obj == length(savedShapes)
                        show('the originating cluster was not found.
SOMETHING WENT WRONG!');
                    end
                end

            end
        end

        clusters = cell(length(savedShapes), 1); % indices of cells belonging
to each cluster
        for cNum = 1 : length(savedShapes)
            clusters{cNum,1} = find(cellInfo(:,2) == cNum);
        end

        % shows what cells withing the cluster are close to each other
        proximityTable = -ones(length(cellInfo));

        %% Calculate Parameters for each cell
        actinMask = actinMask.*(~borderCells);
        show(actinMask);
        hold on;
        for number = 1 : length(cellList)
            proximityTable(number,number) = 0;

            linInd = cell2mat(cellList(number));
            cellActin = zeros(imSizeX, imSizeY);
            cellActin(linInd) = 1;

            area = numel(linInd); % mask area ~ cell area
            [rows, columns] = ind2sub(imSize, linInd);

            centerRow = sum(rows)/area; % it is not necessarily integer!
            centerColumn = sum(columns)/area; % it is not necessarily
integer!
```

```
            if (min([rows; columns]) <= border) || (max([rows; columns]) >=
imSizeX - border)
                actinMask(linInd) = 0;
            else

                numberOfConnections = 0;
                connectedCell = -1; % 0 - fibroblast, 1 - cardiomyocyte
                connectedAngle = 0;
                neighbors = clusters{cellInfo(number,2),1};
                neighbors = neighbors(neighbors > 0);

                for n = 1 : length(neighbors)

                    if(proximityTable(number, neighbors(n)) == -1)
                        neighborInd = cell2mat(cellList(neighbors(n)));
                        neighborActin = zeros(imSizeX, imSizeY);
                        neighborActin(neighborInd) = 1;

                        dilatedCell = bwmorph(cellActin, 'dilate',
3.5/PixelSize); % 3 µm dilation
                        dilatedNeighbor = bwmorph(neighborActin, 'dilate',
3.5/PixelSize); % 3 µm dilation

                        intersection = dilatedCell.*dilatedNeighbor;

                        if nnz(intersection) > 0
                            intArea = nnz(intersection);
                            [r,c] = ind2sub(imSize, find(intersection));

                            ri = sum(r)/intArea;
                            ci = sum(c)/intArea;

                            vectR = ri - centerRow;
                            vectC = ci - centerColumn;
                            vectLength = sqrt(vectR^2 + vectC^2);
                            vectR = vectR/vectLength;
                            vectC = vectC/vectLength;
                            sinA = -vectR;
                            cosA = vectC;
                            angleA = 0;

                            if sinA >= 0
                                angleA = acos(cosA);
                            else
                                if cosA < 0
                                    angleA = pi - asin(sinA);
                                else
                                    angleA = 2*pi + asin(sinA);
                                end
                            end

                            if(angleA == 0)
                                angleA = 2*pi;
                            end

                            if(angleA < 0)
```

```
                                show('Ivan, you messed up the angle');
                            end

                            proximityTable(number,neighbors(n)) = angleA;
                           % proximityTable(neighbors(n),number) = angleA;
                        else
                            proximityTable(number,neighbors(n)) = 0;
                            proximityTable(neighbors(n),number) = 0;
                        end
                    end

                    if(proximityTable(number, neighbors(n)) > 0)
                        numberOfConnections = numberOfConnections + 1;
                        connectedCell = cellInfo(neighbors(n),1);
                        connectedAngle = proximityTable(number,neighbors(n));
                    end
                end

                if numberOfConnections == 1 || numberOfConnections == 0

                    % shifting to the center of inertia
                    rows1 = rows - centerRow;
                    cols1 = columns - centerColumn;

                    Irr = sum(cols1.*cols1); % row = y. Irr ~ Iy
                    Icc = sum(rows1.*rows1); % column = x. Icc = Ix
                    Irc = -sum(rows1.*cols1); % Ixy = Iyx

                    inertiaTensor = [Irr, Irc; Irc, Icc];
                    [eigenVectors, diagMatrix] = eig(inertiaTensor);

                    % orientation - angle between the vector and the regular
                    % x-axis

                    if diagMatrix(1,1) > diagMatrix(2,2)
                        maxMoment = diagMatrix(1,1);
                        minMoment = diagMatrix(2,2);
                        mainDirection = eigenVectors(:,2);
                    else
                        maxMoment = diagMatrix(2,2);
                        minMoment = diagMatrix(1,1);
                        mainDirection = eigenVectors(:,1);
                    end

                    % a - big semi-axis of the fitted ellipse
                    % b - small semi-axis of the fitted ellipse
                    a = sqrt(5*maxMoment/area);
                    b = sqrt(5*minMoment/area);

                    orientation = atan(-
mainDirection(1)/mainDirection(2))*180/pi;
                    if orientation < 0
                        orientation = orientation + 180;
                    end
                    ellipticity = (a - b)/a;
                    aspectRatio = a/b;
                    color = 'g';
                    % quiver(x,y,u,v);
                    % red - cardiomyocytes, green - fibroblasts
                    if(cellInfo(number,1) == 1)
                        quiver(centerColumn, centerRow, mainDirection(2)*a,
mainDirection(1)*a, '-r', 'linewidth',2);
                        color = 'r';
                        cellType = 'cardiomyocyte';
                    else if cellInfo(number,1) == 0
                            quiver(centerColumn, centerRow,
mainDirection(2)*a, mainDirection(1)*a, '-g', 'linewidth',2);
                            cellType = 'fibroblast';
                        end
                    end;
                    ellipse(a,b,atan(mainDirection(1)/mainDirection(2)),
centerColumn, centerRow, color);

                    % converts all the relevant stuff to µm (area, inertia
moments)
                    micronArea = area*(PixelSize^2);
                    micronA = a*PixelSize;
                    mirconB = b*PixelSize;

                    if(connectedCell == 0)
                        connectedCell = 'fibroblast';
                    else
                        if(connectedCell == 1)
                            connectedCell = 'cardiomyocyte';
                        else
                            connectedCell = 'none';
                        end
                    end

                    connectedAngle = connectedAngle*180/pi;

                    fprintf(fileID, [picture ';' cellType
';%3.0f;%3.0f;%3.0f;%3.2f;%3.3f;%3.1f;%3.0f;' connectedCell '\n'],
micronArea, micronA, mirconB, aspectRatio, ellipticity, orientation,
connectedAngle);
                end
            end
        end
        hold off;

    end
end

fclose('all');
clearvars thresh;
```

Appendix 4. MATLAB Code for Creating Biomimetic Heat Maps

Script for generating location-specific alignment data for the biomimetic pattern heat maps

```
%% Request data
SampleInfo = inputdlg({'Number of samples:','Excel file title:','actin
layer:','alpha-actinin layer:', 'fibronectin
layer:'},'Input',1,{'1','heat_map_data','2','3','4'});
SampleCount = str2num(SampleInfo{1});

actinLayerNumber = str2num(SampleInfo{3});
actininLayerNumber = str2num(SampleInfo{4});
fibronectinLayerNumber = str2num(SampleInfo{5});

rootfolder = path;

% get files for each sample
for i=1:SampleCount
    message = sprintf('Select files for sample %i',i);

[file,path]=uigetfile({'*.*';'*.lsm';'*.TIF';'*.tif';'*.bmp';'*.jpg'},message
,rootfolder,'Multiselect','on');
    FileList{i,1}=path;
    rootfolder = path;
    FileList{i,2}= file;
end

% Loop for each sample
for i=1:SampleCount

    disp('Sample 1')
    sample = FileList{i,2};
    if ~isa(sample,'char')
        PictureCount = length(sample);
    else
        PictureCount = 1;
    end

    % Setup
    All_angles = [];
    fbs_All_angles = [];
    average_PercentActinArea = 0;
    average_fbs_PercentActinArea = 0;

    cm_orient_block_data = cell(155/1, 250/1);
    fb_orient_block_data = cell(155/1, 250/1);

    imageData = cell(PictureCount,6);

    % First loop. Align all FN patterns together.
    for j = 1:PictureCount
        disp(['Processing image ', int2str(j), ' out of ',
int2str(PictureCount) '...']);

        discardImage = 0;
        % Open images and load actin channel in a matrix
        if PictureCount == 1
            picture = sample;
            picturename = [FileList{i,1} sample];
        else
            picture = sample{j};
            picturename = [FileList{i,1} sample{j}];
        end

        img = bfopen(picturename);

        actin = img{1,1}{actinLayerNumber,1};
        sarcomeres = img{1,1}{actininLayerNumber,1};
        fibronectin = img{1,1}{fibronectinLayerNumber,1};

        info = img{1,4};  % Load OME metadata
        PixelSize = str2double(info.getPixelsPhysicalSizeX(0));

        fprintf('Image resolution: %5.5f px/µm\n\n', 1/PixelSize);

        % Define blksze (should be 3/pixelsize based on recommendation)
        blksze = floor(3/PixelSize);

        % take into account only actin that is near sarcomeres

        if ~isa(sarcomeres, 'double'),
            sarcomeres = double(sarcomeres);
            actin = double(actin);
            fibronectin = double(fibronectin);
        end

        [imSizeX,imSizeY] = size(sarcomeres);

        fibronectin = fibronectin/max(fibronectin(:));
        fibronectin = medfilt2(fibronectin,[3 3]);
        fibronectin2 = fibronectin > 0.22;
        fibronectin2 = bwmorph(fibronectin2, 'close');
        fibronectin2 = bwmorph(fibronectin2, 'open');
        %fibronectin = fibronectin2 .* fibronectin;
        %fibronectin = fibronectin2;

        if j == 0
            %% Transforming all layers to make the FN pattern rectangular
(250x155 µm)
            imshow(fibronectin);
            disp('choose 2 pattern corners along the vertical axis' );
            [movingPointsX, movingPointsY] = getpts;
            close;
            % use only last 2 points
            movingPoints = [movingPointsX(1:2), movingPointsY(1:2)];
            patternOrigin = movingPoints(1,:);
            distance = sqrt((movingPointsX(1) - movingPointsX(2))^2 +
(movingPointsY(1) - movingPointsY(2))^2);
```

```
            fixedPoints = [movingPoints(1,1), movingPoints(1,2);
                movingPoints(1,1), movingPoints(1,2)+distance];

            transform =
fitgeotrans(movingPoints,fixedPoints,'nonreflectivesimilarity');
            %transform = estimateGeometricTransform(movingPoints,
fixedPoints, 'similarity');
            [outboundsX, outboundsY] = outputLimits(transform,[1 imSizeX],[1
imSizeY]);
            patternOrigin(1) = patternOrigin(1) - (outboundsX(1) - 1);
            patternOrigin(2) = patternOrigin(2) - (outboundsY(1) - 1);
            newPatternOrigin = patternOrigin;
            reference_fn_image = imwarp(fibronectin, transform);
            figure
            imshow(reference_fn_image);

            fn_pattern = reference_fn_image(round(patternOrigin(2)) :
round(patternOrigin(2)) + round(250/PixelSize) - 1, round(patternOrigin(1)) :
round(patternOrigin(1)) + round(155/PixelSize) - 1);
        else
            while 1
                %disp('bad match');

                imshow(fibronectin);
                disp('choose 2 pattern corners' );
                [movingPointsX, movingPointsY] = getpts;
                close;

                pointsEntered = numel(movingPointsX);

                if pointsEntered >= 2
                    if(pointsEntered == 3)
                        % use only last 4 points
                        movingPoints = [movingPointsX(numel(movingPointsX) -
1 : numel(movingPointsX)), movingPointsY(numel(movingPointsY) - 1 :
numel(movingPointsY))];
                        newPatternOrigin = [movingPointsX(1),
movingPointsY(1)];
                    else
                        movingPoints = [movingPointsX, movingPointsY];
                        newPatternOrigin = movingPoints(1,:);
                    end

                    distance = sqrt((movingPoints(1,1) - movingPoints(2,1))^2
+ (movingPoints(1,2) - movingPoints(2,2))^2);
                    fixedPoints = [movingPoints(1,1), movingPoints(1,2);
                        movingPoints(1,1), movingPoints(1,2)+distance];

                    transform =
fitgeotrans(movingPoints,fixedPoints,'nonreflectivesimilarity');
                else
                    if numel(movingPointsX) == 1
                        movingPoints = [movingPointsX, movingPointsY];
                        newPatternOrigin = movingPoints(1,:);
                        % transform stays the same, so empty line here
                    else
```

```
                        discardImage = 1;
                    end
                end

                newPatternOrigin = transformPointsForward(transform,
newPatternOrigin);

                [outboundsX, outboundsY] = outputLimits(transform,[1
imSizeX],[1 imSizeY]);
                newPatternOrigin(1) = newPatternOrigin(1) - (outboundsX(1) -
1);
                newPatternOrigin(2) = newPatternOrigin(2) - (outboundsY(1) -
1);

                new_fn_image = imwarp(fibronectin, transform);
                overlay_patterns(fn_pattern > 0.15, new_fn_image > 0.15,
newPatternOrigin);

                disp('red: first pattern, green: current pattern. If patterns
do not match, choose shift vector (first point - new pattern, second -
old)');
                [inputX, inputY] = getpts;
                while numel(inputX) == 2
                    close;
                    newPatternOrigin(1) = newPatternOrigin(1) + (inputX(2) -
inputX(1));
                    newPatternOrigin(2) = newPatternOrigin(2) + (inputY(2) -
inputY(1));
                    overlay_patterns(fn_pattern > 0.15, new_fn_image > 0.15,
newPatternOrigin);
                    [inputX, inputY] = getpts;
                end
                if isempty(input('Type any key to fix the pattern manually or
Enter to continue'))
                    break;
                end
            end

            clearvars ptsCurrent featuresCurrent validPtsCurrent indexPairs
matchedReference matchedCurrent
        end

        if discardImage == 0
            imageData{j,1} = actin;
            imageData{j,2} = sarcomeres;
            imageData{j,3} = fibronectin;
            imageData{j,4} = PixelSize;
            imageData{j,5} = transform;
            imageData{j,6} = newPatternOrigin;
        end

    end

    %% Second loop. Actin orientation detection.
    for j = 1: size(imageData,1)
        if ~isempty(imageData{j,1})
```

```
            disp(['Analyzing alignment of image # ', int2str(j), ' out of ',
int2str(size(imageData,1)) '...']);

            actin = imageData{j,1};
            sarcomeres = imageData{j,2};
            fibronectin = imageData{j,3};
            PixelSize = imageData{j,4};
            transform = imageData{j,5};
            newPatternOrigin = imageData{j,6};

            blksze = floor(3/PixelSize);
            [imSizeX,imSizeY] = size(sarcomeres);

            d = floor(1.5*blksze);
            disk = strel('disk', d, 0); % used to merge sarcomeres into mask
by expanding and shrinking the image
            actin_disk = strel('disk', floor(d/2), 0); % used to merge actin
into mask

            threshold = (0.1*max(sarcomeres(:))+0.9*min(sarcomeres(:)));
            sarcomereMask = bwmorph(sarcomeres > threshold, 'open');
            sarcomeres_border = bwmorph(sarcomereMask, 'remove');
            % dilate and erode to fill the holes between z-lines of
            % sarcomeres
            sarcomereMask = sarcomereMask | imdilate(sarcomeres_border,
disk);
            % erode only 60%, because cells' area is higher than sarcomere
coverage
            sarcomereMask = bwmorph(sarcomereMask,'erode',floor(d*0.6));

            [imSizeX,imSizeY] = size(actin);

            maxActinArea = (imSizeX - 2*0.6*floor(d/2))*(imSizeY -
2*floor(d/2));
            maxSarcomereArea = (imSizeX - 2*0.6*d)*(imSizeY - 2*d);

            actinThreshold = (0.05*max(actin(:))+0.95*min(actin(:)));
            actinMask = bwmorph(actin > actinThreshold, 'open');
            actin_border = bwmorph(actinMask, 'remove');

            % dilate and erode to fill the holes between actin filaments
            actinMask = actinMask | imdilate(actin_border, actin_disk);
            actinMask = bwmorph(actinMask,'erode',floor(floor(d/2)*0.6));
            if(actininLayerNumber >= 0)
                actinMask = actinMask | sarcomereMask; % expand actin mask to
include sarcomere mask
            end

            if ~exist('thresh','var')
                thresh = actinDetectTest(actin,blksze,PixelSize);
            end

            %%
            actin = imwarp(actin, transform);
            sarcomeres = imwarp(sarcomeres, transform);
            sarcomereMask = imwarp(sarcomereMask, transform);
            actinMask = imwarp(actinMask, transform);

            %% splitting orientations between blocks
            block_x = 0;
            block_y = 0;

            cm_orientim =
create_orientation_image(actin,blksze,thresh,sarcomereMask);
            %fb_orientim =
create_orientation_image(actin,blksze,thresh,~sarcomereMask);

            for row = 1 : size(cm_orientim, 1)
                for col = 1 : size(cm_orientim, 2)
                    if cm_orientim(row,col) ~= 0
                        % || fb_orientim(row,col) ~= 0
                        block_x = floor((col -
newPatternOrigin(1))/(1/PixelSize));
                        block_y = floor((row -
newPatternOrigin(2))/(1/PixelSize));

                        block_x = rem(block_x, 155/1);
                        block_y = rem(block_y, 250/1);

                        if block_x <= 0
                            block_x = block_x + 155/1;
                        end
                        if block_y <= 0
                            block_y = block_y + 250/1;
                        end
                        if cm_orientim(row,col) ~= 0
                            cm_orient_block_data{block_x, block_y} =
[cm_orient_block_data{block_x, block_y}; cm_orientim(row,col)];
                        end
                        %                         if fb_orientim(row,col) ~=
0
                        %
fb_orient_block_data{block_x, block_y} = [fb_orient_block_data{block_x,
block_y}; fb_orientim(row,col)];
                        %                         end
                    end
                end
            end

            % Total number of actin positive pixels in the skeleton image
            % Sarcomere density = total/(image area)
            PercentActinArea = nnz(sarcomereMask)/maxActinArea;

            average_PercentActinArea = average_PercentActinArea +
PercentActinArea;
        else
            disp(['Image # ', int2str(j), ' is discarded!']);
        end
        %}
    end
    cm_alignment_data = zeros(155/1, 250/1, 6);
    all_cm_angles = [];
    for block_x = 1 : 155/1
        for block_y = 1 : 250/1
            angles = [];
            all_cm_angles = [all_cm_angles; cm_orient_block_data{block_x,
block_y}];
            for deltaX = -2 : 2
                for deltaY = -2 : 2
                    indX = rem(block_x + deltaX + 155/1,155/1);
                    indY = rem(block_y + deltaY + 250/1,250/1);
                    if indX <= 0
                        indX = indX + 155;
                    end
                    if indY <= 0
                        indY = indY + 250;
                    end
                    angles = [angles; cm_orient_block_data{indX, indY}];
                end
            end
            if numel(angles) > 5

                % Convert radians to degrees
                angles_deg = rad2deg(angles);

                Mean = mean(angles_deg);
                Std = std(angles_deg);
                Median = median(angles_deg);

                % Create histogram
                [n,xout] = hist(angles_deg,180);
                dx = xout(2)-xout(1);                   % calc a single bin
width
                n = n / sum( n*dx );                    % normalize histogram
to have area of 1

                % Find mode
                [~,I] = max(n);
                Mode = xout(I);

                OrientationOrderParameter = OOP(angles);
                cm_alignment_data(block_x, block_y, :) = [numel(angles),
Mean, Std, Median, Mode, OrientationOrderParameter];
            else
                cm_alignment_data(block_x, block_y, :) = [0, 0, 0, 0, 0, 0];
            end

        end
    end

    average_values = [0; 0; 0; 0; 0; 0]; % average for the pattern location

    for k = 1 : 6
        average_values(k) =
sum(sum(cm_alignment_data(:,:,k)))/nnz(cm_alignment_data(:,:,k));
    end
    all_cm_angles_deg = rad2deg(all_cm_angles);
    Mean = mean(all_cm_angles_deg);
    Std = std(all_cm_angles_deg);
    Median = median(all_cm_angles_deg);
    % Create histogram
    [n,xout] = hist(all_cm_angles_deg,180);
    dx = xout(2)-xout(1);                   % calc a single bin width
    n = n / sum( n*dx );                    % normalize histogram to have
area of 1
    % Find mode
    [~,I] = max(n);
    Mode = xout(I);
    OrientationOrderParameter = OOP(all_cm_angles);

    overall_values = [average_values(1), Mean, Std, Median, Mode,
OrientationOrderParameter];
    %overall_values = average_values;

    average_PercentActinArea = average_PercentActinArea/length(imageData);
    %average_fbs_PercentActinArea =
average_fbs_PercentActinArea/PictureCount;

    data_label = {'Number of orientation vectors in each block';
        'Mean orientation angle';
        'Standatd deviation of the mean angle';
        'Median orientation angle';
        'Most probable orienation angle';
        'OOP'};

    %Save data
    save(strcat(path,['sample_', num2str(i), '_results.mat']), 'fn_pattern',
'cm_alignment_data', 'fb_alignment_data');

end

fclose('all');
clearvars thresh;
```

Script for simultaneously creating low and high density heat maps using the alignment data

```
% to use this script, you need to load alignment maps for both low and high
% density cases named cm_alignment_data_low and cm_alignment_data_high. The
% same for the fn_pattern_low, fn_pattern_high. Modify the paths to those
files below.

load('E:\Dropbox (RBG)\Dropbox (RBG)\Lab stuff\2015\2015-05-07 Matlab heat
map stuff\first results (low density)\combined data\contour
pattern\data.mat', 'fn_pattern','cm_alignment_data');
fn_pattern_low = fn_pattern;
cm_alignment_data_low = cm_alignment_data;
```

```
PixelSize = 155/size(fn_pattern_low, 2);
bin_size = 1/PixelSize;

%load('/Volumes/Macintosh HD 2/Drop-boxes/Dropbox (RBG)/Lab stuff/2015/2015-
05-07 Matlab heat map stuff/high density/results_high_density.mat',
'fn_pattern','cm_alignment_data');
load('E:\Dropbox (RBG)\Dropbox (RBG)\Lab stuff\2015\2015-05-07 Matlab heat
map stuff\high density\results_high_density.mat',
'fn_pattern','cm_alignment_data');
fn_pattern_high = fn_pattern;
cm_alignment_data_high = cm_alignment_data;
fn_pattern_high = imresize(fn_pattern_high, [round(250/PixelSize),
round(155/PixelSize)], 'Method', 'bilinear');

cm_alignment_data_low(:,:,1) =
cm_alignment_data_low(:,:,1)/max(max(cm_alignment_data_low(:,:,1)))*100;
cm_alignment_data_high(:,:,1) =
cm_alignment_data_high(:,:,1)/max(max(cm_alignment_data_high(:,:,1)))*100;

average_values_low = [0; 0; 0; 0; 0; 0]; % average for the pattern location
average_values_high = [0; 0; 0; 0; 0; 0]; % average for the pattern location

for k = 1 : 6
    average_values_low(k) =
sum(sum(cm_alignment_data_low(:,:,k)))/nnz(cm_alignment_data_low(:,:,k));
    average_values_high(k) =
sum(sum(cm_alignment_data_high(:,:,k)))/nnz(cm_alignment_data_high(:,:,k));
end

data_label = {'Normalized Cell Occurrence';
    'Mean Orientation Angle';
    'Standard Deviation of the Mean Angle';
    'Median Orientation Angle';
    'Most Probable Orienation Angle';
    'OOP'};
x_axis_label = { 'Normalized Cell Occurrence';
    'Mean Orientation Angle, Degrees'
    'Std. Dev. of Mean Angle, Degrees'
    'Median Orientation Angle, Degrees'
    'Most Probable Orienation Angle, Degrees';
    'OOP'};

orig_pattern = imread('E:\Dropbox (RBG)\Dropbox (RBG)\Lab stuff\2015\2015-05-
07 Matlab heat map stuff\original
pattern\BM_pattern_for_heat_map_contour_4.png');
orig_pattern = rgb2gray(orig_pattern);
orig_pattern = double(orig_pattern);
orig_pattern = orig_pattern/max(orig_pattern(:));
orig_pattern = imresize(orig_pattern, [round(250/PixelSize),
round(155/PixelSize)], 'Method', 'bilinear');

newParrenOrigin_low = matchPatterns(fn_pattern_low, orig_pattern);
newParrenOrigin_high = matchPatterns(fn_pattern_high,orig_pattern);

shift_vector_low = round(newParrenOrigin_low/bin_size);
shift_vector_high = round(newParrenOrigin_high/bin_size);
```

```
correlation = [0 0 0 0 0 0];

final_images = cell(2,6);

for mapNum = 1 : 6

    map_low = createHeatMap(cm_alignment_data_low(:,:,mapNum),
shift_vector_low, PixelSize);
    map_high = createHeatMap(cm_alignment_data_high(:,:,mapNum),
shift_vector_high, PixelSize);

    correlation(mapNum) = corr2(map_high, map_low);

    max_value = round(max([map_low(:); map_high(:)]),3, 'significant');
    min_value = round(min([map_low(:); map_high(:)]),3, 'significant');

    max_value = round(max_value,3, 'decimals');
    min_value = round(min_value,3, 'decimals');

    avg_value_low = round(average_values_low(mapNum),3, 'significant');
    avg_value_high = round(average_values_high(mapNum),3, 'significant');

    avg_value_low = round(avg_value_low,3, 'decimals');
    avg_value_high = round(avg_value_high,3, 'decimals');

    %% under- and oversaturate maps for higher contrast (optional)
    if(mapNum == 2 || mapNum == 3 || mapNum == 4 || mapNum == 5)
        cut_out_fraction = 0.1; % fraction of data that you want to make
outside the color range (over or undersaturated) to increase contrast of what
is near the average
        avg_value = (avg_value_low + avg_value_high)/2;

        sorted_data_low = sort(map_low(:));
        sorted_data_high = sort(map_high(:));
        data_size = length(sorted_data_low);

        min_value = min(sorted_data_low(round(data_size*cut_out_fraction/2)),
sorted_data_high(round(data_size*cut_out_fraction/2)));
        max_value = max(sorted_data_low(round(data_size*(1-
cut_out_fraction/2))), sorted_data_high(round(data_size*(1-
cut_out_fraction/2))));

        max_value = round(max_value,3, 'significant');
        min_value = round(min_value,3, 'significant');

        max_value = round(max_value,3, 'decimals');
        min_value = round(min_value,3, 'decimals');
        map_low = (map_low - min_value).*(map_low > min_value) + min_value;
%undersaturate data
        map_low = (map_low - max_value).*(map_low < max_value) + max_value;
%oversaturate data

        map_high = (map_high - min_value).*(map_high > min_value) +
min_value; %undersaturate data
```

```
        map_high = (map_high - max_value).*(map_high < max_value) +
max_value; %oversaturate data
    end
    %% show final images
   final_images{1,mapNum} = draw_heat_map(map_low, orig_pattern, min_value,
max_value, avg_value_low, PixelSize, 94, strcat(data_label(mapNum), ', Low
Cell Density'));
   final_images{2,mapNum} = draw_heat_map(map_high, orig_pattern, min_value,
max_value, avg_value_high, PixelSize, 94, strcat(data_label(mapNum), ', High
Cell Density'));

    %% Show correlations:
    %disp(strcat(data_label(mapNum),' correlation:_',
num2str(correlation(mapNum))));

    %% Draw histograms
    bin_width = (max_value - min_value)/250;

    current_plot = figure('units','normalized','position',[0.1 0.1 0.43 0.6])
    hist_low = histogram(map_low(:), 100, 'BinWidth', bin_width);
%    title(strcat(data_label(mapNum), ' Distribution'), 'FontSize', 30);
    xlim([min_value max_value]);
    xlabel(x_axis_label(mapNum), 'FontSize', 30, 'FontName', 'Arial',
'FontWeight', 'bold');
    ylabel('Frequency', 'FontSize', 30, 'FontName', 'Arial', 'FontWeight',
'bold');
    set(gca,'FontSize',25, 'FontName', 'Arial');

    hold on
    hist_high = histogram(map_high(:), 100, 'BinWidth', bin_width);

    hist_low.EdgeColor = 'none';
    hist_high.EdgeColor = 'none';
    legend('\fontsize{30}\fontname{Arial} low cell density',
'\fontsize{30}\fontname{Arial} high cell density', 'Location',
'southoutside');

    saveas(current_plot, strcat('E:\Dropbox (RBG)\Dropbox (RBG)\Lab
stuff\2015\2015-05-07 Matlab heat map stuff\vector graphics histograms\new\',
data_label{mapNum}, '.pdf'));

end
%% save images as png files
folder_name = uigetdir
if(folder_name ~= 0)
    for i = 1 : 6
        imwrite(final_images{1,i}, strcat(folder_name, '/', data_label{i},
'_low_density.png'));
        imwrite(final_images{2,i}, strcat(folder_name, '/', data_label{i},
'_high_density.png'));
    end
end
```

createHeatMap.m

```
function heat_map = createHeatMap(cm_alignment_data, shift_vector, PixelSize)

bin_size = 1/PixelSize;
heat_map = zeros(round(250/PixelSize), round(155/PixelSize));

    for block_x = 1 : 155/1
        for block_y = 1 : 250/1
            transformed_block_x = 155-block_x - shift_vector(1);
            transformed_block_y = 250-block_y - shift_vector(2);

            if transformed_block_x < 1
                transformed_block_x = transformed_block_x + (floor(-
transformed_block_x/155) + 1)*155;
            else if transformed_block_x > 155
                    transformed_block_x = transformed_block_x -
(floor((transformed_block_x - 155)/155) + 1)*155;
                end
            end

            if transformed_block_y <= 0
                transformed_block_y = transformed_block_y + (floor(-
transformed_block_y/250) + 1)*250;
            else if transformed_block_y > 250
                    transformed_block_y = transformed_block_y -
(floor((transformed_block_y - 250)/250) + 1)*250;
                end
            end

            heat_map(max(1,round((block_y - 1)*bin_size)) :
round(min(250*bin_size,(block_y)*bin_size)), max(1,round((block_x -
1)*bin_size)) : round(min(155*bin_size,block_x*bin_size))) =
cm_alignment_data(transformed_block_x, transformed_block_y);
        end
    end
```

draw_heat_map.m

```
function final_image =  draw_heat_map(map_data, orig_pattern, min_value,
max_value, avg_value, PixelSize, font_size, picTitle)

low_hue = 240/360;
high_hue = 0;
palitra_length = 800;
%font_size = 10;

dim1 = size(map_data, 1);
dim2 = size(map_data, 2);

hsv = zeros(dim1, dim2, 3);
hsv(:,:,1) = low_hue + (high_hue - low_hue)*(map_data - min_value)/(max_value
- min_value);
hsv(:,:,2) = 1;
hsv(:,:,3) = orig_pattern*0.8 + 0.1;
```

```
%     figure;
%     imshow(OOP_map);
%     figure
%     imshow(hsv2rgb(hsv));

scale_hsv = zeros(palitra_length, 200,3);
scale_hsv(:,:,3) = 1;
scale_hsv(:,:,2) = 0;
scale_hsv(:,5:40,2) = 1;

for row = 1 : palitra_length
    scale_hsv(row,5:40,1) = high_hue - (high_hue -
low_hue)*(row/palitra_length);
end

min_y = 1;
max_y = palitra_length;
avg_y = round(1 + (palitra_length - 1)*(max_value - avg_value)/(max_value -
min_value));

min_text_y = 1;
max_text_y = max_y - font_size - 1;
avg_text_y = avg_y - round(font_size/2);

scale_hsv(max_y-3:max_y,5:43,3) = 0;
scale_hsv(min_y:min_y+3,5:43,3) = 0;
scale_hsv(avg_y-1:avg_y+1,5:43,1) = 0;
scale_hsv(avg_y-1:avg_y+1,5:43,2) = 1;

textInserter = vision.TextInserter('%s', 'LocationSource', 'Input port',
'Color',  [0, 0, 0], 'FontSize', font_size);
strings = uint8([num2str(max_value) 0 num2str(min_value) 0
num2str(avg_value)]);
labeled = step(textInserter, hsv2rgb(scale_hsv), strings, int32([45,
min_text_y; 45, max_text_y; 45, avg_text_y]));

scale_bar_microns = floor(120*PixelSize/10)*10;
scale_bar_px = scale_bar_microns/PixelSize;
left_margin = round((120 - scale_bar_px)/2);

scale_bar = ones(120,120,3);
scale_bar(80:100, left_margin:round(left_margin + scale_bar_px), :) = 0;
string = uint8([num2str(scale_bar_microns) ' um']);
textInserter2 = vision.TextInserter('%s', 'LocationSource', 'Input port',
'Color',  [0, 0, 0], 'FontSize', 30);
scale_bar = step(textInserter2, scale_bar, string, int32([16 40]));
%     figure;
%     imshow(scale_bar);

final_image = ones(dim1, dim2 + size(labeled,2),3);
final_image(:,1:dim2,:) = hsv2rgb(hsv);
final_image(1:size(labeled,1),dim2 + 1:dim2 + size(labeled,2),:) = labeled;
final_image(size(final_image,1) - size(scale_bar,1) +
1:size(final_image,1),size(final_image,2) - size(scale_bar,2) +
1:size(final_image,2),:) = scale_bar;
figure;
```

```
imshow(final_image);
title(picTitle, 'FontSize', 15);
```

matchPatterns.m

```
function newPatternOrigin = matchPatterns(fn_pattern, orig_pattern)
tform = affine2d([-1 0 0; 0 -1 0; 0 0 1]);
fn_pattern1 = imwarp(fn_pattern,tform);
newPatternOrigin = [1 1];

overlay_patterns(orig_pattern, fn_pattern1 > 0.15, newPatternOrigin);

disp('red: first pattern, green: current pattern. If patterns do not match,
choose shift vector (first point - new pattern, second - old)');
[inputX, inputY] = getpts;
while numel(inputX) == 2
    close;
    newPatternOrigin(1) = newPatternOrigin(1) + (inputX(2) - inputX(1));
    newPatternOrigin(2) = newPatternOrigin(2) + (inputY(2) - inputY(1));
    newPatternOrigin = overlay_patterns(orig_pattern, fn_pattern1 > 0.15,
newPatternOrigin);
    [inputX, inputY] = getpts;
end
close;
```

Appendix 5. MATLAB Code for Creating 20x20 Heat Maps

Script for generation of the orientation data to be used to create the 20x20 heat maps

```
if(~exist('rootfolder', 'var'))
    rootfolder = '/Volumes/Macintosh HD 2/Drop-boxes/Dropbox (RBG)/Lab
stuff';
end

heatMapWidth = 40; %in microns
heatMapHeight = 40; %in microns

SampleInfo = inputdlg({'Number of samples:','Excel file title:','actin
layer:','alpha-actinin layer:', 'fibronectin
layer:'},'Input',1,{'1','heat_map_data','2','3','4'});
SampleCount = str2num(SampleInfo{1});
%xlsname = ['\DATA\' SampleInfo{2}];

actinLayerNumber = str2num(SampleInfo{3});
actininLayerNumber = str2num(SampleInfo{4});
fibronectinLayerNumber = str2num(SampleInfo{5});

% get files for each sample
for i=1:SampleCount
    message = sprintf('Select files for sample %i',i);

[file,path]=uigetfile({'*.*';'*.lsm';'*.TIF';'*.tif';'*.bmp';'*.jpg'},message
,rootfolder,'Multiselect','on');
    if(exist(path, 'dir'))
        rootfolder = path;
    end
    FileList{i,1}=path;
    rootfolder = path;
    FileList{i,2}= file;
end

% Loop for each sample
for i=1:SampleCount

    disp('Sample 1')
    sample = FileList{i,2};
    if ~isa(sample,'char')
        PictureCount = length(sample);
    else
        PictureCount = 1;
    end

    % Setup
    All_angles = [];
    fbs_All_angles = [];
    average_PercentActinArea = 0;
    average_fbs_PercentActinArea = 0;

    cm_orient_block_data = cell(heatMapWidth/1);
    fb_orient_block_data = cell(heatMapWidth/1);

    imageData = cell(PictureCount,6);

    % Loop over each picture for sample #i
    for j = 1:PictureCount
        disp(['Processing image ', int2str(j), ' out of ',
int2str(PictureCount) '...']);

        discardImage = 0;

        % Open images and load actin channel in a matrix
        if PictureCount == 1
            picture = sample;
            picturename = [FileList{i,1} sample];
        else
            picture = sample{j};
            picturename = [FileList{i,1} sample{j}];
        end

        img = bfopen(picturename);

        actin = img{1,1}{actinLayerNumber,1};
        sarcomeres = img{1,1}{actininLayerNumber,1};
        fibronectin = img{1,1}{fibronectinLayerNumber,1};

        info = img{1,4};  % Load OME metadata
        PixelSize = double(info.getPixelsPhysicalSizeX(0).value());

        fprintf('Image resolution: %5.5f px/µm\n\n', 1/PixelSize);

        % Define blksze (should be 3/pixelsize based on recommendation)
        blksze = floor(3/PixelSize);

        % take into account only actin that is near sarcomeres

        if ~isa(sarcomeres, 'double'),
            sarcomeres = double(sarcomeres);
            actin = double(actin);
            fibronectin = double(fibronectin);
        end

        [imSizeX,imSizeY] = size(sarcomeres);

        fibronectin = fibronectin/max(fibronectin(:));
        fibronectin = medfilt2(fibronectin,[3 3]);
        fibronectin2 = fibronectin > 0.22;
        fibronectin2 = bwmorph(fibronectin2, 'close');
        fibronectin2 = bwmorph(fibronectin2, 'open');
        %fibronectin = fibronectin2 .* fibronectin;
        %fibronectin = fibronectin2;

        if j == 1
            %%
            imshow(fibronectin);
            disp('choose 2 pattern corners along the same side of a line' );
            [movingPointsX, movingPointsY] = getpts;
            close;
            % use only last 2 points
            movingPoints = [movingPointsX(1:2), movingPointsY(1:2)];
            patternOrigin = movingPoints(1,:);
            distance = sqrt((movingPointsX(1) - movingPointsX(2))^2 +
(movingPointsY(1) - movingPointsY(2))^2);

            fixedPoints = [movingPoints(1,1), movingPoints(1,2);
                movingPoints(1,1), movingPoints(1,2)+distance];

            transform =
fitgeotrans(movingPoints,fixedPoints,'nonreflectivesimilarity');
            [outboundsX, outboundsY] = outputLimits(transform,[1 imSizeX],[1
imSizeY]);
            patternOrigin(1) = patternOrigin(1) - (outboundsX(1) - 1);
            patternOrigin(2) = patternOrigin(2) - (outboundsY(1) - 1);
            newPatternOrigin = patternOrigin;
            reference_fn_image = imwarp(fibronectin, transform);
            figure
            imshow(reference_fn_image);

            fn_pattern = reference_fn_image(round(patternOrigin(2)) :
round(patternOrigin(2)) + round(heatMapHeight/PixelSize) - 1,
round(patternOrigin(1)) : round(patternOrigin(1)) +
round(heatMapWidth/PixelSize) - 1);

        else
            while 1

                imshow(fibronectin);
                disp('choose 2 pattern corners along the same side of a line'
);
                [movingPointsX, movingPointsY] = getpts;
                close;

                pointsEntered = numel(movingPointsX);

                if pointsEntered == 2

                    movingPoints = [movingPointsX, movingPointsY];
                    newPatternOrigin = movingPoints(1,:);

                    distance = sqrt((movingPoints(1,1) - movingPoints(2,1))^2
+ (movingPoints(1,2) - movingPoints(2,2))^2);
                    fixedPoints = [movingPoints(1,1), movingPoints(1,2);
                    movingPoints(1,1), movingPoints(1,2)+distance];

                    transform =
fitgeotrans(movingPoints,fixedPoints,'nonreflectivesimilarity');
                else
                    if numel(movingPointsX) == 1
                        movingPoints = [movingPointsX, movingPointsY];
                        newPatternOrigin = movingPoints(1,:);
                        % transform stays the same, so empty line here
                    else
                        discardImage = 1;
                    end
                end

                newPatternOrigin = transformPointsForward(transform,
newPatternOrigin);

                [outboundsX, outboundsY] = outputLimits(transform,[1
imSizeX],[1 imSizeY]);
                newPatternOrigin(1) = newPatternOrigin(1) - (outboundsX(1) -
1);
                newPatternOrigin(2) = newPatternOrigin(2) - (outboundsY(1) -
1);

                new_fn_image = imwarp(fibronectin, transform);
                overlay_patterns(fn_pattern > 0.15, new_fn_image > 0.15,
newPatternOrigin);

                disp('red: first pattern, green: current pattern. If patterns
do not match, choose shift vector (first point - new pattern, second -
old)');
                [inputX, inputY] = getpts;
                while numel(inputX) == 2
                    close;
                    newPatternOrigin(1) = newPatternOrigin(1) + (inputX(2) -
inputX(1));
                    newPatternOrigin(2) = newPatternOrigin(2) + (inputY(2) -
inputY(1));
                    overlay_patterns(fn_pattern > 0.15, new_fn_image > 0.15,
newPatternOrigin);
                    [inputX, inputY] = getpts;
                end
                if isempty(input('Type any key to fix the pattern manually or
Enter to continue'))
                    break;
                end
            end


            clearvars ptsCurrent featuresCurrent validPtsCurrent indexPairs
matchedReference matchedCurrent
        end

        if discardImage == 0
            imageData{j,1} = actin;
            imageData{j,2} = sarcomeres;
            imageData{j,3} = fibronectin;
            imageData{j,4} = PixelSize;
            imageData{j,5} = transform;
            imageData{j,6} = newPatternOrigin;
        end

    end

    %%
    for j = 1: size(imageData,1)
        if ~isempty(imageData{j,1})

            disp(['Analyzing alignment of image # ', int2str(j), ' out of ',
int2str(size(imageData,1)) '...']);

            actin = imageData{j,1};
            sarcomeres = imageData{j,2};
            fibronectin = imageData{j,3};
            PixelSize = imageData{j,4};
            transform = imageData{j,5};
            newPatternOrigin = imageData{j,6};

            blksze = floor(3/PixelSize);
            [imSizeX,imSizeY] = size(sarcomeres);

            sarcomereMask = zeros(imSizeX, imSizeY);
            actinMask = zeros(imSizeX, imSizeY);

            threshold = (0.22*max(sarcomeres(:))+0.78*min(sarcomeres(:)));
            actinThreshold = (0.05*max(actin(:))+0.95*min(actin(:)));
            d = floor(1.5*blksze);

            %disp('making sarcomere mask');

            circle = zeros(2*d+1);
            for x = -d:d
                for y = -d:d
                    circle(x+d+1,y+d+1) = x*x+y*y < d*d;
                end
            end

            for x = d+1:imSizeX-d
                for y = d+1:imSizeY-d
                    if sarcomeres(x,y) > threshold
                        sarcomereMask(x-d:x+d, y-d:y+d) = sarcomereMask(x-
d:x+d, y-d:y+d) | circle;
                    end
                    if actin(x,y) > actinThreshold
                        actinMask(x-d:x+d, y-d:y+d) = actinMask(x-d:x+d, y-
d:y+d) | circle;
                    end
                end
            end

            % erode 60%, because cells' area is higher than sarcomere
coverage
            sarcomereMask = bwmorph(sarcomereMask,'erode',floor(d*0.6));
            actinMask = bwmorph(actinMask,'erode',floor(d*0.6));

            if ~exist('thresh','var')
                thresh = actinDetectTest(actin,blksze,PixelSize);
            end

            %%
            actin = imwarp(actin, transform);
            sarcomeres = imwarp(sarcomeres, transform);
            sarcomereMask = imwarp(sarcomereMask, transform);
            actinMask = imwarp(actinMask, transform);

            %fibronectin = imwarp(fibronectin, transform);
            %fn_pattern = fibronectin(round(newPatternOrigin(2)) :
round(newPatternOrigin(2)) + round(heatMapHeight/PixelSize) - 1,
round(newPatternOrigin(1)) : round(newPatternOrigin(1)) +
round(heatMapWidth/PixelSize) - 1);
            %% splitting orientations between blocks
            block_num = 0;
            block_y = 0;

            cm_orientim =
create_orientation_image(actin,blksze,thresh,sarcomereMask);
            %fb_orientim =
create_orientation_image(actin,blksze,thresh,~sarcomereMask);

            disp('starting to sort data');
            for col = 1 : size(cm_orientim, 2)
                    block_num = floor((col -
newPatternOrigin(1))/(1/PixelSize));

                    block_num = rem(block_num, heatMapWidth/1);

                    if block_num <= 0
                        block_num = block_num + heatMapWidth/1;
                    end

                    if nnz(cm_orientim(:,col)) > 0
                        cm_orient_block_data{block_num} =
[cm_orient_block_data{block_num}; cm_orientim(cm_orientim(:,col) ~= 0 ,col)];
                    end

            end
            disp('data sorted');

            % Total number of actin positive pixels in the skeleton image
            % Sarcomere density = total/(image area)
            PercentActinArea = nnz(sarcomereMask)/((imSizeX-20)*(imSizeY-
20));
            %PercentFibroblastActinArea =
nnz(actinMask.*(~sarcomereMask))/((imSizeX-20)*(imSizeY-20));

            average_PercentActinArea = average_PercentActinArea +
PercentActinArea;
            %average_fbs_PercentActinArea = average_fbs_PercentActinArea +
PercentFibroblastActinArea;
        else
            disp(['Image # ', int2str(j), ' is discarded!']);
        end
        %}
    end

    cm_alignment_data = zeros(heatMapWidth/1, 6);
    all_cm_angles = [];
    %fb_alignment_data = zeros(heatMapWidth/1, heatMapHeight/1, 6);
    for block_num = 1 : heatMapWidth/1
            angles = cm_orient_block_data{block_num};
            all_cm_angles = [all_cm_angles; angles];

            if numel(angles) > 5

                % Convert radians to degrees
                angles_deg = rad2deg(angles);

                Mean = mean(angles_deg);
                Std = std(angles_deg);
                Median = median(angles_deg);

                % Create histogram
                [n,xout] = hist(angles_deg,180);
                dx = xout(2)-xout(1);                   % calc a single bin
width
                n = n / sum( n*dx );                    % normalize histogram
to have area of 1

                % Find mode
                [~,I] = max(n);
                Mode = xout(I);

                OrientationOrderParameter = OOP(angles);
                cm_alignment_data(block_num, :) = [numel(angles), Mean, Std,
Median, Mode, OrientationOrderParameter];
            else
                cm_alignment_data(block_num, :) = [0, 0, 0, 0, 0, 0];
            end
    end

    average_values = [0; 0; 0; 0; 0; 0]; % average for the pattern location

    for k = 1 : 6
        average_values(k) =
sum(cm_alignment_data(:,k))/nnz(cm_alignment_data(:,k));
    end

    all_cm_angles_deg = rad2deg(all_cm_angles);
    Mean = mean(all_cm_angles_deg);
    Std = std(all_cm_angles_deg);
    Median = median(all_cm_angles_deg);
    % Create histogram
    [n,xout] = hist(all_cm_angles_deg,180);
    dx = xout(2)-xout(1);                   % calc a single bin width
    n = n / sum( n*dx );                    % normalize histogram to have
area of 1
    % Find mode
    [~,I] = max(n);
    Mode = xout(I);
    OrientationOrderParameter = OOP(all_cm_angles);

    overall_values = [average_values(1), Mean, Std, Median, Mode,
OrientationOrderParameter];
```

```
    %overall_values = average_values;

    average_PercentActinArea = average_PercentActinArea/length(imageData);
    %average_fbs_PercentActinArea =
average_fbs_PercentActinArea/PictureCount;

    %Save data
    save(strcat(path, ['Sample_', num2str(i), '_results.mat']), 'fn_pattern',
'cm_alignment_data', 'fb_alignment_data');
end

fclose('all');
clearvars thresh;
```

Script for simultaneously generating low and high density heat maps using orientation data

```
% to use this script, you need to load alignment maps for both low and high
% density cases named cm_alignment_data_low and cm_alignment_data_high. The
% same for the fn_pattern_low, fn_pattern_high
if(~exist('rootfolder', 'var'))
    rootfolder = '/Volumes/Macintosh HD 2/Drop-boxes/Dropbox (RBG)/Lab
stuff';
end

[file1, path1] = uigetfile({'*.mat', '*.mat select low density data'},
'select low density data', rootfolder);
load([path1 file1], 'fn_pattern','cm_alignment_data');
fn_pattern_low = fn_pattern;
cm_alignment_data_low = cm_alignment_data;

PixelSize = 40/size(fn_pattern_low, 2);
bin_size = 1/PixelSize;

[file2, path2] = uigetfile({'*.mat', '*.mat select high density data'},
'select high density data', path1);
load([path2 file2], 'fn_pattern','cm_alignment_data');
fn_pattern_high = fn_pattern;
cm_alignment_data_high = cm_alignment_data;

cm_alignment_data_low(:,1) =
cm_alignment_data_low(:,1)/max(cm_alignment_data_low(:,1))*100;
cm_alignment_data_high(:,1) =
cm_alignment_data_high(:,1)/max(cm_alignment_data_high(:,1))*100;

average_values_low = [0; 0; 0; 0; 0; 0]; % average per location
average_values_high = [0; 0; 0; 0; 0; 0]; % average per location

for k = 1 : 6
    nnz_values_low = cm_alignment_data_low(:,k);
    nnz_values_low = nnz_values_low(nnz_values_low ~= 0);
    nnz_values_high = cm_alignment_data_high(:,k);
    nnz_values_high = nnz_values_high(nnz_values_high ~= 0);
    average_values_low(k) = sum(nnz_values_low(:))/numel(nnz_values_low);
    average_values_high(k) = sum(nnz_values_high(:))/numel(nnz_values_high);
end
```

```
data_label = {'Normalized Cell Occurrence';
    'Mean Orientation Angle';
    'Standard Deviation of the Mean Angle';
    'Median Orientation Angle';
    'Most Probable Orienation Angle';
    'OOP'};
x_axis_label = { 'Normalized Cell Occurrence';
    'Mean Orientation Angle, Degrees'
    'Std. Dev. of Mean Angle, Degrees'
    'Median Orientation Angle, Degrees'
    'Most Probable Orienation Angle, Degrees';
    'OOP'};

[file, path] = uigetfile({'*.*', '*.* select original pattern image'},
'select low density data', rootfolder);
orig_pattern = imread([path file]);
orig_pattern = rgb2gray(orig_pattern);
orig_pattern = double(orig_pattern);
orig_pattern = orig_pattern/max(orig_pattern(:));

correlation = [0 0 0 0 0 0];

final_images = cell(2,6);

for mapNum = 1 : 6

    map_low = controlCreateHeatMap2(cm_alignment_data_low(:,mapNum), 11,
size(orig_pattern, 1), [80, 155]);
    map_high = controlCreateHeatMap2(cm_alignment_data_high(:,mapNum), 10,
size(orig_pattern, 1), [80, 155]);

    correlation(mapNum) = corr2(cm_alignment_data_low(:,mapNum),
cm_alignment_data_high(:,mapNum));

    max_value = round(max([map_low(:); map_high(:)]),3, 'significant');
    min_value = round(min([map_low(:); map_high(:)]),3, 'significant');

    max_value = round(max_value,3, 'decimals');
    min_value = round(min_value,3, 'decimals');

    avg_value_low = round(average_values_low(mapNum),3, 'significant');
    avg_value_high = round(average_values_high(mapNum),3, 'significant');

    avg_value_low = round(avg_value_low,3, 'decimals');
    avg_value_high = round(avg_value_high,3, 'decimals');

    %% under- and oversaturate maps for higher contrast (optional)
    if(mapNum == 2 || mapNum == 3 || mapNum == 4 || mapNum == 5)
        cut_out_fraction = 0.2; % fraction of data that you want to make
outside the color range (over or undersaturated) to increase contrast of
what is near the average
        avg_value = (avg_value_low + avg_value_high)/2;

        sorted_data_low = sort(map_low(:));
        sorted_data_high = sort(map_high(:));
        data_size = length(sorted_data_low);

        min_value =
min(sorted_data_low(round(data_size*cut_out_fraction/2)),
sorted_data_high(round(data_size*cut_out_fraction/2)));
        max_value = max(sorted_data_low(round(data_size*(1-
cut_out_fraction/2))), sorted_data_high(round(data_size*(1-
cut_out_fraction/2))));

        max_value = round(max_value,3, 'significant');
        min_value = round(min_value,3, 'significant');

        max_value = round(max_value,3, 'decimals');
        min_value = round(min_value,3, 'decimals');

        %half_range = min(avg_value - min_value, max_value - avg_value)*(1 -
cut_out_fraction);
        %min_thresh = avg_value - half_range;
        %max_thresh = avg_value + half_range;
        map_low = (map_low - min_value).*(map_low > min_value) + min_value;
%undersaturate data
        map_low = (map_low - max_value).*(map_low < max_value) + max_value;
%oversaturate data

        map_high = (map_high - min_value).*(map_high > min_value) +
min_value; %undersaturate data
        map_high = (map_high - max_value).*(map_high < max_value) +
max_value; %oversaturate data
    end
    %% show final images
    final_images{1,mapNum} = control_draw_heat_map(map_low, orig_pattern,
min_value, max_value, avg_value_low, 40/size(orig_pattern, 1), 60,
strcat(data_label(mapNum), ', Low Cell Density'));
    final_images{2,mapNum} = control_draw_heat_map(map_high, orig_pattern,
min_value, max_value, avg_value_high, 40/size(orig_pattern, 1), 60,
strcat(data_label(mapNum), ', High Cell Density'));

    %% Show correlations:
%    disp(strcat(data_label(mapNum),' correlation:_',
num2str(correlation(mapNum))));

    %% Draw histograms
    bin_width = (max_value - min_value)/50;

    current_plot = figure('units','normalized','position',[0.1 0.1 0.43
0.6])
    hist_low = histogram(cm_alignment_data_low(:,mapNum), 10, 'BinWidth',
bin_width);
%    title(strcat(data_label(mapNum), ' Distribution'), 'FontSize', 30);
    xlim([min_value max_value]);
    xlabel(x_axis_label(mapNum), 'FontSize', 30, 'FontName', 'Times New
Roman', 'FontWeight', 'bold');
    ylabel('Frequency', 'FontSize', 30, 'FontName', 'Times New Roman',
'FontWeight', 'bold');
    set(gca,'FontSize',25, 'FontName', 'Times New Roman');
    hold on
    hist_high = histogram(cm_alignment_data_high(:,mapNum), 10, 'BinWidth',
bin_width);

    hist_low.EdgeColor = 'none';
    hist_high.EdgeColor = 'none';
    legend('\fontsize{30}\fontname{Times New Roman} low cell density',
'\fontsize{30}\fontname{Times New Roman} high cell density', 'Location',
'southoutside');

    if(~exist('save_dir', 'var'))
        save_dir = uigetdir(rootfolder, 'select low density data');
    end
    saveas(current_plot, strcat('/Volumes/Macintosh HD 2/Drop-boxes/Dropbox
(RBG)/Lab stuff/2015/2015-05-07 Matlab heat map stuff/vector graphics
histograms/', data_label{mapNum}, '.pdf'));

end
%% save images as png files
folder_name = uigetdir
if(folder_name ~= 0)
    for i = 1 : 6
        imwrite(final_images{1,i}, strcat(folder_name, '/', data_label{i},
'_low_density.png'));
        imwrite(final_images{2,i}, strcat(folder_name, '/', data_label{i},
'_high_density.png'));
    end
end
```

control_draw_heat_map.m

```
function final_image =  control_draw_heat_map(map_data, orig_pattern,
min_value, max_value, avg_value, PixelSize, font_size, picTitle)

low_hue = 240/360;
high_hue = 0;

dim1 = size(map_data, 1);
dim2 = size(map_data, 2);
palitra_length = round(dim1*5/7);
palitra_width = round(dim2/5.5);
%font_size = round(min(dim1, dim2)/30);

hsv = zeros(dim1, dim2, 3);
temp_hue = low_hue + (high_hue - low_hue)*(map_data - min_value)/(max_value
- min_value);
temp_hue = temp_hue - min(low_hue, high_hue);
temp_hue = temp_hue.*(temp_hue > 0) + min(low_hue, high_hue);

temp_hue = temp_hue - max(low_hue, high_hue);
temp_hue = temp_hue.*(temp_hue < 0) + max(low_hue, high_hue);

fn_pattern_fill = repmat(orig_pattern, ceil(dim1/size(orig_pattern, 1)),
ceil(dim2/size(orig_pattern, 2)));
```

```
fn_pattern_fill = fn_pattern_fill(1:dim1, 1:dim2);
hsv(:,:,1) = temp_hue;
hsv(:,:,2) = 1;
hsv(:,:,3) = fn_pattern_fill*0.8 + 0.1;
%     figure;
%     imshow(OOP_map);
%     figure
%     imshow(hsv2rgb(hsv));

scale_hsv = zeros(palitra_length, palitra_width,3);
scale_hsv(:,:,3) = 1;
scale_hsv(:,:,2) = 0;
scale_hsv(:,round(palitra_width/24):round(palitra_width/3),2) = 1;

for row = 1 : palitra_length
    scale_hsv(row,round(palitra_width/24):round(palitra_width/3),1) =
high_hue - (high_hue - low_hue)*(row/palitra_length);
end

min_y = 1;
max_y = palitra_length;
avg_y = round(1 + (palitra_length - 1)*(max_value - avg_value)/(max_value -
min_value));

min_text_y = 1;
max_text_y = max_y - font_size;
avg_text_y = avg_y - round(font_size/2);

shift = 0;
min_dist = round(font_size*1.5 + 2);
if(abs(avg_y - min_y) < min_dist)
    shift = min_dist - (avg_y - min_y); % this shift is more than 0
else if(abs(avg_y - max_y) < min_dist)
    shift = (max_y - avg_y) - min_dist; % this shift is less than 0
    end
end

avg_text_y = avg_text_y + shift;

scale_hsv(max_y-5:max_y,round(palitra_width/24):round(palitra_width/2.79),3)
= 0;
scale_hsv(min_y:min_y+5,round(palitra_width/24):round(palitra_width/2.79),3)
= 0;
scale_hsv(avg_y-
2:avg_y+2,round(palitra_width/24):round(palitra_width/2.79),1) = 0;
scale_hsv(avg_y-
2:avg_y+2,round(palitra_width/24):round(palitra_width/2.79),2) = 1;

textInserter = vision.TextInserter('%s', 'LocationSource', 'Input port',
'Color',  [0, 0, 0], 'FontSize', font_size);
strings = uint8([num2str(max_value) 0 num2str(min_value) 0
num2str(avg_value)]);
labeled = step(textInserter, hsv2rgb(scale_hsv), strings,
int32([round(palitra_width/2.67), min_text_y; round(palitra_width/2.67),
max_text_y; round(palitra_width/2.67), avg_text_y]));
```

```
scale_bar_microns = floor((palitra_width - 4)*PixelSize/10)*10;
scale_bar_px = scale_bar_microns/PixelSize;
left_margin = round((palitra_width - scale_bar_px)/2);

scale_bar = ones(min(palitra_width, round(dim1*2/7)),palitra_width,3);
scale_bar(round(size(scale_bar, 1)*0.8):size(scale_bar, 1),
left_margin:round(left_margin + scale_bar_px), :) = 0;
string = uint8([num2str(scale_bar_microns) ' um']);
textInserter2 = vision.TextInserter('%s', 'LocationSource', 'Input port',
'Color',  [0, 0, 0], 'FontSize', round(font_size));
scale_bar = step(textInserter2, scale_bar, string,
int32([round(left_margin*1.1) round(size(scale_bar, 1)*0.8 -
font_size*1.15)]));
%     figure;
%     imshow(scale_bar);

final_image = ones(dim1, dim2 + size(labeled,2),3);
final_image(:,1:dim2,:) = hsv2rgb(hsv);
final_image(1:size(labeled,1),dim2 + 1:dim2 + size(labeled,2),:) = labeled;
final_image(size(final_image,1) - size(scale_bar,1) +
1:size(final_image,1),size(final_image,2) - size(scale_bar,2) +
1:size(final_image,2),:) = scale_bar;
figure;
imshow(final_image);
title(picTitle, 'FontSize', 15);
```

controlCreateHeatMap2.m

```
function heat_map = controlCreateHeatMap2(cm_alignment_data, shift, mapSize,
mapSizeMicrons)

bin_size = mapSize/40;

micronSizePx = bin_size;
widthMicron = mapSizeMicrons(2);
heightMicron = mapSizeMicrons(1);
widthPx = widthMicron*micronSizePx;
heightPx = heightMicron*micronSizePx;

heat_map = zeros(heightPx, widthPx);

for block = 1 : widthMicron

    transformed_block = block - shift;
    transformed_block = mod(transformed_block, 40);

    if(transformed_block == 0)
        transformed_block = 40;
    end

    col1 = max(1,round((block - 1)*bin_size));
    col2 = min(widthPx,round(block*bin_size));

    heat_map(:, col1 : col2) = cm_alignment_data(transformed_block);
```

```

end
end
```

Appendix 5. MATLAB Script for Measuring Cardiomyocyte Contraction Propagation Speed Based on Confocal Line Scan Videos of Intracellular Calcium Dynamic

Main script

```
if(~exist('rootfolder', 'var'))
    %rootfolder = 'F:\Images\Ivan';
    rootfolder = 'E:\Dropbox (RBG)\Dropbox (RBG)\Lab stuff';
end

[file,path]=uigetfile('*.*', 'open a video file',rootfolder);


if(exist(path, 'dir'))
    rootfolder = path;
    img = bfopen([path file]);

    imageCount = size(img, 1);

    slicesToAnalyze = input('enter slices to analyze: ');

    for sliceNum = slicesToAnalyze
        disp(['analyzing slice ' sliceNum]);

        channelNum = 1;
        stimulationFrequency = 1; % Hz

        calciumImage = double(img{sliceNum,1}{channelNum,1});

        OME = img{1,4};  % Load OME metadata

        xPixelSize = double(OME.getPixelsPhysicalSizeX(sliceNum -
1).value());
        tPixelSize = double(OME.getPixelsPhysicalSizeY(sliceNum -
1).value());

        xUnits = OME.getPixelsPhysicalSizeX(sliceNum -
1).unit().getSymbol();
        tUnits = OME.getPixelsPhysicalSizeY(sliceNum -
1).unit().getSymbol();

        sizeXpixels = size(calciumImage, 2);
        sizeTpixels = size(calciumImage, 1);

        sizeX = sizeXpixels*xPixelSize; % in microns
        sizeT = sizeTpixels*tPixelSize; % in seconds

        v0 = 2000; % microns/second = 0.1 * v(cm/sec)
        v0pix = v0/xPixelSize*tPixelSize; % pix/pix

        scaleFactor = 10;
        loop_done = false;
        avTime = 200;
        avLength = 50;

        while ~loop_done

            %avLength = round(avTime*v0pix/scaleFactor);
            gaussT = fspecial('gaussian', [avTime 1], avTime/6 ); % vertical
filter, three sigma crop to each side
            gaussX = fspecial('gaussian', [1 avLength], avLength/6); %
horizontal
            window = 1/avLength/avTime*ones(avLength, avTime);
            smoothedImage = imfilter(calciumImage, gaussT*gaussX,
'replicate');
            calciumData = smoothedImage.';
            derivativeData = diff(calciumData, 1, 2);

            %% Plot example of the signal and averaging
%             figure
%             plot(calciumImage(1:1500, 255), 'k.');
%             hold on
%             plot(smoothedImage(1:1500, 255), 'r-', 'LineWidth', 1);
%
%             set(gca,'fontsize',16);
%             set(gca,'fontname','Arial');
%
%             xlabel('Frame number', 'FontSize', 20, 'FontName', 'Arial',
'FontWeight', 'bold');
%             ylabel('Brightness', 'FontSize', 20, 'FontName', 'Arial',
'FontWeight', 'bold');
%             hold off

            plot(derivativeData(255,:)/max(derivativeData(1,:)));

%%
            if(input('are you satisfied? 1 = yes, any key = no: ') ~= 1)
               avLength = input(['enter new length average (now '
num2str(avLength) '): ']);
               avTime = input(['enter new time average (now '
num2str(avTime) '): ']);
            else
                loop_done = true;
            end
        end

        timePoints = linspace(0, sizeT, sizeTpixels);
        derivativeTimePoints = timePoints(1 : end - 1) + 0.5*tPixelSize;

        xPoints = [];
        peakLocations = [];

        minPeakProminence = 0.5
        numberOfPeaks = -1;
        %
        figure
        for pointNum = 1 : sizeXpixels
            findpeaks(derivativeData(pointNum,
:)/max(derivativeData(pointNum, :)), 'MinPeakDistance',
0.8/stimulationFrequency/tPixelSize, 'MinPeakProminence', minPeakProminence)
% 0.3-0.6 seem to work fine

            if(numberOfPeaks < 1)
                while(input('change min peak prominence? 1 = yes, any key =
no: ') == 1)
                    minPeakProminence = input('enter min peak prominence:
');
                    findpeaks(derivativeData(pointNum,
:)/max(derivativeData(pointNum, :)), 'MinPeakDistance',
0.8/stimulationFrequency/tPixelSize, 'MinPeakProminence', minPeakProminence)
% 0.3-0.6 seem to work fine
                end

                [~, peaks] = findpeaks(derivativeData(pointNum,
:)/max(derivativeData(pointNum, :)), 'MinPeakDistance',
0.8/stimulationFrequency/tPixelSize, 'MinPeakProminence',
minPeakProminence); % 0.3-0.6 seem to work fine

                if(~input('cancel frame (1 = yes)? '))
                    numberOfPeaks = size(peaks,2);
                end

            else
                [~, peaks] = findpeaks(derivativeData(pointNum,
:)/max(derivativeData(pointNum, :)), 'MinPeakDistance',
0.8/stimulationFrequency/tPixelSize, 'MinPeakProminence',
minPeakProminence); % 0.3-0.6 seem to work fine
            end

            if(numberOfPeaks ~= size(peaks,2))
                cancelFrame = 1;
            end

            if(~cancelFrame)
                peakLocations = [peakLocations;
derivativeTimePoints(peaks)];
                xPoints = [xPoints; pointNum];
            else
                cancelFrame = 0;
            end

        end

        peakDelays = repmat(peakLocations(1,:),size(peakLocations, 1),1); %
peak locations of the first pixel
        peakDelays = peakLocations - peakDelays; % this calculates the
delays

        xPoints = xPoints.'.*xPixelSize;
        %xPoints = linspace(0,sizeX, sizeXpixels);
        allXPoints = repmat(xPoints.', 1, numberOfPeaks);

        %shiftedDelays = peakDelays(:,10:end);
        avDelays = mean(peakDelays, 1);
        peakDelays = peakDelays - repmat(avDelays, size(peakDelays, 1), 1);
        regressionDataY = peakDelays(:);
        regressionDataX = allXPoints(:);

        %regressionDataY = regressionDataY2;
        %regressionDataX = regressionDataX2;

        modelFun = @(b,x) x/b(1)/1000+b(2);
        modelFun2 = @(b,x) b(1) + b(2)*x + b(3)*x.^2 + b(4)*x.^3 +
b(5)*x.^4;

        nlm = fitnlm(regressionDataX,regressionDataY,modelFun,[1 0]);
        coefCI(nlm, 0.05) % calculate confidence intervals
        %figure;
        %plot(regressionDataX, regressionDataY, '.');
        %line(xPoints,predict(nlm,xPoints.'),'linestyle','--','color','k')

        % predict the response of the system to the input arguments xPoints
        % 'Simultaneous' = true means that confidence intervals will be
        % calculated
        [predictedY, predictedCI] = predict(nlm, xPoints.', 'Simultaneous',
true);
        %figure;
        %plot(regressionDataX,regressionDataY,'k.', xPoints,predictedY,'b-',
xPoints,predictedCI,'b:');

        figure;
        plot(regressionDataY*1000,regressionDataX,'k.', predictedY*1000,
xPoints,'b-', predictedCI*1000, xPoints,'r-');

        %figure;
        %plot(regressionDataY*1000,regressionDataX,'k.');
        set(gca,'fontsize',16);
        set(gca,'fontname','Arial');
        xlabel('Time, ms', 'FontSize', 20, 'FontName', 'Arial',
'FontWeight', 'bold');
        ylabel('Distance, \mum', 'FontSize', 20, 'FontName', 'Arial',
'FontWeight', 'bold');
    end
end
```

Appendix 6. List of Publictions, Presentations, and Posters.

PUBLICATIONS

1. I Batalov, Q Jallerat, S Kim, AW Feinberg. Understanding the Role of Cell-Cell and Cell-Matrix Interactions in the Formation of 2D Myocardium Using Embryonic-Inspired Scaffolds. In preparation
2. I Batalov, S Kim, AW Feinberg. Recapitulating the Sub-micron Structure of the Embryonic Myocardium to Guide the Alignment of Cardiac Monolayers. In preparation.
3. I Batalov, AW Feinberg. Differentiation of Cardiomyocytes from Human Pluripotent Stem Cells Using Monolayer Culture. Biomarker Insights. 2015;10(Suppl 1):71-76. doi:10.4137/BMI.S20050.

PRESENTATIONS

1. I Batalov, S Kim, AW Feinberg. Biomimetic Micropatterns Based on the Embryonic Heart for 2D Cardiac Tissue Engineering (talk), SFB Annual Meeting, Charlotte, NC, USA, 2015.
2. I Batalov, Q Jallerat, S Kim, AW Feinberg. Biomimetic Micropatterns Based on the Embryonic Heart for 2D Cardiac Tissue Engineering. CMECS Meeting, Pittsburgh, PA, USA, 2015
3. I Batalov, Q Jallerat, S Kim, AW Feinberg. Using Biomimetic Fibronectin Micropatterns to Guide the Alignment of iPSC-derived Cardiac Monolayers. CMECS Meeting, Pittsburgh, PA, USA, 2016

POSTERS

1. I Batalov, S Kim, AW Feinberg. Using fibronectin architecture in the embryonic heart to engineer developmentally inspired human iPS-derived cardiac tissues. International Society for Stem Cell Research Conference, San Francisco, CA, USA, 2016.
2. I Batalov, Q Jallerat, S Kim, AW Feinberg. The Role of Cell-Cell and Cell-ECM Interactions in Cardiomyocyte Alignment on Biomimetic Micropatterned Surfaces. McGowan Institute for Regenerative Medicine Retreat, Farmington, PA, USA, 2016.
3. I Batalov, Q Jallerat, AW Feinberg. Using the Embryonic Heart as an Instructive Template for Cardiac Tissue Engineering. American Heart Association's BCVS, New Orleans, LA, USA, 2015.
4. I Batalov, S Kim, Q Jallerat, AW Feinberg. Engineering 2D Cardiac Tissues using Biomimetic Protein Micropatterns Based on the ECM in the Embryonic Heart. 4th TERMIS World Congress, Boston, MA, USA, 2015.
5. I Batalov, S. Kim, AW Feinberg. Biomimetic Micropatterns Based on the Embryonic Heart for 2D Cardiac Tissue Engineering. BMES Annual Meeting, Tampa, FL, 2015.
6. I Batalov, Q Jallerat, S. Kim, AW Feinberg. Biomimetic Protein Micropatterns Based on the Embryonic Heart for 2D Cardiac Tissue Engineering. Annual Materials Science & Engineering Symposium, Carnegie Mellon University, Pittsburgh, PA, USA, 2015.