2009

Application Of Alkylsilane Self-assembled Monolayers For Cellpatterning And Development Of Biolocial Microelectromechanicalsyste

Kerry Wilson
University of Central Florida

APPLICATION OF ALKYLSILANE SELF-ASSEMBLED MONOLAYERS FOR CELL PATTERNING AND DEVELOPMENT OF BIOLOCIAL MICROELECTROMECHANICAL SYSTEMS

by

KERRY A. WILSON
B.S Clemson University, 1998
M.S. Clemson University, 2002

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Chemistry
in the College of Sciences
at the University of Central Florida
Orlando, Florida

Spring Term
2009

Major Professor: James J. Hickman

ABSTRACT

Advances in microfabrication and surface chemistry techniques have provided a new paradigm for the creation of *in vitro* systems for studying problems in biology and medicine in ways that were previously not practical. The ability to create devices with micro- to nano-scale dimensions provides the opportunity to non-invasively interrogate and monitor biological cells and tissue in large arrays and in a high-throughput manner. These systems hold the potential to, in time, revolutionize the way problems in biology and medicine are studied in the form of point-of-care devices, lab-on-chip devices, and biological microelectromechanical systems (BioMEMS). With new *in vitro* models, it will be possible to reduce the overall cost of medical and biological research by performing high-throughput experiments while maintaining control over a wide variety of experimental variables. A critical aspect of developing these sorts of systems, however, is controlling the device/tissue interface. The surface chemistry of cell-biomaterial and protein-biomaterial interactions is critical for long-term efficacy and function of such devices.

The work presented here is focused on the application of surface and analytical chemistry techniques for better understanding the interface of biological elements with silica substrates and the development a novel Bio-MEMS device for studying muscle and neuromuscular biology. A novel surface patterning technique based on the use of a polyethylene glycol (PEG) silane self-assembled monolayer (SAM) as a cytophobic surface and the amine-terminated silane diethyeletriamine (DETA) as a cytophilic surface was developed for patterning a variety of cell types (e.g. skeletal muscle, and neural cells) over long periods of time (over 40 days) with high fidelity to the patterns. This method was then used to pattern embryonic rat skeletal muscle and

motor neurons onto microfabricated silicon cantilevers creating a novel biological microelectromechanical system (BioMEMS) for studying muscle and the neuromuscular junction. This device was then used to study the effect of exogenously applied substances such as growth factors and toxins. Furthermore, a whispering-gallery mode (WGM) biosensor was developed for measuring the adsorption of various proteins onto glass microspheres coated with selected silane SAMS commonly used in BioMEMS system. With this biosensor it was possible to measure the kinetics of protein adsorption onto alkylsilane SAMS, in a real-time and label-free manner.

I dedicate this dissertation to my family. Without their loving support I could never have achieved all that I have today.

ACKNOWLEDGMENTS

The author would like to acknowledge Drs. Kathy Wahl and Richard Colton of the United States Naval Research Laboratory for their assistance in the construction of the AFM detection system used during the development of the BioMEMS device as well as analysis of the data obtained. The author would also like to acknowledge Dr. Frank Vollmer for his gracious help in development of our whispering-gallery mode biosensor and flow cell. Further assistance was provided by Mr. Craig Finch and Dr. Frank Sommerhage in the development of data analysis software and Matlab programs (respectively) for analyzing the data generated from the WGM biosensor system. Finally author would like to acknowledge all the members of the Hybrid-Systems laboratory for not just their technical expertise, but also their friendship and emotional support during the course of this work. The work presented in this dissertation was funded through: DARPA ITO through SPAWAR F30602-02-2-0541 and NIH 5RO1NS050452-03. Finally the author would like to acknowledge the assistance provided by the MEMS-Exchange during the design and fabrication of our silicon microcantilevers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AFM	Atomic Force Microscopy
BSA	Bovine Serum Albumin
CA	Contact angle
DETA	(3-Trimethoxysilyl propyl) diethylenetriamine
DFB	Distributed feedback
DUV	Deep ultraviolet
FN	Fibronectin
GO	Glucose Oxidase
IPA	Isopropanol
PDMS	Polydimethylsiloxane
PEG	2-[Methoxypoly(ethyleneoxy)propyl]trimethoxysilane
VI	Virtual Instrument
WGM	Whispering Gallery Mode
XPS	X-ray Photoelectron Spectroscopy

CHAPTER ONE: INTRODUCTION

The boom in the semiconductor manufacturing industry of the past three decades has yielded a vast array of tools and methods for fabricating devices with micro- to nano-scale features. Concomitantly, advances in biotechnology, i.e. the sequencing of the human genome, and looming proteomic era have opened new avenues for the application of technologies from the semiconductor industry in the form of gene and protein arrays, lab-on-a-chip devices and biological micro-electromechanical systems (Bio-MEMS). To date, most applications of these technologies have focused on studying biomolecular interactions (DNA-DNA, DNA-RNA, protein-protein, etc.) [1-8], individual cells or pure cultures of a single cell type [9-34]. These methods, while suitable for answering specific questions about a limited number of factors within a biological system, do not necessarily reflect the overall complexity of the whole biological context in which the system of interest exists and can be poor predictors of the *in vivo* reality. Typically animal studies are required for understanding the complex interactions of cells and biomolecules *in vivo*. These types of studies, however, can yield results that are difficult to interpret due to the lack of sufficient control over all experimental variables. Also, animal studies can be ethically questionable causing unnecessary pain and suffering to the animals used. Ethical concerns regarding animal studies in and of themselves can prevent researchers from being able to perform studies that can yield significant findings. For example, in Europe legislation has been passed to limit the use of animal studies to medical research, and even so the use of animal studies for medical purposes has become more difficult to justify. For these reasons and more the development of *in vitro* models for drug discovery, disease research, and basic science has become increasingly fertile ground. New devices and techniques are being developed to study the complex interactions of multiple cell types in lab-on-a-chip systems that allow stringent

control of environmental variables, as well as real-time, non-invasive monitoring and interrogation[21, 30, 35-48].

One critical capability to have when developing *in vitro* model systems is the ability to control the spatial orientation of the different cell types with respect to each other. One common method for doing this modifying the surface of a substrate to present a 2-dimensional pattern consisting of cytophilic (cell adhesive) regions surrounded by cytophobic (cell repulsive) regions. Methods to achieve this include microcontact printing, microfluidic patterning, dip-pen nanolithography, and photolighography. Various chemistries can be used with these methods such as, physical adsorption of biopolymers, alkanethiol, and alkylsilane chemistry. Alkylsilanes are a broad class of compounds that can be covalently linked to silica substrates and hence a wide variety of solid state devices. This is a particularly advantageous type of chemistry to use as a wide variety of silanes area commercially available and provide stable covalently linked films that are suitable for use in aqueous environments. For these reasons alkylsilane self-assemble monolayers were chosen as model substrates for the development of a novel surface patterning technique as a broadly applicable method for the long-term culture of multiple cell types on silica substrates and the development of a novel BioMEMS device for studying muscle biology and neuromuscular junction formation.

The work presented here demonstrates the development of a photolithographic method for patterning PEG and amine terminated alkylsilane monolayers for patterning embryonic neurons and muscle cells in long-term cultures. This technique was then applied to patterning cells on a novel BioMEMS device based on microfabricated silicon cantilevers and an AFM detection system. Embryonic skeletal myocytes were plated onto patterned cantilevers and differentiated *in situ* into functional myotubes that were capable of bending

the cantilever. The contractile stress generated by the myotubes was quantified using a modified Stoney's equation. Furthermore, it was shown that this system could be used to quantify differences in muscle behavior and development due to exogenously applied factors. This system was further applied to the coculture of myotubes with embryonic motor neurons for creating a model system for studying neuromuscular junction formation and function. The adsorption of proteins onto alkylsilane SAMS was also studied. To this end a WGM biosensor and novel fluidic system was contsructed. Using this system the adsorption of proteins onto DETA, PEG, 13F and glass was quantified.

CHAPTER TWO: DIRECT PATTERNING OF COPLANAR ALKYLSILANE MONOLAYERS BY DEEP-ULTRAVIOLET PHOTOLITHOGRAPHY AS A GENERAL METHOD FOR LONG-TERM CELL PATTERNING AND CULTURE

Introduction

Surface modification and patterning have been staple techniques in bioengineering and cell biology for decades. This is due primarily to the fact that the interaction of a cell or biomolecule with a particular material is dictated by the properties of the first few nanometers of the surface of the bulk material[49]. Various methods have been developed to modify the surface characteristics of bulk materials to tune the biocompatibility of a material for promoting or preventing the attachment of cells and biomolecules[25]. Such methods include, but are not limited to, physical adsorption of polymers, microcontact printing, microfluidic patterning, and photolithographic pattering. Patterning cells onto various substrates has been used extensively to create cellular microarrays in which the spatial orientation of the cells with respect to one another can be tightly controlled, thereby creating a unique microenvironment for each cell. In this way individual cells can be studied in isolation, in networks, or in complex coculutures where the interaction of multiple cell types can be comparatively studied.

The basis of cell patterning lies in the interaction of membrane bound receptors, known as integrins, with adsorbed biomolecules on the underlying substrate[50-54]. Cells secrete adhesion proteins (fibronectin, laminin, collagen, etc.), which adsorb onto a material by physical interactions (electrostatic, van Der Waals, hydrophobic interactions, etc.). The nature and extent of the adsorption is determined by the chemical characteristics of the surface. Cytophobic surfaces, those that resist cell adhesion, tend either to adsorb very little protein or adsorb it in such a way that it looses its biological activity. Cytophilic surfaces, those that

promote cell adhesion and growth, tend to adsorb protein in such a way that its biological activity is retained allowing it to bind to membrane bound integrin receptors. Surfaces that tend to resist cell adhesion are non-polar and either strongly hydrophobic (such as hydrocarbon or perfluorinated surfaces) or strongly hydrophilic (such as PEG). Surfaces that tend to promote cell adhesion usually consist of polar, hydrophilic molecules. Aminated molecules, such as DETA and poly-lysine are considered to be strongly cytophilic and promote the growth and differentiation of cells.

Alkylsilane self-assembled monolayers are a class of compounds that are used extensively for modifying the surface properties of silica substrates (silicon, glass, etc.)[55]. These compounds come in a variety of chemistries that are used for a number of different applications and are often used to modulate the biocompatibility of silica substrates [55]. Work by Stenger and coworkers[56, 57] showed that it is possible to pattern the aminated alkylsilane DETA using deep ultraviolet (DUV) photolithography. By this method DETA SAMs were exposed to intense ultraviolet light from an ArF excimer laser (emission wavelength 193 nm). Upon exposure the DETA molecule underwent a photochemical reaction that cleaved the carbon-nitrogen bonds within the molecule, thereby removing the cytophilic surface coating. The ablated regions were then re-derivitized with the perfluorinated silane 13F to create a surface that resisted cell adhesion. This method was extended to produce high contrast micrometer scale patterns that were used to pattern cells into microarrays. Furthermore, it was shown that the geometric cues designed into the silane patterns induced specific polarity in neural cells and direct the outgrowth of axonal processes and the development of the dendritic field.

A novel cell patterning method has been developed using the same DUV photolithography method of Stenger and coworkers. First a polyethylene glycol (PEG)

monolayer was formed on a silica substrate. The PEG monolayer was then patterned using DUV photolithography creating ablated regions suitable for re-derivitization. The patterned PEG monolayers were then reacted with DETA to form cell adhesive islands. These patterned PEG-DETA substrates were then used for culturing embryonic myocytes, motor neurons, and hippocampal neurons. It has been shown that these surfaces can support attachment and growth of a variety of cell types for periods in excess of 40 days. This ability to create patterned arrays of cells in long-term cultures enhances the ability to create cellular microarrays and Bio-MEMS devices.

Materials and methods

PEG-silane preparation protocol

Substrate Cleaning

Silica substrates (glass and/or silicon wafers) were cleaned using serial acid baths. Substrates were arranged in a porcelain coverslip holder (Thomas Scientific, Swedesboro, NJ). The substrates were then immersed in a 1:1 (vol:vol) solution of methanol and concentrated HCl for at least 1 hour. This step removed surface contaminants. After 1 hour the substrates were rinsed 3x in diH_2O and transferred to a solution of concentrated sulfuric acid for at least 1 hour. This step oxidized the surface of the silica substrates leaving a hydrophilic surface suitable for reaction of the silane derivatives. After at least one hour in sulfuric acid the substrates were washed 3x in diH_2O. The rinsed substrates were then boiled in diH_2O for 30 minutes. After boiling the samples were place in a 120^oC oven for at least 3 hours. The resulting surfaces were analyzed using contact angle goniometry and XPS to verify hydrophilicity of the surfaces (CA < 5.0^o) and the elemental composition of the

surfaces respectively. Surfaces with a CA of less that 5.0^{o} and an elemental carbon content of approximately 5.0% were considered suitable for derivitization.

PEG-silane coating procedure

Silica substrates were coated with a PEG-terminated silane by a modified protocol from Papra et al [58]. Dry toluene was prepared by distillation over metallic sodium to remove any water or other contaminants. The toluene was refluxed for a minimum of 2 hours prior to collection. The first toluene fraction was discarded to ensure a minimum of contamination. Toluene was collected in clean Pyrex bottles that had been placed in a 120^{o}C oven for at least 2 hours to drive off any water adsorbed to the surface of the glass. The remaining air in the Pyrex bottle was replaced with dry ultra-pure nitrogen to ensure a minimum of gaseous oxygen in the reaction mixture. The toluene was then placed in the antechamber of an MBraun glove box (MBraun, Stratham, NH). The chamber was evacuated and refilled with dry nitrogen 3x before bringing the toluene into the main chamber. The alkylsilane 2-[Methoxypoly(ethyleneoxy)propyl]trimethoxysilane (Gelest, Tullytown, PA), as seen in Figure 1, was added to the toluene to a final concentration of 0.1% by volume. The PEG-toluene solution was then removed from the glove box and brought into a chemical fume hood. Concentrated HCl was added to a final volume of 0.08 % (0.8 ml HCl/L) and the solution briefly stirred. The cleaned silica substrates were removed from the oven and allowed to cool to room temperature before incubation in the PEG-toluene solution. Samples were incubated in the PEG-toluene solution for 1 hour at room temperature. The reaction vessel was loosely covered to prevent excessive exposure to atmosphere. After 1 hour the samples were removed and rinsed in serial washes of toluene (1x), ethanol (2x), and diH2O

(1x). The washed samples were blown dry under a stream of ultrapure nitrogen and were used immediately or stored in a desiccator until needed.

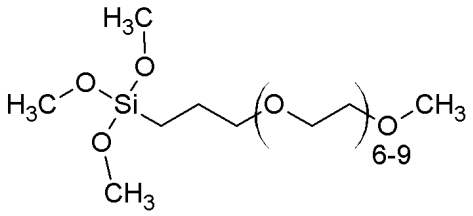

Figure 1: Chemical structure for 2-[Methoxypoly(ethyleneoxy)propyl]trimethoxysilane.

Deep-UV photolithography of PEG-silane monolayers

PEG-silane modified silica substrates were patterned using DUV photolithography. The samples were patterned in a photolithography system of our own design (Figure 2), which was based on a mask aligner, 193 nm ArF excimer laser (Lambda Physik, Santa Clara, CA) with an in-line beam homogenizer. A beam homogenizer was necessary for patterning of the monolayers. Light emitted from the laser had a parabolic intensity profile (Figure 2) that yielded uneven ablation. The beam homogenizer refocused the laser such that it yielded a top-hat profile with an even distribution of light intensity across the ablation area. Samples were placed on the stage of the mask aligner under a 5x5 inch chrome plated photomask, which contained the pattern to be ablated. The masks were written in dark-field polarity such that the areas corresponding to the ablated pattern were transparent and the remaining areas were coated with chrome. If necessary the substrate was precision aligned using the aligner stage to ensure micrometer precision placement of the pattern. The substrate was then brought into contact with the mask and a vacuum applied between the stage and mask to ensure a hard contact. A hard contact was used to minimize the gap between the substrate and mask to ensure a high contrast pattern with minimal edge effects due to refringence of the laser light. The substrates were then exposed to 193 nm ultraviolet laser light for 15 to 120 seconds with a pulse intensity of 200 mJ/pulse and a frequency of 10Hz. After ablation

the samples were removed from the aligner stage and stored for subsequent processing. When necessary, the orientation of the pattern on the substrate was marked by scoring or breaking the upper right-hand corner of the substrate. This enabled easy recognition of the orientation of the pattern on the substrate during subsequent processing and cell culture.

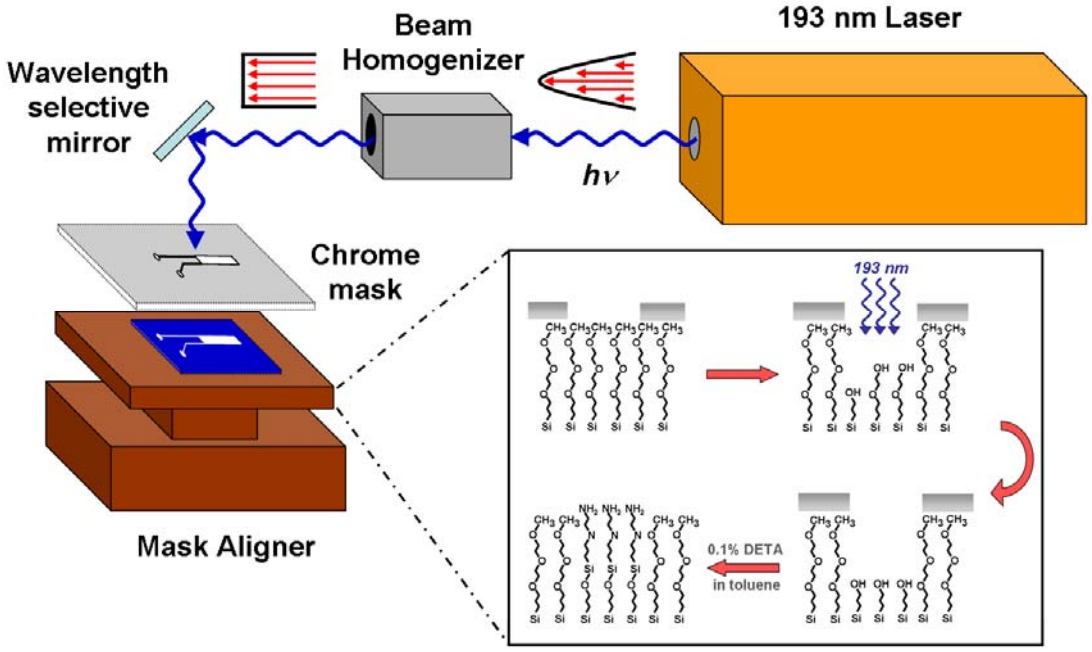

Figure 2: DUV photolithography system and proposed reaction scheme

Back-fill of patterned PEG-silane monolayers with DETA-silane

After ablation the patterned PEG-silane substrates were reacted with the alkylsilane (3-Trimethoxysilyl propyl) diethylenetriamine (DETA), seen in Figure 3. Fresh distilled toluene was transferred into a Pyrex bottle that had been dried in an 120^{o}C oven to dry off excess surface water. Dry nitrogen was used to replace the air in the remaining volume of the bottle to minimize free oxygen. The bottle was sealed and placed in the antechamber of an MBraun glovebox, which was evacuated and refilled with dry nitrogen 3 times. The toluene was transferred into the main chamber. DETA was added to the toluene to a final concentration of 0.1% (vol:vol). The DETA-toluene solution was removed from the glove box and

transferred to a pyrex beaker and the samples were immersed in the solution. To drive the reaction forward the solution was gently heated to no more than 65°C. Optimal reaction time was analyzed for these conditions by incubating the samples 10, 20, and 30 minutes. After reaction with DETA the samples were allowed to cool to room temperature, washed 3 time with dry tolune and heated too 65°C in fresh toluene for 30 more minutes, then dried under ultrapure nitrogen. The resulting samples were analyzed by XPS and contact angle goniometry.

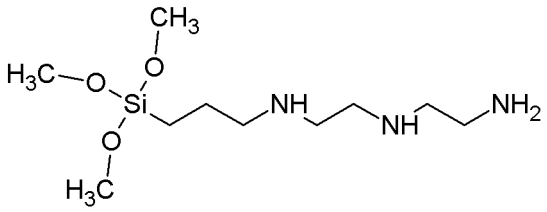

Figure 3: Chemical structure for (3-Trimethoxysilyl propyl) diethylenetriamine (DETA).

Characterization of unpatterned and patterned silane monolayers

Contact angle goniometry analysis

The surface contact angle of the modified substrates was measured by contact angle goniometry. The samples were place on the stage of a Ramé Hart (Netcong, NJ) contact angle goniometer. The sample stage was raised to close proximity with the outlet nozzle of the drop-dispensing pump. A 5 ul drop of reagent grade ultra-pure water was placed on the surface of the sample. A side-on image of the water drop on the surface was taken using an imaging video CCD camera and processed with DropImage Advance software (Ramé Hart, Netcong, NJ). Normal contact angle for PEG was considered to be 38±2°, and 49±2° for DETA.

X-ray photoelectron spectroscopy analysis

XPS was performed on unpatterned and patterned samples to verify the relative amounts of the elements comprising the different monolayers. All analysis were performed on a Fisons ESCALab 220-XL with a monochromatic single-anode K-α source. Samples were first introduced into the preplock chamber which was then closed and evacuated for at least 30 minutes to a vacuum of less than $5x10^{-3}$ Torr by a roughing pump. The samples were then transferred into the main chamber. After introduction the main chamber was allowed to settle to a pressure of less than $1x10^{-8}$ Torr. Elemental analysis was performed for identification of all relevant constituents of the silanes used. Elements were identified according to their characteristic binding energies. A survey scan from 10 to 1200 eV was first performed to identify the major elemental constituents of the samples using a step increment of 1 eV, a dwell time of 100 ms, and a pass energy of 50 eV. The strong line peaks were analyzed using high resolution scans for Si 2p (103.3eV) C 1s (284.5 eV), N 1s (402.5 eV), and O 1s (533.0 eV). High resolution scans were performed over a minimum spectral width of +/- 5 eV of the elemental center binding energy, a step increment of 0.1 eV, a dwell time of 20 ms, and pass energy of 50 eV. Analysis for the presence of the strong line of fluorine, F 1s (689.0 eV), was also performed as a negative control to ensure there was no contamination from fluorinated silanes that are stored in the glove box. Relative elemental composition of the various elements was measured by calculating the area under the curves of the different elements and normalizing those areas by the sensitivity factor of the particular element, and the transmission function of the instrument. Relative atom percent was calculated from normalized peak areas.

Palladium-catalyzed metallization of patterned silane monolayers

Patterned samples were visualized using a palladium-catalyzed copper reduction reaction, modified from Kind et al.[59], that specifically deposits metallic copper on regions containing the amine terminated silane DETA. The patterned substrates were immersed in a solution containing 0.8 mM Palladium chloride and 0.6 mM NaCl for 10 minutes. The substrates were then rinsed in 3 x in diH_20. A solution of 1 part 0.6 M dimethylamiobutyrate (DMAB) and 4 parts diH_2O was prepared, and the samples were immersed for 10 minutes. The samples were again washed in diH_20 then immersed in 10 ml of a 0.2 M solution of Copper (II) sulfate with 10 μl of formaldehyde (37.2%) until copper deposition can be seen. When the solutions used for this reaction are made fresh the final copper reduction may only take a minute to visualize the pattern. As the solutions age, however, the reduction reaction may take longer so care must be taken to monitor the development of pattern. Metallized patterns were imaged and analyzed using light microscopy.

Cell Culture Methods

Embryonic Skeletal Muscle

Skeletal muscle was dissected from the hind limb thighs of a rat fetus at embryonic day 18 (Charles River Laboratories, Wilmington, MA) according to previously published protocol [37] with some modification. Tissue samples were collected in a sterile 15-ml centrifuge tube containing 1 ml of calcium and magnesium free phosphate buffered saline (PBS). Tissue samples were enzymatically disassociated using 3 ml of 0.05% of trypsin–EDTA (Invitrogen, Carlsbad, CA) solution for 60 min in a 37°C water bath with agitation of 100 rpm. After 60 min, the trypsin solution was removed and 6 ml of L15 media (Invitrogen, Carlsbad, CA) containing 10% fetal bovine serum (FBS) was added to terminate the trypsin

action. The tissue was then mechanically triturated using a sterile narrow bore Pasteur pipette, allowed to settle for 3 min, and transferred to a 15-ml centrifuge tube. This was repeated three times. The dissociated tissue was then centrifuged at 300g for 10 minutes at 4°C on 6 ml of a 4% (wt/vol) cushion of bovine serum albumin (BSA). The pellet was resuspended in 10 ml L15 + 10% FBS and plated in uncoated 100-mm Petri dishes for 20–30 min depending on the amount of tissue, to allow contaminating fibroblasts to settle out. After 20–30 minutes the supernatant was layered on 6 ml of a 4% BSA cushion, and centrifuged at 300g for 10 min at 4°C. The pellet was resuspended in 1.5 ml of medium.

Purified myocytes were plated at a density of 500–800 cells per square millimeter. Myocytes were allowed to attach for 1 hour after which time 3 ml of culture medium (Neurobasal media containing B-27 [Invitrogen, Carlsbad, CA], Glutamax [Invitrogen, Carlsbad, CA], and Pencillin/Streptavidin) was added. Cultures were maintained in a 5% CO_2 incubator (relative humidity, 85%). Culture medium was exchanged every 4 days. Cantilever/myocyte constructs were allowed to culture for 10-13 days. During this time myocytes fuse into functional myotubes capable of spontaneous contraction, as well as evoked contraction under electrical field stimulation.

Embryonic motor neuron

Spinal motoneurons were purified from ventral cords of embryonic day 14 (E14) rat pups. Briefly, rats were euthanized by CO_2 asphyxiation. Ventral spinal cells from the embryo were collected in cold Hibernate E (BrainBits, Springfield, IL, SA)/GlutaMAX™/antibiotic-antimycotic/ B27 (Invitrogen, Carlsbad, CA, USA). The cells were dissociated with 0.05% trypsin–EDTA (Invitrogen) treatment. The dissociated cells were layered over a 4 ml step gradient (Optipep diluted 1:1 (vol/vol) with Hibernate E/GlutaMAX™/antibioticantimycotic/

B27 and then made to 15%, 20%, 25% and 35% (vol:vol) in Hibernate E/GlutaMAX™/antibiotic-antimycotic/B27) followed by centrifugation for 15 min, using 800g, at 4 °C. This is modified from the previously described protocols due to non-availability of metrizamide [36, 60, 61]. After centrifugation, four bands of cells were obtained, the topmost of which contained the motor neurons. These cells were further purified by immunopanning. The motoneurons were selected using the immune interaction between the motoneurons and MAB192 antibody (1:2 dilution, ICN Biomedicals, Akron, OH, USA) coated on the dishes [36, 62]. The antibody recognized the low affinity NGF receptor that is only expressed by ventral motoneurons at this age [63].

Embryonic hippocampal neuron

Rat pups at embryonic day 18 dissected from timed pregnant rats which were euthanized using CO_2 asphyxiation. Embryos were collected in ice cold Hibernate E/ B27/ Glutamax™/ Antibiotic-Antimycotic. The hippocampi were isolated from the embryonic brain and collected in tube containing 1ml of Hibernate E/ B27/ Glutamax™/ Antibiotic-Antimycotic. The embryonic hippocampal neurons were obtained by triturating the tissue using a Pasteur pipette. The 1ml cell suspension was layered over a 4 ml step gradient (Optipep diluted 1:1 (vol:vol) with Hibernate E/ GlutaMAX™ / antibiotic-antimycotic/ B27 and then made to 15%, 20%, 25% and 35% (v/v) in Hibernate E/ GlutaMAX™/ antibiotic-antimycotic/ B27) followed by centrifugation for 15 min, using 800g, at 4°C. This additional step helped to remove the debris arise during dissection from the damaged cells. After centrifugation, one strong bands of cells were obtained at the top. The pyramidal hippocampal neurons constituted this band with large somas. The cells were resuspended in culture medium (

Neurobasal / B27 / Glutamax™ / Antibiotic-Antimycotic) and plated at a density of 75 cells/mm^2 [57, 64-68]. Half of the medium was changed after every 3-4 days.

Results and discussion

Contact angle measurements

Contact angle measurements were performed on samples from all reaction conditions: Control PEG monolayer, control DETA monolayer, and DETA backfilled onto unablated and ablated PEG monolayer, as well as PEG backfilled onto a control DETA surface. The results are seen in Figure 4. For the control PEG monolayers, the contact angle was measured to be $37.0^o \pm 1.1^o$ which is consistent with a silane PEG monolayer. The control DETA samples also showed a normal contact angle of $48.6^o \pm 1.1^o$. Figure 4c shows the contact angle for DETA backfilled onto an unablated PEG monolayer. While the reaction conditions were identical to those of plain clean glass immersed in the same reaction mixture the contact angle, $37.2^o \pm 0.6^o$, does not show a significant degree of change from the control PEG sample. This is due to the presence of the methoxy-terminal group of the PEG silane protecting the monolayer from reaction with the DETA silane. The methoxy-terminal group is a poor nucleophile compared to the hydroxylated surface of the clean glass coverslips, therefore little to no reaction takes place on the PEG monolayer. Conversely, when the PEG silane is reacted onto a DETA surface the resulting contact angle and XPS data show evidence of PEG reacting onto the unprotected DETA forming a hybrid surface with a contact angle of $42.6^o \pm 0.5^o$ which is intermediate between that of PEG (~38^o) and DETA (~49^o). Figure 4d shows the contact angle resulting for DETA reacted onto a PEG monolayer ablated by DUV photolithography for 45 seconds. The contact angle for this condition, $49.3^o \pm 1.2^o$, is statistically indistinguishable ($p<.01$) from the control DETA sample. This

demonstrates that the PEG monolayer is ablated sufficiently to leave a surface suitable for formation of a normal DETA monolayer.

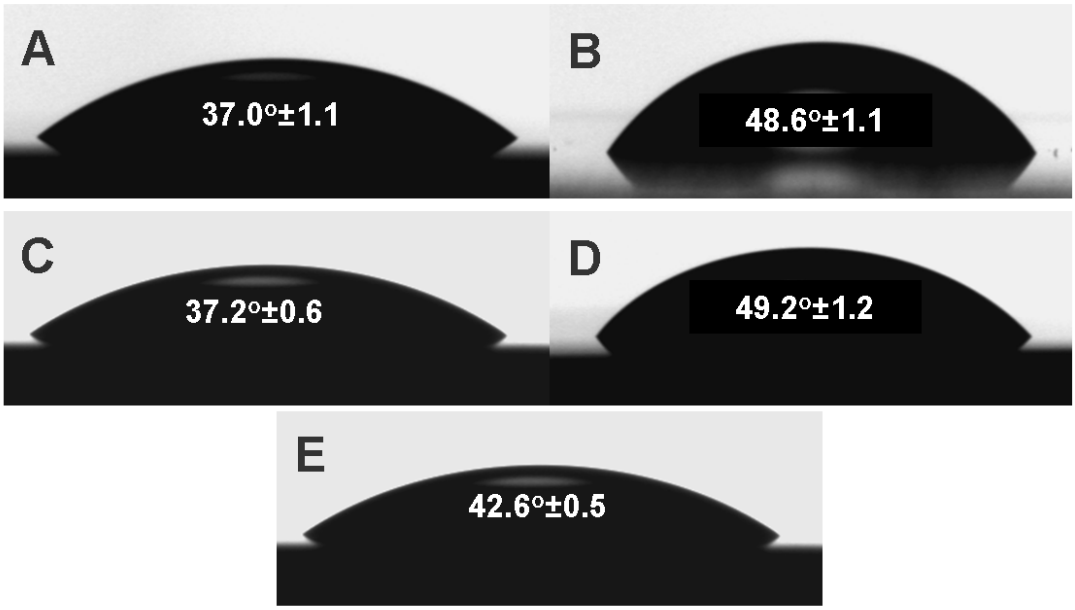

Figure 4: Contact angle measurements for, a) PEG control, b) DETA control, c) DETA backfilled into unablated PEG control, d) DETA backfilled into ablated PEG, e) PEG backfilled into a DETA control.

XPS analysis

Analysis was done by XPS to verify the chemical composition of the various surfaces tested in this study. Figure 5 shows representative spectra (survey spectrum, and high resolution spectra for C 1s and N 1s peaks) for a control PEG monolayer. The characteristic peak for this surface is the C 1s peak. In figure 5 it can be seen that the C 1s spectrum is comprised of to partially overlapping peaks. The smaller peak, at ~284.6 eV is characteristic of an aliphatic carbon peak which corresponds to the 3 carbon spacer between the silane and the PEG units. The larger peak, at ~286.6 eV corresponds to a carbon bound to an oxygen or nitrogen group. This is expected as the PEG silane used for this study is a 6-9 mer of PEG

groups. The ratio of these two peaks is approximately 6:1 indicating the proportion of aliphatic to ether linkages expected the PEG silane.

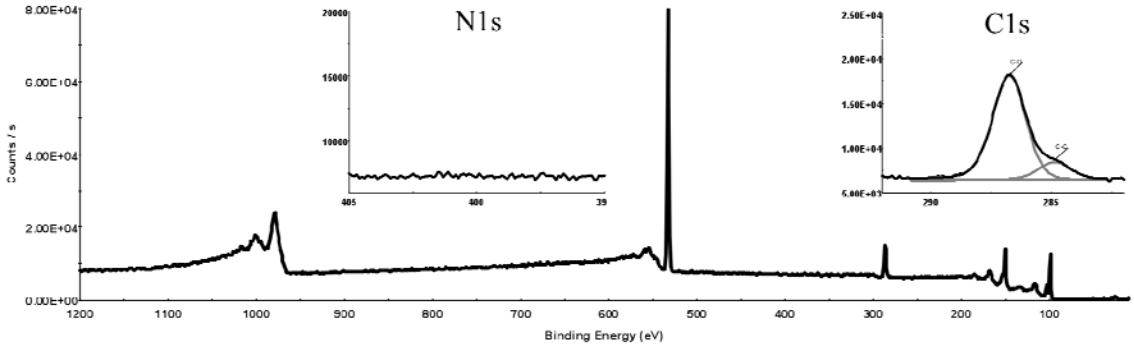

Figure 5: Representative XPS spectra of PEG monolayer on silicon. Survey spectrum and high resolution C 1s and N 1s (inset).

A time course study was then performed to determine the optimal ablation time for the PEG monolayer. Figure 6a shows the atom percent values for the C 1s spectra as a function of ablation time. It can be seen that at time zero the total carbon content is ~21%. After only 30 seconds of ablation time the carbon content is reduced to values comparable to those of clean silicon (dummy point at 240 seconds). After 30 seconds of ablation little change is seen in the carbon content of the samples tested.

The ablated PEG samples were then reacted with DETA to determine which ablation times yield optimal backfill of DETA into the patterned PEG monolayer. Figure 6b shows the percent nitrogen content versus ablation time. As seen in Figure 6a, at thirty seconds of ablation time nitrogen content values consistent with those of a normal DETA monolayer are observed. Beyond 30 seconds nitrogen content values are consistent with those found in a normal DETA monolayer. Figure 7 shows representative XPS spectra of a PEG coated silicon wafer ablated for 45 seconds by DUV photolithography. It should be noted that in these spectra the C-O component of the C 1s spectrum is drastically reduced compared to the

C-C component (figure 7). This is evidence that the C-O bond of the PEG silane is photo-labile at 193 nm and is the site of reaction during photoablation.

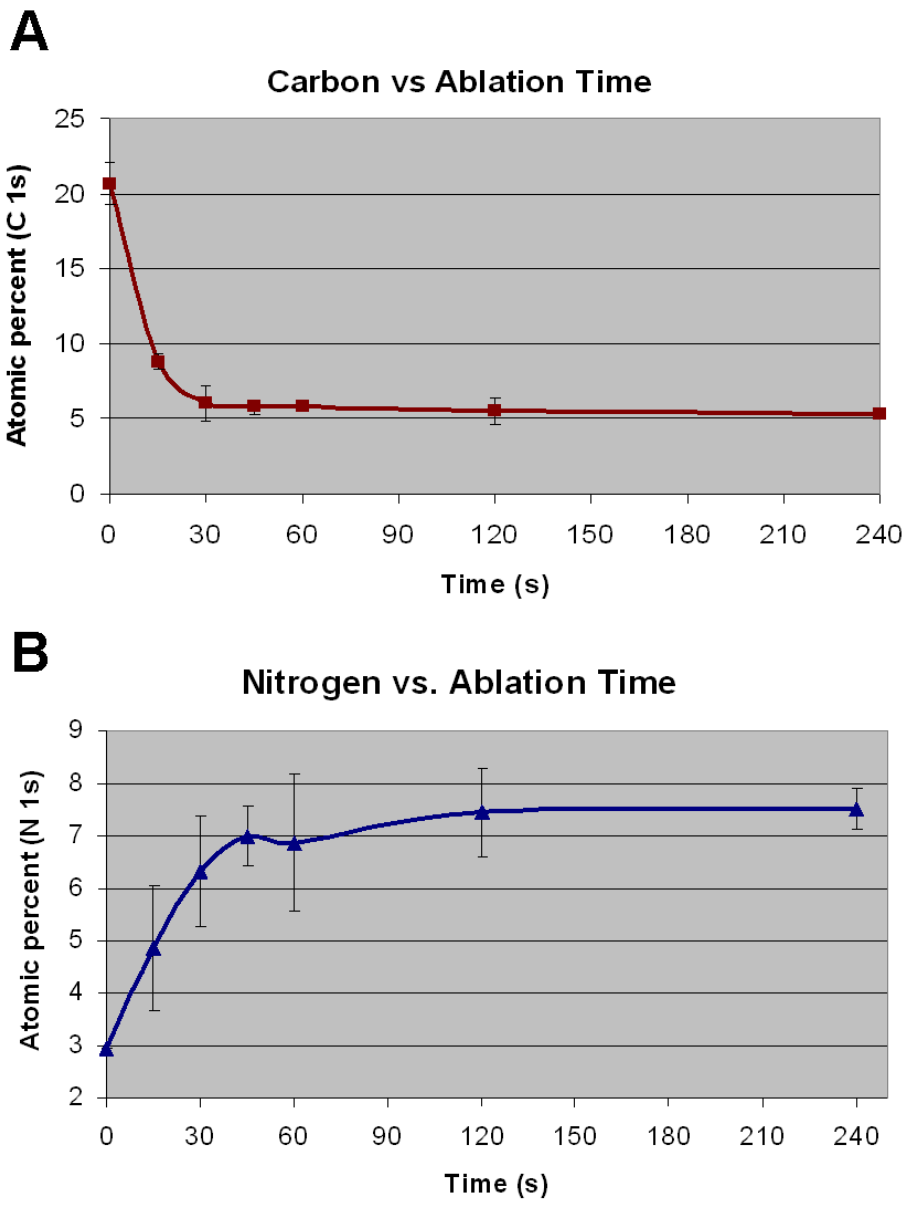

Figure 6: A) Atomic percent of C1s signal vs. ablation time. After 30 seconds atomic percent of C1s is comparable to that of clean Si. B) Atomic percent of N1s signal vs. ablation time. Ablated samples were reacted with 0.1% DETA for 30 minutes to form a monolayer of DETA in ablated regions.

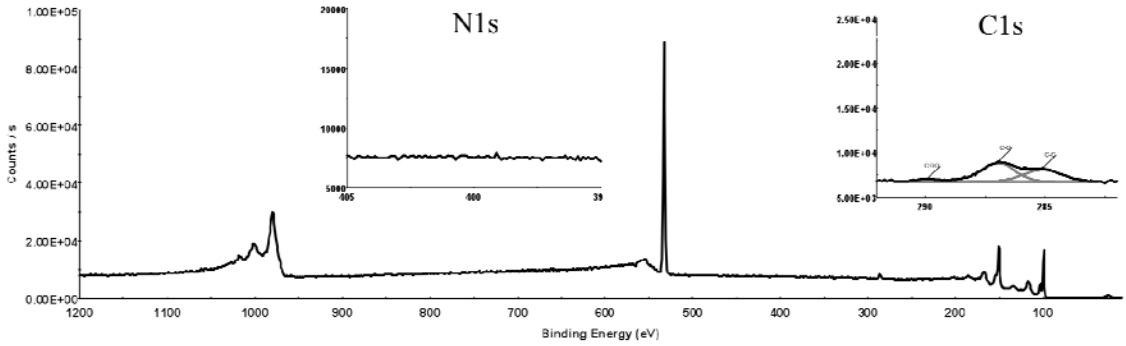

Figure 7: Representative XPS spectra of ablated PEG monolayer on silicon. Survey spectrum and high resolution C 1s and N 1s (inset).

For the culture of cells on patterned PEG-DETA surfaces it was important to verify a high level of contrast between the PEG and DETA portions of the pattern. Therefore, it is important that a minimum of DETA be incorporated into the PEG portion of the pattern. If DETA were able to freely react with the PEG monolayer, an over-layer of DETA would be formed on the PEG and effectively eliminate the surface contrast and hence the cytophobicity of the PEG surface. Also, it was necessary to optimize the reaction time to ensure DETA monolayer formation on the ablated regions while minimizing incorporation into the PEG regions. Figure 8 shows the nitrogen content as measured by XPS on control PEG samples as well as samples that had been ablated for 45 seconds versus reaction time. The blue trace shows the nitrogen content of DETA backfilled onto ablated samples. Here it can bee seen that at 30 minutes the nitrogen content reaches a level suitable for cell culture. Samples incubated for 20 minutes or less, while showing substantial DETA incorporation, are slightly lower than what is optimal for culture (~7.5). Conversely it can be seen in the blue trace that after 10 minutes of reaction time, the relative nitrogen content incorporated onto a control PEG sample does not change. DETA incorporation into a PEG monolayer reaches its maximum before 10 minutes reaction time. This is important to know as it becomes clear

that longer reaction times can be used to optimize DETA coating on ablated samples while not sacrificing incorporation of DETA into the PEG coating.

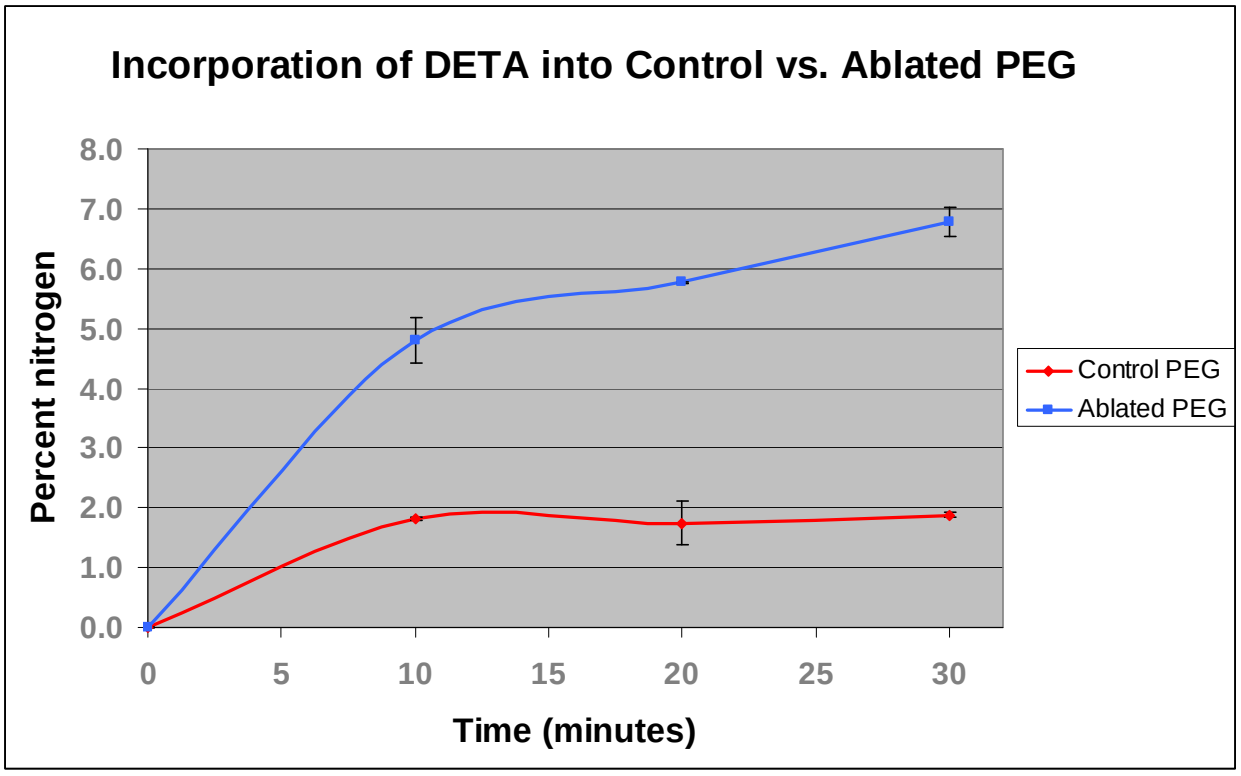

Figure 8: Nitrogen content incorporated into control vs. ablated PEG monolayers as a function of reaction time.

As noted in the previous section the contact angle of an unablated PEG surface reacted with DETA does not significantly differ from that of a control PEG surface. XPS analysis was performed to compare spectra from control PEG monolayers to that of unablated PEG monolayers reacted with DETA for 30 minutes at 65°C (Figure 9). It can be seen from both the survey and high resolution spectra that only a very small amount of nitrogen can be detected (~1.9%). This evidence lends support to the hypothesis that the methoxy-terminated PEG-silane effectively protects the PEG monolayer from reaction with the DETA silane allowing the formation of high contrast alkylsilane monolayer patterns.

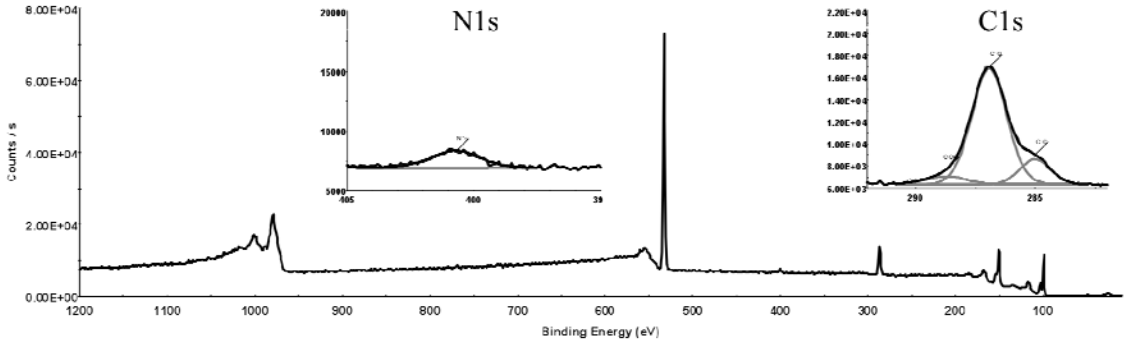

Figure 9: Representative XPS of unablated PEG monolayer reacted with DETA for 30 minutes at 65°C. Survey spectrum and high resolution C 1s and N 1s (inset).

In order to test whether the nitrogen detected on the control PEG samples was due to covalently linked or physically adsorbed DETA, samples were sonicated for 30 minutes in dry toluene to attempt to remove adsorbed silane. XPS analysis showed that after sonication the measured nitrogen content was identical to that in unsonicated samples. This suggests that the DETA was covalently linked to the surface although it is unclear what mechanism it occurs by (Data not shown).

Figure 10 shows XPS spectra for a control DETA surface. The C 1s component of the XPS spectrum shows two components at 285.5 eV and 286.7 eV, which correspond to aliphatic carbon (C-C) and carbon in an amine (C-N) bonded environment respectively. The primary component of the carbon signal comes from the C-N component, which is expected as 5 of the 7 carbon atoms in DETA are bonded to nitrogen. The ratio of the two peaks reflects this (Table 1).

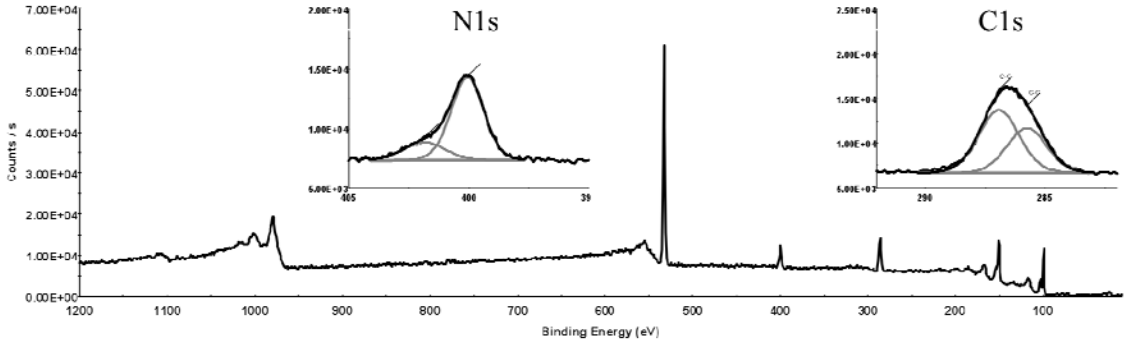

Figure 10: Representative XPS of control DETA monolayer. Survey spectrum and high resolution C 1s and N 1s (inset).

Figure 11 shows XPS spectra of DETA backfilled into an ablated PEG monolayer. The DETA monolayer deposited onto the ablated PEG surface was comparable to that of the control DETA deposited onto clean glass with some slight differences. The analysis of the N 1s spectrum shows that the amount of nitrogen present, 7.4 % (Table 1), was suitable for cell cultur. The C 1s spectrum shows, as with the control sample, the primary component of the C1s signal is centered at 286.7 eV, which corresponds to an amine bonded carbon. An aliphatic component is present at 285.5 eV and a small component at 288.9 eV which corresponds to carbon in a highly oxidized state such as an amide or carboxylic acid group. This component of the signal may be due to oxidized carbon residue resulting from the ablation reacting with the amine containing DETA. The precise nature of carbons in this chemical state cannot be deduced from the data presented here.

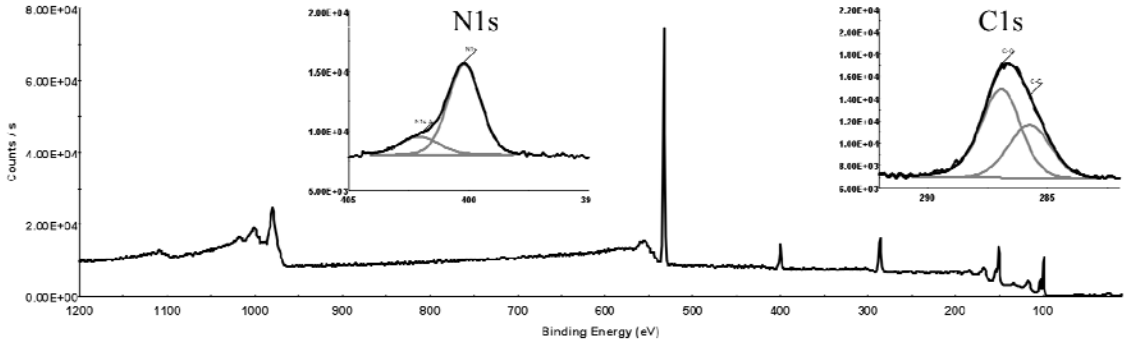

Figure 11: Representative XPS of ablated PEG monolayer reacted with DETA for 30 minutes at 65°C. Survey spectrum and high resolution C 1s and N 1s (inset).

Table 1: Relative content and chemical states of carbon and nitrogen on silane surfaces.

	Clean Si	Control PEG	Ablated PEG	DETA on control PEG	Ablated PEG-DETA backfill	PEG on control DETA	Control DETA
N1s	0.0±0.0%	0.0±0.0%	0.0±0.0%	2.2±0.6%	7.4±0.7%	5.9±0.1%	7.8±0.8%
C-O/C-N	0.0±0.0%	18.3±1.3%	1.6±0.2%	16.8±0.8%	14.6±1.5%	18.0±2.8%	12.1±.03%
C-C	5.1±0.5%	2.9±0.3%	3.9±0.7%	3.6±0.8%	6.0±2.1%	10.4±2.6%	9.4±0.8%
COO	0.0±0.0%	0.0±0.0%	0.3±0.1%	0.9±0.9%	2.8±0.3%	2.1±0.2%	0.0±0.0%

Figure 12 shows representative XPS spectra of PEG reacted onto a control DETA surface. Samples reacted with DETA then PEG respectively yielded surface properties intermediate to those of either PEG or DETA alone. The high resolution spectrum of the C 1s region shows that the dominant peak is found at 286.9 eV, which corresponds to either the ether or amine group found in PEG and DETA surfaces respectively. However, the ratio of the ether/amine peak to that of the aliphatic carbon is intermediate to that of PEG and DETA. This suggest that this set of conditions yields a mixed monolayer that is neither completely

PEG of DETA. As previously stated, the contact angle measurements support this conclusion as the contact angle lies in between that of pure PEG and DETA surfaces. It is possible that this is due to PEG silane reacting directly to the free terminal amine of DETA, as it is a good nucleophile. This data also serves as further evidence that the methoxy-terminus of the PEG silane effectively protects it from reaction with the DETA silane during backfilling.

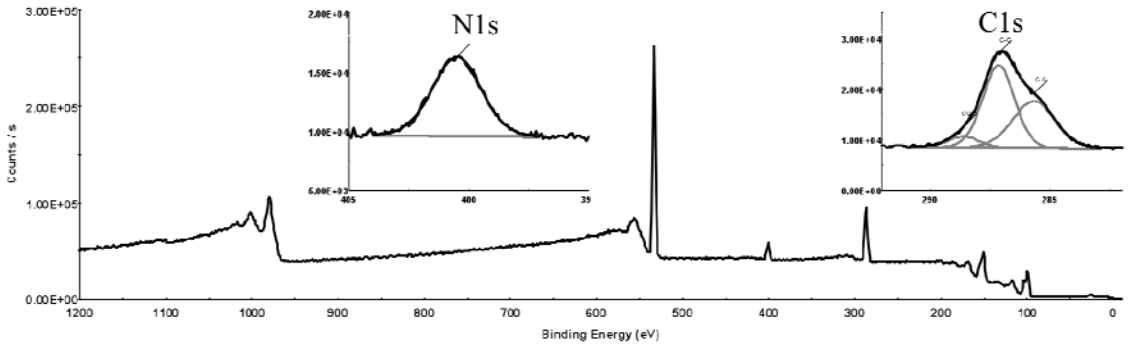

Figure 12: Representative XPS of PEG reacted onto a control DETA surface for 30 minutes at 65°C. Survey spectrum and high resolution C 1s and N 1s (inset).

Metallization and cell culture

Various types of cells were cultured on PEG-DETA surfaces to verify the usefulness of this method for cell patterning. Figure 13 shows the results for both metallization and culture of different cell types on PEG-DETA patterns. Figure 13a-c-e show palladium catalyzed metallization of DETA. In these images the metalized pattern corresponds to the dark regions of the picture. The pattern in Figure 13a consists of a series of 100 um wide lines used for culturing embryonic skeletal muscle. The patterns were used to mimic the geometry that would be used in future experiments on microfabricated silicon cantilevers. Figure 13c consists of an array of 30 μm x 30 μm squares and Figure 13e is a pattern designed specifically for creating 2-cell networks of hippocampal neurons. The metallization results confirm that the combination of PEG and DETA creates high contrast patterns that should be suitable for cell culture. Since the copper reduction reaction only occurs on regions

containing the amine-terminated DETA, it can be seen that the DETA is confined primarily to regions where the PEG had been ablated. This result is in agreement with the XPS analysis and the hypothesis that the methoxy-terminated PEG used for these studies resists incorporation of DETA. A limited amount copper of deposition can be seen on the PEG regions of the DETA backfilled patterns, which indicates a small amount of nitrogen present in the PEG regions. This also is in agreement with XPS analysis, which shows a low level of nitrogen signal in unablated PEG samples reacted with DETA. It was possible that this low-level copper deposition could be due to reduced copper precipitating out onto the background in a non-specific manner. For this reason, metallization experiments were performed on unablated PEG surfaces (results not shown), which confirmed that no copper deposition was seen on native PEG surfaces. Thus, it can be concluded that a small yet detectable amount of DETA is being incorporated into the PEG regions. Both XPS and contact angle analysis indicated that the amount of DETA incorporated in to the PEG regions is not sufficient to alter the surface properties from cytophobic to cytophilic, since the nitrogen content is well below that used in cell culture and the contact angle was indistinguishable from control PEG surfaces. Cell culture experiments confirmed that these surfaces were suitable for cell patterning.

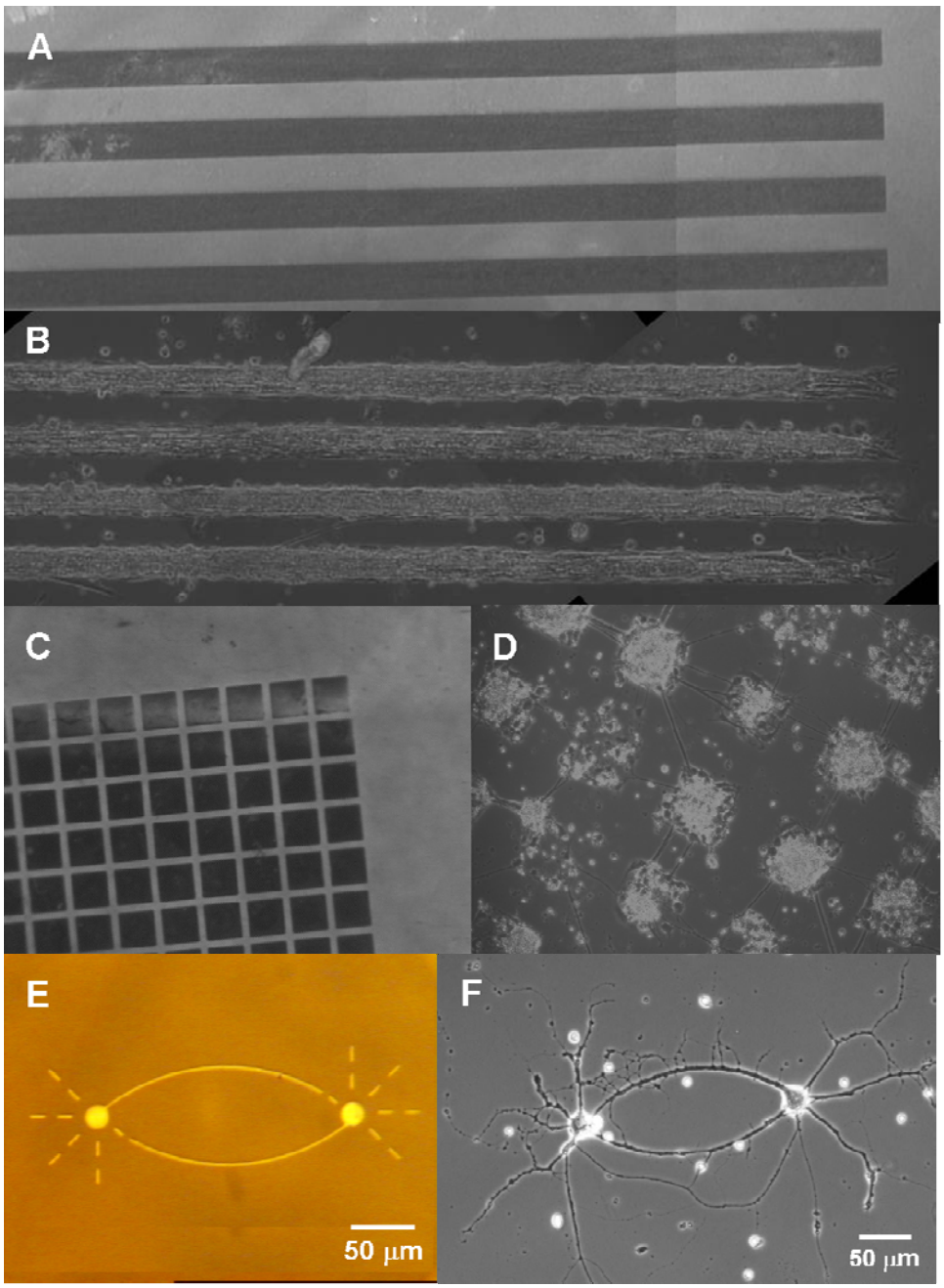

Figure 13: Metalized PEG-DETA patterns and resulting cell cultures with varying cell types. A-C-E) Metallization results for three different patterns (A, 100 µm lines; B, 30 µm x 30 µm squares; C, two-cell circuit pattern). B-D-F) cells cultured on PEG-DETA patterns (B, embryonic skeletal muscle; D, embryonic motorneuron; F, hippocampal neurons).

Multiple cell types were cultured on the PEG-DETA surface to verify that this method is broadly applicable to patterning cells on two dimension surfaces. Figure 13b-d-f shows phase contrast microscopy images embryonic skeletal muscle, motorneuron, and

hippocampal neuron respectively. In all cases the cells were confined to the DETA regions of the patterns and adhered to the patterns for the duration of the culture. Figure 13b shows embryonic skeletal muscle cultured on 100 μm wide lines of DETA. It can be seen that the cells adhered and differentiated on the DETA regions of the lines. Myotube assembly was anisotropic with myotubes orienting themselves along the long axis of the pattern. This behavior has been demonstrated before using mouse skeletal muscle cell lines on microcontact printing patterned surfaces[17]. The myocytes then fused and differentiated into functional myotubes that exhibited spontaneous contraction as well as contraction under electrical stimulation.

Embryonic motorneurons were also cultured on PEG-DETA surfaces. Figure 13d shows phase contrast images of motorneurons cultured for 10 days on surfaces patterned with an array of 30 μm x 30 μm DETA squares. It can be seen that it is primarily the cell bodies that adhere to the square pattern, while processes can be seen to extend across the PEG regions to contact cells on other squares. It was common to see cells extend processes onto the PEG while the cell bodies remained attached to the DETA regions. This indicates that while the PEG is a cytophobic surface, it is not completely repulsive to attachment of cell processes. A discussion of the difference between the mechanisms of cell soma versus process adhesion is beyond the scope of the current study.

Figure 13f shows embryonic hippocampal neurons cultured on PEG-DETA patterns designed for creating two-cell circuits. The pattern consisted of two circular regions (20 μm diameter) intended for adhesion of the cell bodies. These circular regions were connected by 2 μm lines that extend in an arc from one somal adhesion site to another. The somal adhesion sites were further surrounded by dashed lines extending radially from the somal adhesion sites. This geometry served to induce a specific polarity in the neurons similar to

Ravenscroft et al.[12], such that they would extend a dominant process (which would become the axon), which would contact the soma of the opposing cell. The image in Figure 13f shows cells adhering to the pattern shown in figure 12e. Cell bodies can be adhered to the circular regions for somal adhesion and sending out axonal processes which follow the lines connecting the two cell bodies

Figure 14 shows results from embryonic skeletal myoblasts cultured on PEG-DETA patterns over 41 days. Embryonic skeletal myoblasts were cultured on 100 μm wide lines of DETA patterned on PEG surfaces. After 5 days the myoblasts fused and differentiated into myotubes. Figure 14a-d shows myotubes formed on the DETA lines after 9, 16, 26, and 41 days in culture respectively. It can be seen that after 26 days in culture the number of myotubes adhering to the patterns is reduced when compared to earlier culture days. This is in part due to the functional myotubes spontaneously contracting on the substrate and detaching. Despite this, a significant number of myotubes remain attached to the patterns. It can also be seen that the areas between the DETA lines remain relatively free of migrating cells and cell debris. This long-term maintenance pattern contrast and fidelity will prove critical in the application of this technique for creating long-term patterned cultures that can be used to create *in vitro* models for the study of cellular development, cell-cell interactions, and cellular response to exogenously applied compounds(growth factors, drugs, toxins, etc).

Conclusions

A novel method for patterning cells with alkylsilane monolayers was developed. PEG-terminated monolayers were used a cytophobic surface to prevent cell adhesion. The PEG monolayers were then patterned using DeepUV photolithography according to Stenger et al [56]. The patterned monolayer was then reacted with the amine-terminated silane DETA to create cell-adhesive islands in the non-adhesive PEG regions. Patterned and unpatterned

surfaces were analyzed by XPS, contact angle goniometry, and palladium catalyzed copper reduction metallization. This technique was used to pattern various cell types (motorneuron, hippocampal neuron, and skeletal muscle). Furthermore it was shown that cells were confined to the patterns for time periods in excess of 40 days.

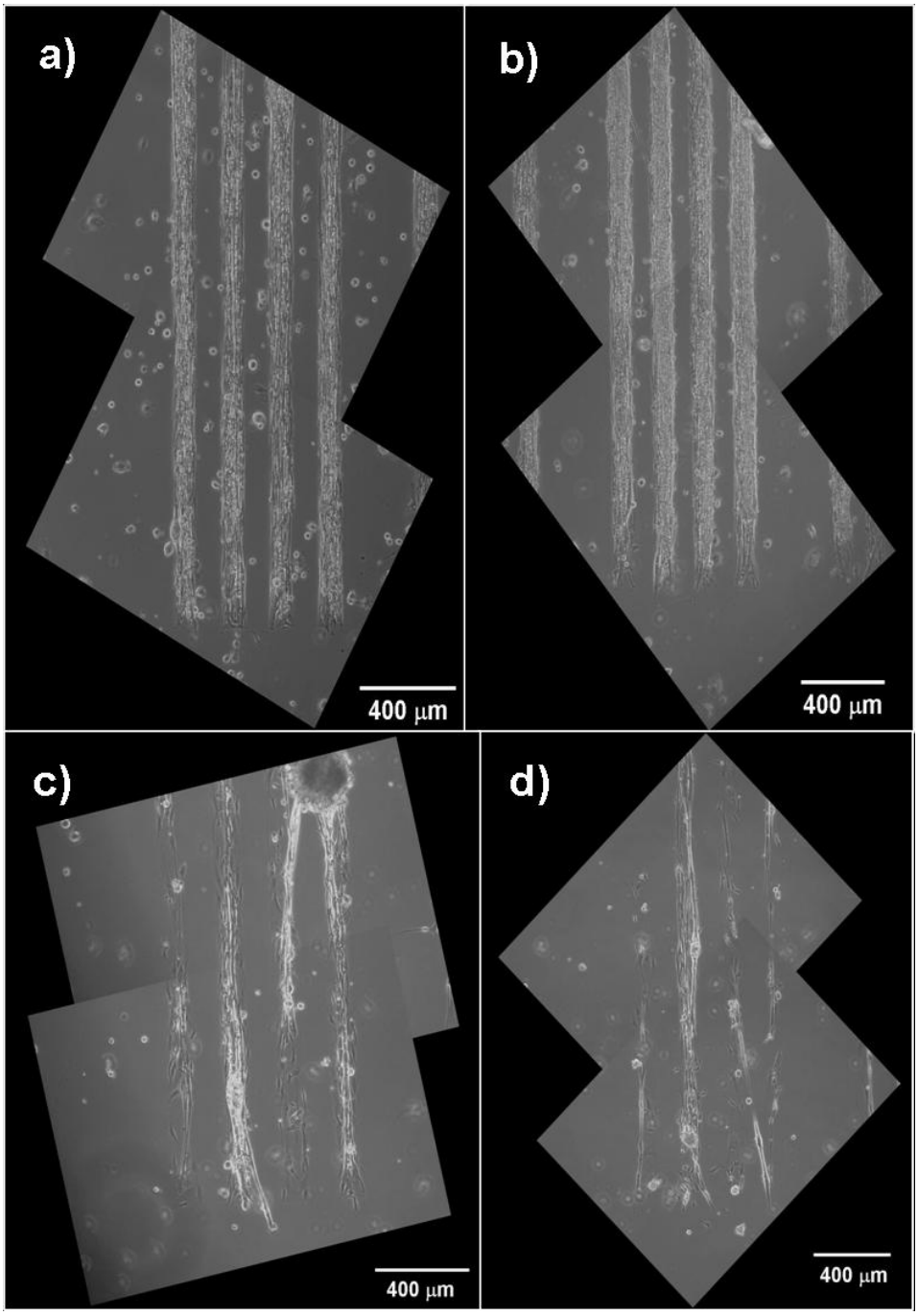

Figure 14: Skeletal muscle on patterned PEG-DETA surface remained confined to the DETA regions of the pattern up to 41 days on 100 μm wide lines. A) 9 days in culture, B) 16 days in culture, C) 26 days in culture, D) 41 days in culture. Although many of the myotubes had pulled off the surface due to spontaneous contraction, the remaining myotubes were still confined to the patterns..

Characterization of unpatterned and patterned surfaces showed that this method produces high-contrast 2-dimensional silane patterns with characteristics suitable for cell culture. XPS analysis showed that the PEG silane can be ablated sufficiently to create a surface suitable for rederivitization with a DETA monolayer capable of supporting cell adhesion, growth, and differentiation. Furthermore, it was shown that the DETA did not significantly incorporate into the PEG regions. Because of this the contrast between adhesive and non-adhesive regions was maintained. Contact angle measurements show that the wetting properties of the unablated PEG monolayers reacted with DETA are identical to those of control PEG surfaces, while the contact angle of DETA monolayers formed on ablated PEG are identical to that of control DETA surfaces. Also, metallization of the patterned DETA regions demonstrated that the amine silane is confined to the ablated PEG regions resulting in a high-contrast pattern. It is hypothesized that the lack of incorporation of DETA into the PEG regions is due to the methoxy-terminus on the PEG silane. Control experiments in which the PEG silane was reacted onto a DETA monolayer resulted in a hybrid surface with contact angle and elemental composition intermediate to that of PEG and DETA.

Cell culture experiments showed that this technique creates surfaces suitable for culturing a wide variety of cell types. Culture results for hippocampal neurons, motorneurons, and skeletal muscle are presented. The applicability of this technique to such a broad range of cell types make it ideal for creating complex cocultures with multiple cell types. This can be used for creating *in vitro* systems that can be used as models for *in vivo* biological circuits

such as neuronal networks and neuromuscular constructs. In this manner *in vitro* test beds can be created that control the spatial orientation of the cells in question on a well-defined substrate and in a highly controlled culture environment.

CHAPTER THREE: PROTEIN ADSORPTION TO ALKYLSILANE MONOLAYERS AS MEASURED WITH A WHISPERING-GALLERY MODE BIOSENSOR

Introduction

Protein adsorption

Non-specific binding (adsorption) of biomolecules at solid-liquid interfaces is an important phenomenon that affects the function of materials and devices intended for use with biological systems. Biomolecule adsorption is a crucial factor in determining detection limits, biocompatibility, and long-term efficacy of lab-on-a-chip, microfluidic, and Bio-MEMS devices due to loss of analyte by adsorption and fouling of microchannels and active sensors. Proteins are particularly notorious for their ability to non-specifically stick to materials. An excellent review of protein adsorption is provided by Andrade and Hlady [69]. The inherent variability of protein sequence and structure make the prediction of protein adsorption from first principles an intractable problem. Thus, it is necessary to devise elegant experimental solutions for making empirical observations that can be used to develop better models for predicting how various surface/protein combinations will interact.

There are many different methods for studying adsorption of biomolecules, such as radio-labeling, fourier transform infrared spectroscopy (FTIR), X-ray photoelectron spectroscopy (XPS), ellipsometry, and the surface plasmon resonance (SPR) technique, all of which have strengths and weaknesses. Ideally a technique for studying adsorption should be highly sensitive, label free, suitable for high throughput applications, and allow for real-time monitoring of adsorption kinetics. Currently the standard method for studying adsorption of biomolecules at liquid-solid interfaces is SPR. SPR is a mature technique that is label free, allows real-time monitoring, and boasts an ultimate detection limit of ~10 pg/mm^2 [2,70]. However, SPR is not suitable as a high throughput technique. Furthermore, due to the small

volumes and low sample concentrations used in many applications sensitivity approaching the single molecule regime is highly desirable.

Surface Chemistry Effects

It is well known that the chemical composition of the outermost few nanometers of a material's surface largely determines the biocompatibility of that material [71]. The Gibbs free energy of interaction between a surface and protein, ΔG, can be used as a general predictor of adsorption. The Gibbs free energy equation is written as: $\Delta G = \Delta H - T\Delta S$, where ΔH is the change in enthalpy upon adsorption, ΔS is the change in entropy, and T is the temperature of the system in Kelvin. Surfaces can be loosely classified into three categories that have been shown to have distinct behavior with respect to protein adsorption: 1) neutral hydrophobic, 2) neutral hydrophilic, 3) and charged hydrophilic.

Hydrophobic surfaces (particularly those consisting of alkane moieties) generally exhibit high levels of protein adsorption. This is largely attributed to the entropic benefits gained from a hydrophobic material shedding its hydration layer in favor of interacting with hydrophobic residues on a protein surface. Proteins with hydrophobic internal residues may also denature, exposing those internal residues to a hydrophobic surface thus, reducing the solvent accessible surface of the entire system, further increasing entropy. This denaturation of the protein, however, leads to irreversible binding and an undesirable loss of function. This phenomenon is well documented in the biomaterial literature [69, 72, 73].

Hydrophilic surfaces (-OH, PEG) are considered the most inactive towards protein adsorption. The ability of a hydrophilic surface to hydrogen bond with water creates an enthalpic barrier to protein adsorption. It is energetically unfavorable for both the material and protein surface to shed their bound water in favor of an adsorbed state, which in the case

of a neutral surface would be dominated only by weak Van der Waals forces. These surfaces cannot be said to be completely passive towards adsorption, however. Various papers have shown that proteins do adsorb somewhat on hydrophilic surfaces, just in comparatively small amounts. Proteins can retain their native conformation on a hydrophilic surface as well, allowing them to retain their function [53, 54, 74, 75].

Protein adsorption on charged surfaces is dominated by the electrostatic interactions between the surface and protein. The extent of protein adsorption has been shown to vary with pH [69]. At the isoelectric point (i.e.p.) of a protein, adsorption varies mainly with temperature indicating that entropic contributions dominate in the absence of charge/charge interactions. Bovine serum albumin (BSA), for example, has a net negative charge at pH 7.4 and will adsorb strongly to a positively charged surface while adsorbing much less on a negatively charged surface. At the i.e.p. of BSA little difference can be measured. The extent to which a protein adsorbs to a charged surface is not strictly dictated by net charge, however. The charge distribution on a protein surface may be inhomogeneous and posses enough local charge for it to adsorb, thus affecting not only the extent of adsorption but also the accessible binding surface of the protein. This effect has been reported in the biomaterials literature for several proteins, such as fibronectin (FN) [53, 74, 75] and osteopontin (Opn) [54]. It has been shown that, while the total amount of protein adsorbing to negatively and positively charged surfaces is comparable, there are significant differences in the binding of antibodies to specific binding motifs as well as effects on cellular adhesion, proliferation, and differentiation.

WGM biosensors

The so-called Whispering gallery mode (WGM) sensors derive their name from the whispering gallery of St. Paul's cathedral in London, where two people standing at

diametrically opposed sides of the room can clearly hear whispers from each other due to the resonance of sound waves within the gallery. WGM sensors are based on the total internal reflection of light within a dielectric resonator, usually a glass microsphere. Light is evanescently coupled to a glass microsphere with a radius of 100-500 μm, from an acid etched or flame drawn optical fiber connected to a tunable laser at one end and an InGaAs photodetector at the other. The resonant wavelength of the incident light is related to the radius, r, of the glass microsphere by the simple equation, $N \times \lambda = 2\pi r$, where λ is the wavelength of light, and N is an integer number. Upon resonance the incident photons couple with the microsphere reducing the transmitted light through the optical fiber to almost zero. As the photons orbit the microsphere they form a standing evanescent wave of light that extends ~200 nm from the surface[76, 77]. The tunable laser source is then swept through a range of wavelengths and the shift of the resonant peak is monitored by computer software. The change in the resonant peak, $\Delta\lambda$, is related to the thickness, t, of the adsorbed layer by the equation $\Delta\lambda/\lambda=t/r$. Thus as material is adsorbed from solution, increasing the effective radius of the microsphere, the resonant peak shifts to longer wavelengths. Even with a simple experimental configuration it has been shown that a detection limit of ~1 pg/mm^2 can be achieved (10 times more sensitive than SPR) and theoretical calculations predict the ultimate detection limit of the method to be in the single molecule regime[78, 79].

The limit of detection for WGM biosensors is determined by the number of orbits that a photon can make in a resonator before being absorbed or scattered [78]. In a perfect resonator with no loss of photons a sharp resonant line would be observed at the resonant wavelength, λ. In reality, however, any resonator experiences losses that limit the number of orbits to a finite number and the resonance to a certain line width, $\Delta\lambda$. A resonant shift can only be measured if it is greater than the width of the resonant line. For this reason a quality factor

has been defined, $Q = \lambda/\Delta\lambda$, to compare resonators and determine the limit of detection for a particular resonator. Under ideal conditions, atmospheric conditions where nascent water vapor adsorbed to the silica surface has been prevented, ultimate Q's of 10^{10} have been reported [80]. Due to vibrational light absorption by H_2O molecules, resonators in an aqueous environment are typically limited to Q values of 10^6. Even with these relatively modest Q values, it is possible to detect an adsorbed layer thickness of 10 picometers or 0.1 Å. This is well below the dimensions of even the smallest proteins.

Using this technique it is possible to quantitatively analyze a variety of parameters related to protein adsorption, such as adsorption kinetics, binding and debinding constants, and total surface density. Vollmer and coworkers [79] have developed a method based on first order perturbation theory that allows the calculation of the surface density of the adsorbed species, σ_s, based on the measured change in resonant wavelength, $\Delta\lambda$:

$$\frac{\Delta\lambda}{\lambda} = \frac{\alpha_{ex}\sigma_s}{\varepsilon_o\left(n_1^2 - n_2^2\right)R} \qquad \text{Equation 1}$$

where λ is the nominal resonant wavelength, α_{ex} is the excess polarizability (which is related to molecular weight), ε_o is the vacuum permittivity, R is the microsphere (resonator) radius, and n_1 and n_2 are the refractive indices of the sphere and the buffer solution respectively. With this method it is possible to monitor the development of the adsorbed protein layer as a function of time. Furthermore, since the dimensions of a protein molecule (usually less than 20 nm) are much smaller than the length of the evanescent field of the WGM, the resonant shift can be considered to be linearly proportional to the thickness of the adsorbed layer.

WGM biosensors are highly suitable for high throughput applications. WGM devices can be fabricated in large arrays on silicon wafers using standard photolithographic techniques. Resonators can be fabricated as simple disk [81], toroid [82], or microcavities in large arrays [83].

Variations in diameter give each of the microcavities a distinct resonant wavelength that can be tracked using a single waveguide or fiber optic. One particularly promising application for such an array is for label-free detection of DNA for sequencing and genetic analysis [81, 84]. The creation of a WGM biosensor array is beyond the scope of the current work, but provides tantalizing possibilies for future applications. Furthermore, as the resonators are made of Si or SiO_2, they are amenable to functionalization with alkylsilanes monolayers, which are broadly used in biomaterials and biosensing applications. With this method it is possible to perform real-time, label-free measurements of protein adsorption onto silanized microspheres and characterize the interactions of the protein with various SAMs.

Here a WGM biosensor has been constructed for real-time measurement of the adsorption of proteins onto alkylsilane self-assembled monolayers. The system is based on that of Vollmer and coworkers, but incorporates a novel flow cell for the delivery of sample to the resonator surface under laminar flow conditions. Adsorption of the extracellular matrix protein Fibronectin (FN) and the enzymatic protein glucose oxidase (GO) were studied at varying concentrations on PEG, DETA and 13F alkylsilanes SAMs.

Materials and methods

WGM instrument

A simplified schematic representation of the instrument used in these experiments is shown in Figure 15. The instrument can be broken down into three components: the fluidic system, the laser and detector, and the data acquisition and analysis system. The fluidics system is comprised of a peristaltic pump for delivery of the buffer and protein solutions to the sensor, the flow cell for housing the resonator and waveguide and focusing the buffer and protein solutions onto the resonator, as well as a waterbath for keeping solutions at a constant temperature. The laser and detection system consist of a tunable distributed feed-back (DFB)

diode laser, an arbitrary function generator, a current and temperature control unit for controlling the DFB laser, and an InGaAs photodetector. The data acquisition and analysis system was comprised of a computer with a data acquisition card with Labview software for acquiring the data, and data analysis software for calculating the surface concentration of adsorbed protein.

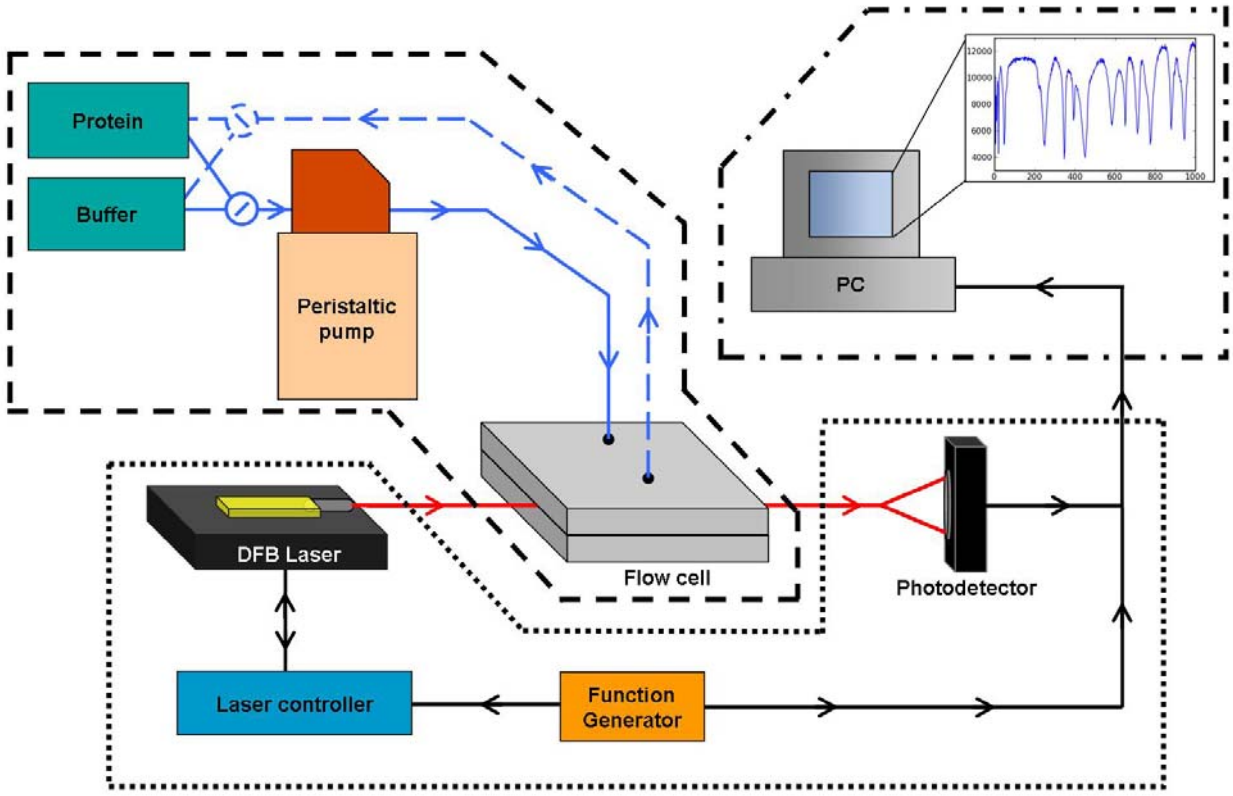

Figure 15: Schematic representation of WGM biosensor system. The instrument was comprised of three components, a fluidic system for driving buffer and protein solutions (dashed lines), the laser and detection system (dotted lines), and the data acquisition and analysis system (dotted and dashed lines).

Laser and detection system

The laser and detection system was based from that described by Vollmer et al [79]. A Lucent D2304G DFB laser diode with a nominal wavelength of 1310 nm and maximum power output of 10 mW was used as the excitation source in for all experiments. This laser contained a built-in optical isolator to prevent backscattering of the emitted light back into

the laser module itself and was connectorized with an FC/PC connector. The laser was mounted on an LDM-4984 butterfly laser diode mount (ILX Lightwave Corp, Bozeman, MT) and controlled with a LDC 3724B single channel current and temperature controller (ILX Lightwave Corp, Bozeman, MT). The wavelength of the DFB laser was modulated using an HP 33120A arbitrary waveform generator. The waveform generator was connected to the modulation input of the laser controller and the transfer coefficient was set to 20 mA/V. Under these conditions, for a change of 1 V the laser current was modulated by 20 mA, thus changing the wavelength emitted by the DFB laser in a current dependent manner. For all experiments a saw-tooth function with a variable peak-to-peak height was used to modulate the laser at a frequency of 100 Hz. The laser output was coupled to an FC/PC optical fiber, the end of which was connected, via a BNC cable, to a Thorlabs model PDA10CF InGaAs photodetector. The photodetector was connected to a Labview M-series data acquisition card and the signal from the detector was analyzed using a virtual instrument software written Labview 7.0 (National Instruments, Austin, TX).

The data acquisition software was modified from virtual instrument (VI) files kindly provided by Dr. Frank Vollmer of the Rowland Institute at Harvard University. The data acquisition VI tracked all resonant valleys in the acquired spectrum using a peak fitting algorithm that selected all valleys with a minimum FWHM value set within the VI and determined the position of the valley minimum using a Bessel function. The data acquisition was synchronized with the saw-tooth function created by the function generator such that acquisition began at the minimum and ended at the peak of the function. The position of each resonance over time was saved to a binary file to be analyzed later.

Microsphere fabrication

Microspheres were fabricated according to Vollmer et al [79]. A single mode fiber optic with a 250 μm acrylate polymer coating and 125 μm cladding with a 9 μm core (Fiber instrument sales, Oriskany, NY) was used to fabricate resonators. The acrylate coating was first removed using a fiber optic stripper and the stripped region wiped with a Kimwipe soaked with IPA to remove any residual acrylate. The end of the stripped fiber was then placed in the flame of a nitrous-butane Microflame torch (Azuremoon trading company, Cordova, TN). A nitrous-butane flame was used due to the very high temperatures needed to melt the glass and form the resonator (~2500^{o}C). The tip of the fiber was placed in the flame until the glass was seen to glow a bright white color and begin to melt. As the glass fiber melted the surface tension of the molten glass caused it to form into a spheroidal droplet suitable for exciting WGM resonances. As the tip melted the fiber was rotated to ensure that the resonator remained centered on the stalk of the fiber. This process was repeated until a resonator of the desired size was obtained (Figure 16). Resonator diameters used for these studies ranged from 250 to 350 μm. For studies of protein adsorption on glass, the resonators were used immediately after fabrication. Glass resonators made by this method had a contact angle of <5^{o}.

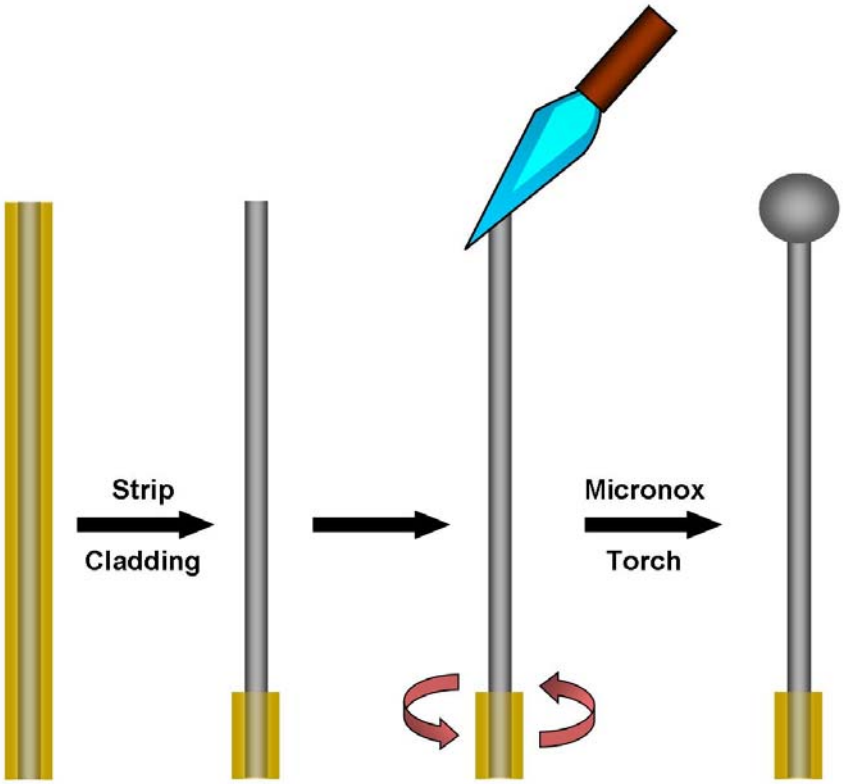

Figure 16: Fabrication of glass microspheres. The cladding was stripped from a section of the fiber and wiped with IPA. Then the tip of the stripped end was placed in a nitrous-butane flame while being rotated. Surface tension of the melting glass formed a spherical droplet that was used as a microresonator.

Surface modification of microspheres with silane monolayers

Glass microspheres were mounted into a resonator holder made from a perforated PDMS block which was adhered to a glass microscope slide. The resonators were inserted into the PDMS block with the resonator protruding out of the block (Figure 17). Glass microspheres, and glass coverslip controls for XPS and CA analysis, were then placed in a Harrick model PDC-32G plasma cleaner (Harrick, Ithaca, NY). The door of the plasma cleaner was closed and the chamber was evacuated to a pressure of 300 millitorr. Ultrapure oxygen was then purged into the system to a pressure of 800 millitorr, and evacuated again to 300 millitorr. Oxygen plasma was initiated by applying a RF field around the chamber. After initiation of the plasma, the pressure in the chamber was adjusted to ~550 millitorr. Cleaning was

allowed to proceed for 20 minutes. After cleaning the resonators and coverslips were removed for subsequent silane reaction.

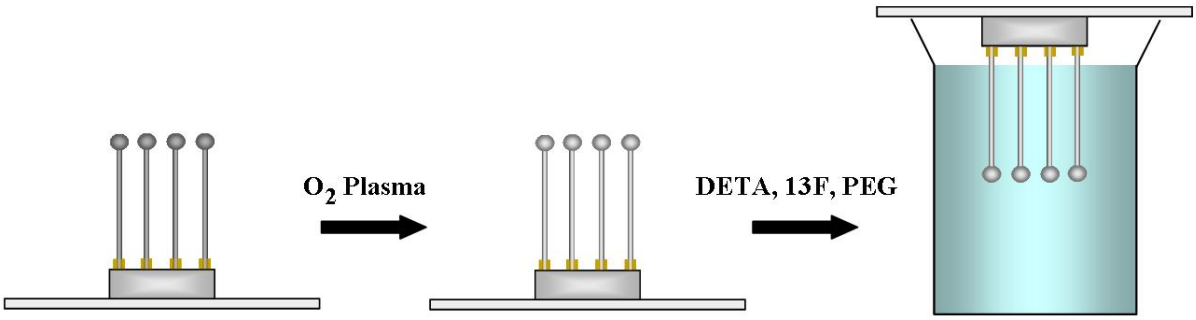

Figure 17: Surface modification of microspheres. Microspheres were first mounted in a PDMS block. They were then cleaned using an O_2 plasma. Cleaned microspheres were then immersed in solutions containing DETA, 13F, or PEG.

Modification with PEG and DETA silanes was performed as previously described in Chapter 2, with slight variation. Solutions of silane in toluene were prepared as described and transferred to 250 ml Pyrex beakers. The microspheres mounted onto the resonator holder were then inverted and immersed in the silane solution such that the microspheres were immersed in the solution and suspended from the holder. Microspheres coated with 13F were immersed in a 0.1% (vol:vol) solution of 13F silane in dry toluene. Dry toluene, resonators, and control coverslips were transferred into an MBraun glovebox (Stratham, NH) to perform the 13F modification under anhydrous, low oxygen conditions. This step was necessary to prevent polymerization of the 13F monomer, as it is a trichloro-silane which is highly reactive in the presence of water vapor. 13F was added to the dry toluene, mixed and transferred to a 250 ml Pyrex beaker. The coverslips and microspheres were then immersed in the silane solution for 30 minutes. 5 minutes prior to completion of the reaction, the beaker was removed from the glovebox and place in a chemical fume hood. Upon completion of the reaction, the microspheres and coverslips were washed 3x in dry toluene

and dried in an 80^{o}C oven for 1 hour. Control coverslips were analyzed by XPS and contact angle goniometry.

Waveguide fabrication

A 1.5 m section a single-mode fiber optic (125 μm cladding/ 9 μm core) (Fiber instrument sales, Oriskany, NY) was prepared by stripping a 1-2 cm section in the middle of the fiber and cleaning it with IPA. The fiber was mounted on a syringe pump (KD Scientific, Holliston, MA) such that tension was applied on the stripped section of the fiber when the pump was running. As tension was applied to the fiber and a nitrous-butane Microflame torch (Azuremoon trading company, Cordova, TN) was used to briefly melt a 0.5 to 1 cm section of the stripped portion of fiber. As the syringe pump pulled the fiber taut the section was again melted with the torch, thus thinning the fiber in this short section. This procedure was repeated until the tapered region of the fiber was approximately 10 μm thick.

Flow-cell fabrication and assembly

A flow-cell was fabricated for PA experiments using polycarbonate blocks (l =50 mm, w=50 mm, h=6 mm). The final device configuration can be seen in Figure 18. The body of the flow cell was fabricated by milling out a channel (l =30 mm, w=3 mm, h=3 mm) in one of the polycarbonate blocks, Figure 18, this served as the fluid channel. A channel for mounting the resonator (l =10 mm, w=3 mm, h=1.5 mm) was created at one end of the main channel. Another channel for mounting the resonator (l =50 mm, w=3 mm, h=1.5 mm) was also created. A lid was fabricated using a similar polycarbonate block with 6-32 tapped holes aligned to the ends of the main channel, these served as the inlet and outlet ports. Female Luer connectors with 6-32 tapped ends were threaded into the inlet and outlet ports, and sealed with hot glue.

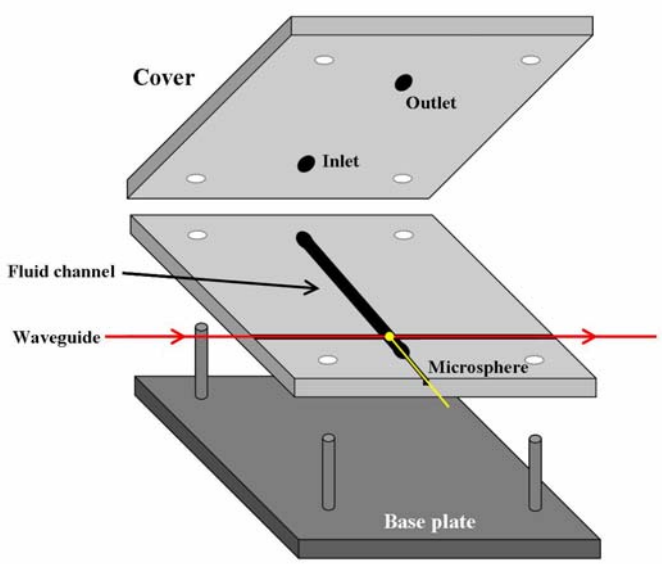

Figure 18: Schematic of flow cell with waveguide and resonator mounted in place.

After fabrication of the waveguide the cross-channel of the flow cell was aligned to the waveguide such that the narrowest region of the waveguide was centered in the main fluidic channel. The flow cell was raised into place using a 3-axis micromanipulator (Newport, Irvine, CA) such that the waveguide was resting on the channel bottom. The waveguide was sealed in place with Kwik-sil silicon elastomer adhesive (World Precision Instruments Inc., Sarasota, FL). After the adhesive was set (approximately 10 minutes) the waveguide was connected to the laser and detection system. The polymer coating on the free ends of the waveguide was stripped using a fiber optic stripper and cleaned with IPA. The ends of the waveguide were then cleaved using a Fitel fiber optic cleaver (Furukawa Electric Co. Ltd, Tokyo) to ensure a clean flat break at the end of the fiber. Having a clean break at the ends of the waveguide was critical to minimize insertion loss due to splicing optical fibers. One end of the waveguide was inserted into mechanical fiber optic splice (Fiber instrument sales, Oriskany, NY) connected to the DFB laser and the other end was inserted into another mechanical fiber optic splice that was connected to the detector. The laser was then turned

on to ensure sufficient laser intensity was being conducted through the waveguide. The gain on the DAQ card and detector was adjusted to optimize signal to noise.

Either a plain glass or silane-coated microsphere was then mounted into the flow cell. A glass microsphere was taped to a 3-axis micrometer and aligned over the waveguide. The microsphere was slowly lowered into contact with the tapered section of the waveguide. Upon contact with the waveguide resonances could be seen in the resulting spectrum on the data acquisition software. The detector gain was adjusted again to optimize the signal intensity. The microsphere was then aligned so that it was centered in the fluid channel and the stalk of the fiber was resting on the bottom of the microsphere channel. The microsphere was then secured into place using Kwik-sil elastomer adhesive as with the waveguide. After the adhesive had set the channels were primed with 50 mM PBS (pH 7.4). Priming the channels ensured that no bubbles would be present in the flow cell when it was sealed.

A 1 mm thick PDMS gasket was created to place between the lid and the body of the flow cell. A 10:1 (wt:wt) mixture of Sylgard (Dow Corning, Midland, MI) and initiator was mixed and degassed the poured over a polished silicon wafer. The mixture was allowed to settle on a flat surface then cured in a 65^{o}C oven for 1 hour. A 50 mm x 50 mm square was then cut out and holes were cut out for the inlet and outlet ports. The gasket was aligned on the lid such that fluid could freely flow through the inlet and outlet. The lid was then aligned using the mounting posts of the flow cell base plate and brought into contact with the body of the flow cell. The lid was sealed to the flow cell using wing nuts to apply even compression across the lid. More PBS was added through the inlet port using a hypodermic needle and syringe to displace remaining air in the flow cell.

Buffer and protein solution were re-circulated using a peristaltic pump that flowed the solutions from 50 ml Pyrex bottles through the flow cell and back into the Pyrex bottle. 3-

way stopcocks were used to switch between buffer and protein solutions. Silastic tubing (1.6 mm O.D., 0.76 mm I.D., Dow Corning) was used to connect the flow cell with the buffer and protein solution containers. Holes were drilled in the lids of the Pyrex bottles for feed and return lines. The tubing was run through the holes and secured into place using hot glue. Male Luer connectors (Harvard Apparatus, Cambridge, MA) were connected to the free ends of the feed and return tubes that were then connected to separate 3-way stopcocks with female Luer connections. One stopcock was used for feed lines from the buffer and protein bottles and one was used for the return lines. A length of tubing was then connected to the stopcock for the feed lines and threaded through the roller of a peristaltic pump. The tubing was then primed with buffer to ensure no bubbles were present in the line. The free end of the feed line was then fitted with a male Luer connector, which was then attached to the inlet connector of the flow cell. A similar piece of tubing was connected to the outlet port of the flow cell and subsequently the stopcock for the return lines. Using this system it was possible to switch between the buffer and protein lines without disturbing the flow field.

Fibronectin adsorption experiments

FN adsorption was measured on resonators modified with DETA, 13F, and PEG as previously described. Adsorption experiments using Fibronectin (FN), M.W. 500kD, from bovine plasma (Sigma-Aldrich, St. Louis, MO) were performed to determine the extent to which it adsorbs onto the silanes chosen for this study, as it is a critical mediator of cell adhesion, focal contact formation and differentiation in many cell types. Solutions of 10 μg/ml, 1 μg/ml, 0.5 μg/ml, and 0.25 μg/ml in PBS (pH 7.4) were used for the WGM experiments. Buffer solution was first flowed through the flow cell at rate of 150 ml/hr to equilibrate the system for at least 15 minutes. After a stable baseline had been achieved the

protein solution was introduced. Data was collected until a stable equilibrium had been achieved and no further adsorption was evident.

Cell Culture

To determine the biological activity of the protein on the various silanes, cell culture experiments were performed on silane-coated coverslips that had been treated with 1 μg/ml of FN in PBS (pH 7.4). Embryonic hippocampal neurons and skeletal myoblasts were used. Cells were plated on FN coated coverslips and allowed to adhere for 1 hour. After plating 3 ml of culture media was added and cultures were maintained in a water-jacketed incubator at 37^{o}C and 5% CO_2 for seven days. Phase-contrast microscopy images were taken during the course of the culture to document the morphology of the cells and, in the case of skeletal myoblasts, the differentiation of the cells into functional myotubes.

Embryonic Skeletal Muscle

Skeletal muscle was dissected from the hind limb thighs of a rat fetus at embryonic day 18 (Charles River Laboratories, Wilmington, MA) according to previously published protocol [37] with some modification. Tissue samples were collected in a sterile 15-ml centrifuge tube containing 1 ml of calcium and magnesium free phosphate buffered saline (PBS). Tissue samples were enzymatically disassociated using 3 ml of 0.05% of trypsin–EDTA (Invitrogen, Carlsbad, CA) solution for 60 min in a 37°C water bath with agitation of 100 rpm. After 60 min, the trypsin solution was removed and 6 ml of L15 media (Invitrogen, Carlsbad, CA) containing 10% fetal bovine serum (FBS) was added to terminate the trypsin action. The tissue was then mechanically triturated using a sterile narrow bore Pasteur pipette, allowed to settle for 3 min, and transferred to a 15-ml centrifuge tube. This was repeated three times. The dissociated tissue was then centrifuged at 300g for 10 minutes at

4°C on 6 ml of a 4% (wt/vol) cushion of bovine serum albumin (BSA). The pellet was resuspended in 10 ml L15 + 10% FBS and plated in uncoated 100-mm Petri dishes for 20–30 min depending on the amount of tissue, to allow contaminating fibroblasts to settle out. After 20–30 minutes the supernatant was layered on 6 ml of a 4% BSA cushion, and centrifuged at 300g for 10 min at 4°C. The pellet was resuspended in 1.5 ml of medium. Purified myocytes were plated at a density of 500–800 cells per square millimeter onto the cantilevers. Myocytes were allowed to attach for 1 hour after which time 3 ml of culture medium (Neurobasal media containing B-27 [Invitrogen, Carlsbad, CA], Glutamax [Invitrogen, Carlsbad, CA], and Pencillin/Streptavidin) was added. Cultures were maintained in a 5% CO_2 incubator (relative humidity, 85%). Culture medium was exchanged every 4 days.

Embryonic hippocampal neuron

Rat pups at embryonic day 18 dissected from timed pregnant rats that were euthanized using CO_2 asphyxiation. Embryos were collected in ice cold Hibernate E/ B27/ Glutamax™/ Antibiotic-Antimycotic. The hippocampi were isolated from the embryonic brain and collected in tube containing 1ml of Hibernate E/ B27/ Glutamax™/ Antibiotic-Antimycotic. The embryonic hippocampal neurons were obtained by triturating the tissue using a Pasteur pipette. The 1ml cell suspension was layered over a 4 ml step gradient (Optipep diluted 1:1 (vol:vol) with Hibernate E/ GlutaMAX™ / antibiotic-antimycotic/ B27 and then made to 15%, 20%, 25% and 35% (vol:vol) in Hibernate E/ GlutaMAX™/ antibiotic-antimycotic/ B27) followed by centrifugation for 15 min, using 800g, at 4°C. This additional step helped to remove the debris arise during dissection from the damaged cells. After centrifugation, one strong band of cells was obtained at the top. The pyramidal hippocampal neurons constituted this band with large somas. The cells were resuspended in culture medium (Neurobasal / B27

/ Glutamax™ / Antibiotic-antimycotic) and plated at a density of 75 cells/mm^2 [57, 64-68]. Half of the medium was changed after every 3-4 days.

Live-dead assay

A Live-dead assay (Invitrogen, Carlsbad, CA) was performed at day 7 to determine the amount of living versus dead cells on the coverslips. Briefly, a solution containing 5 μM of casein and 20 μM ethidium bromide was prepared in 50 mM PBS. Cells were washed 3x in PBS and incubated in the live-dead solution for 30 minutes. Random images were taken on an epifluorescence microscope (Zeiss) and the number of live cells (green fluorescing) and dead cells (red-fluorescing) were counted.

GO activity assay

To further probe the extent of denaturation of proteins on silane surfaces the enzymatic protein glucose oxidase (GO), M.W, 160kD, was used in adsorption experiments. By using GO it was possible to measure the enzymatic activity of the protein in solution versus when it was adsorbed to a surface. GO was purchased from Sigma (St. Louis, MO). Solutions of 100 μg/ml and 10 μg/ml were prepared in PBS (pH 7.4). Adsorption of GO was measured on DETA, PEG, 13F and glass as described in previous sections. Enzymatic activity was measured using the Amplex Red glucose oxidase assay (Invitrogen, Carlsbad, CA). Resonators were soaked in 100 μg/ml and 10 μg/ml GO solutions for two hours in a round-bottomed 96-well ELISA plate. After two hours the resonators were washed 3x with PBS. The resonators were then transferred to a new 96-well plate and allowed to soak in 100 μl of PBS for 2 hours to allow any reversibly bound GO to desorb from the surface of the resonator. After 2 hours 50 μl of the PBS was transferred to another well in the 96-well plate. A solution containing 0.1 mM Amplex red, 0.1 mM Glucose, and 0.5 U/ml horseradish

peroxidase (HRP) was prepared. 50 μl of the Amplex red solution was added to each of the wells containing the resonators and soak buffer. The reaction mixtures were then placed in a Synergy HT multi-mode microplate reader (Bio-Tek, Winooski, VT) and the absorbance at 530 nm was read for each well at 1-minute intervals. Standard dilutions of GO of 100 ng/ml, 50 ng/ml, 10 ng/ml, 5 ng/ml, 1 ng/ml, and 0.5 ng/ml were prepared and reacted with Amplex red reagent for all experiments.

Data analysis

Data analysis software was written using the Python programming language (Appendix A). The binary file from an experiment was loaded into the software and the traces were reconstructed from the raw data. During the reconstruction the change in position of the trace was plotted as the absolute wavelength shift versus time. The trace from one resonance was chosen for further data analysis. Continuous traces from resonances with the lowest FWHM value were chosen for analysis. A linear baseline subtraction was applied to correct for baseline drift. Surface concentrations were then calculated using Equation 1 from Vollmer and Arnold[85], where σ_s is the average surface concentration (in g/m^2), λ_r is the nominal wavelength of the resonance (1310 nm), $\Delta\lambda_r$ is the fractional wavelength shift of the resonance, n_s is the refractive index of the sphere (1.46), n_m is the refractive index of the medium surrounding the sphere (1.3357), α_{ex} is the excess polarizability of the protein molecule (0.184 cm^3/g), ε_o is the permittivity of free space, and R is the radius of the sphere. Sphere radii were measured from images taken by brightfield microscopy. Multiple traces for each set of experimental conditions were averaged using MatLab data analysis software (Appendix B) and plotted using Excel.

Results and discussion

Fibronectin adsorption onto alkylsilane monolayers

WGM measurements

Adsorption of FN on DETA, 13F, and PEG coated microspheres was measured using the WGM biosensor. Figure 19 shows averaged sensograms for FN adsorption onto silane coated resonators (A,B,C) and the isotherms plotting the measured saturation values versus solution concentration (D). Saturation values at the higher concentrations are in excellent agreement with previously published results [50, 51, 86] for FN adsorption measured by SPR onto alkanethiol SAMS with similar surface chemistries and contact angles. It can be seen that for DETA, 13F, and PEG saturation at 10 μg/ml solution concentration occurs at ~1.9 ng/mm^2, 2.1 ng/mm^2, and 0.5 ng/mm^2 respectively. However, the amount of protein adsorption measured by our system at lower concentrations is significantly greater than that of previously published results. At solution concentrations of 1 μg/ml, FN adsorption onto DETA and 13F is measured to be ~1.5 ng/mm^2 and ~1.4 ng/mm^2 respectively. Values reported for amine-terminated and methyl-terminated SAMS show saturation values of ~0.25 ng/mm^2. Furthermore, adsorption measurements performed at solution concentrations as low as 0.25 ng/ml showed saturation values for DETA and 13F (~1.0 ng/mm^2) that are higher than those previously published on similar surfaces, while those measure for PEG modified microspheres are below 0.1 ng/mm^2. These results indicate that, while the limiting surface coverage on silane monolayers is comparable to those of alkanethiol SAMS, FN has a higher affinity for silane SAMs. SPR measurements by Michael et al. were performed under conditions that were shown not to be mass transport limited as with our system (data not shown). Thus, these measurements likely reflect the true affinity of FN for the SAMs studied here.

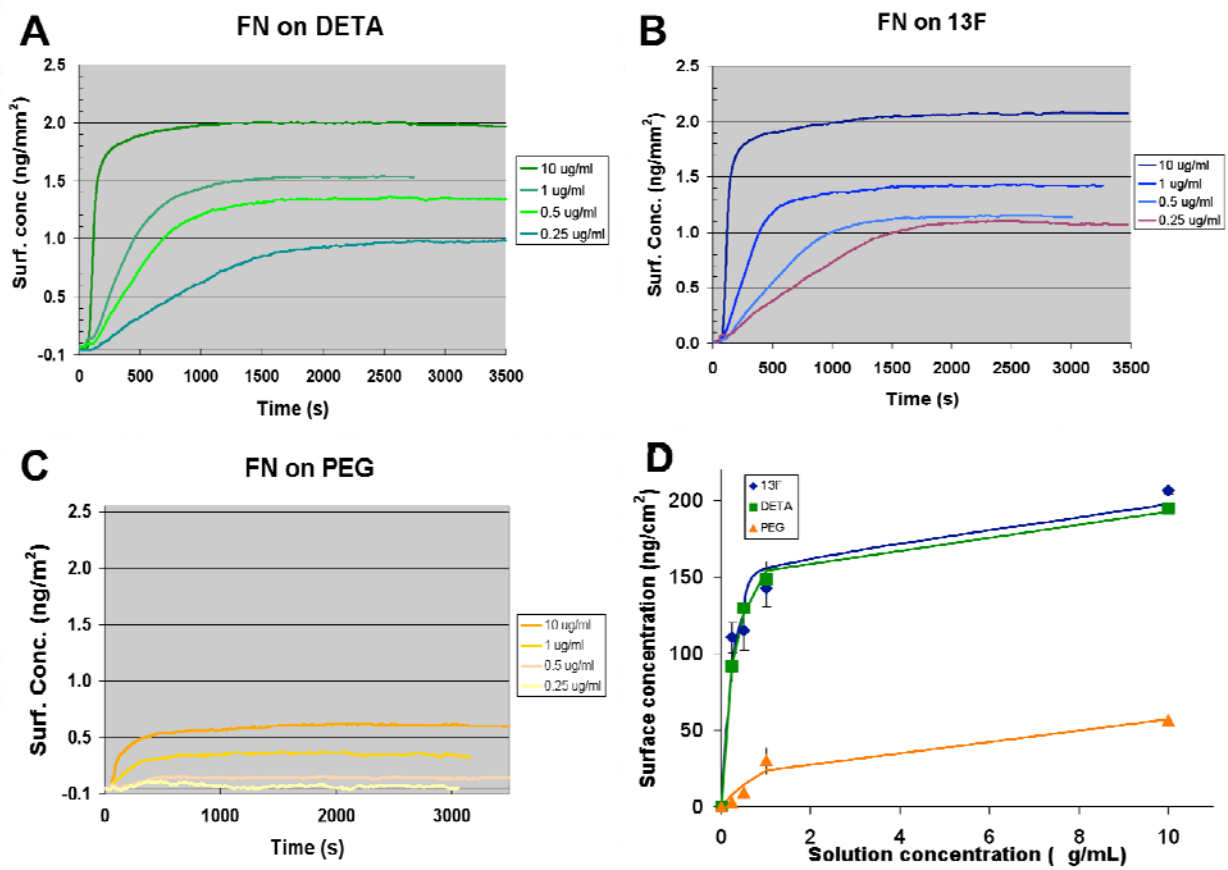

Figure 19: Adsorption of FN onto DETA, 13F, and PEG coated microspheres. Panels A,B,C shows adsorption sensograms of FN acquired with WGM biosensor on DETA, 13F and PEG microspheres respectively. Panel D shows saturation values of FN versus solution concentration on DETA, 13F, and PEG.

Cell culture on fibronectin adsorbed to alkylsilane monolayers

Cell culture experiments were performed to determine the biological activity of FN on the silanes used for this study. Embryonic hippocampal neurons (EHipp) and skeletal myocytes (ESM) were plated onto DETA, 13F, and PEG substrates that had been coated with 1 μg/ml of FN in PBS. The cultures were maintained for 7 days before live/dead assay was performed to determine the extent of cell survival. Figure 20 shows results for EHipps cultured on silanes. Panels A-C and D-F show phase contrast images of cells at 1 day and 7 days in culture respectively. It can be seen from these pictures that EHipp cells survive on DETA significantly more on DETA than on 13F or PEG surfaces. Table 2 shows the results

for live versus dead EHipp cells counted on the various surfaces. The lack of survival of cells on PEG surfaces can be attributed to a low amount of adsorbed FN. However, on the amount of adsorbed protein measured on 13F surfaces is comparable to that of DETA. Thus, the lack of cell is likely due to decreased bioactivity of FN on the strongly hydrophobic 13F. To further probe the activity of FN on the silanes used, embryonic skeletal myocytes (ESM) were cultured on substrates coated with FN.

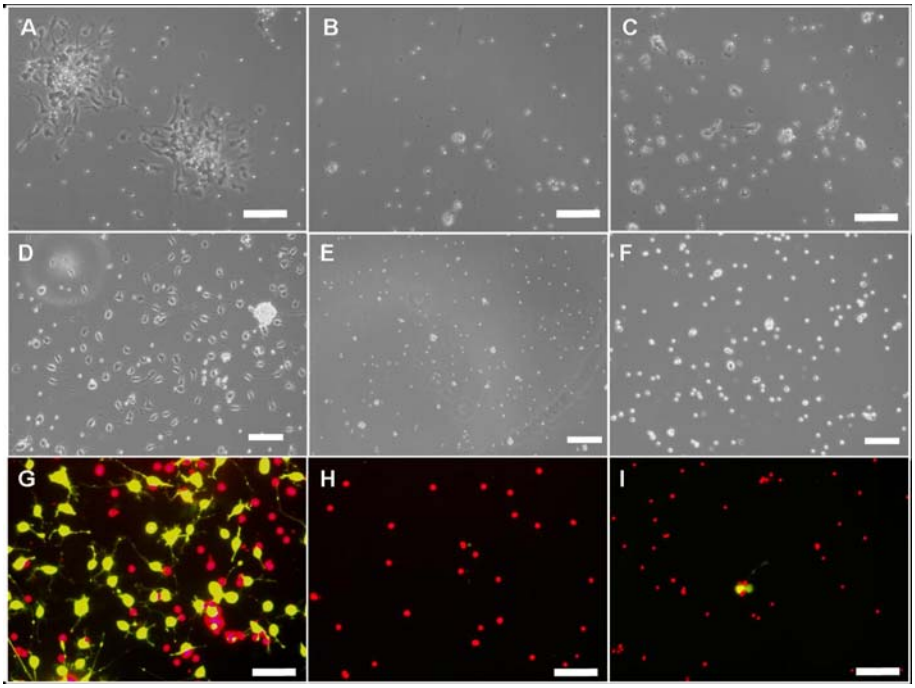

Figure 20: Embryonic hippocampal cells cultured on DETA (A-D-G), 13F (B-E-H), and PEG (C-F-I)coated substrates. All scale bars represent 100 μm.

Table 2: Live/dead assay for embryonic hippocampal cells on silanes. All values are units of cells/mm^2.

	DETA	13F	PEG
Live	212±102	1±3	4±5
Dead	340±75	218±81	265±86

Skeletal myocytes are precursor cells that fuse and differentiate into contractile myotubes. This differentiation is mediated by, among other factors, the interaction of the α5β1 integrin receptors on the surface of the myocytes with the cell binding domain of the FN molecule [87, 88]. Without this interaction the formation of myotubes does not take place. Figure 21 shows results from cell culture experiments a 1 day and 7 days after plating. At day 1 in culture dense myoctye adhesion can be seen on DETA coverslips coated with FN, while significantly less adhered cells can be seen on 13F and PEG. Furthermore, on DETA coverslips myocytes can be seen to be taking on the spindle shaped morphology that is characteristic of activation of α5β1 integrin receptors. Cells on 13F and PEG maintain an unelongated morphology. After 7 days in culture myocytes on DETA have begun to form long cylindrical myotubes. The myotubes could be seen to spontaneously twitch indicating that they were in fact functional myotubes. After 7 days no myotube formation had occurred on either 13F or PEG coated substrates. Table 3 show the resulting cells counts using the live/dead assay. On DETA 178±43 live cells/mm^2 and 35±13 myotubes/mm^2 were counted, while 63±66 dead cells/mm^2 were counted. On PEG no live cells or myotubes were observe, but 111±59 dead cells/mm^2 were counted. The results for 13F showed that while a significant number of cells survived 50±32 cells/mm^2 no myotubes formed and the number of dead cells 18±14 cells/mm^2 was actually less than that of PEG or DETA. The fact that so many cells survived on the 13F substrates indicates that there is enough protein adsorbed to the surface to promote adhesion, however, the lack of myotube formation indicates that FN has a reduced biological activity and does not activate the α5β1 integrin signaling pathways necessary for myotube differentiation.

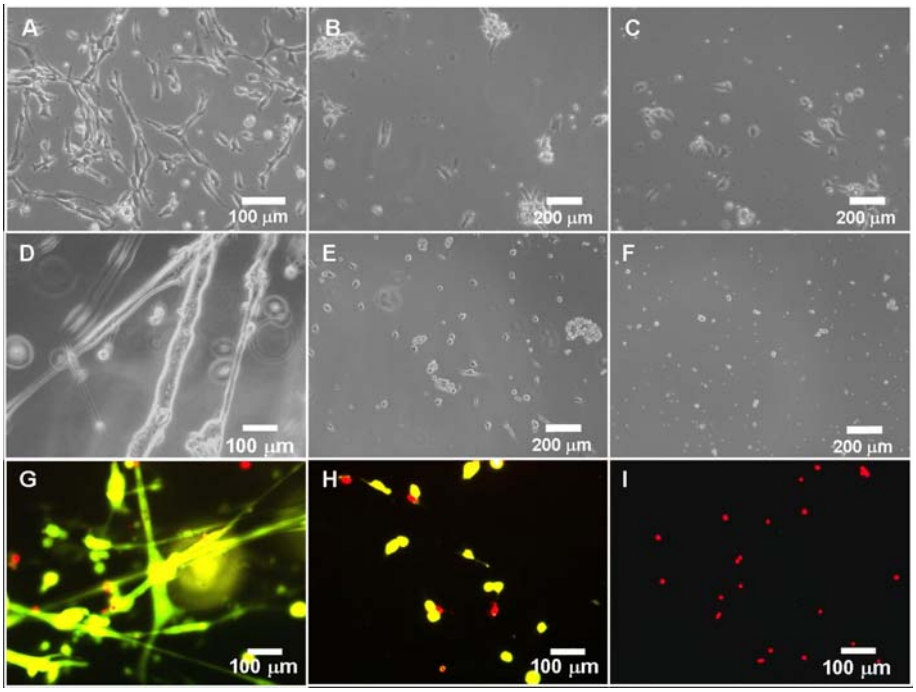

Figure 21: Embryonic skeletal muscle cultured on DETA (A-D-G), 13F (B-E-H), and PEG (C-F-I)coated substrates.

Table 3: Live/dead assay for embryonic skeletal muscle on silanes. All values are units of cells/mm^2.

	DETA	13F	PEG
Myotubes	35±13	0±0	0±0
Live	178±43	50±32	0±0
Dead	63±66	18±14	111±59

Glucose oxidase adsorption onto alkylsilane monolayers

To further characterize the amount of denaturation occurring during adsorption of proteins onto silane SAMs, the enzymatic protein glucose oxidase (GO) was studied. By

using an enzymatic protein it was possible to use its catalytic activity as an indicator of its conformation on the surface. Figure 22 shows WGM measurements of GO adsorption onto silane-modified resonators from solution concentrations of 100 μg/ml and 10 μg/ml respectively. The data shown here vary qualitatively from the FN adsorption data in that there are significant differences in the amount of protein adsorbed on the DETA and 13F surfaces. For the 100 μg/ml data the saturation values can be seen to vary on the surfaces according to the trend DETA > 13F > Glass > PEG, while the trend for the 10 μg/ml data follows the trend 13F > DETA > Glass > PEG. It is as of yet unclear why this would be the case. One possible interpretation is that on DETA, GO can form multiple layers at high solution concentrations. As can be seen in Figure 22a, the adsorption curve for DETA starts at an initially high rate of adsorption then begins to slow at around 250 seconds. At ~400 seconds, however, the rate of adsorption begins to increase again. The graph gives the impression of two super-imposed adsorption curves that would indicate multilayer formation. It is interesting to note that this behavior is only seen on DETA, which at physiological pH is positively charged. This may be indicative of an electrostatically mediated adsorption process as GO has a net negative charge at physiological pH. It should be noted, however, that the error bars on the DETA isotherm are quite significant. This is most likely due to the variability inherent in the DETA coating procedure and more replicates of this set of conditions are required. Saturation values for each set of conditions were tabulated for subsequent use in the activity calculations.

The enzymatic activity of GO on the resonators was measured using the Amplex-red glucose oxidase assay[89] (Invitrogen, Carlsbad, CA). In this assay the oxidation of glucose and O_2 to gluconate and H_2O_2 is coupled to the oxidation of the Amplex reagent by HRP to the reaction product resorufin, which has peak absorbance at 530 nm. Silane coated

resonators with adsorbed GO (n=6 for all conditions) were immersed in the Amplex reagent and incubated at room temperature at room along with glucose standards for 30 minutes. The absorbance at 530 nm was monitored over 30 minutes. Figure 23 shows the change in absorbance over time for protein adsorbed on silanes from solution concentrations of 10 μg/ml and 100 μg/ml. It can be seen that for the 100 μg/ml data, the protein adsorbed to the DETA, 13F, and glass still retain significant catalytic activity and follows the trend DETA = 13F = glass >> PEG. The activity does not follow the same trend as the saturation values, as there is no statistically significant difference between them. Very little activity is seen for GO adsorbed to PEG. The activity assay for protein adsorbed from solutions of 10 μg/ml, however, shows a markedly different trend, DETA > 13F = glass = PEG. For these data it can be seen that while significant activity remains for GO adsorbed to DETA, the activity on 13F and glass is drastically reduced and is comparable to that of PEG. These results also do not follow the adsorption trend. For this reason it was necessary to normalize the activity data to the saturation values measured by WGM. The normalized activity was used as an indicator of denaturation.

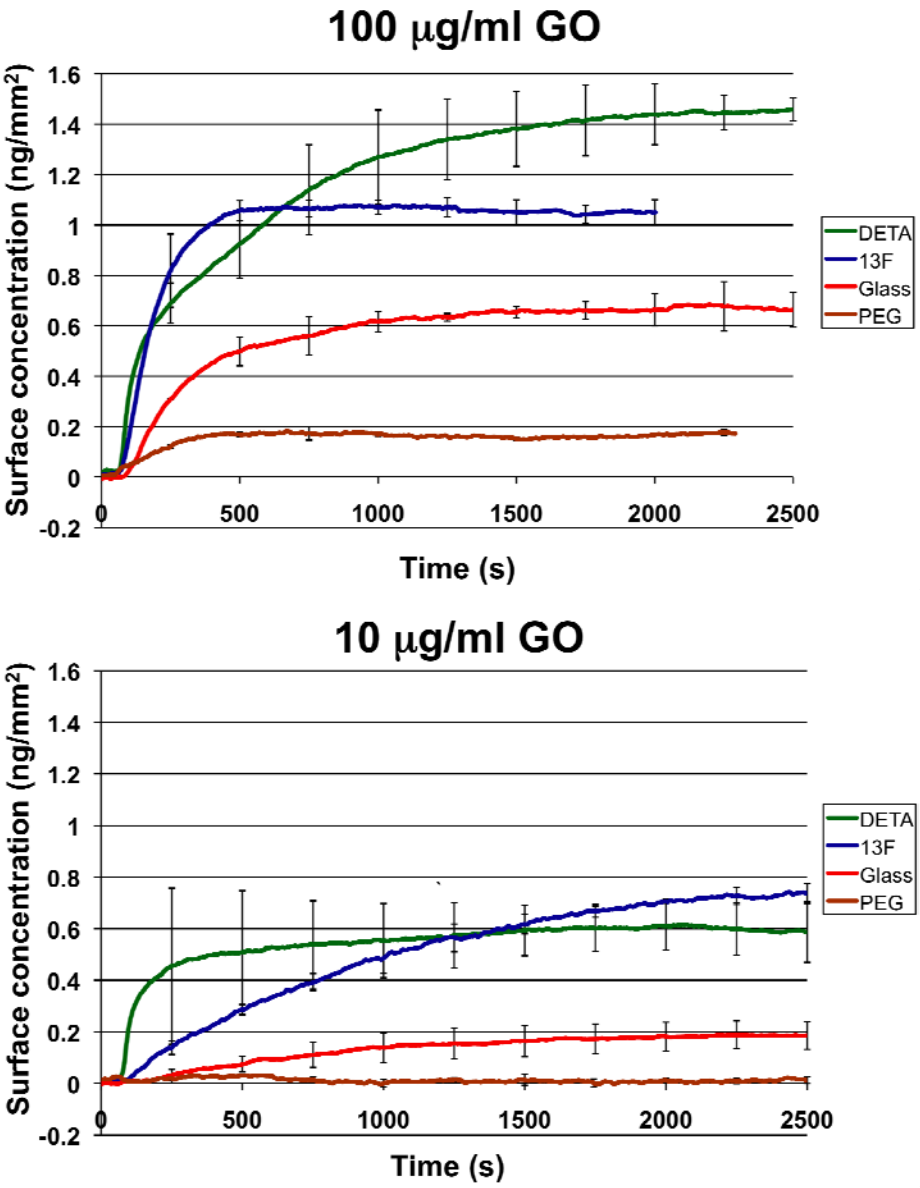

Figure 22: Glucose oxidase adsorption onto DETA, 13F, Glass, and PEG resonators from solutions of 100 µg/ml (top) and 10 µg/ml (bottom).

In order to calculate the normalized activity a standard curve was first created using the glucose standards run with each experiment. A linear regression line was fitted to the absorbance curves to calculate the rate of change in absorbance for each set of conditions. The slope of the line is the activity of the GO (units = min^{-1}). However as can be seen in Figure 23, the absorbance curves for conditions with higher activity were not linear. This is

most likely due to the fact that the protein was localized to the resonator rather than being in solution. This means that the rate of change in absorbance was limited by the diffusion of glucose to the resonator surface. As the GO oxidizes the glucose around the resonator it becomes depleted, and since the catalytic rate of glucose oxidation by GO is limited only by the diffusion rate of glucose into the active site of the protein, depletion of the substrate around the resonator can slow the reaction rate. Another possible source of this artifact is a competing reaction between the HRP, which oxidizes the Amplex reagent, and excess H_2O_2 produced by GO which can oxidize the reaction product resorufin to resazurin[89], which has different absorbance characteristics. For these reason the linear regression line was fitted to the initial slope of the absorbance curves to capture the true catalytic activity of the GO adsorbed to the resonators. Figure 24 shows the standard curve (Figure 24a) and representative linear regression fit (Figure 24b).

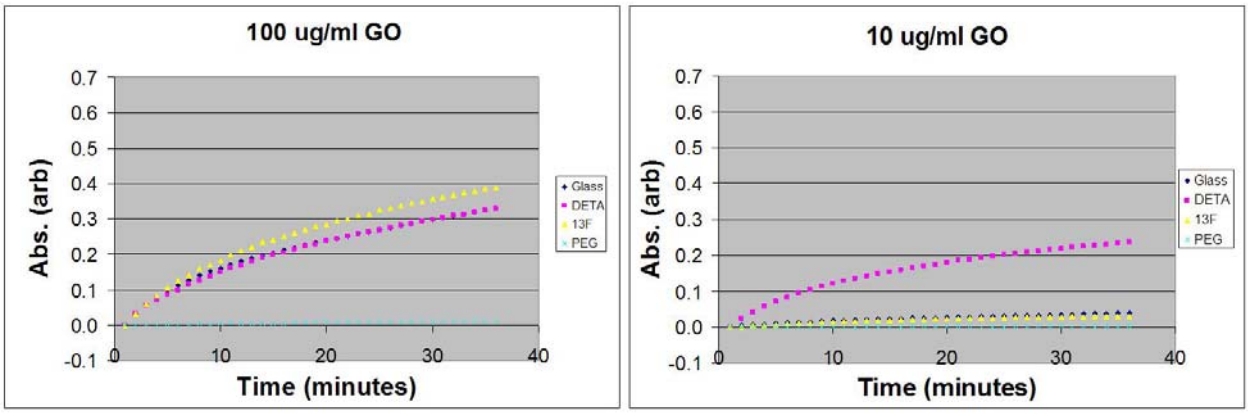

Figure 23: Kinetic measurements of enzymatic activity of protein adsorbed onto silane modified resonators from solutions of from solutions of 100 μg/ml (left) and 10 μg/ml (right).

To calculate the normalized activity, the slope of the regression line for the absorbance curve was used to calculate the corresponding solution concentration from the standard curve, and subsequently the mass of protein. The average mass of protein adsorbed onto the

resonator was calculated using the surface concentration measured by the WGM biosensor multiplied by the surface area of the resonator. The mass calculated from the activity assay was then divided by the average mass measured by WGM to yield the normalized activity. Figure 24c-d shows the resulting normalized activity data for GO adsorbed to the silanes used. For GO adsorbed from solutions of 100 μg/ml it can be seen that the normalized activity follows the trend glass > 13F = DETA > PEG. These results show that the highest activity, and therefore the least amount of denaturation, occurs on glass at this concentration, while the lowest activity is found, surprisingly on PEG. The normalized activities of GO on DETA and 13F were found not to be statistically different. The trend found for GO adsorbed from solutions of 10 μg/ml, however, show a different trend, DETA > 13F = glass = PEG. These results indicate that at different solution concentrations the activity of GO on DETA, while reduced, remains the same (normalized activity of GO on DETA not significantly different at $p = 0.05$), while the activity on glass and 13F is reduced at lower concentrations. One possible explanation for this is would be due to increased spreading of the proteins (i.e. denaturation) on these surfaces at lower solution concentrations. Work by van der Veen et al. [90] and Michael et al [52] have shown that there is evidence for protein spreading on glass and strongly hydrophilic surfaces respectively. It has been hypothesized by van der Veen et al. that at higher solution concentrations, protein adsorbs to a surface more quickly creating a "crowded" monolayer that prevents neighboring protein molecules from denaturing due to lateral interactions. At lower solution concentrations, however, adsorption occurs more slowly resulting in a less densely packed layer, in which individual protein molecules can denature and spread more resulting in a lower saturating surface concentration. The work by Michael et al. further shows reduced spreading and denaturation on amine-terminated alkanethiol surfaces. The activity values for GO on PEG, however, are contrary to what is

expected for a surface of this type. On PEG one would expect to see very little denaturation. This result may be artifactual. Due to the very small amounts of protein adsorbed to the PEG, the activity measured is exceedingly low (approximately the same as blank samples) which could result in significant error in the measurement. Thus, since the measured activity is so close to zero, the resulting normalized activities are artifactually low. The difference in activity between the 100 μg/m and 10 μg/ml data on DETA is somewhat surprising, however. While the two values are not statistically different ($p>0.05$), one would not expect to see a decrease in enzyme activity on DETA, as previous experiments have shown aminated surfaces to be non-denaturing. Also, experiments with Fn in the previous section have shown that DETA allows integrin-mediated differentiation of skeletal myocytes, indicating biologically active Fn on the surface. One possible interpretation is the previously mentioned multilayer formation of GO on DETA. It is possible that if GO forms multiple layers at higher concentrations, the diffusion of the compounds in the activity assay may be hindered and limit the rate of substrate turnover. Further experiments are required to confirm this hypothesis, however.

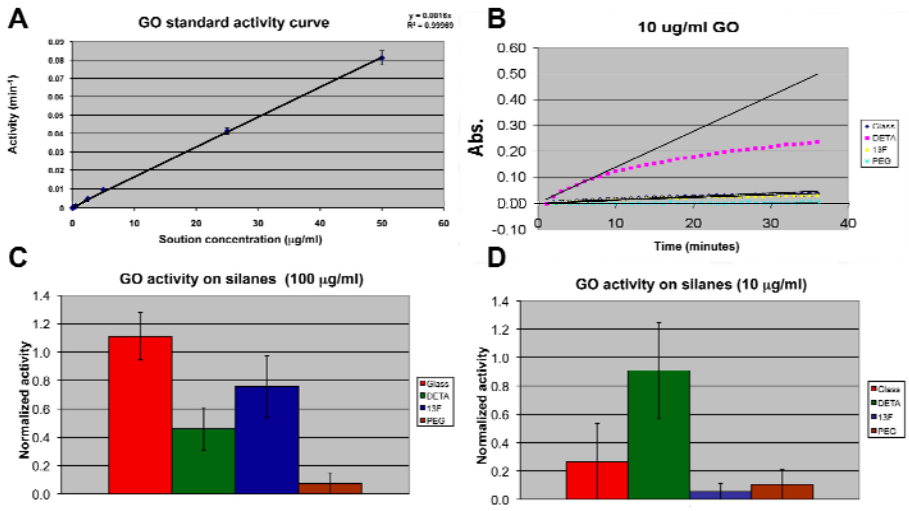

Figure 24: Enzymatic activity of GO (10 μg/ml) adsorbed onto silane surfaces normalized by adsorbed surface concentration.

Conclusions

The data presented in this chapter outline the use of a novel WGM biosensor and flow cell to characterize the adsorption of protein onto silane SAMs. Adsorption of the extracellular matrix protein Fibronectin (FN) and the enzymatic protein glucose oxidase (GO) were studied at varying concentrations and flow conditions. For FN it was shown that comparable amounts of protein adsorb to DETA monolayers compared to 13F monolayers over a range of solution concentrations. In contrast to previously published results of FN adsorption onto alkanethiol SAM, however, significantly more adsorption was measured on alkylsilane surfaces at lower solution concentrations [52]. Measurements on PEG monolayers show a drastic reduction in the amount of adsorbed protein, as expected. Cell culture experiments showed that despite the comparable amounts of protein adsorbed to DETA and 13F, the biological activity of the protein was not retained on the highly hydrophobic 13F indicating that the protein was in fact denatured. Similarly, cells tended not to grow on PEG surfaces due to the lack of adsorbed protein. To further probe the extent of denaturation caused by the various surfaces the enzymatic protein GO was studied. GO catalyzes the oxidation of beta-D-glucose and O_2 into D-glucono-1,5-lactone and H_2O_2, which can then be detected by the horseradish peroxidase mediated oxidation of the colorimetric substrate Amplex-red into resorufin [89]. The catalytic activity of GO adsorbed onto silane surfaces was measured by this reaction and it was found that significant denaturation does indeed occur on 13F, while significant catalytic activity remains on GO adsorbed to DETA. Furthermore, denaturation of GO adsorbed to glass was found to be strongly concentration dependent.

CHAPTER FOUR: DEVELOPMENT OF A NOVEL BIO-MEMS DEVICE FOR FUNCTIONAL STUDIES OF NEUROMUSCULAR FUNCTION AND DEVELOPMENT

Introduction

Microelectro-mechanical systems (MEMS) have received a great deal of attention in recent years due to their promise for miniaturizing systems for a variety of applications. One particularly alluring facet of MEMS technologies is the possibility of coupling solid state devices with biological components (Bio-MEMS) such as biomolecules, cells, and tissues for creating novel bio-analytical systems. To date biological components have been incorporated into MEMS devices to create cell-based sensors and assays [10, 20, 44, 91-93], motors and actuators [5, 24, 94], and pumps [95]. Bio-MEMS technologies present a unique opportunity to study fundamental biological processes at a level unrealized with previous methods. The capability to miniaturize analytical systems enables researchers to perform multiple experiments in parallel and with a high degree of control over experimental variables. This capacity will allow a high throughput approach for studying a wide variety of problems in biology.

One tissue of particular interest with respect to a variety of diseases is skeletal muscle. Diseases affect skeletal muscle in different ways. Some diseases, such as amyotrophic lateral sclerosis (ALS), affect the stimulating inputs from the neuromuscular junction [96]. Other diseases affect the muscle directly such as muscular dystrophy and muscular atrophy [97], which cause deterioration of the muscles' ability to generate force. Thus, it is advantageous to have a system that allows the real-time interrogation of the physiological properties of muscle as well as the controlled addition of exogenous factors for comparative experimentation. However, it is first necessary to be able to apply the measurements to statistical analysis with regard to physiological factors such as peak stress generated, time to peak stress, the time needed for the muscle to relax to half of the peak stress, and the average

rate of stress generation [98, 99]. All of these factors give information about the condition of the muscle and can be compared to published values.

The present study outlines a novel method for performing real-time quantitative measurements of the physiological properties of cultured skeletal muscle using a Bio-MEMS device. Stresses generated by myotubes were measured using a modified Stoney's equation, which quantifies stresses generated by a thin film on a cantilever with known physical properties [100]. By this method it has been shown that it is possible to quantitatively measure stress on cantilevers that are in agreement with values previously published in the literature for cultured skeletal muscle. Furthermore, a method for selectively seeding and coculturing neuronal and muscle cells on these devices using microfluidic chambers was developed. By this method it was possible to create a model for studying neuromusclular junction development and function. This work validates the use of this system as a foundation for a high-throughput Bio-MEMS device.

Materials and methods

Cantilever Fabrication

The layout for the cantilevers was generated using AutoCAD 2004 (Figure 25). The patterns were written to chrome coated 4-5 inch soda-lime glass masks for front and back side photolithography. Cantilevers were fabricated from 6 inch double-sided polished silicon-on-insulator (SOI) wafers with a 5 μm crystalline silicon layer (front side) and a 500 μm silicon dioxide layer (back side). The front side was primed with a 10 nm layer of hexamethyldisilazane (HMDS) to promote resist adhesion. A 5 μm layer of the photoresist AZ 5214 E (Clariant, Muttenz, Switzerland) was spun onto the device layer followed by softbake, alignment, exposure, and development. The device layer was etched using the deep reactive ion etch (DRIE) process at a rate of 2 μm/min. Resist was stripped and a 0.5 μm

thick layer of silicon dioxide was deposited via Plasma Enhanced Chemical Vapor Deposition (PECVD) process to protect the device layer during subsequent processing. The wafer was then flipped over and was primed with a 100 Å layer of HMDS and spun with 4.15 μm layer of AZ 9245 photoresist (Clariant, Muttenz, Switzerland). Coating was followed by softbake, front-back alignment, development, and DRIE etch at 4 μm/min until the bulk of the back side had been etched through leaving only the buried native oxide layer. The devices were then immersed in a buffered HF dip to remove the buried native oxide layer as well as the protective silicon dioxide that had been deposited onto the device layer. Individual devices were separated by breaking connecting tabs that were incorporated into the device design. Cantilever dimensions were measured using a JEOL 6400 scanning electron microscope (SEM) at a take-off angle of 50^{o} off normal.

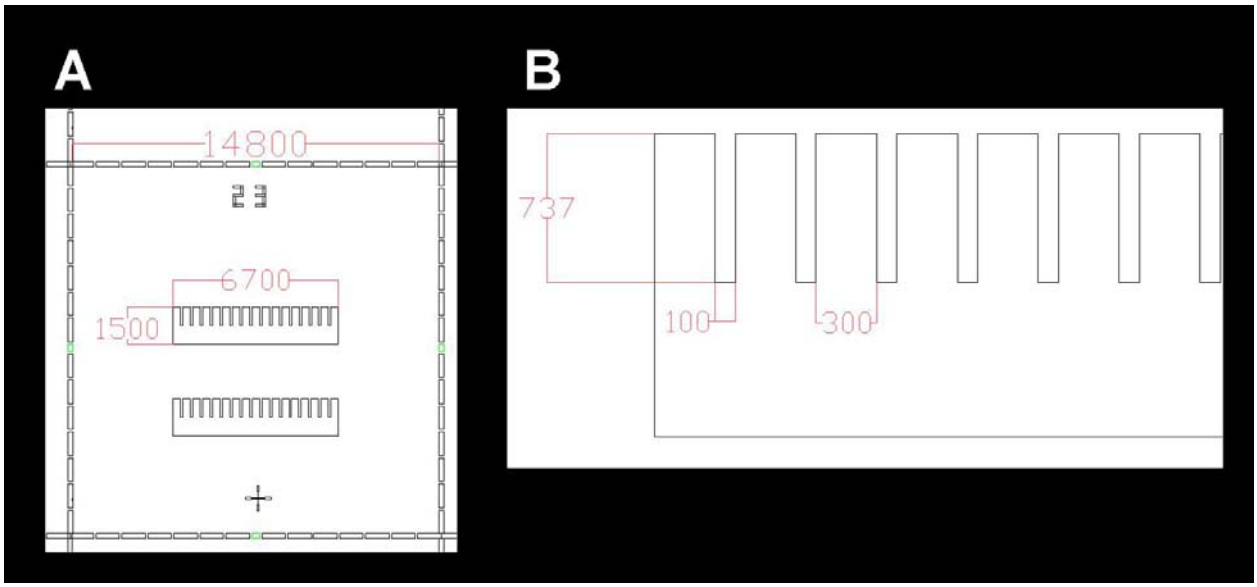

Figure 25: Layout of cantilever devices generated in AutoCAD, all units are shown in microns. A) The layout of a single die. The outer boundaries, delimited by dashed lines which formed connecting tabs allowing the die to be easily separated, were designed to be 14.8 mm x 14.8 mm. B) Close-up view of a partial cantilever row showing cantilever dimensions (737 μm x 100 μm) and spacing between cantilevers (300 μm).

PEG-DETA surface modification

The silicon cantilevers were coated with the PEG-terminated alkylsilane, 2-[Methoxypoly(ethyleneoxy)propyl]trimethoxysilane (Gelest, Tullytown, PA) to prevent nonspecific cell adhesion and differentiation. Cantilevers were cleaned in serial acid baths of concentrated HCl in methanol (1:1 dilution) for 30 minutes and concentrated H_2SO_4 for 1 hr, followed by 30 minutes in boiling de-ionized water. Cleaned cantilevers were dried overnight in an 120^{o}C oven. Surface modification with PEG silane was performed according to the protocol outlined in Chapter 2. Briefly, the cantilevers were incubated in 0.1% (vol:vol) solution of PEG silane in toluene for 60 minutes at room temperature, followed by washing in fresh toluene, 2x wash in 95% ethanol, and 1x wash in deionized water. They were then dried under a stream of dry nitrogen. Cantilevers were then patterned using Deep UV photolithography and backfilled with DETA, also as described in chapter 2. X-ray photoelectron spectroscopy (XPS) and contact angle measurements were used to characterize the surface coating.

Microfluidic chamber fabrication

Chamber molds fabricated by SU-8 photolithography

Fabrication of chamber molds was performed in a class 10,000 clean room. Square pieces of a single-side polished silicon <100> wafer (65mm x65mm) were rinsed with methanol, isopropanol, and acetone respectively, and then blown dry under filtered compressed air. The final chamber geometry was achieved using three layers of SU-8 patterned with different photomasks. Five such molds were fabricated onto each wafer. The wafer pieces were placed on a spin coating device and secured with vacuum. Approximately 3 ml of the negative photoresist SU-8 100 (Microchem, Newton MA) was deposited on the

center of the wafer piece. SU-8 100 is a negative resist (one that polymerizes under exposure to light energy) and was chosen due to its viscosity and ability to form high aspect ration structures up to 200 μm thick. Any bubbles present in the resist were manually removed prior to spinning. The wafer and resist were then spun at three sequentially increasing rates of 500 rpm (10 seconds with a ramp of 500 rpm/second), 1500 rpm (30 seconds with a ramp of 500 rpm/second), and 2750 rpm (30 seconds with a ramp of 500 rpm/second). By increasing the spin rate over multiple steps it was possible to ensure an even coating of the resist. The final spin resulted in an SU-8 layer that was ~100 μm thick. The wafer pieces were removed from the spin coater and place on a level surface to allow the SU-8 resist to relax. Allowing the resist to relax was critical to ensure that the layer was flat and even with no edge beads. The resist was then baked on a 65^{o}C hotplate for 10 minutes followed by a bake in a 100^{o}C oven for 45 minutes. This "prebake" step drives off excess solvent in the resist and hardens it for photolithography. The wafer pieces were then place in a Karl Suss MJB3-1 mask aligner (Garching, Germany) and aligned under a photomask containing the pattern for the first layer of the micro-chamber mold (figure 16a). The resist was then exposed to light from a UV lamp for 30 seconds (200 mJ/second). The patterned wafer was then placed on a 65^{o}C hotplate for 1 minute followed by baking in a 100^{o}C oven for 14 minutes to develop the exposed resist. This process was repeated 2 more times to fabricate the second and third layers of the mold (Figure 26a-b respectively). Fabricating molds with greater than three layers yielded poor resolution for the smaller structures in the patterns. After the third layer was developed, the wafer was then immersed in SU-8 developer (Microchem, Newton MA) for 45 minutes or until all of the unexposed resist had dissolved away. The developed mold was rinsed with excess IPA. After rinsing with IPA any undeveloped resist remaining on the surface was evident as a white film. If undeveloped

resist was evident the wafer was returned to the SU-8 developer until it was completely removed. After development was complete the wafer was rinsed again with IPA and was blown dry with filtered compressed air.

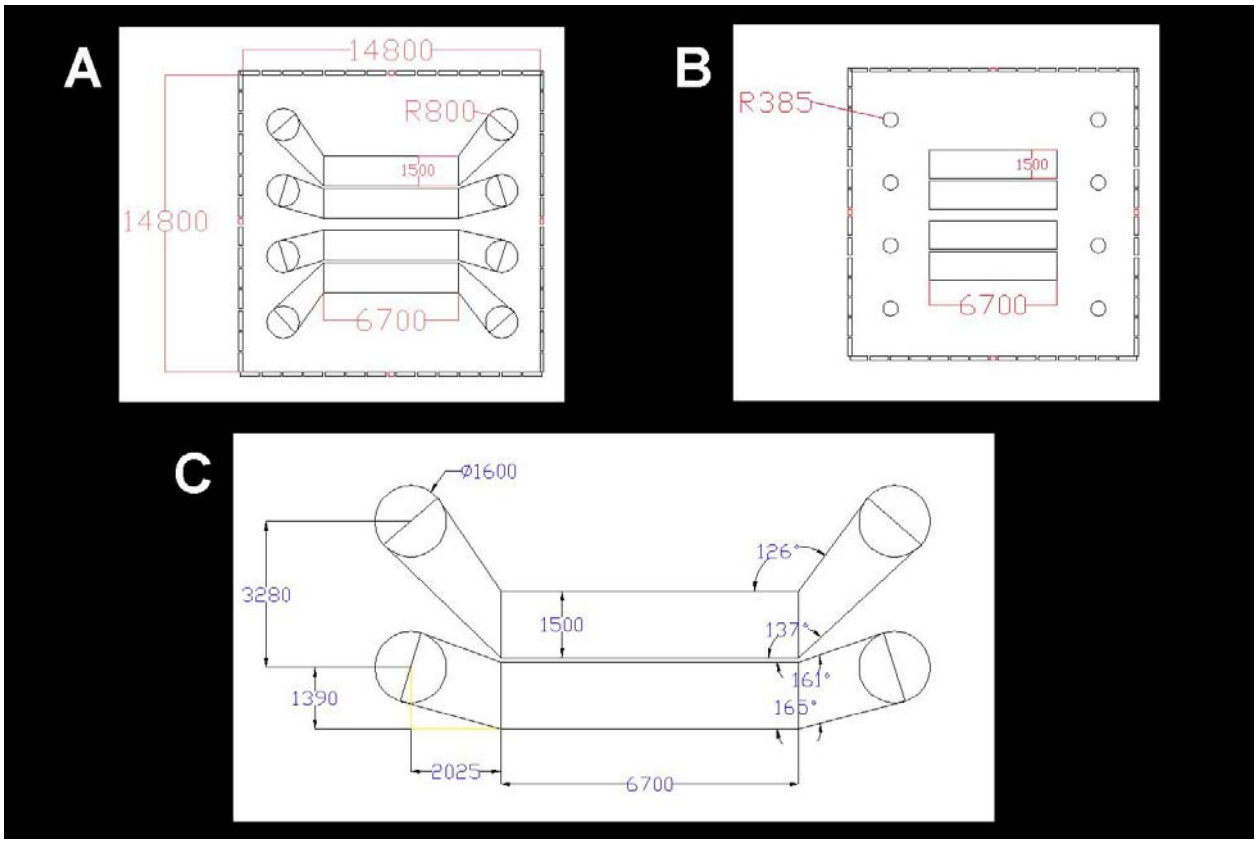

Figure 26: AutoCAD designs for chamber molds (all units given in microns or degress). A) The first two layers were exposed with a mask containing the pattern seen here. A slight variation of this pattern was used for the first layer where 10 µm vias connected the chambers. B) The top layer was fabricated using the pattern here. The smaller diameter circles result in raised structures with the same dimensions as the inner diameter of the tubing used for introducing cell suspensions. C) Expanded view of one half of the chamber pattern showing all dimensions of the design.

The resulting molds were then coated with the alkylsilane 13F to create a non-adhesive surface for subsequent steps. Briefly a solution of 0.1% (vol:vol) 13F in dry toluene was prepared in glove box under a dry, nitrogen atmosphere. The 13F solution was transferred to a Pyrex Petri dish. The wafers were then placed into the 13F solution for 5-10 minutes. The

wafers were then removed and rinsed with toluene and blown dry under a stream of filtered dry nitrogen. The 13F coated molds were placed in a desiccator until used.

Microfluidic chamber fabrication

The microfluidic chambers were fabricated by casting PDMS (Dow Corning, Midland MI) over the 13F coated SU-8 chamber molds. Prior to casting, the chamber mold was taped onto a casting form to confine the PDMS to a square the same area as the cantilever die to enable subsequent alignment of the chamber and cantilevers. The molds were fitted with 1 cm segments of silicone tubing, I.D. 0.75 mm and O.D. 1.6 mm, (Dow Corning, Midland MI) at the inlet and outlet portions of the individual chambers. A small amount of Duco cement was placed on the end of the tube segment to hold it into place prior to casting. Sylgard 184, PDMS monomer, was mixed with initiator in a 10:1 (wt:wt) ratio and mixed thoroughly. The mixture was degassed under vacuum for ~25 minutes to remove any air bubbles present. The degassed mixture was then poured into the casting form. Care was taken to ensure none of the Sylgard 184 entered the tubing, as this would clog the tube and render the device useless. The entire assembly was then degassed again to remove any air bubbles between the Sylgard 184 and chamber mold. After degassing, Sylgard was added to compensate for any loss that occurred during the degassing process. The uncured chambers were then place in a 65^{o}C oven for 1 hour to polymerize the Sylgard. After curing the casting form was removed from the oven and allowed to cool for 15 minutes and allow the PDMS to contract, making extracting the microfluidic chambers easier. A scalpel blade was used to trim excess PDMS from the edges of the chambers. IPA was then injected into the spaces between the casting form, molds and PDMS chambers to facilitate removal of the chambers. Once the chambers

were removed, remaining Duco cement was removed from the ends of the tubing to create a continuous chamber through which solution could flow.

Cell culture

Cell harvesting and preparation

Skeletal muscle was dissected from the hind limb thighs of a rat fetus at embryonic day 18 (Charles River Laboratories, Wilmington, MA) according to previously published protocol [37] with some modification. Tissue samples were collected in a sterile 15-ml centrifuge tube containing 1 ml of calcium and magnesium free phosphate buffered saline (PBS). Tissue samples were enzymatically disassociated using 3 ml of 0.05% of trypsin–EDTA (Invitrogen, Carlsbad, CA) solution for 60 min in a 37°C water bath with agitation of 100 rpm. After 60 min, the trypsin solution was removed and 6 ml of L15 media (Invitrogen, Carlsbad, CA) containing 10% fetal bovine serum (FBS) was added to terminate the trypsin action. The tissue was then mechanically triturated using a sterile narrow bore Pasteur pipette, allowed to settle for 3 min, and transferred to a 15-ml centrifuge tube. This was repeated three times. The dissociated tissue was then centrifuged at 300g for 10 minutes at 4^{o}C on 6 ml of a 4% (wt/vol) cushion of bovine serum albumin (BSA). The pellet was resuspended in 10 ml L15 + 10% FBS and plated in uncoated 100-mm Petri dishes for 20–30 min depending on the amount of tissue, to allow contaminating fibroblasts to settle out. After 20–30 minutes the supernatant was layered on 6 ml of a 4% BSA cushion, and centrifuged at 300g for 10 min at 4°C. The pellet was resuspended in 1.5 ml of medium.

Purified myocytes were plated at a density of 500–800 cells per square millimeter onto the cantilevers. Myocytes were allowed to attach for 1 hour after which time 3 ml of culture medium (Neurobasal media containing B-27 [Invitrogen, Carlsbad, CA], Glutamax

[Invitrogen, Carlsbad, CA], and Pencillin/Streptavidin) was added. Cultures were maintained in a 5% CO_2 incubator (relative humidity, 85%). Culture medium was exchanged every 4 days. Cantilever/myocyte constructs were allowed to culture for 10-13 days. During this time myocytes fuse into functional myotubes capable of generating contractile stresses sufficient to deflect the cantilever. These cultures were used in experiments for validating the use of Stoney's equations equation for calculating contractile stress of the myotubes.

Cell seeding using microfluidic chambers

Cantilever die and microfluidic chambers were sterilized in absolute ethanol for 5 minutes and dried in a sterile biosafety cabinet prior to cell seeding. Cantilever die were placed on a sterile piece of PDMS, cantilever side up, to create a sealed base on which the chambers could be placed. The microfluidic chambers were then aligned by hand onto the cantilever die as shown in Figure 27 and pressed to seal the PDMS to the silicon. The PDMS sealed to the silicon wafer sufficiently to allow confinement of cell suspensions within the boundaries of the chamber. The assembled devices were then seeded with motorneuron and skeletal muscle cell suspensions (Figure 27). The cells were introduced by injecting 200 μl of the respective suspensions using a programmable Eppendorf pipettor (Hamburg, Germany). The cell suspensions were aspirated into the pipettes, and the tips of the pipettes were placed in the end of the tube embedded in the PDMS chamber. The cells were injected into the chamber simultaneously. The pipette tips were then removed from the tubing and the chamber/cantilever assembly was kept in a water jacketed incubator for one hour at a temperature of 37^{o}C. After one hour the chamber and PDMS backing were removed from the cantilever die, and 3 ml of culture media were added. The resulting Bio-MEMS device was then cultured for 10-13 days at 37^{o}C and a 5% CO_2.

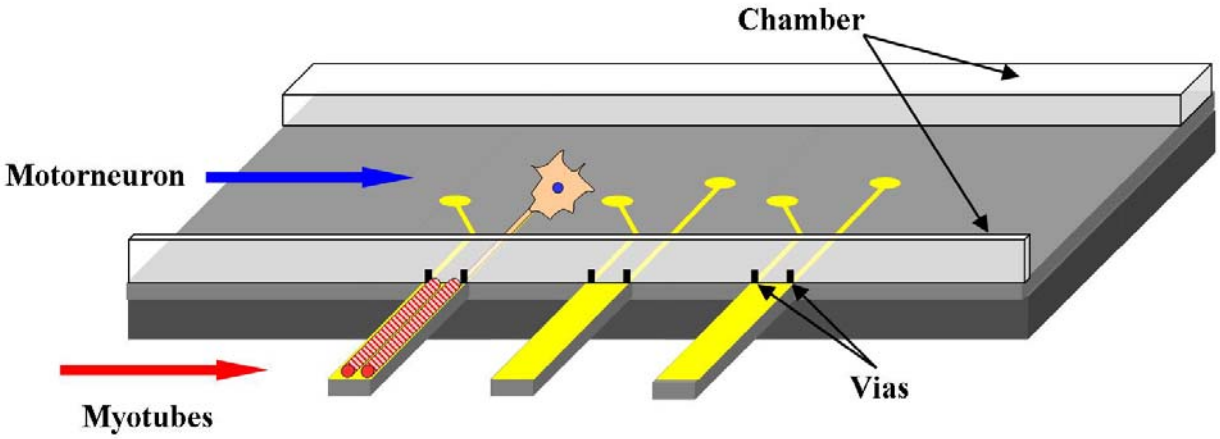

Figure 27: Simplified schematic of cell seeding procedure on surface patterned microcantilevers. The cantilevers were isolated from the bulk substrate by a PDMS barrier with 10 μm vias to allow passage of axonal processes through to the developing muscle. After the chambers were sealed onto the substrate, myocytes and motorneuron suspensions were introduced simultaneously into their respective compartments.

AFM setup

An atomic force microscope (AFM) detection system was designed for measuring deflection of the cantilevers during myotube contraction (Figure 28). The entire system was assembled around an upright Olympus BX51WI electrophysiology microscope (Olympus Inc., Center Valley, PA). The AFM consisted of a class 2 red photodiode laser (Newport, Irvine, CA), a stimulation chamber, a 4-quadrant photodetector (Noah Industries, Melbourne, FL), and a computer with pClamp 10.0 data acquisition software (Molecular Devices, Union City, CA). The laser and photodetector (PD) were mounted on x-y-z-θ translators (Newport, Irvine, CA) which were mounted on the underside of the microscope stage. The stimulation chamber was fabricated from 5 mm thick polycarbonate sheet. An approximately 15 mm x 15 mm square chamber was milled out of the sheet and fitted with silver wires (0.015 inch diameter) for field stimulation. The silver wires were mounted parallel to each other with a separation of 15 mm. The bottom of the chamber was sealed using a 22 mm x 22 mm glass coverslip. This created a transparent base through which the laser beam could easily pass.

The silver wires were connected to an external pulse generator (A-M systems, Sequim, WA) capable of producing field stimulation pulses of varying intensity, frequency, and waveform. Both the pulse generator and PD were connected to an Axon Instruments series 1440 digitizer (Molecular Devices, Union City, CA) which was interfaced with the computer.

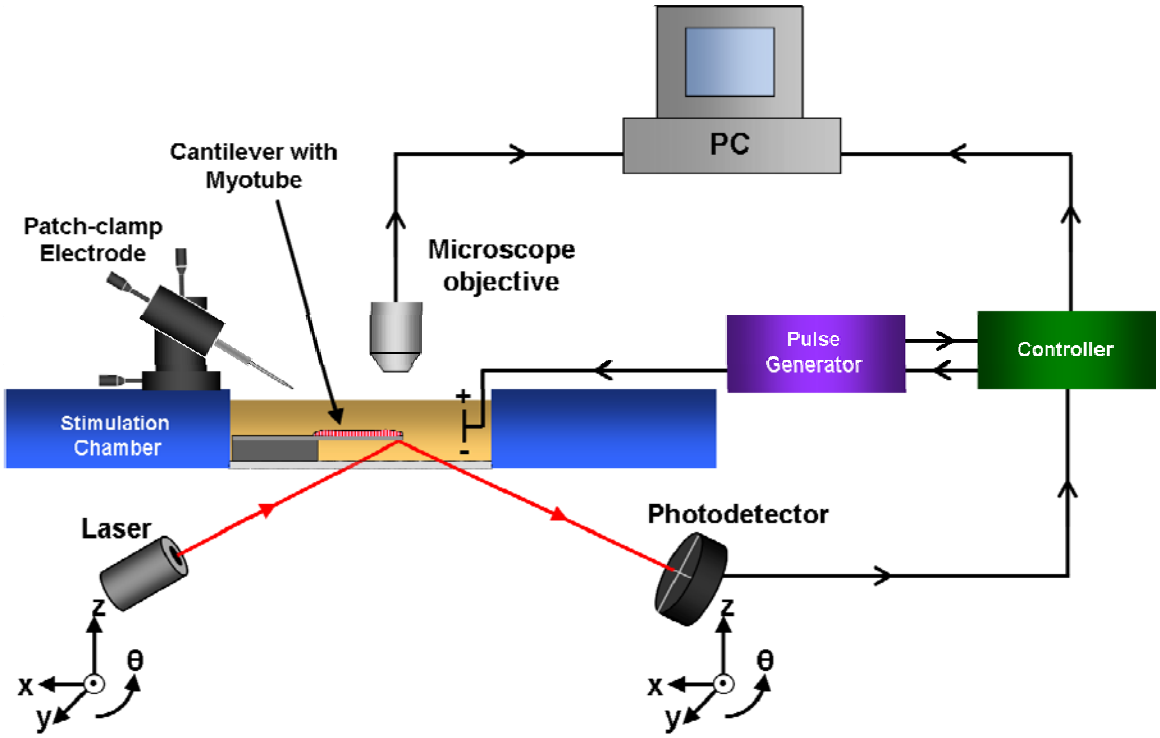

Figure 28: Schematic representation of AFM detection system

AFM calibration

The AFM system was calibrated using a modified version of the optical lever method [101]. A bare, uncultured cantilever was placed in the stimulation chamber. The laser was focused on one of the cantilevers and the PD was adjusted so that the laser fell on the diode surface. Using a digital volt meter to monitor the output voltage, the PD was lowered so that the voltage being read was less than -7 volts. The PD was then raised in 5 μm increments and the voltage recorded at each position. The results were plotted in Excel and a linear regression line was fitted to the linear region of the calibration curve, which was the region between -5 and 5 volts. The slope of this region was the detectors sensitivity ($y_{detector}$). This

value was used to calculated the angle, θ, of the deflection at the end of the cantilever using the equation:

$$\theta = \frac{y_{measured}}{2\cos(\varphi) l \times y_{\det ector}} \qquad \text{Equation 2}$$

where, $y_{measured}$ is the voltage measured from the PD, φ, is the angle of the detector to normal, and l, is the path length of the reflected laser beam.

Stress calculation

The stress exerted by a myotube attached along its length to a cantilever can be estimated by considering the system as a cantilever bimorph and using Stoney's equation [100]. Stoney's equation relates the stress in a bimorph system (film on substrate) to curvature of the substrate and the mechanical properties and thicknesses of the substrate and adherent film layer. The film stress, σ_{film}, is:

$$\sigma_{film} = \frac{1}{6Rt_{film}}\left[\frac{E_{beam}t_{beam}^3}{(1-\nu_{beam})(t_{beam}+t_{film})} + \frac{E_{film}t_{film}^3}{(1-\nu_{film})(t_{beam}+t_{film})}\right] \qquad \text{Equation 3}$$

where E_{beam} and ν_{beam} are the cantilever material modulus (130 GPa) and Poisson's ratio (0.28), respectively, t_{beam} is the cantilever thickness, t_{film} is the myotube thickness, R is the effective radius of curvature of the beam caused by the stress in the myotube layer, σ_{film}.

Many applications of Stoney's formula, most recently for studies of deposited and adsorbed films on thin substrates [102] or cantilevers [103-105], neglect the second term in the brackets because the films are much thinner than the substrate. In the present Bio-MEMS system, this assumption is not satisfied (t_{film} ~ 10 μm compared to the cantilever thickness, t_{beam} ≈ 5 μm). However, for the system we also neglect this term because the modulus of the myotube cells comprising the film on the cantilever are expected to be in the kPa range, at

least 6 orders of magnitude lower than the modulus of the beam substrate Si (130 GPa). Thus we write:

$$\sigma_{film} \approx \frac{E_{beam}(t_{beam}{}^{3})}{6(1-\nu_{beam})t_{film}(t_{film}+t_{beam})}\frac{1}{R} \qquad \text{Equation 4}$$

The radius of curvature of the cantilever during contraction was calculated using the raw voltage data collected from the PD. This was done taking into account the geometry of the system (path length of the reflected laser, sensitivity of the detector, etc.) [106, 107]. From the raw data the change in angle of the end of the cantilever, θ, was calculated using equation 1. Using θ it was then possible to calculate the deflection of the free end of the cantilever,δ , by the relation from Butt et al [103]:

$$\delta = \frac{2\theta L}{3} \qquad \text{Equation 5}$$

where L is the length of the cantilever. Experimentally, 1/R is estimated using the measured beam deflection and the geometric approximation from Ratieri et al. [108]:

$$\frac{1}{R} \approx \frac{3\delta}{2L^{2}} \qquad \text{Equation 6}$$

From Figure 28, tensile or compressive stress in the myotube film will result in an upward or downward vertical deflection of the cantilever beam. Measured deflections from the photodiode detector will be reported as positive and negative deflection δ, respectively. Since the myotube film is grown on the top face of the cantilever array, deflections due to tensile (positive values) or compressive (negative values) stresses in the film are consistent with standard conventions [109]. All calculations were performed using Matlab (Appendix C)

Immunostaining and Confocal Microscopy

After AFM measurements the tissue samples were washed 3x with PBS, then fixed for 15 minutes in 4% (vol/vol) paraformaldehyde at room temperature. Tissues were permeabilized and blocked in a single step using a solution of 0.1% Triton-X100 in PBS, with 10% donkey serum. Blocking and permeabilization was allowed to proceed for 1-2 hours. Afterward, the samples were washed 3x in PBS and incubated with a mouse anti-myosin heavy chain primary antibody (Developmental Studies Hybridoma Bank, Iowa City, IA) overnight at 20^{o}C. For cocultures with both muscle and motorneuron, a second goat anti-neurofilament primary antibody was used. Following incubation with the primary antibody, the tissues were washed 3x with PBS and incubated with a donkey anti-mouse secondary conjugated with Alexa Fluor 594 and donkey anti-goat Alexa Fluor 488 (Invitrogen, Carlsbad, CA) at room temperature for 1-2 hours. The final stained samples were washed again with PBS and imaged under PBS using confocal microscopy.

Myotube thickness was measured by optical sectioning with a Perkin Elmer Ultraview spinning disc confocal microscope (Perkin Elmer, Waltham, MA) under a 40x water immersion lens. The 40x lens was mounted on a piezoelectric z-step motor with a minimum step size of 0.4 μm and a total travel length of 60 μm. Images were collected in 0.5 μm steps from the surface of the cantilever to the top of the tissue. The "z-stack" of images was reconstructed using a 3-D rendering program provided with the microscope. The thickness of the myotube was then measured using the reconstructed image and an internal reference scale.

Exogenous factors added to muscle culture

In order to demonstrate the usefulness of this device for studying the biology of muscle development and function, experiments were conducted using exogenously applied factors to

elicit a measurably different response of the muscle compared to control conditions. The sodium channel agonist veratridine was added to normally cultured myotube on cantilevers and the response was measure with the AFM detection system. After 10 days of culture the myotube/cantilever constructs were place in the AFM detection system and stimulated with a 1 Hz pulse to elicit synchronous, detectable contraction. Upon confirmation of synchronous contraction veratridine was added to a final concentration of 5 μM, and the resulting contractile behavior recorded.

Cultures were also performed in order to enhance the contractile capacity of the myotubes. The culture medium NbActiv4 was used *in lieu* of the Neurobasal/B27 formulation used for control cultures. Cantilevers were seeded normally and cultured under conditions identical to those previously stated. After 10-13 days cantilever/myotube constructs were placed in the AFM detection system and stimulated with a 1 Hz pulse train. The calculated values were then compared to previous experiments and published literature.

Results and discussion

Characterization of cantilevers

When using Stoney's equation to estimate film stress on cantilevers it is critical to have precise knowledge of the thickness of both the beam and the film. Figure 29 shows representative SEM micrographs of the cantilevers used for the experiments. The cantilevers were measured to have a mean length and width of 755 +/- 3 μm and 109 +/- 1 μm. As shown in Figure 29B the mean thickness of the cantilevers was measured to be 5.27 +/- 0.07 μm. Given these values one can expect a ~4% error in the stress estimation from experiment to experiment due to variation in beam thickness.

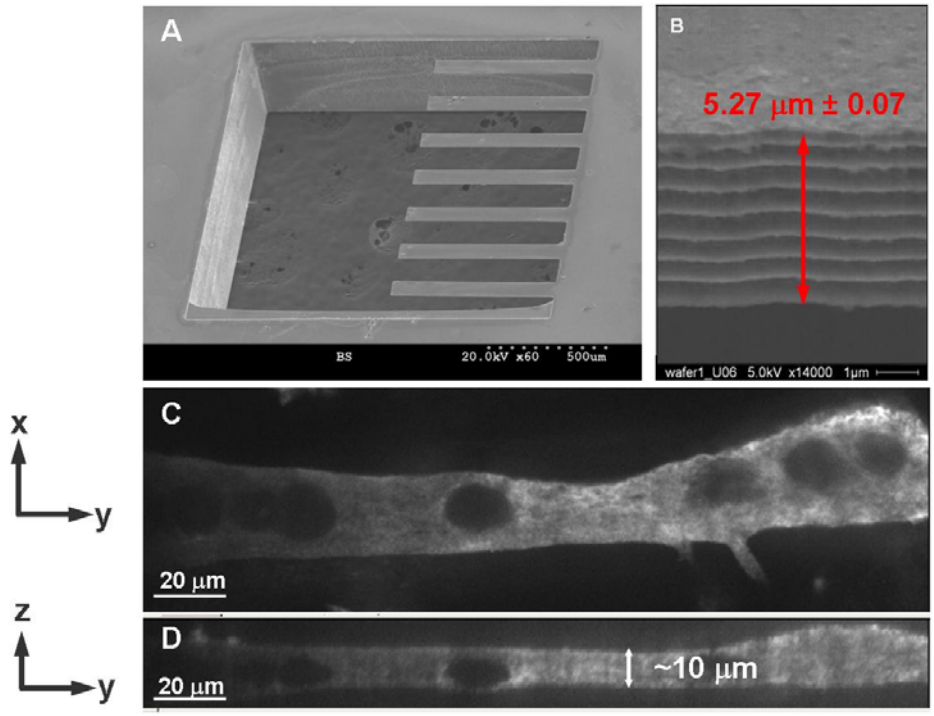

Figure 29: SEM and confocal microscopy measurements of cantilever and tissue thickness. A) Low magnification view (50° takeoff angle) of silicon cantilevers, B) High magnification view (50° takeoff angle) of cantilever showing the measured thickness, C) Top-down view of cultured myotube taken by confocal microscopy, D) Digitally reconstructed side-view of cultured myotube showing measured thickness.

The spring constant of the cantilevers was calculated theoretically and measured experimentally. The calculated spring constant, 1.21 N/m, was determined from the measured dimensions and Young's modulus for crystalline silicon, using formula for the spring constant of a rectangular cantilever. The spring constant was determined experimentally using the method of Sader et al. [110]. In short the resonant frequency was measured via ring-down experiment, and the resulting data processed by the spectrum analysis routines in the pClamp software. The resonant frequency in air was determined to be 88.5 kHz. Corrected for damping by air, the resonant frequency of the cantilevers was found to be 88.7 kHz. This value was then applied to Sader's equation for calculating the cantilever spring constant, which was found to be 1.26 N/m. Due to the high resonance

frequency of the cantilevers, it is expected that the resulting data reflects only the behavior of the myotube contraction as the response time of the cantilevers is on the order of microseconds, whereas the time scale of the muscle contraction is on the order of milliseconds.

Myotube culture

After plating the dissociated myocytes on the cantilevers, the Bio-MEMS constructs were allowed to culture for 10-13 days during which time the myocytes fused into functional myotubes. During fusion and differentiation, the myotubes spontaneously orient along the long axis of the cantilever, facilitating bending of the cantilever. It should be noted, however, that the orientation of the myotubes was not always directly parallel to the long axis of the cantilever. This configuration resulted in some torsional bending, and hence a possible underestimation of the total contractile stress. Typically the coverage of myotubes on cantilevers was greater than 95%. Occasionally, tissue coverage was less due to tissue processing, suboptimal surface modification, and other systematic errors. Only robust cultures with morphologically normal looking myotubes were used for AFM experiments. Figure 29c shows a confocal microscope image of a section of a representative myotube cultured for 13 days on a DETA modified cantilever (not visible). Figure 3a shows the top down projection of the z-stack in the plane of the cantilever. The data from the z-stack of images were reconstructed into a 3-dimensional representation of the myotube geometry. Figure 29b is a side view showing the thickness of the myotube along a section of the cantilever. The mean thickness of the myotube was ~10 μm. Due to the morphology of the myotube, however, the thickness is not necessarily uniform throughout the length of the cantilever. The thickness has been measured to range between 5 μm to 15 μm on an

individual cantilever. This variation in film thickness throughout the tissue can potentially lead to discrepancies between true and calculated stress values. In this study we used the average value of 10 μm for calculations. The effect of the thickness variation on the calculated stress will be considered in a later section.

Stress Calculation

Figure 30 shows both the raw voltage data from the PD and the resulting stress calculated using the Stoney's equation. Figure 30a shows the raw data collected in free-run mode from myotubes cultured for 13 days and stimulated with a 5 volt DC pulse at a frequency of 1 Hz. As shown previously [38] this allowed selective stimulation of the myotubes to actuate the cantilevers. Each trigger pulse, Figure 30b, corresponds precisely with the onset of a myotube contraction. The myotubes responded to the stimulation in a frequency dependent manner; increasing or decreasing the stimulation frequency would result in a corresponding change in the frequency of myotube contraction, Figure 30b-c. As with previously published results, stimulation at or above a frequency of 10 Hz induced a state of fused tetanus [38]. Figure 31 shows the resulting Stoney's calculation using the raw data. The stresses calculated from this data set range between 1.1 kPa and 1.4 kPa. These values are in excellent agreement with previously published literature for cultured skeletal muscle [111], which report average peak twitch stress values of 2.9 kPa (reported as specific peak twitch force in units of kN/m^2), but less than 1% those expected for adult muscle, ~300 kPA [99, 111]. This is not surprising due to the fact that the tissue used in this study was collected from embryonic stage rat pups and cultured *in vitro* for only 13 days after dissection. It is possible that the culture conditions, as published here, are not sufficient for the development of myotubes with adult phenotype. Similar observations were made by Dennis and Kosnik [111]

for cultured adult rat myoids noting the possibility of developmental arrest in culture preventing the development of adult isoforms of myosin.

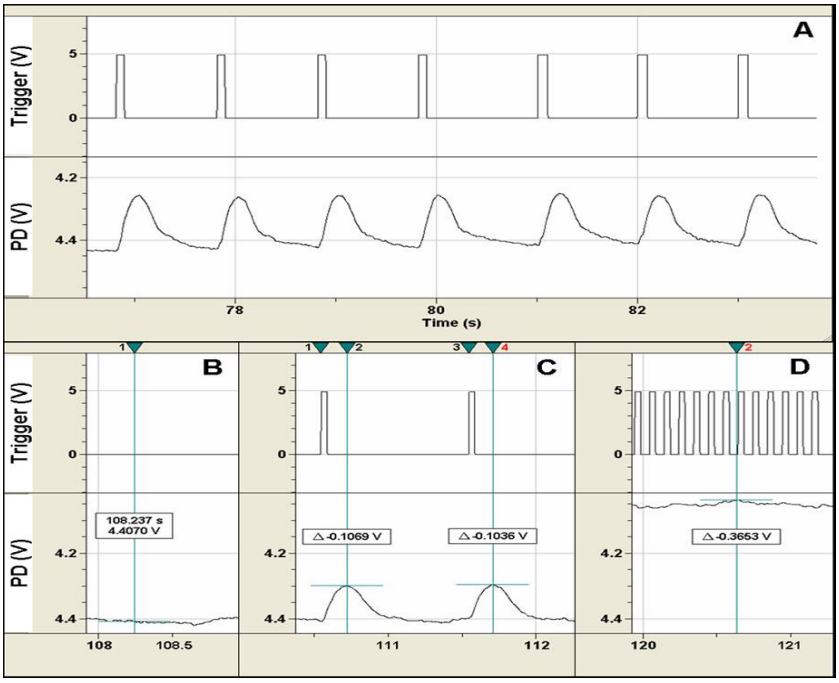

Figure 30: Real time interrogation and monitoring of myotube contraction with a Bio-MEMS device. In each picture the top trace shows the stimulus trigger and the lower trace is the raw data recorded from the PD. A) 1 Hz stimulus resulting in synchronous contraction, B) In the absence of stimulus no contraction was recorded, C) 10 Hz stimulation induced a state of fused tetanus, where the muscle was unable to relax.

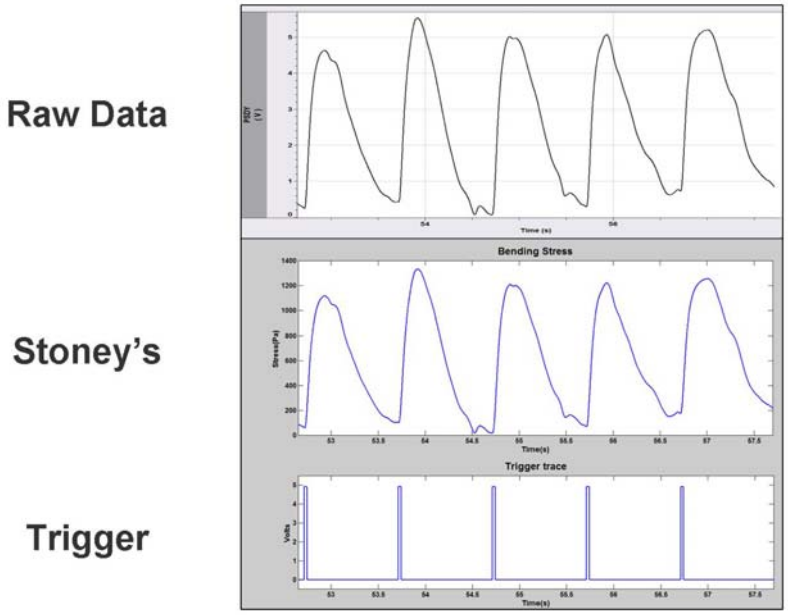

Figure 31: Raw data versus calculated stress for cultured embryonic muscle.

To further characterize the myotubes, three other parameters were analyzed (Figure 32): the time to peak twitch stress (TPT) was measured, which is the time required to reach stress from the onset of contraction, time to half relaxation (½RT), which is the time require to relax to 50% of peak tension, and the average stress generation ($d\sigma/dt$), which is the slope of the force curve between 20% and 80% of peak tension. Figure 31 shows both the raw data (Figure 31a) and calculated stress from 5 myotube contractions (Figure 31b). The resulting average contractile stress for these data is ~1.2 kPa which is in agreement with previously stated results. The calculated values for TPT, and ½RT (Table 4) were significantly longer than those published for cultured muscle by Dennis and Kosnik [111] and for adult rat muscle as published by Close [98]. The average TPT for the cultured myotubes was 236.8 ± 26.1 ms. This value is considerably slower than that of 69.3 ± 9.4 ms published by Dennis for cultured rat myoids as well as values of 65.0 ± 3.8 ms and 36.0 ± 2.3 ms, for neonatal and adult rat respectively, published by Close. The ½RT values for cultured myotubes were also prolonged compared to those reported by Dennis and Kosnik and Close. The average ½RT for the data presented in Table 4 was measured to be 233.6 ± 23.8 ms. Dennis reported ½RT values of 116.4 ± 19.4 ms for myoids, while Close reported values of 70.0 ± 4.9 ms for neonatal muscle and 48.0 ± 3.4 ms for adult muscle. It is interesting to note, however, that the TPT:½RT (~1:1) ratio for the cultured myotubes was closer to that of the neonatal and adult rat muscle than that of the cultured myoids (~1:1.7). It can be concluded from these data that the myotubes cultured in the Bio-MEMS system, while exhibiting contractile stress magnitudes comparable with those previously published for cultured rat muscle, show evidence of a more embryonic phenotype with regard to other important physiological

parameters. This is further reinforced by previously published results showing staining of similarly cultured myotubes for embryonic myosin heavy chain, but not adult or fetal isoforms [35].

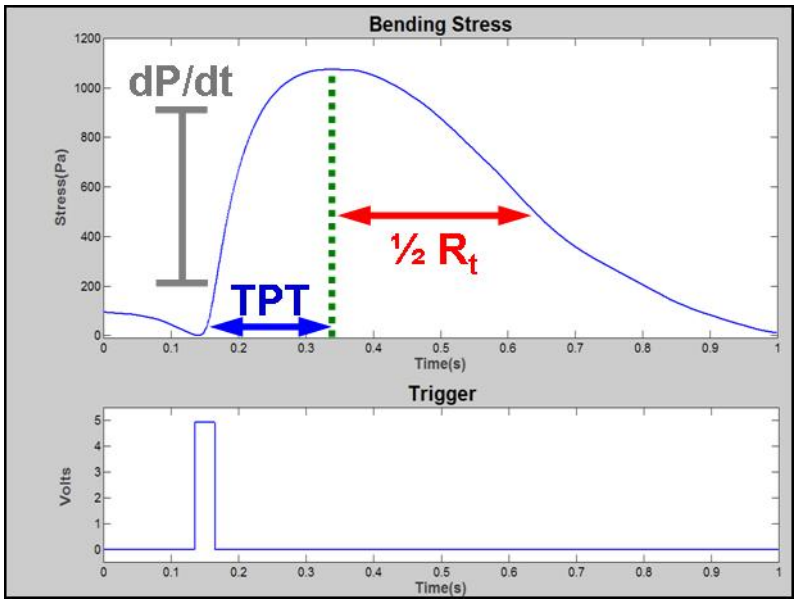

Figure 32: Critical parameters for muscle characterization.

Table 4: Comparison of calculated stress values to published literature. Values for dσ/dt are not available in Close et al., but average force generation has been reported to be more than 1000 fold higher that measured in Dennis and Kosnik

	σ_F (kPa)	TPT (ms)	½ RT (ms)	dσ/dt (Pa/ms)
ESM	**~1.1**	**236.8 ± 26.1**	**233.6 ± 23.8**	**7.15**
Cultured Myoids	**~2.9**	**69.3 ± 9.4**	**116.4 ± 19.4**	**75.3 ± 10.0**
Adult	**> 300**	**36.0 ± 2.3**	**48.0 ± 3.4**	**X**

Variation in Stress Calculation due to film thickness

As previously stated the thickness of the myotube film on the cantilever has been measured to vary between 5 to 15 μm (Figure 29). Figure 33 shows the variation in calculated stress due to film thickness. Figure 33a is a plot of the variation in calculated stress using the average of 11 contractions versus the film thicknesses used in the Stoney's calculation ranging from 5 to 15 μm. It is clear from this graph that there is a significant variation in the calculated stress due to the measured film thickness. Figure 33b shows a plot of the calculated peak contractile stress vs. film thickness. In this plot it can be seen that the stress values range from ~0.5 kPa to ~3.2 kPa over the selected film thickness values. It is interesting to note that the Stoney's calculation is particularly sensitive to variations in the film thickness in the range encountered here. Below 5 μm the stress values increase exponentially. Above 15 μm the change in stress due to film thickness slows considerably. This reinforces the need for accurate measurements of the myotube thickness and standardization of the culture methods to minimize variations of the same. It should be noted, however, that even though there is obviously significant variation in calculated stress these values are still within the range of those published by Dennis and Kosnik (0.9 kN/m^2 to 5.0 kN/m^2). These results validate this approach as a method for measuring contractile stress generated by cultured skeletal muscle in a Bio-MEMS device.

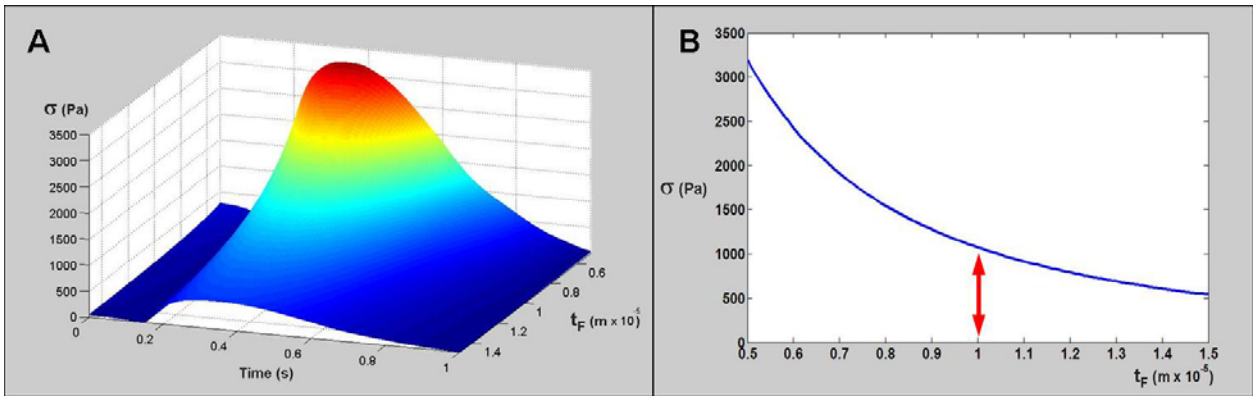

Figure 33: Stress variation with measured film thickness. A) Data recorded from a contracting myotube plotted as a function of time and measured film thickness. B) Peak stress plotted as a function of film thickness. Arrow indicates calculated stress value for 10 μm film thickness.

Cell patterning using microfluidic chambers and surface chemistry

Microfluidic chambers were prepared as previously described. Figure 34 shows SEM images of the resulting PDMS chambers. The lower micrograph shows the low magnification view of the entire device. In each device there are two chambers, one for seeding muscle on the cantilevers and one for seeding motorneurons on the bulk silicon. The two chambers are separated by a barrier with small vias (top right image) incorporated to allow cellular processes from the motorneurons to interact with the muscle. The vias were made small relative to the size of a cell body (top left image), about 10 μm. This was necessary to prevent migrating cell bodies from crossing over onto the cantilever while allowing cell processes to pass through.

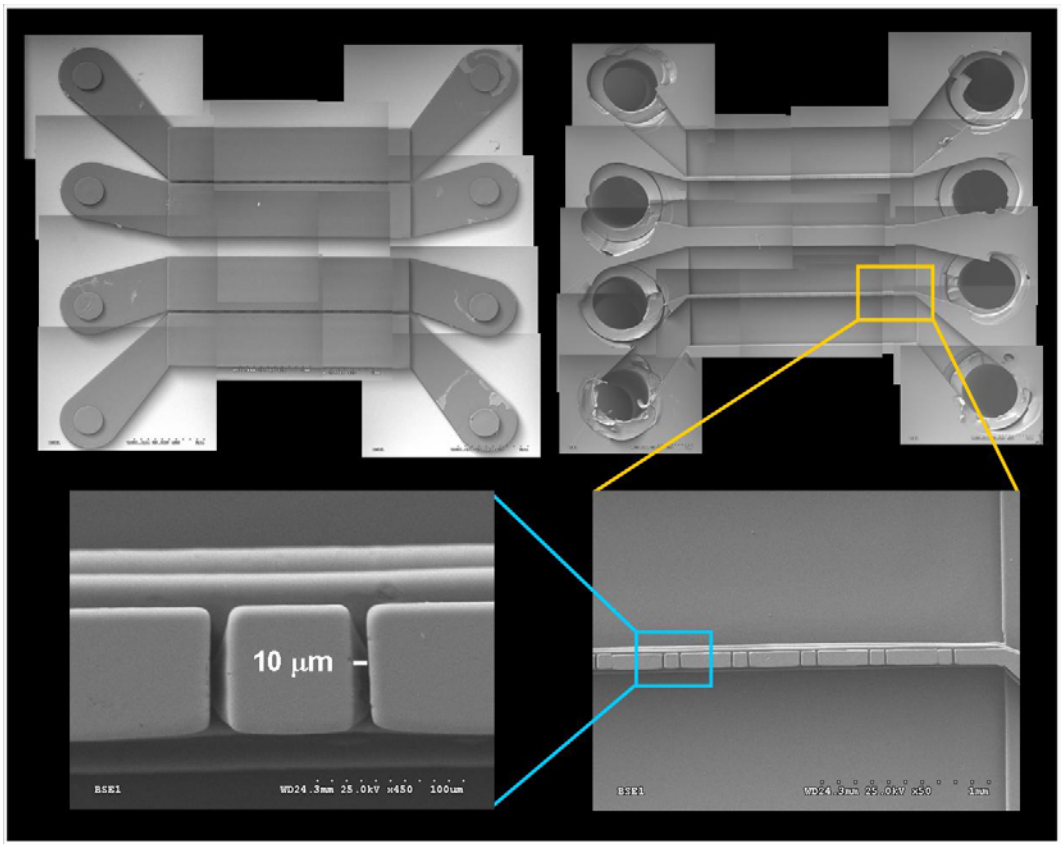

Figure 34: SEM images of PDMS chambers made from SU-8 molds. The lower image shows a low magnification view of the entire chamber area. The inset at the top right shows vias for allowing cell processes to penetrate the barrier. The inset at the top left shows the width of the vias.

Cells were seeded into the chambers as previously described. Cell growth and differentiation was monitored throughout the course of the culture. Figure 35 shows representative pictures of myoblasts and motorneurons growing on patterned PEG-DETA substrates. The pattern, seen in Figure 35a, corresponds to the DETA portion of the pattern which was surrounded by a cytophobic PEG region. The chamber was aligned over the pattern such that the barrier fell approximately over the region where lines from the somal adhesion sites meet the myoblast adhesion region (large rectangles). Differentiating myoblasts can be seen in Figure 35b after seeding. Fidelity of myoblasts to the pattern was very good. Myotubes begin to form after 5 days and align themselves along the long axis of the rectangular pattern. Confinement of the motorneurons to the somal adhesion site was also good, Figure 35c. Motorneurons would adhere to the 30 μm diameter circle and send out processes down the lines leading to the developing myotubes. However, migration of motorneurons down the DETA lines was occasionally problematic. Due to the height of the chambers (~300 um) it was impractical to leave them on the cultures beyond the time required for seeding. The small volume of the chambers prevented long term culture, due to nutrient depletion and lack of diffusion of gasses through the thick PDMS chamber. Perfusing the cells with media was also problematic due to the high shear rates encountered in such small chambers. For this reason chambers were removed after seeding. In the absence of a physical barrier it was not uncommon to see motorneurons migrating down the lines to the differentiating myoblasts. However, the majority of cultures yielded cells confined to their respective regions.

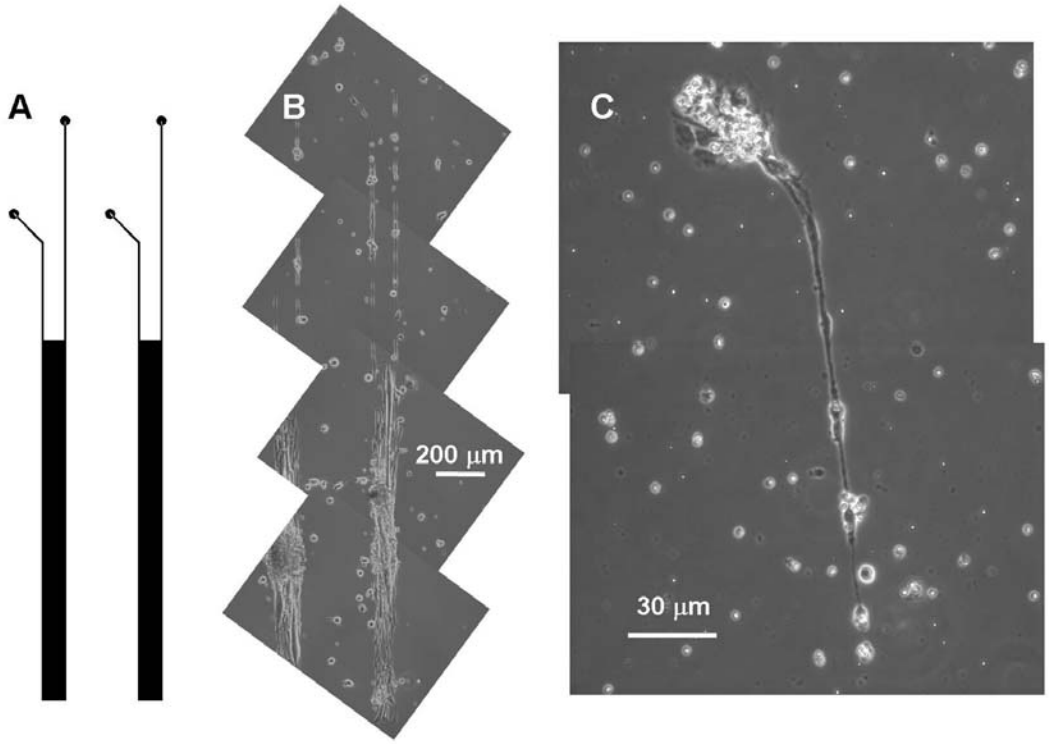

Figure 35: Cocultured motorneurons and myoblasts seeded with microfluidic chambers. A) Pattern used for PEG-DETA photolithography. The pattern consists of a 100 μm wide rectangle for myoblast adhesion, with 6 μm wide lines leading to 30 μm diameter circular somal adhesion sites for motorneurons. B) Montaged micrograph to the left shows differentiating myoblasts on the patterned substrate. C) High magnification view of motorneurons adhered to somal adhesion site.

Addition of exogenous factors modulates function of cultured myotubes

Addition of the sodium channel agonist veratridine

Given the ability of this method for quantifying muscle contraction force and dynamics in real-time, it is ideal for studying the effect of exogenously applied factors on muscle physiology. One such example was the addition of the toxin veratridine to the stimulation chamber during electrical stimulation. Veratridine is an agonist that causes the persistent opening of voltage-gated sodium channels. Normally, upon depolarization of the cell membrane from either field stimulation or neurotransmitter release from neural inputs

voltage-gated sodium channels open allowing an influx of sodium ions into the cytoplasm which further depolarizes the sarcoplasmic reticulum causing calcium release and contraction. After a certain refractory period the voltage-gated sodium channels close and the resting membrane potential is restored. Veratridine binds to the voltage-gated sodium channels causing an abnormally high release of sodium into the cytoplasm, tetanic contraction, and if it is not removed cell death. Figure 36 shows the recording of contracting skeletal muscle before and after addition of veratridine. Before addition the muscle was contracting normally in synchronization with the one Hz stimulus. At 84 seconds veratridine was injected into the stimulation chamber and allowed to diffuse to the tissue. As seen in figure 18 upon injection of the veratridine the muscle began to contract in an asynchronous, tetanic, manner with a peak stress far beyond those of the synchronized contractions. After the initial tetanic contraction, the muscle then lost the ability to further contract and the stress exerted on the cantilever returned to baseline. This is the reaction that is expected upon exposure to this toxin.

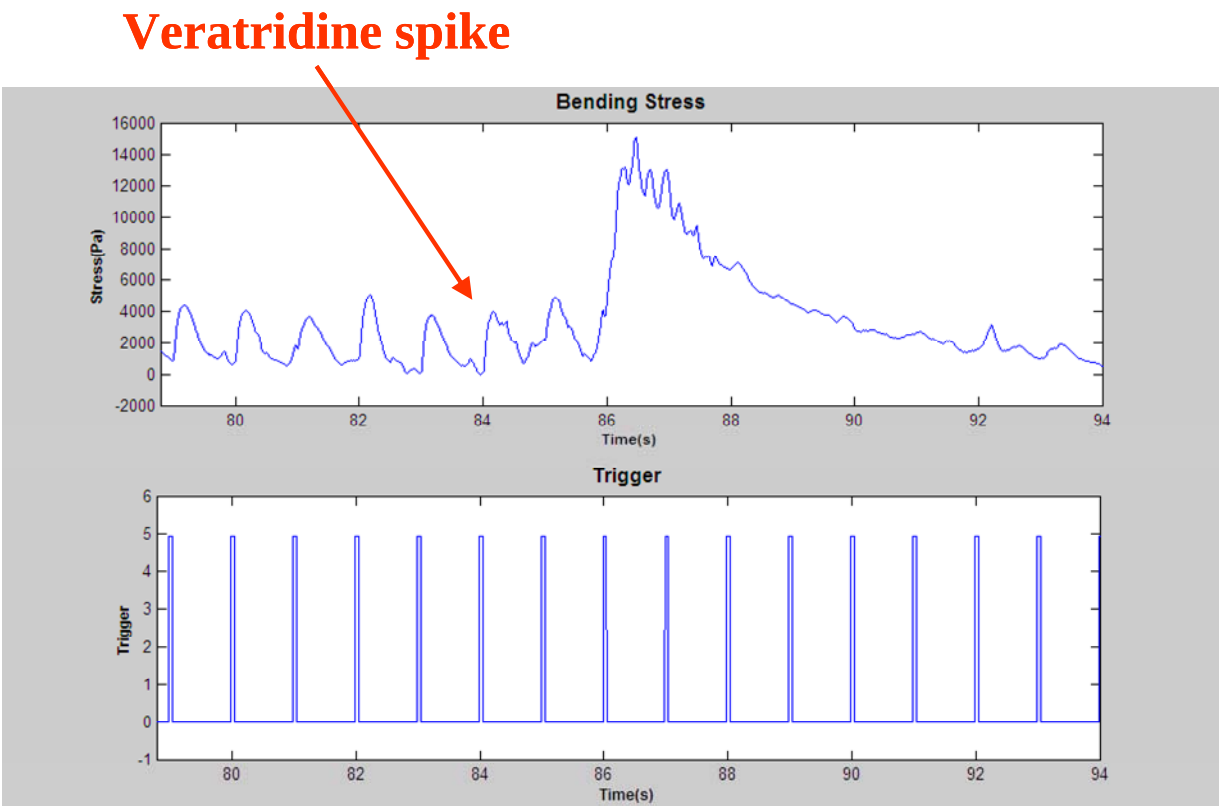

Figure 36: Contractile myotubes were exposed to the sodium channel agonist veratridine. Myotube were contracting synchronously with the 1 Hz stimulus when, at 84 seconds into the recording, veratridine was injected. After injection of the toxin the muscle tissue contracted in a tetanic manner, and lost the ability to contract further.

Growth of myotube in NbActiv4 to enhance muscle contractility

As stated in previous sections, the contactile phenotype of the muscle cultured in this system was of an embryonic nature. For this device to serve as a model system for the study of normal muscle it is necessary to be able to culture muscle of a more adult phenotype. In order to do this it is necessary to supply additional factors that promote the development of more mature contractile properties in the myotubes. The culture medium NbActiv4 is a proprietary formulation based on Neurobasal medium and the growth factor cocktail B27[112]. NbActiv4 contains three additional growth factors (creatine, cholesterol, and estrogen) that have been shown to produce an eight-fold increase in spike activity in cultured neurons.

However, these extra growth factors are also critical for the development of the contractile mechanism of skeletal muscle. For this reason we cultured embryonic skeletal myotubes grown on silicon cantilevers in NbActiv4 to quantify the changes in myotube development due to the added growth factors. Figure 37 shows representative contraction data for myotube. Figure 37a shows the raw data recorded by AFM for NbActiv4 cultured muscle. Here it can be seen that the TPT is measured to 172.1 ms and the ½ RT 175.7 ms.

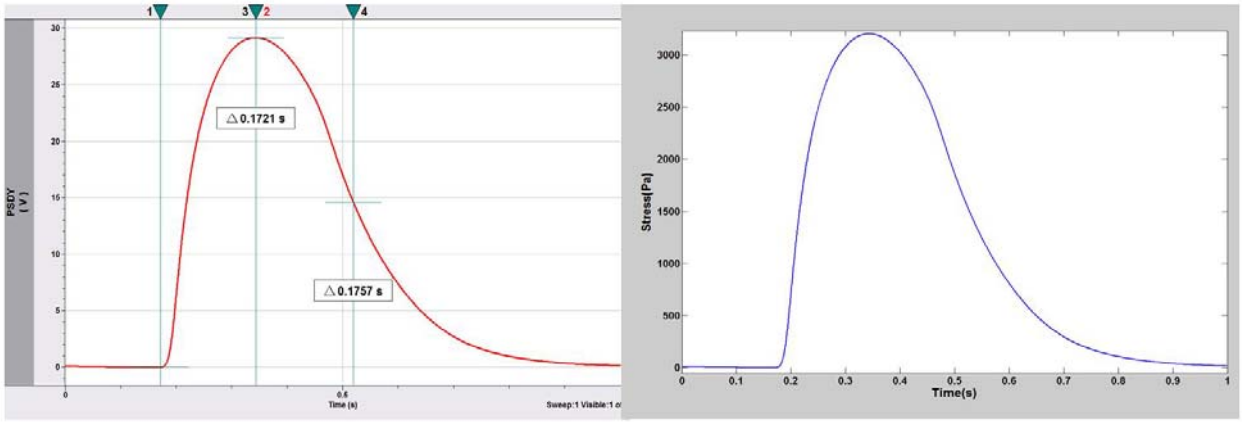

Figure 37: Contraction kinetics from muscle tissue cultured in NB4 media. A) raw data recorded from Bio-MEMS device showing TPT and ½ RT, B) Stress values calculated using Stoney's equation.

Table 5 shows comparison of NbActiv4 cultured muscle with previously published results as well as myotubes cultured in Neurobasal/B27. It can be seen that the addition of NbActiv4 enhances the contractile properties of the myotubes significantly. Most notably the contractile stress generated by NbActiv4 myotubes, 3.2 kPa, is approximately 3 fold higher than those cultured in Neurobasal/B27, 1.1 kPa. Although this value is still much less than the stress generated by adult muscle, it is comparable to that published by Dennis et al [111]. Also, TPT and ½ RT values for NbActiv4 myotubes have decreased significantly compared to muscle cultured in Neurobasal/B27. This decrease in contraction time demonstrates that

the myotubes are being pushed down a path towards a more mature phenotype, and developing fast-twitch isoforms of myosin, while increasing the speed of contraction. Furthermore, the increase in average stress generation ($d\sigma/dt$) by almost five fold reinforces the argument that the contractile apparatus of myotubes grown in NbActiv4 is more mature and capable of greater stress generation. A more detailed treatment of the effects of NbActiv4 on embryonic myotube development are presented in work by Das (dissertation).

Table 5: Contractile properties of NbActiv4 cultured muscle versus previous results and published literature. Values for $d\sigma/dt$ are not available in Close et al., but average force generation has been reported to be more than 1000 fold higher that measured in Dennis and Kosnik

	σF (kPa)	**TPT (ms)**	**½ RT (ms)**	**$d\sigma/dt$ (Pa/ms)**
ESM	**~1.1**	**236.8 ± 26.1**	**233.6 ± 23.8**	**7.2**
NbActiv4	**~3.2**	**172.1 ± 4.7**	**175.6 ± 3.6**	**35.4**
Cultured Myoids	**~2.9**	**69.3 ± 9.4**	**116.4 ± 19.4**	**75.3 ± 10.0**
Adult	**>300**	**36.0 ± 2.3**	**48.0 ± 3.4**	**X**

The work presented in this chapter demonstrates the development of a novel Bio-MEMS device for studying skeletal muscle and its development using microfabricated silicon cantilevers and alkylsilane surface chemistry. The usefulness of this device has been demonstrated for real-time interrogation of cultured skeletal muscle and the quantification contractile stress and kinetics. It has also been shown that physiological phenomena can be monitored and quantified such as tetanus, and response to exogenously applied factors.

Cultured myocytes spontaneously differentiate into functional myotubes on silicon cantilevers coated with DETA that produce contractile stress sufficient to deflect the

cantilevers. These deflections were then detected by AFM. By applying electrical field stimulation, it was possible to selectively actuate the myotubes on cantilevers in a frequency and intensity dependent manner. This ability to selectively actuate a cantilever is advantageous as it allows a high degree of control over the timing and nature of contraction. This method could also be applied to create bio-robotic devices using skeletal muscle as on actuator on a microfabricated device. Previous studies have utilized cardiomyocytes to provide mechanical force. However, cardiac tissue contracts in a primarily spontaneously manner unlike skeletal muscle which will remain inactive in the absence of in the absence of stimulating inputs. Also, skeletal muscle is preferable over cardiac muscle due to its rate-response characteristics. As stimulation frequencies increase, contraction frequency and force generation of skeletal muscle will also increase until tetanus is induced, resulting in tonic contraction. Cardiac muscle, on the other hand, will cease to contract under high frequency stimulation, a situation similar to that of cardiac infarction (see appendices).

Furthermore it has been shown that neuronal cell types can be patterned and cocultured with skeletal muscle using surface chemistry and microfluidics. The ability to create organized neural/muscle cocultures will enable the creation of *in vitro* biological circuits that can be used for a variety of applications (pharmacology, basic science, biorobotics/bioprosthetics). Still it must be understood that many technological hurdles remain to be overcome to realize the full potential of this technology.

This technique holds particular promise for applications in drug discovery and as a model for various diseases involving skeletal muscle. The development of an *in vitro* model for functional biological circuits would greatly benefit the broader scientific community and society in general. By creating lab-on-chip systems that allow high-throughput, real-time experimentation, research costs would be reduced, data collection and analysis would be

simplified, and the need for costly and ethically questionable animal studies would be reduced.

CHAPTER FIVE: CONCLUDING REMARKS

The work presented in this dissertation is highly interdisciplinary in nature and draws upon such diverse fields as surface and analytical chemistry, cell biology, and engineering. The unifying theme throughout is the application of alkylsilane self-assembled monolayers as model substrates for controlling cell adhesion and differentiation and creating functional BioMEMS that can serve as *in vitro* models for studying cell function and interaction.

A new method for patterning cells using PEG and DETA terminated alkylsilane monolayers was demonstrate. By this method, cell repulsive PEG monolayers were photolithographically patterned to selectively remove regions of the PEG where cell adhesive islands of DETA monolayers could then be formed. These patterned substrates were then used for controlling the spatial orientation of cells in both pure and co-cultured systems. Multiple cell types were cultured on patterned PEG-DETA substrates demonstrating its broad applicability different tissues of interest. Furthermore it was shown that this combination of silane surfaces was suitable for creating long-term cultures that could be maintained up to 41 days. This is a particularly attractive capability to have, particularly when trying to create *in vitro* models for studying cellular development and function cellular process *in vivo* occur over a broad range of time scales (milliseconds to years). Thus, in order to create predictive models for cell biology it is critical to be able to maintain patterned cultures over the necessary time scales.

As the adsorption of biomolecules, such as proteins, to a substrate is critical to the attachment and function of cells growing on it, a novel WGM biosensor and fluidic delivery system was constructed to study how different proteins adsorb to alkylsilane monolayers. The adsorption of the extracellular matrix protein fibronectin was characterized with respect to the amount and nature of its interaction with our model surfaces. Results from WGM

measurements were then correlated with cell culture studies to verify the biological activity of FN on the various silanes. To further probe the affect of the underlying substrate on the biological activity of the protein the enzymatic protein glucose oxidase was used. By measuring the amount of protein adsorbed with the WGM system and correlating those results with those of a colorimetric activity assay for GO it was shown that measurable differences in the activity of the protein on DETA, 13F, PEG and glass were present. This is first known instance of being able to directly correlate quantitative protein adsorption measurements with enzymatic activity. Furthermore, for the case of GO on DETA it, evidence was presented for the possible formation of multiple layers of protein on the surface of the resonator. These results lend further evidence to the role of surface-protein interaction with biological activity.

Finally a novel BioMEMS device was constructed based on silicon microcantilevers and using the surface chemistry methods outlined previously. Embryonic skeletal myoblasts were grown on patterned microcantilevers and differentiate into functional myotubes capable of producing contractile stress sufficient to produce detectable bending of the cantilever. A custom built AFM detection system was devised to measure the contractile stress generated by the myotubes. Stress generation was quantified using a modified Stoney's. Values calculated using Stoney's equation were in excellent agreement with previously published results. This method was also used to study the effect of exogenously applied factors on muscle behavior. The results of this study validated the use of this system as the foundation for further work in developing *in vitro* models of cell behavior and development.

The chapters presented in this dissertation represent significant milestones towards developing *in vitro* systems for studying biology using alkylsilane modified substrates and BioMEMS devices. Much work remains to be done towards creating predictive model that

could be used as replacements for systems that are now in use. Current methods tend to focus on either simplifying experimental systems (cell culture, biochemical assays, etc.) for tightly controlling experimental variables, or using animal models that sacrifice experimental control for including complex cellular, tissue and systemic interactions that are otherwise difficult to recapitulate. The alternative to these approaches is to create methods and devices that can incorporate the best of both worlds. With BioMEMS and *in vitro* model systems it is possible to study complex interactions of multiple cell and tissue types while retaining a high degree of control over experimental variables, while interrogating and monitoring the components of interest in a real-time high-throughput manner. This is the ultimate potential of what this technology can achieve. Small, yet significant steps have been presented here towards this end.

APPENDIX A: PYTHON CODE FOR WGM SENSOR DATA ANALYSIS

```
# Name: wxmpl
# Purpose: painless matplotlib embedding for wxPython
# Author: Ken McIvor <mcivor@iit.edu>
#
# Copyright 2005-2006 Illinois Institute of Technology
#
# See the file "LICENSE" for information on usage and redistribution
# of this file, and for a DISCLAIMER OF ALL WARRANTIES.

"""
Embedding matplotlib in wxPython applications is straightforward, but the
default plotting widget lacks the capabilities necessary for interactive use.
WxMpl (wxPython+matplotlib) is a library of components that provide these
missing features in the form of a better matplolib FigureCanvas.
"""


import wx
import sys
import os.path
import weakref

import matplotlib
matplotlib.use('WXAgg')
import matplotlib.numerix as Numerix
from matplotlib.axes import PolarAxes, _process_plot_var_args
from matplotlib.backend_bases import FigureCanvasBase
from matplotlib.backends.backend_agg import FigureCanvasAgg, RendererAgg
from matplotlib.backends.backend_wxagg import FigureCanvasWxAgg
from matplotlib.figure import Figure
from matplotlib.font_manager import FontProperties
from matplotlib.transforms import Bbox, Point, Value
from matplotlib.transforms import bound_vertices, inverse_transform_bbox

__version__ = '1.2.9'

__all__ = ['PlotPanel', 'PlotFrame', 'PlotApp', 'StripCharter', 'Channel',
   'FigurePrinter', 'EVT_POINT', 'EVT_SELECTION']

# If you want to use something other than `lpr' to print under linux you may
# specify that command here.
LINUX_PRINTING_COMMAND = 'lpr'

# Work around some problems with the pre-0.84 WXAgg backend
BROKEN_WXAGG_BACKEND = matplotlib.__version__ < '0.84'

# Work around an API change in 0.90's matplotlib.axes._process_plot_var_args
```

```
PROCESS_PLOT_ARGS_REQUIRED_AXES = matplotlib.__version__ >= '0.90'

#
# Utility functions and classes
#

def is_polar(axes):
    """
    Returns a boolean indicating if C{axes} is a polar axes.
    """
    return isinstance(axes, PolarAxes)


def find_axes(canvas, x, y):
    """
    Finds the C{Axes} within a matplotlib C{FigureCanvas} contains the canavs
    coordinates C{(x, y)} and returns that axes and the corresponding data
    coordinates C{xdata, ydata} as a 3-tuple.

    If no axes contains the specified point a 3-tuple of C{None} is returned.
    """

    axes = None
    for a in canvas.get_figure().get_axes():
        if a.in_axes(x, y):
            if axes is None:
                axes = a
            else:
                return None, None, None

    if axes is None:
        return None, None, None

    xdata, ydata = axes.transData.inverse_xy_tup((x, y))
    return axes, xdata, ydata


def get_bbox_lims(bbox):
    """
    Returns the boundaries of the X and Y intervals of a C{Bbox}.
    """
    return bbox.intervalx().get_bounds(), bbox.intervaly().get_bounds()


def find_selected_axes(canvas, x1, y1, x2, y2):
    """
    Finds the C{Axes} within a matplotlib C{FigureCanvas} that overlaps with a
```

```
    canvas area from C{(x1, y1)} to C{(x1, y1)}.  That axes and the
    corresponding X and Y axes ranges are returned as a 3-tuple.

    If no axes overlaps with the specified area, or more than one axes
    overlaps, a 3-tuple of C{None}s is returned.
    """
    axes = None
    bbox = bound_vertices([(x1, y1), (x2, y2)])

    for a in canvas.get_figure().get_axes():
        if bbox.overlaps(a.bbox):
            if axes is None:
                axes = a
            else:
                return None, None, None

    if axes is None:
        return None, None, None

    xymin, xymax = limit_selection(bbox, axes)
    xrange, yrange = get_bbox_lims(
        inverse_transform_bbox(axes.transData, bound_vertices([xymin, xymax])))
    return axes, xrange, yrange


def limit_selection(bbox, axes):
    """
    Finds the region of a selection C{bbox} which overlaps with the supplied
    C{axes} and returns it as the 2-tuple C{((xmin, ymin), (xmax, ymax))}.
    """
    bxr, byr = get_bbox_lims(bbox)
    axr, ayr = get_bbox_lims(axes.bbox)

    xmin = max(bxr[0], axr[0])
    xmax = min(bxr[1], axr[1])
    ymin = max(byr[0], ayr[0])
    ymax = min(byr[1], ayr[1])
    return (xmin, ymin), (xmax, ymax)


def format_coord(axes, xdata, ydata):
    """
    A C{None}-safe version of {Axes.format_coord()}.
    """
    if xdata is None or ydata is None:
        return ''
    return axes.format_coord(xdata, ydata)
```

```
class AxesLimits:
    """
    Alters the X and Y limits of C{Axes} objects while maintaining a history of
    the changes.
    """
    def __init__(self):
        self.history = weakref.WeakKeyDictionary()

    def _get_history(self, axes):
        """
        Returns the history list of X and Y limits associated with C{axes}.
        """
        return self.history.setdefault(axes, [])

    def zoomed(self, axes):
        """
        Returns a boolean indicating whether C{axes} has had its limits
        altered.
        """
        return not (not self._get_history(axes))

    def set(self, axes, xrange, yrange):
        """
        Changes the X and Y limits of C{axes} to C{xrange} and {yrange}
        respectively.  A boolean indicating whether or not the
        axes should be redraw is returned, because polar axes cannot have
        their limits changed sensibly.
        """
        if is_polar(axes):
            return False

        history = self._get_history(axes)
        if history:
            oldRange = axes.get_xlim(), axes.get_ylim()
        else:
            oldRange = None, None

        history.append(oldRange)
        axes.set_xlim(xrange)
        axes.set_ylim(yrange)
        return True

    def restore(self, axes):
        """
        Changes the X and Y limits of C{axes} to their previous values.  A
```

```
        boolean indicating whether or not the axes should be redraw is
        returned.
        """
        hist = self._get_history(axes)
        if not hist:
            return False
        else:
            xrange, yrange = hist.pop()
            if xrange is None and yrange is None:
                axes.autoscale_view()
            else:
                axes.set_xlim(xrange)
                axes.set_ylim(yrange)
            return True


class DestructableViewMixin:
    """
    Utility class to break the circular reference between an object and its
    associated "view".
    """
    def destroy(self):
        """
        Sets this object's C{view} attribute to C{None}.
        """
        self.view = None


#
# Director of the matplotlib canvas
#

class PlotPanelDirector(DestructableViewMixin):
    """
    Encapsulates all of the user-interaction logic required by the
    C{PlotPanel}, following the Humble Dialog Box pattern proposed by Michael
    Feathers:
    U{http://www.objectmentor.com/resources/articles/TheHumbleDialogBox.pdf}
    """

    # TODO: merge all of the self.view.XYZ.something() methods into
    #       accessor methods of the PlotPanel (Law of Demeter fixes).
    # TODO: make `rightClickUnzoom' an option on PlotPanel, PlotFrame, etc
    # TODO: add a programmatic interface to zooming

    def __init__(self, view, zoom=True, selection=True, rightClickUnzoom=True):
        """
```

```
        Create a new director for the C{PlotPanel} C{view}.  The keyword
        arguments C{zoom} and C{selection} have the same meanings as for
        C{PlotPanel}.
        """
        self.view = view
        self.zoomEnabled = zoom
        self.selectionEnabled = selection
        self.rightClickUnzoom = rightClickUnzoom
        self.limits = AxesLimits()
        self.leftButtonPoint = None

    def setSelection(self, state):
        """
        Enable or disable left-click area selection.
        """
        self.selectionEnabled = state

    def setZoomEnabled(self, state):
        """
        Enable or disable zooming as a result of left-click area selection.
        """
        self.zoomEnabled = state

    def setRightClickUnzoom(self, state):
        """
        Enable or disable unzooming as a result of right-clicking.
        """
        self.rightClickUnzoom = state

    def canDraw(self):
        """
        Returns a boolean indicating whether or not the plot may be redrawn.
        """
        return self.leftButtonPoint is None

    def zoomed(self, axes):
        """
        Returns a boolean indicating whether or not the plot has been zoomed in
        as a result of a left-click area selection.
        """
        return self.limits.zoomed(axes)

    def keyDown(self, evt):
        """
        Handles wxPython key-press events.  These events are currently skipped.
        """
        evt.Skip()
```

```
def keyUp(self, evt):
    """
    Handles wxPython key-release events.  These events are currently
    skipped.
    """
    evt.Skip()

def leftButtonDown(self, evt, x, y):
    """
    Handles wxPython left-click events.
    """
    self.leftButtonPoint = (x, y)

    view = self.view
    axes, xdata, ydata = find_axes(view, x, y)

    if self.selectionEnabled and not is_polar(axes):
        view.cursor.setCross()
        view.crosshairs.clear()

def leftButtonUp(self, evt, x, y):
    """
    Handles wxPython left-click-release events.
    """
    if self.leftButtonPoint is None:
        return

    view = self.view
    axes, xdata, ydata = find_axes(view, x, y)

    x0, y0 = self.leftButtonPoint
    self.leftButtonPoint = None
    view.rubberband.clear()

    if x0 == x:
        if y0 == y and axes is not None:
            view.notify_point(axes, x, y)
            view.crosshairs.set(x, y)
        return
    elif y0 == y:
        return

    xdata = ydata = None
    axes, xrange, yrange = find_selected_axes(view, x0, y0, x, y)

    if axes is not None:
```

```
            xdata, ydata = axes.transData.inverse_xy_tup((x, y))
            if self.zoomEnabled:
                if self.limits.set(axes, xrange, yrange):
                    self.view.draw()
            else:
                bbox = bound_vertices([(x0, y0), (x, y)])
                (x1, y1), (x2, y2) = limit_selection(bbox, axes)
                self.view.notify_selection(axes, x1, y1, x2, y2)

        if axes is None:
            view.cursor.setNormal()
        elif is_polar(axes):
            view.cursor.setNormal()
            view.location.set(format_coord(axes, xdata, ydata))
        else:
            view.crosshairs.set(x, y)
            view.location.set(format_coord(axes, xdata, ydata))

    def rightButtonDown(self, evt, x, y):
        """
        Handles wxPython right-click events.  These events are currently
        skipped.
        """
        evt.Skip()

    def rightButtonUp(self, evt, x, y):
        """
        Handles wxPython right-click-release events.
        """
        view = self.view
        axes, xdata, ydata = find_axes(view, x, y)
        if (axes is not None and self.zoomEnabled and self.rightClickUnzoom
        and self.limits.restore(axes)):
            view.crosshairs.clear()
            view.draw()
            view.crosshairs.set(x, y)

    def mouseMotion(self, evt, x, y):
        """
        Handles wxPython mouse motion events, dispatching them based on whether
        or not a selection is in process and what the cursor is over.
        """
        view = self.view
        axes, xdata, ydata = find_axes(view, x, y)

        if self.leftButtonPoint is not None:
            self.selectionMouseMotion(evt, x, y, axes, xdata, ydata)
```

```
    else:
        if axes is None:
            self.canvasMouseMotion(evt, x, y)
        elif is_polar(axes):
            self.polarAxesMouseMotion(evt, x, y, axes, xdata, ydata)
        else:
            self.axesMouseMotion(evt, x, y, axes, xdata, ydata)

def selectionMouseMotion(self, evt, x, y, axes, xdata, ydata):
    """
    Handles wxPython mouse motion events that occur during a left-click
    area selection.
    """
    view = self.view
    x0, y0 = self.leftButtonPoint
    view.rubberband.set(x0, y0, x, y)
    if axes is None:
        view.location.clear()
    else:
        view.location.set(format_coord(axes, xdata, ydata))

def canvasMouseMotion(self, evt, x, y):
    """
    Handles wxPython mouse motion events that occur over the canvas.
    """
    view = self.view
    view.cursor.setNormal()
    view.crosshairs.clear()
    view.location.clear()

def axesMouseMotion(self, evt, x, y, axes, xdata, ydata):
    """
    Handles wxPython mouse motion events that occur over an axes.
    """
    view = self.view
    view.cursor.setCross()
    view.crosshairs.set(x, y)
    view.location.set(format_coord(axes, xdata, ydata))

def polarAxesMouseMotion(self, evt, x, y, axes, xdata, ydata):
    """
    Handles wxPython mouse motion events that occur over a polar axes.
    """
    view = self.view
    view.cursor.setNormal()
    view.location.set(format_coord(axes, xdata, ydata))
```

```
#
# Components used by the PlotPanel
#

class Painter(DestructableViewMixin):
    """
    Painters encapsulate the mechanics of drawing some value in a wxPython
    window and erasing it.  Subclasses override template methods to process
    values and draw them.

    @cvar PEN: C{wx.Pen} to use (defaults to C{wx.BLACK_PEN})
    @cvar BRUSH: C{wx.Brush} to use (defaults to C{wx.TRANSPARENT_BRUSH})
    @cvar FUNCTION: Logical function to use (defaults to C{wx.COPY})
    @cvar FONT: C{wx.Font} to use (defaults to C{wx.NORMAL_FONT})
    @cvar TEXT_FOREGROUND: C{wx.Colour} to use (defaults to C{wx.BLACK})
    @cvar TEXT_BACKGROUND: C{wx.Colour} to use (defaults to C{wx.WHITE})
    """

    PEN = wx.BLACK_PEN
    BRUSH = wx.TRANSPARENT_BRUSH
    FUNCTION = wx.COPY
    FONT = wx.NORMAL_FONT
    TEXT_FOREGROUND = wx.BLACK
    TEXT_BACKGROUND = wx.WHITE

    def __init__(self, view, enabled=True):
        """
        Create a new painter attached to the wxPython window C{view}.  The
        keyword argument C{enabled} has the same meaning as the argument to the
        C{setEnabled()} method.
        """
        self.view = view
        self.lastValue = None
        self.enabled = enabled

    def setEnabled(self, state):
        """
        Enable or disable this painter.  Disabled painters do not draw their
        values and calls to C{set()} have no effect on them.
        """
        oldState, self.enabled = self.enabled, state
        if oldState and not self.enabled:
            self.clear()

    def set(self, *value):
        """
```

```
        Update this painter's value and then draw it.  Values may not be
        C{None}, which is used internally to represent the absence of a current
        value.
        """
        if self.enabled:
            value = self.formatValue(value)
            self._paint(value, None)

    def redraw(self, dc=None):
        """
        Redraw this painter's current value.
        """
        value = self.lastValue
        self.lastValue = None
        self._paint(value, dc)

    def clear(self, dc=None):
        """
        Clear the painter's current value from the screen and the painter
        itself.
        """
        if self.lastValue is not None:
            self._paint(None, dc)

    def _paint(self, value, dc):
        """
        Draws a previously processed C{value} on this painter's window.
        """
        if dc is None:
            dc = wx.ClientDC(self.view)

        dc.SetPen(self.PEN)
        dc.SetBrush(self.BRUSH)
        dc.SetFont(self.FONT)
        dc.SetTextForeground(self.TEXT_FOREGROUND)
        dc.SetTextBackground(self.TEXT_BACKGROUND)
        dc.SetLogicalFunction(self.FUNCTION)
        dc.BeginDrawing()

        if self.lastValue is not None:
            self.clearValue(dc, self.lastValue)
            self.lastValue = None

        if value is not None:
            self.drawValue(dc, value)
            self.lastValue = value
```

```
        dc.EndDrawing()

    def formatValue(self, value):
        """
        Template method that processes the C{value} tuple passed to the
        C{set()} method, returning the processed version.
        """
        return value

    def drawValue(self, dc, value):
        """
        Template method that draws a previously processed C{value} using the
        wxPython device context C[113].  This DC has already been configured, so
        calls to C{BeginDrawing()} and C{EndDrawing()} may not be made.
        """
        pass

    def clearValue(self, dc, value):
        """
        Template method that clears a previously processed C{value} that was
        previously drawn, using the wxPython device context C[113].  This DC has
        already been configured, so calls to C{BeginDrawing()} and
        C{EndDrawing()} may not be made.
        """
        pass


class LocationPainter(Painter):
    """
    Draws a text message containing the current position of the mouse in the
    lower left corner of the plot.
    """

    PADDING = 2
    PEN = wx.WHITE_PEN
    BRUSH = wx.WHITE_BRUSH

    def formatValue(self, value):
        """
        Extracts a string from the 1-tuple C{value}.
        """
        return value[0]

    def get_XYWH(self, dc, value):
        """
        Returns the upper-left coordinates C{(X, Y)} for the string C{value}
        its width and height C{(W, H)}.
```

```
        """
        height = dc.GetSize()[1]
        w, h = dc.GetTextExtent(value)
        x = self.PADDING
        y = int(height - (h + self.PADDING))
        return x, y, w, h

    def drawValue(self, dc, value):
        """
        Draws the string C{value} in the lower left corner of the plot.
        """
        x, y, w, h = self.get_XYWH(dc, value)
        dc.DrawText(value, x, y)

    def clearValue(self, dc, value):
        """
        Clears the string C{value} from the lower left corner of the plot by
        painting a white rectangle over it.
        """
        x, y, w, h = self.get_XYWH(dc, value)
        dc.DrawRectangle(x, y, w, h)


class CrosshairPainter(Painter):
    """
    Draws crosshairs through the current position of the mouse.
    """

    PEN = wx.WHITE_PEN
    FUNCTION = wx.XOR

    def formatValue(self, value):
        """
        Converts the C{(X, Y)} mouse coordinates from matplotlib to wxPython.
        """
        x, y = value
        return int(x), int(self.view.get_figure().bbox.height() - y)

    def drawValue(self, dc, value):
        """
        Draws crosshairs through the C{(X, Y)} coordinates.
        """
        dc.CrossHair(*value)

    def clearValue(self, dc, value):
        """
        Clears the crosshairs drawn through the C{(X, Y)} coordinates.
```

```
        """
        dc.CrossHair(*value)


class RubberbandPainter(Painter):
    """
    Draws a selection rubberband from one point to another.
    """

    PEN = wx.WHITE_PEN
    FUNCTION = wx.XOR

    def formatValue(self, value):
        """
        Converts the C{(x1, y1, x2, y2)} mouse coordinates from matplotlib to
        wxPython.
        """
        x1, y1, x2, y2 = value
        height = self.view.get_figure().bbox.height()
        y1 = height - y1
        y2 = height - y2
        if x2 < x1: x1, x2 = x2, x1
        if y2 < y1: y1, y2 = y2, y1
        return [int(z) for z in (x1, y1, x2-x1, y2-y1)]

    def drawValue(self, dc, value):
        """
        Draws the selection rubberband around the rectangle
        C{(x1, y1, x2, y2)}.
        """
        dc.DrawRectangle(*value)

    def clearValue(self, dc, value):
        """
        Clears the selection rubberband around the rectangle
        C{(x1, y1, x2, y2)}.
        """
        dc.DrawRectangle(*value)


class CursorChanger(DestructableViewMixin):
    """
    Manages the current cursor of a wxPython window, allowing it to be switched
    between a normal arrow and a square cross.
    """
    def __init__(self, view, enabled=True):
        """
```

```
        Create a CursorChanger attached to the wxPython window C{view}.  The
        keyword argument C{enabled} has the same meaning as the argument to the
        C{setEnabled()} method.
        """
        self.view = view
        self.cursor = wx.CURSOR_DEFAULT
        self.enabled = enabled

    def setEnabled(self, state):
        """
        Enable or disable this cursor changer.  When disabled, the cursor is
        reset to the normal arrow and calls to the C{set()} methods have no
        effect.
        """
        oldState, self.enabled = self.enabled, state
        if oldState and not self.enabled and self.cursor != wx.CURSOR_DEFAULT:
            self.cursor = wx.CURSOR_DEFAULT
            self.view.SetCursor(wx.STANDARD_CURSOR)

    def setNormal(self):
        """
        Change the cursor of the associated window to a normal arrow.
        """
        if self.cursor != wx.CURSOR_DEFAULT and self.enabled:
            self.cursor = wx.CURSOR_DEFAULT
            self.view.SetCursor(wx.STANDARD_CURSOR)

    def setCross(self):
        """
        Change the cursor of the associated window to a square cross.
        """
        if self.cursor != wx.CURSOR_CROSS and self.enabled:
            self.cursor = wx.CURSOR_CROSS
            self.view.SetCursor(wx.CROSS_CURSOR)


#
# Printing Framework
#

# TODO: Map print quality settings onto PostScript resolutions automatically.
#       For now, it's set to something reasonable to work around the fact that
#       it defaults to `72' rather than `720' under wxPython 2.4.2.4
wx.PostScriptDC_SetResolution(300)


class FigurePrinter(DestructableViewMixin):
```

```
"""
Provides a simplified interface to the wxPython printing framework that's
designed for printing matplotlib figures.
"""

def __init__(self, view, printData=None):
    """
    Create a new C{FigurePrinter} associated with the wxPython widget
    C{view}.  The keyword argument C{printData} supplies a C{wx.PrintData}
    object containing the default printer settings.
    """
    self.view = view

    if printData is None:
        self.pData = wx.PrintData()
    else:
        self.pData = printData

def getPrintData(self):
    """
    Return the current printer settings in their C{wx.PrintData} object.
    """
    return self.pData

def setPrintData(self, printData):
    """
    Use the printer settings in C{printData}.
    """
    self.pData = printData

def pageSetup(self):
    dlg = wx.PrintDialog(self.view)
    pdData = dlg.GetPrintDialogData()
    pdData.SetPrintData(self.pData)
    pdData.SetSetupDialog(True)

    if dlg.ShowModal() == wx.ID_OK:
        self.pData = pdData.GetPrintData()
    dlg.Destroy()

def previewFigure(self, figure, title=None):
    """
    Open a "Print Preview" window for the matplotlib chart C{figure}.  The
    keyword argument C{title} provides the printing framework with a title
    for the print job.
    """
    window = self.view
```

```
        while not isinstance(window, wx.Frame):
            window = window.GetParent()
            assert window is not None

        fpo = FigurePrintout(figure, title)
        fpo4p = FigurePrintout(figure, title)
        preview = wx.PrintPreview(fpo, fpo4p, self.pData)
        frame = wx.PreviewFrame(preview, window, 'Print Preview')
        if self.pData.GetOrientation() == wx.PORTRAIT:
            frame.SetSize(wx.Size(450, 625))
        else:
            frame.SetSize(wx.Size(600, 500))
        frame.Initialize()
        frame.Show(True)

    def printFigure(self, figure, title=None):
        """
        Open a "Print" dialog to print the matplotlib chart C{figure}.  The
        keyword argument C{title} provides the printing framework with a title
        for the print job.
        """
        pdData = wx.PrintDialogData()
        pdData.SetPrintData(self.pData)
        printer = wx.Printer(pdData)
        fpo = FigurePrintout(figure, title)
        if printer.Print(self.view, fpo, True):
            self.pData = pdData.GetPrintData()


class FigurePrintout(wx.Printout):
    """
    Render a matplotlib C{Figure} to a page or file using wxPython's printing
    framework.
    """

    ASPECT_RECTANGULAR = 1
    ASPECT_SQUARE = 2

    def __init__(self, figure, title=None, size=None, aspectRatio=None):
        """
        Create a printout for the matplotlib chart C{figure}.  The
        keyword argument C{title} provides the printing framework with a title
        for the print job.  The keyword argument C{size} specifies how to scale
        the figure, from 1 to 100 percent.  The keyword argument C{aspectRatio}
        determines whether the printed figure will be rectangular or square.
        """
        self.figure = figure
```

```
        figTitle = figure.gca().title.get_text()
        if not figTitle:
            figTitle = title or 'Matplotlib Figure'

        if size is None:
            size = 100
        elif size < 0 or size > 100:
            raise ValueError('invalid figure size')
        self.size = size

        if aspectRatio is None:
            aspectRatio = self.ASPECT_RECTANGULAR
        elif (aspectRatio != self.ASPECT_RECTANGULAR
        and aspectRatio != self.ASPECT_SQUARE):
            raise ValueError('invalid aspect ratio')
        self.aspectRatio = aspectRatio

        wx.Printout.__init__(self, figTitle)

    def GetPageInfo(self):
        """
        Overrides wx.Printout.GetPageInfo() to provide the printing framework
        with the number of pages in this print job.
        """
        return (0, 1, 1, 1)

    def OnPrintPage(self, pageNumber):
        """
        Overrides wx.Printout.OnPrintPage to render the matplotlib figure to
        a printing device context.
        """
        # % of printable area to use
        imgPercent = max(1, min(100, self.size)) / 100.0

        # ratio of the figure's width to its height
        if self.aspectRatio == self.ASPECT_RECTANGULAR:
            aspectRatio = 1.61803399
        elif self.aspectRatio == self.ASPECT_SQUARE:
            aspectRatio = 1.0
        else:
            raise ValueError('invalid aspect ratio')

        # Device context to draw the page
        dc = self.GetDC()

        # PPI_P: Pixels Per Inch of the Printer
```

```
wPPI_P, hPPI_P = [float(x) for x in self.GetPPIPrinter()]
PPI_P = (wPPI_P + hPPI_P)/2.0

# PPI: Pixels Per Inch of the DC
if self.IsPreview():
   wPPI, hPPI = [float(x) for x in self.GetPPIScreen()]
else:
   wPPI, hPPI = wPPI_P, hPPI_P
PPI = (wPPI + hPPI)/2.0

# Pg_Px: Size of the page (pixels)
wPg_Px,  hPg_Px  = [float(x) for x in self.GetPageSizePixels()]

# Dev_Px: Size of the DC (pixels)
wDev_Px, hDev_Px = [float(x) for x in self.GetDC().GetSize()]

# Pg: Size of the page (inches)
wPg = wPg_Px / PPI_P
hPg = hPg_Px / PPI_P

# minimum margins (inches)
# TODO: make these arguments to __init__()
wM = 0.75
hM = 0.75

# Area: printable area within the margins (inches)
wArea = wPg - 2*wM
hArea = hPg - 2*hM

# Fig: printing size of the figure
# hFig is at a maximum when wFig == wArea
max_hFig = wArea / aspectRatio
hFig = min(imgPercent * hArea, max_hFig)
wFig = aspectRatio * hFig

# scale factor = device size / page size (equals 1.0 for real printing)
S = ((wDev_Px/PPI)/wPg + (hDev_Px/PPI)/hPg)/2.0

# Fig_S: scaled printing size of the figure (inches)
# M_S: scaled minimum margins (inches)
wFig_S = S * wFig
hFig_S = S * hFig
wM_S = S * wM
hM_S = S * hM

# Fig_Dx: scaled printing size of the figure (device pixels)
# M_Dx: scaled minimum margins (device pixels)
```

```
        wFig_Dx = int(S * PPI * wFig)
        hFig_Dx = int(S * PPI * hFig)
        wM_Dx = int(S * PPI * wM)
        hM_Dx = int(S * PPI * hM)

        image = self.render_figure_as_image(wFig, hFig, PPI)

        if self.IsPreview():
            image = image.Scale(wFig_Dx, hFig_Dx)
        self.GetDC().DrawBitmap(image.ConvertToBitmap(), wM_Dx, hM_Dx, False)

        return True

    def render_figure_as_image(self, wFig, hFig, dpi):
        """
        Renders a matplotlib figure using the Agg backend and stores the result
        in a C{wx.Image}.  The arguments C{wFig} and {hFig} are the width and
        height of the figure, and C{dpi} is the dots-per-inch to render at.
        """
        figure = self.figure

        old_dpi = figure.dpi.get()
        figure.dpi.set(dpi)
        old_width = figure.figwidth.get()
        figure.figwidth.set(wFig)
        old_height = figure.figheight.get()
        figure.figheight.set(hFig)
        old_frameon = figure.frameon
        figure.frameon = False

        wFig_Px = int(figure.bbox.width())
        hFig_Px = int(figure.bbox.height())

        agg = RendererAgg(wFig_Px, hFig_Px, Value(dpi))
        figure.draw(agg)

        figure.dpi.set(old_dpi)
        figure.figwidth.set(old_width)
        figure.figheight.set(old_height)
        figure.frameon = old_frameon

        image = wx.EmptyImage(wFig_Px, hFig_Px)
        image.SetData(agg.tostring_rgb())
        return image


#
```

```
# wxPython event interface for the PlotPanel and PlotFrame
#

EVT_POINT_ID = wx.NewId()


def EVT_POINT(win, id, func):
    """
    Register to receive wxPython C{PointEvent}s from a C{PlotPanel} or
    C{PlotFrame}.
    """
    win.Connect(id, -1, EVT_POINT_ID, func)


class PointEvent(wx.PyCommandEvent):
    """
    wxPython event emitted when a left-click-release occurs in a matplotlib
    axes of a window without an area selection.

    @cvar axes: matplotlib C{Axes} which was left-clicked
    @cvar x: matplotlib X coordinate
    @cvar y: matplotlib Y coordinate
    @cvar xdata: axes X coordinate
    @cvar ydata: axes Y coordinate
    """
    def __init__(self, id, axes, x, y):
        """
        Create a new C{PointEvent} for the matplotlib coordinates C{(x, y)} of
        an C{axes}.
        """
        wx.PyCommandEvent.__init__(self, EVT_POINT_ID, id)
        self.axes = axes
        self.x = x
        self.y = y
        self.xdata, self.ydata = axes.transData.inverse_xy_tup((x, y))

    def Clone(self):
        return PointEvent(self.GetId(), self.axes, self.x, self.y)


EVT_SELECTION_ID = wx.NewId()


def EVT_SELECTION(win, id, func):
    """
    Register to receive wxPython C{SelectionEvent}s from a C{PlotPanel} or
    C{PlotFrame}.
```

```
    """
    win.Connect(id, -1, EVT_SELECTION_ID, func)


class SelectionEvent(wx.PyCommandEvent):
    """
    wxPython event emitted when an area selection occurs in a matplotlib axes
    of a window for which zooming has been disabled.  The selection is
    described by a rectangle from C{(x1, y1)} to C{(x2, y2)}, of which only
    one point is required to be inside the axes.

    @cvar axes: matplotlib C{Axes} which was left-clicked
    @cvar x1: matplotlib x1 coordinate
    @cvar y1: matplotlib y1 coordinate
    @cvar x2: matplotlib x2 coordinate
    @cvar y2: matplotlib y2 coordinate
    @cvar x1data: axes x1 coordinate
    @cvar y1data: axes y1 coordinate
    @cvar x2data: axes x2 coordinate
    @cvar y2data: axes y2 coordinate
    """
    def __init__(self, id, axes, x1, y1, x2, y2):
        """
        Create a new C{SelectionEvent} for the area described by the rectangle
        from C{(x1, y1)} to C{(x2, y2)} in an C{axes}.
        """
        wx.PyCommandEvent.__init__(self, EVT_SELECTION_ID, id)
        self.axes = axes
        self.x1 = x1
        self.y1 = y1
        self.x2 = x2
        self.y2 = y2
        self.x1data, self.y1data = axes.transData.inverse_xy_tup((x1, y1))
        self.x2data, self.y2data = axes.transData.inverse_xy_tup((x2, y2))

    def Clone(self):
        return SelectionEvent(self.GetId(), self.axes, self.x1, self.y1,
            self.x2, self.y2)


#
# Matplotlib canvas in a wxPython window
#

class PlotPanel(FigureCanvasWxAgg):
    """
    A matplotlib canvas suitable for embedding in wxPython applications.
```

```
"""
def __init__(self, parent, id, size=(6.0, 3.70), dpi=96, cursor=True,
 location=True, crosshairs=True, selection=True, zoom=True):
    """
    Creates a new PlotPanel window that is the child of the wxPython window
    C{parent} with the wxPython identifier C{id}.

    The keyword arguments C{size} and {dpi} are used to create the
    matplotlib C{Figure} associated with this canvas.  C{size} is the
    desired width and height of the figure, in inches, as the 2-tuple
    C{(width, height)}.  C{dpi} is the dots-per-inch of the figure.

    The keyword arguments C{cursor}, C{location}, C{crosshairs},
    C{selection}, and C{zoom} enable or disable various user interaction
    features that are descibed in their associated C{set()} methods.
    """
    FigureCanvasWxAgg.__init__(self, parent, id, Figure(size, dpi))

    self.insideOnPaint = False
    self.cursor = CursorChanger(self, cursor)
    self.location = LocationPainter(self, location)
    self.crosshairs = CrosshairPainter(self, crosshairs)
    self.rubberband = RubberbandPainter(self, selection)
    self.director = PlotPanelDirector(self, zoom, selection)

    self.figure.set_edgecolor('black')
    self.figure.set_facecolor('white')
    self.SetBackgroundColour(wx.WHITE)

    # find the toplevel parent window and register an activation event
    # handler that is keyed to the id of this PlotPanel
    topwin = self._get_toplevel_parent()
    topwin.Connect(-1, self.GetId(), wx.wxEVT_ACTIVATE, self.OnActivate)

    wx.EVT_ERASE_BACKGROUND(self, self.OnEraseBackground)
    wx.EVT_WINDOW_DESTROY(self, self.OnDestroy)

def _get_toplevel_parent(self):
    """
    Returns the first toplevel parent of this window.
    """
    topwin = self.GetParent()
    while not isinstance(topwin, (wx.Frame, wx.Dialog)):
        topwin = topwin.GetParent()
    return topwin

def OnActivate(self, evt):
```

```
    """
    Handles the wxPython window activation event.
    """
    if not evt.GetActive():
        self.cursor.setNormal()
        self.location.clear()
        self.crosshairs.clear()
        self.rubberband.clear()
    evt.Skip()

def OnEraseBackground(self, evt):
    """
    Overrides the wxPython backround repainting event to reduce flicker.
    """
    pass

def OnDestroy(self, evt):
    """
    Handles the wxPython window destruction event.
    """
    if self.GetId() == evt.GetEventObject().GetId():
        objects = [self.cursor, self.location, self.rubberband,
            self.crosshairs, self.director]
        for obj in objects:
            obj.destroy()

        # unregister the activation event handler for this PlotPanel
        topwin = self._get_toplevel_parent()
        topwin.Disconnect(-1, self.GetId(), wx.wxEVT_ACTIVATE)

def _onPaint(self, evt):
    """
    Overrides the C{FigureCanvasWxAgg} paint event to redraw the
    crosshairs, etc.
    """
    if not isinstance(self, FigureCanvasWxAgg):
        return

    self.insideOnPaint = True
    FigureCanvasWxAgg._onPaint(self, evt)
    self.insideOnPaint = False

    dc = wx.PaintDC(self)
    self.location.redraw(dc)
    self.crosshairs.redraw(dc)
    self.rubberband.redraw(dc)
```

```
def get_figure(self):
    """
    Returns the figure associated with this canvas.
    """
    return self.figure

def set_cursor(self, state):
    """
    Enable or disable the changing mouse cursor.  When enabled, the cursor
    changes from the normal arrow to a square cross when the mouse enters a
    matplotlib axes on this canvas.
    """
    self.cursor.setEnabled(state)

def set_location(self, state):
    """
    Enable or disable the display of the matplotlib axes coordinates of the
    mouse in the lower left corner of the canvas.
    """
    self.location.setEnabled(state)

def set_crosshairs(self, state):
    """
    Enable or disable drawing crosshairs through the mouse cursor when it
    is inside a matplotlib axes.
    """
    self.crosshairs.setEnabled(state)

def set_selection(self, state):
    """
    Enable or disable area selections, where user selects a rectangular
    area of the canvas by left-clicking and dragging the mouse.
    """
    self.rubberband.setEnabled(state)
    self.director.setSelection(state)

def set_zoom(self, state):
    """
    Enable or disable zooming in when the user makes an area selection and
    zooming out again when the user right-clicks.
    """
    self.director.setZoomEnabled(state)

def zoomed(self, axes):
    """
    Returns a boolean indicating whether or not the C{axes} is zoomed in.
    """
```

```
    return self.director.zoomed(axes)

def draw(self, repaint=True):
    """
    Draw the associated C{Figure} onto the screen.
    """
    if (not self.director.canDraw()
    or  not isinstance(self, FigureCanvasWxAgg)):
        return

    # Before matplotlib 0.84, FigureCanvasWxAgg.draw() always called
    # gui_repaint(), which redrew the plot using a ClientDC.  This is
    # a workaround that lets us repaint the plot decorations in a sane
    # manner.

    doRepaint = repaint and not self.insideOnPaint
    if BROKEN_WXAGG_BACKEND:
        FigureCanvasAgg.draw(self)
        s = self.tostring_rgb()
        w = int(self.renderer.width)
        h = int(self.renderer.height)
        image = wx.EmptyImage(w, h)
        image.SetData(s)
        self.bitmap = image.ConvertToBitmap()

        # Don't repaint when called by _onPaint()
        if doRepaint:
            self.gui_repaint()
    else:
        FigureCanvasWxAgg.draw(self, repaint)

    # Don't redraw the decorations when called by _onPaint()
    if doRepaint:
        self.location.redraw()
        self.crosshairs.redraw()
        self.rubberband.redraw()

def notify_point(self, axes, x, y):
    """
    Called by the associated C{PlotPanelDirector} to emit a C{PointEvent}.
    """
    wx.PostEvent(self, PointEvent(self.GetId(), axes, x, y))

def notify_selection(self, axes, x1, y1, x2, y2):
    """
    Called by the associated C{PlotPanelDirector} to emit a
    C{SelectionEvent}.
```

```
        """
        wx.PostEvent(self, SelectionEvent(self.GetId(), axes, x1, y1, x2, y2))

    def _get_canvas_xy(self, evt):
        """
        Returns the X and Y coordinates of a wxPython event object converted to
        matplotlib canavas coordinates.
        """
        return evt.GetX(), int(self.figure.bbox.height() - evt.GetY())

    def _onKeyDown(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} key-press event handler, dispatching
        the event to the associated C{PlotPanelDirector}.
        """
        self.director.keyDown(evt)

    def _onKeyUp(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} key-release event handler,
        dispatching the event to the associated C{PlotPanelDirector}.
        """
        self.director.keyUp(evt)

    def _onLeftButtonDown(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} left-click event handler,
        dispatching the event to the associated C{PlotPanelDirector}.
        """
        x, y = self._get_canvas_xy(evt)
        self.director.leftButtonDown(evt, x, y)

    def _onLeftButtonUp(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} left-click-release event handler,
        dispatching the event to the associated C{PlotPanelDirector}.
        """
        x, y = self._get_canvas_xy(evt)
        self.director.leftButtonUp(evt, x, y)

    def _onRightButtonDown(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} right-click event handler,
        dispatching the event to the associated C{PlotPanelDirector}.
        """
        x, y = self._get_canvas_xy(evt)
        self.director.rightButtonDown(evt, x, y)
```

```
    def _onRightButtonUp(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} right-click-release event handler,
        dispatching the event to the associated C{PlotPanelDirector}.
        """
        x, y = self._get_canvas_xy(evt)
        self.director.rightButtonUp(evt, x, y)

    def _onMotion(self, evt):
        """
        Overrides the C{FigureCanvasWxAgg} mouse motion event handler,
        dispatching the event to the associated C{PlotPanelDirector}.
        """
        x, y = self._get_canvas_xy(evt)
        self.director.mouseMotion(evt, x, y)


#
# Matplotlib canvas in a top-level wxPython window
#

class PlotFrame(wx.Frame):
    """
    A matplotlib canvas embedded in a wxPython top-level window.

    @cvar ABOUT_TITLE: Title of the "About" dialog.
    @cvar ABOUT_MESSAGE: Contents of the "About" dialog.
    """

    ABOUT_TITLE = 'About wxmpl.PlotFrame'
    ABOUT_MESSAGE = ('wxmpl.PlotFrame %s\n' %  __version__
        + 'Written by Ken McIvor <mcivor@iit.edu>\n'
        + 'Copyright 2005 Illinois Institute of Technology')

    def __init__(self, parent, id, title, size=(6.0, 3.7), dpi=96, cursor=True,
     location=True, crosshairs=True, selection=True, zoom=True, **kwds):
        """
        Creates a new PlotFrame top-level window that is the child of the
        wxPython window C{parent} with the wxPython identifier C{id} and the
        title of C{title}.

        All of the named keyword arguments to this constructor have the same
        meaning as those arguments to the constructor of C{PlotPanel}.

        Any additional keyword arguments are passed to the constructor of
        C{wx.Frame}.
```

```
        """
        wx.Frame.__init__(self, parent, id, title, **kwds)
        self.panel = PlotPanel(self, -1, size, dpi, cursor, location,
            crosshairs, selection, zoom)

        pData = wx.PrintData()
        pData.SetPaperId(wx.PAPER_LETTER)
        if callable(getattr(pData, 'SetPrinterCommand', None)):
            pData.SetPrinterCommand(LINUX_PRINTING_COMMAND)
        self.printer = FigurePrinter(self, pData)

        self.create_menus()
        sizer = wx.BoxSizer(wx.VERTICAL)
        sizer.Add(self.panel, 1, wx.ALL|wx.EXPAND, 5)
        self.SetSizer(sizer)
        self.Fit()

        wx.EVT_WINDOW_DESTROY(self, self.OnDestroy)

    def create_menus(self):
        mainMenu = wx.MenuBar()
        menu = wx.Menu()

        id = wx.NewId()
        menu.Append(id, '&Save As...\tCtrl+S',
            'Save a copy of the current plot')
        wx.EVT_MENU(self, id, self.OnMenuFileSave)

        # Printing under OSX doesn't work well because the DPI of the
        # printer is always reported as 72.  It will be disabled until print
        # qualities are mapped onto wx.PostScriptDC resolutions.

        if not sys.platform.startswith('darwin'):
            menu.AppendSeparator()

            id = wx.NewId()
            menu.Append(id, 'Page Set&up...',
                'Set the size and margins of the printed figure')
            wx.EVT_MENU(self, id, self.OnMenuFilePageSetup)

            id = wx.NewId()
            menu.Append(id, 'Print Pre&view...',
                'Preview the print version of the current plot')
            wx.EVT_MENU(self, id, self.OnMenuFilePrintPreview)

            id = wx.NewId()
            menu.Append(id, '&Print...\tCtrl+P', 'Print the current plot')
```

```
        wx.EVT_MENU(self, id, self.OnMenuFilePrint)

    menu.AppendSeparator()

    id = wx.NewId()
    menu.Append(id, '&Close Window\tCtrl+W',
        'Close the current plot window')
    wx.EVT_MENU(self, id, self.OnMenuFileClose)

    mainMenu.Append(menu, '&File')
    menu = wx.Menu()

    id = wx.NewId()
    menu.Append(id, '&About...', 'Display version information')
    wx.EVT_MENU(self, id, self.OnMenuHelpAbout)

    mainMenu.Append(menu, '&Help')
    self.SetMenuBar(mainMenu)

def OnDestroy(self, evt):
    if self.GetId() == evt.GetEventObject().GetId():
        self.printer.destroy()

def OnMenuFileSave(self, evt):
    """
    Handles File->Save menu events.
    """
    fileName = wx.FileSelector('Save Plot', default_extension='png',
        wildcard=('Portable Network Graphics (*.png)|*.png|'
            + 'Encapsulated Postscript (*.eps)|*.eps|All files (*.*)|*.*'),
        parent=self, flags=wx.SAVE|wx.OVERWRITE_PROMPT)

    if not fileName:
        return

    path, ext = os.path.splitext(fileName)
    ext = ext[1:].lower()

    if ext != 'png' and ext != 'eps':
        error_message = (
            'Only the PNG and EPS image formats are supported.\n'
            'A file extension of `png\' or `eps\' must be used.')
        wx.MessageBox(error_message, 'Error - plotit',
            parent=self, style=wx.OK|wx.ICON_ERROR)
        return

    try:
```

```
            self.panel.print_figure(fileName)
        except IOError, e:
            if e.strerror:
                err = e.strerror
            else:
                err = e

            wx.MessageBox('Could not save file: %s' % err, 'Error - plotit',
                parent=self, style=wx.OK|wx.ICON_ERROR)

    def OnMenuFilePageSetup(self, evt):
        """
        Handles File->Page Setup menu events
        """
        self.printer.pageSetup()

    def OnMenuFilePrintPreview(self, evt):
        """
        Handles File->Print Preview menu events
        """
        self.printer.previewFigure(self.get_figure())

    def OnMenuFilePrint(self, evt):
        """
        Handles File->Print menu events
        """
        self.printer.printFigure(self.get_figure())

    def OnMenuFileClose(self, evt):
        """
        Handles File->Close menu events.
        """
        self.Close()

    def OnMenuHelpAbout(self, evt):
        """
        Handles Help->About menu events.
        """
        wx.MessageBox(self.ABOUT_MESSAGE, self.ABOUT_TITLE, parent=self,
            style=wx.OK)

    def get_figure(self):
        """
        Returns the figure associated with this canvas.
        """
        return self.panel.figure
```

```
    def set_cursor(self, state):
        """
        Enable or disable the changing mouse cursor.  When enabled, the cursor
        changes from the normal arrow to a square cross when the mouse enters a
        matplotlib axes on this canvas.
        """
        self.panel.set_cursor(state)

    def set_location(self, state):
        """
        Enable or disable the display of the matplotlib axes coordinates of the
        mouse in the lower left corner of the canvas.
        """
        self.panel.set_location(state)

    def set_crosshairs(self, state):
        """
        Enable or disable drawing crosshairs through the mouse cursor when it
        is inside a matplotlib axes.
        """
        self.panel.set_crosshairs(state)

    def set_selection(self, state):
        """
        Enable or disable area selections, where user selects a rectangular
        area of the canvas by left-clicking and dragging the mouse.
        """
        self.panel.set_selection(state)

    def set_zoom(self, state):
        """
        Enable or disable zooming in when the user makes an area selection and
        zooming out again when the user right-clicks.
        """
        self.panel.set_zoom(state)

    def draw(self):
        """
        Draw the associated C{Figure} onto the screen.
        """
        self.panel.draw()


#
# wxApp providing a matplotlib canvas in a top-level wxPython window
#
```

```
class PlotApp(wx.App):
    """
    A wxApp that provides a matplotlib canvas embedded in a wxPython top-level
    window, encapsulating wxPython's nuts and bolts.

    @cvar ABOUT_TITLE: Title of the "About" dialog.
    @cvar ABOUT_MESSAGE: Contents of the "About" dialog.
    """

    ABOUT_TITLE = None
    ABOUT_MESSAGE = None

    def __init__(self, title="WxMpl", size=(6.0, 3.7), dpi=96, cursor=True,
     location=True, crosshairs=True, selection=True, zoom=True, **kwds):
        """
        Creates a new PlotApp, which creates a PlotFrame top-level window.

        The keyword argument C{title} specifies the title of this top-level
        window.

        All of other the named keyword arguments to this constructor have the
        same meaning as those arguments to the constructor of C{PlotPanel}.

        Any additional keyword arguments are passed to the constructor of
        C{wx.App}.
        """
        self.title = title
        self.size = size
        self.dpi = dpi
        self.cursor = cursor
        self.location = location
        self.crosshairs = crosshairs
        self.selection = selection
        self.zoom = zoom
        wx.App.__init__(self, **kwds)

    def OnInit(self):
        self.frame = panel = PlotFrame(None, -1, self.title, self.size,
            self.dpi, self.cursor, self.location, self.crosshairs,
            self.selection, self.zoom)

        if self.ABOUT_TITLE is not None:
            panel.ABOUT_TITLE = self.ABOUT_TITLE

        if self.ABOUT_MESSAGE is not None:
            panel.ABOUT_MESSAGE = self.ABOUT_MESSAGE
```

```
        panel.Show(True)
        return True

    def get_figure(self):
        """
        Returns the figure associated with this canvas.
        """
        return self.frame.get_figure()

    def set_cursor(self, state):
        """
        Enable or disable the changing mouse cursor.  When enabled, the cursor
        changes from the normal arrow to a square cross when the mouse enters a
        matplotlib axes on this canvas.
        """
        self.frame.set_cursor(state)

    def set_location(self, state):
        """
        Enable or disable the display of the matplotlib axes coordinates of the
        mouse in the lower left corner of the canvas.
        """
        self.frame.set_location(state)

    def set_crosshairs(self, state):
        """
        Enable or disable drawing crosshairs through the mouse cursor when it
        is inside a matplotlib axes.
        """
        self.frame.set_crosshairs(state)

    def set_selection(self, state):
        """
        Enable or disable area selections, where user selects a rectangular
        area of the canvas by left-clicking and dragging the mouse.
        """
        self.frame.set_selection(state)

    def set_zoom(self, state):
        """
        Enable or disable zooming in when the user makes an area selection and
        zooming out again when the user right-clicks.
        """
        self.frame.set_zoom(state)

    def draw(self):
        """
```

```
        Draw the associated C{Figure} onto the screen.
        """
        self.frame.draw()


#
# Automatically resizing vectors and matrices
#

class VectorBuffer:
    """
    Manages a Numerical Python vector, automatically growing it as necessary to
    accomodate new entries.
    """
    def __init__(self):
        self.data = Numerix.zeros((16,), Numerix.Float)
        self.nextRow = 0

    def clear(self):
        """
        Zero and reset this buffer without releasing the underlying array.
        """
        self.data[:] = 0.0
        self.nextRow = 0

    def reset(self):
        """
        Zero and reset this buffer, releasing the underlying array.
        """
        self.data = Numerix.zeros((16,), Numerix.Float)
        self.nextRow = 0

    def append(self, point):
        """
        Append a new entry to the end of this buffer's vector.
        """
        nextRow = self.nextRow
        data = self.data

        resize = False
        if nextRow == data.shape[0]:
            nR = int(Numerix.ceil(self.data.shape[0]*1.5))
            resize = True

        if resize:
            self.data = Numerix.zeros((nR,), Numerix.Float)
            self.data[0:data.shape[0]] = data
```

```
        self.data[nextRow] = point
        self.nextRow += 1

    def getData(self):
        """
        Returns the current vector or C{None} if the buffer contains no data.
        """
        if self.nextRow == 0:
            return None
        else:
            return self.data[0:self.nextRow]


class MatrixBuffer:
    """
    Manages a Numerical Python matrix, automatically growing it as necessary to
    accomodate new rows of entries.
    """
    def __init__(self):
        self.data = Numerix.zeros((16, 1), Numerix.Float)
        self.nextRow = 0

    def clear(self):
        """
        Zero and reset this buffer without releasing the underlying array.
        """
        self.data[:, :] = 0.0
        self.nextRow = 0

    def reset(self):
        """
        Zero and reset this buffer, releasing the underlying array.
        """
        self.data = Numerix.zeros((16, 1), Numerix.Float)
        self.nextRow = 0

    def append(self, row):
        """
        Append a new row of entries to the end of this buffer's matrix.
        """
        row = Numerix.asarray(row, Numerix.Float)
        nextRow = self.nextRow
        data = self.data
        nPts = row.shape[0]

        if nPts == 0:
```

```
            return

        resize = True
        if nextRow == data.shape[0]:
            nC = data.shape[1]
            nR = int(Numerix.ceil(self.data.shape[0]*1.5))
            if nC < nPts:
                nC = nPts
        elif data.shape[1] < nPts:
            nR = data.shape[0]
            nC = nPts
        else:
            resize = False

        if resize:
            self.data = Numerix.zeros((nR, nC), Numerix.Float)
            rowEnd, colEnd = data.shape
            self.data[0:rowEnd, 0:colEnd] = data

        self.data[nextRow, 0:nPts] = row
        self.nextRow += 1

    def getData(self):
        """
        Returns the current matrix or C{None} if the buffer contains no data.
        """
        if self.nextRow == 0:
            return None
        else:
            return self.data[0:self.nextRow, :]


#
# Utility functions used by the StripCharter
#

def make_delta_bbox(X1, Y1, X2, Y2):
    """
    Returns a C{Bbox} describing the range of difference between two sets of X
    and Y coordinates.
    """
    return make_bbox(get_delta(X1, X2), get_delta(Y1, Y2))


def get_delta(X1, X2):
    """
    Returns the vector of contiguous, different points between two vectors.
```

```
    """
    n1 = X1.shape[0]
    n2 = X2.shape[0]

    if n1 < n2:
        return X2[n1:]
    elif n1 == n2:
        # shape is no longer a reliable indicator of change, so assume things
        # are different
        return X2
    else:
        return X2


def make_bbox(X, Y):
    """
    Returns a C{Bbox} that contains the supplied sets of X and Y coordinates.
    """
    if X is None or X.shape[0] == 0:
        x1 = x2 = 0.0
    else:
        x1 = min(X)
        x2 = max(X)

    if Y is None or Y.shape[0] == 0:
        y1 = y2 = 0.0
    else:
        y1 = min(Y)
        y2 = max(Y)

    return Bbox(Point(Value(x1), Value(y1)), Point(Value(x2), Value(y2)))


#
# Strip-charts lines using a matplotlib axes
#

class StripCharter:
    """
    Plots and updates lines on a matplotlib C{Axes}.
    """
    def __init__(self, axes):
        """
        Create a new C{StripCharter} associated with a matplotlib C{axes}.
        """
        self.axes = axes
        self.channels = []
```

```
    self.lines = {}

def setChannels(self, channels):
    """
    Specify the data-providers of the lines to be plotted and updated.
    """
    self.lines = None
    self.channels = channels[:]

    # minimal Axes.cla()
    self.axes.legend_ = None
    self.axes.lines = []

def update(self):
    """
    Redraw the associated axes with updated lines if any of the channels'
    data has changed.
    """
    axes = self.axes
    figureCanvas = axes.figure.canvas
    zoomed = figureCanvas.zoomed(axes)

    redraw = False
    if self.lines is None:
        self._create_plot()
        redraw = True
    else:
        for channel in self.channels:
            redraw = self._update_channel(channel, zoomed) or redraw

    if redraw:
        if not zoomed:
            axes.autoscale_view()
        figureCanvas.draw()

def _create_plot(self):
    """
    Initially plot the lines corresponding to the data-providers.
    """
    self.lines = {}
    axes = self.axes

    if PROCESS_PLOT_ARGS_REQUIRED_AXES:
        styleGen = _process_plot_var_args(axes)
    else:
        styleGen = _process_plot_var_args()
```

```
        for channel in self.channels:
            self._plot_channel(channel, styleGen)

        if self.channels:
            lines  = [self.lines[x] for x in self.channels]
            labels = [x.get_label() for x in lines]
            self.axes.legend(lines, labels, pad=0.1, axespad=0.0, numpoints=2,
                handlelen=0.02, handletextsep=0.01,
                prop=FontProperties(size='xx-small'))

#       # Draw the legend on the figure instead...
#       handles = [self.lines[x] for x in self.channels]
#       labels = [x._label for x in handles]
#       self.axes.figure.legend(handles, labels, 'upper right',
#           pad=0.1, handlelen=0.02, handletextsep=0.01, numpoints=2,
#           prop=FontProperties(size='xx-small'))

    def _plot_channel(self, channel, styleGen):
        """
        Initially plot a line corresponding to one of the data-providers.
        """
        empty = False
        x = channel.getX()
        y = channel.getY()
        if x is None or y is None:
            x = y = []
            empty = True

        line = styleGen(x, y).next()
        line._wxmpl_empty_line = empty

        if channel.getColor() is not None:
            line.set_color(channel.getColor())
        if channel.getStyle() is not None:
            line.set_linestyle(channel.getStyle())
        if channel.getMarker() is not None:
            line.set_marker(channel.getMarker())
            line.set_markeredgecolor(line.get_color())
            line.set_markerfacecolor(line.get_color())

        line.set_label(channel.getLabel())
        self.lines[channel] = line
        if not empty:
            self.axes.add_line(line)

    def _update_channel(self, channel, zoomed):
        """
```

```
        Replot a line corresponding to one of the data-providers if the data
        has changed.
        """
        if channel.hasChanged():
            channel.setChanged(False)
        else:
            return False

        axes = self.axes
        line = self.lines[channel]
        newX = channel.getX()
        newY = channel.getY()

        if newX is None or newY is None:
            return False

        oldX = line._x
        oldY = line._y

        x, y = newX, newY
        line.set_data(x, y)

        if line._wxmpl_empty_line:
            axes.add_line(line)
            line._wxmpl_empty_line = False
        else:
            if line.get_transform() != axes.transData:
                xys = axes._get_verts_in_data_coords(
                    line.get_transform(), zip(x, y))
                x = Numerix.array([a for (a, b) in xys])
                y = Numerix.array([b for (a, b) in xys])
            axes.update_datalim_numerix(x, y)

        if zoomed:
            return axes.viewLim.overlaps(
                make_delta_bbox(oldX, oldY, newX, newY))
        else:
            return True


#
# Data-providing interface to the StripCharter
#

class Channel:
    """
    Provides data for a C{StripCharter} to plot.  Subclasses of C{Channel}
```

```
override the template methods C{getX()} and C{getY()} to provide plot data
and call C{setChanged(True)} when that data has changed.
"""
def __init__(self, name, color=None, style=None, marker=None):
    """
    Creates a new C{Channel} with the matplotlib label C{name}.  The
    keyword arguments specify the strings for the line color, style, and
    marker to use when the line is plotted.
    """
    self.name = name
    self.color = color
    self.style = style
    self.marker = marker
    self.changed = False

def getLabel(self):
    """
    Returns the matplotlib label for this channel of data.
    """
    return self.name

def getColor(self):
    """
    Returns the line color string to use when the line is plotted, or
    C{None} to use an automatically generated color.
    """
    return self.color

def getStyle(self):
    """
    Returns the line style string to use when the line is plotted, or
    C{None} to use the default line style.
    """
    return self.style

def getMarker(self):
    """
    Returns the line marker string to use when the line is plotted, or
    C{None} to use the default line marker.
    """
    return self.marker

def hasChanged(self):
    """
    Returns a boolean indicating if the line data has changed.
    """
    return self.changed
```

```
def setChanged(self, changed):
    """
    Sets the change indicator to the boolean value C{changed}.

    @note: C{StripCharter} instances call this method after detecting a
    change, so a C{Channel} cannot be shared among multiple charts.
    """
    self.changed = changed

def getX(self):
    """
    Template method that returns the vector of X axis data or C{None} if
    there is no data available.
    """
    return None

def getY(self):
    """
    Template method that returns the vector of Y axis data or C{None} if
    there is no data available.
    """
    return None
```

APPENDIX B: MATLAB CODE FOR AVERAGING AND PLOTTING WGM DATA

```
%% load Excel file
global oldpath
if ~isvarname('oldpath') | length(oldpath) == 0
   oldpath = 'u:\messungen\tida\';
end
[filename, pathname] = uigetfile({'*.xls'},'Open Excel file',oldpath);
if ~filename, break, end
filename = [pathname filename];
oldpath = pathname;
data = xlsread(filename);
clear oldpath
%% split time and data columns
t = data(:,1:2:end);
d = data(:,2:2:end);
clear data
%% crate linear time and interpolate data
time = 1:max(max(t));
for i = 1:size(d')
   temp = d(:,i);
   temp(isnan(temp))=[];
   data(:,i) = interp1(t(1:length(temp),i), temp, time, 'linear', temp(end));
end
clear t d temp
```

```
%% open windows and fetch links to axes
f1 = figure('name', '(Press ESC to exit)');
a1 = subplot(2,1,1);
a2 = subplot(2,1,2);
%% GUI
b = 0;
while b(1) ~= 27 % ESC-key
    % plot data
    p = plot(data, 'parent', a1);
    xlabel('Time [s]', 'parent', a1);
    ylabel('something [idk]', 'parent', a1);
    title('-- raw data --', 'parent', a1);

    % calculate mean and std
    m = mean(data,2);
    s = std(data,0,2);

    % plot results
    plot([m+s m-s], 'color', [.8 .8 .8], 'parent', a2); hold(a2);
    plot(mean(data,2), 'k', 'linewidth', 2.5, 'parent', a2); hold(a2);
    xlabel('Time [s]', 'parent', a2);
    ylabel('something [idk]', 'parent', a2);
    title('-- results --', 'parent', a2);
```

```
    % actual GUI-stuff
    b = 0;
    while (b(1) ~= 27) && (b(1) ~= 127) % wait for either ESC- or DEL-key
        k = waitforbuttonpress;
        if k
            b = uint8(get(gcf,'CurrentCharacter'));
        end
    end
    data = [];
    for i = 1:length(p) % check which data set has been deleted
        try
            data(:,end+1) = get(p(i), 'YData')';
        catch
        end
    end
end
clear data b a1 a2
%% close windows
try
    close(f1);
catch
end
clear f1
%% write results to an Excel file
```

```
    disp(['writing average data and std to "' filename(length(pathname)+1:end-4)
'_averaged.xls" ...']);
    xlswrite([filename(1:end-4) '_averaged.xls'], [time' m s]);
    clear time m s pathname
```

APPENDIX C: MATLAB SCRIPT FOR PERFORMING STONEY'S CALCULATION

```
%This script converts raw data from the PSD into forces generated
%by cultured myotubes on rectangular cantilevers fabricated from
%crystalline silicon.
function [time,sigma]=cantilever(fileName)
file=fopen(fileName,'r');
data=char(fread(file));
data=str2num(reshape(data,1,length(data)));
ymeas=data(:,2)*100;%voltage from PSDY
xmeas=data(:,3)*100;%voltage from PSDX
time=data(:,1);%Time in seconds
trigger=data(:,4)/10;%Stimulus Trigger
ESi=130*10^9;%Elastic modulus of crystalline silicon
tSi=4*10^-6;%Thickness of the cantilever in meters
vf=0.3;%Poisson's ratio of muscle tissue
vSi=0.28;%poisson's ratio of silicon
L=7.55*10^-4;%length of the cantilever
l=10.5*10^-2;%path length of laser
width=1.0*10^-4;%width of the cantilever
Emuscle=10^4;%Elastic modulus of muscle tissue
Ef=Emuscle;
tF=1*10^-5;%Thickness of the muscle tissue
I=width*tSi^3/12;%Bending moment of cantilever
deltay=ymeas/(cos(.45379)*97.5*10^6);%change of the laser position in the y-direction
deltax=xmeas/(cos(.45379)*97.5*10^6);%change of the laser position in the x-direction
```

```
thetay=deltay/(2*l);%Theta calculation y axis
thetax=deltax/(2*l);%Theta calculation for x axis
Ry=L./thetay;%radius of curvature in the y axis
Rx=L./thetax;%radius of curvature in the x axis
if(Ry==0)%avoids divde by zero errors for ymeas
  sigmay=0;
else
  sigmay=1/(6*tF)./Ry*(ESi/(1-vSi)*tSi^3/(tF+tSi));%stress generated in the y axis
end;
if(Rx==0)%avoids divide by zero errors for xmeas
  sigmax=0;
else
  sigmax=1/(6*tF)./Rx*(ESi/(1-vSi)*tSi^3/(tF+tSi));%stress generated in the x axis
  end;
Fy=sigmay*tF*width;%converts sigmay into a force
Fx=sigmax*tF*width;%converts sigmax into a force
%subplot(3,3,1);plot(time,sigmay);%plots sigmay vs. time
%subplot(3,3,4);plot(time,sigmax);%plots sigmax vs. time
%subplot(3,3,7);plot(time,trigger);%plots the trigger voltage vs. time
%subplot(3,2,1);plot(time,Fy)
%xlabel('time(s)')
%ylabel('Fy(Newtons)')
%title('Bending Force');%plots Fy vs. time
%subplot(3,2,3);plot(time,Fx)
```

```
%label('time(s)')
%ylabel('Fx(Newtons)')
%title('Torsional Force');%plots Fx vs. time
%subplot(3,2,5);plot(time,trigger)
%xlabel('time(s)')
%ylabel('Trigger')
%title('Trigger trace');%plots the trigger voltage vs. time
subplot(3,1,1);plot(time,sigmay)
xlabel('time(s)')
ylabel('Stress(Pa)')
title('Bending Stress');%plots Fy vs. time
subplot(3,1,2);plot(time,sigmax)
xlabel('time(s)')
ylabel('Stress(Pa)')
title('Torsional Stress');%plots Fx vs. time
subplot(3,1,3);plot(time,trigger)
xlabel('time(s)')
ylabel('Trigger')
title('Trigger trace');
%save GapFree_10umfilm_sigmax1.xls sigmax -ASCII
%save GapFree_10umfilm_sigmay1.xls sigmay -ASCII
%save GapFree_10umfilm_Fx1.xls Fx -ASCII
%save GapFree_10umfilm_Fy1.xls Fy -ASCII
%save GapFree_10umfilm_Trigger1.xls trigger -ASCII
```

LIST OF REFERENCES

1. Dittrich, P. S.; Manz, A., Lab-on-a-chip: microfluidics in drug discovery. *Nat. Rev. Drug Discov.* **2006,** *5*, 210.

2. Duffy, D.; McDonald, J.; Schueller, O.; Whitesides, G. M., Rapid prototyping of microfluidic systems in poly(dimethylsiloxane). *Anal. Chem.* **1998,** *70*, 4974.

3. Easley, C. J.; Karlinsey, J. M.; Bienvenue, J. M.; Legendre, L. A.; Roper, M. G., A fully integrated microfluidic genetic analysis system with sample-in-answer-out capability. *Proc. Natl. Acad. Sci. USA* **2006,** *103*, 19272.

4. Harrison, D. J.; Fluri, K.; Seiler, K.; Fan, Z.; Effenhauser, C. S.; Manz, A., Micromachining a miniaturized capillary electrophoresis-based chemical analysis system on a chip. *Science* **1993,** *261*, 895.

5. Spetzler, D.; York, J.; Dobbin, C.; Martin, J.; Ishmukhametov, R.; Day, L.; Yu, J.; Kang, H.; Porter, K.; Hornung, T.; Frasch, W. D., Recent developments of bio-molecular motors as on-chip devices using single molecule techniques. *Lab on a Chip* **2007,** *7* (12), 1633-1643.

6. Vilkner, T.; Janasek, D.; Manz, A., Micro Total Analysis Systems. Recent Developments. *Anal. Chem.* **2004,** *76* (12), 3373-3386.

7. Voldman, J.; Gray, M. L.; Schmidt, M. A., Microfabrication in biology and medicine. *Annu. Rev. Biomed. Eng.* **1999,** *1*, 401.

8. Whitesides, G. M., The origins and the future of microfluidics. *Nature* **2006,** *442* (7101), 368-373.

9. Mao, H.; Cremer, P. S.; Manson, M. D., A sensitive, versatile microfluidic assay for bacterial chemotaxis. *Proc. Natl. Acad. Sci. USA* **2003,** *100*, 5449.

10. Pancrazio, J. J.; Whelan, J. P.; Borkholder, D. A.; Ma, W.; Stenger, D. A., Development and Application of Cell-Based Biosensors. *Annals of Biomedical Engineering* **1999,** *27* (6), 697-711.

11. Park, J.; Bansal, T.; Pinelis, M.; Maharbiz, M. M., A microsystem for sensing and patterning oxidative microgradients during cell culture. *Lab Chip* **2006,** *6*, 611.

12. Ravenscroft, M. S.; Bateman, K. E.; Shaffer, K. M.; Schessler, H. M.; Jung, D. R.; Schneider, T. W.; Montgomery, C. B.; Custer, T. L.; Schaffner, A. E.; Liu, Q. Y.; Li, Y. X.; Barker, J. L.; Hickman, J. J., Developmental Neurobiology Implications from Fabrication and Analysis of Hippocampal Neuronal Networks on Patterned Silane-Modified Surfaces. *J. Am. Chem. Soc.* **1998,** *120* (47), 12169-12177.

13. Schaff, U. Y.; Xing, M. M.; Lin, K. K.; Pan, N.; Jeon, N. L.; Simon, S. I., Vascular mimetics based on microfluidics for imaging the leukocyte-endothelial inflammatory response. *Lab Chip* **2007,** *7*, 448.

14. Tan, J. L.; Tien, J.; Pirone, D. M.; Gray, D. S.; Bhadriraju, K.; Chen, C. S., Cells lying on a bed of microneedles: an approach to isolate mechanical force. *Proc. Natl. Acad. Sci. USA* **2003,** *100*, 1484.

15. Toh, Y. C.; Zhang, C.; Zhang, J.; Khong, Y. M.; Chang, S., A novel 3D mammalian cell perfusion-culture system in microfluidic channels. *Lab Chip* **2007,** *7*, 302.

16. Torisawa, Y.; Chueh, B. H.; Huh, D.; Ramamurthy, P.; Roth, T. M., Efficient formation of uniform-sized embryoid bodies using a compartmentalized microchannel device. *Lab Chip* **2007,** *7*, 770.

17. Tourovskaia, A.; Figueroa-Masot, X.; Folch, A., Long-term microfluidic cultures of myotube microarrays for high-throughput focal stimulation. *Nat. Protocols* **2006,** *1* (3), 1092-1104.

18. Tourovskaia, A.; Kosar, T. F.; Folch, A., Local Induction of Acetylcholine Receptor Clustering in Myotube Cultures Using Microfluidic Application of Agrin. *Biophys. J.* **2006,** *90* (6), 2192-2198.

19. Wang, Z.; Kin, M.; Marquez, M.; Thorsen, T., High-density microfluidic arrays for cell cytotoxicity analysis. *Lab Chip* **2007,** *7*, 740.

20. Wheeler, A. R.; Throndset, W. R.; Whelan, R. J.; Leach, A. M.; Zare, R. N.; Liao, Y. H.; Farrell, K.; Manger, I. D.; Daridon, A., Microfluidic Device for Single-Cell Analysis. *Anal. Chem.* **2003,** *75* (14), 3581-3586.

21. Yoo, S. K.; Lee, J. H.; Yun, S.-S.; Gu, M. B.; Lee, J. H., Fabrication of a bio-MEMS based cell-chip for toxicity monitoring. *Biosensors and Bioelectronics* **2007,** *22* (8), 1586-1592.

22. Klauke, N.; Smith, G.; Cooper, J. M., Microfluidic systems to examine intercellular coupling of pairs of cardiac myocytes. *Lab Chip* **2007,** *7*, 731.

23. El-Ali, J.; Sorger, P. K.; Jensen, K. F., Cells on chips. *Nature* **2006,** *442*, 403.

24. Feinberg, A. W.; Feigel, A.; Shevkoplyas, S. S.; Sheehy, S.; Whitesides, G. M.; Parker, K. K., Muscular Thin Films for Building Actuators and Powering Devices. *Science* **2007,** *317* (5843), 1366-1370.

25. Folch, A.; Toner, M., Microengineering of cellular interactions. *Annu. Rev. Biomed. Eng.* **2000,** *2*, 227.

26. Hui, E. E.; Bhatia, S. N., Micromechanical control of cell-cell interactions. *Proc. Natl. Acad. Sci. USA* **2007,** *104*, 5722.

27. Hung, P. J.; Lee, P. J.; Sabounchi, P.; Lin, R.; Lee, L. P., Continuous perfusion microfluidic cell culture array for high-throughput cell-based assays. *Biotechnol. Bioeng.* **2005,** *89*, 1.

28. Irimia, D.; Liu, S. Y.; Tharp, W. G.; Samadani, A.; Toner, M.; Poznansky, M. C., Microfluidic system for measuring neutrophil migratory responses to fast switches of chemical gradients. *Lab Chip* **2006,** *6*, 191.

29. Jeon, N. L.; Baskaran, H.; Dertinger, S. K. W.; Whitesides, G. M.; Van de Water, L.; Toner, M., Neutrophil chemotaxis in linear and complex gradients of interleukin-8 formed in a microfabricated device. *Nat. Biotechnol.* **2002,** *20*, 826.

30. Kim, M. S.; Yeon, J. H.; Park, J. K., A microfluidic platform for 3-dimensional cell culture and cell-based assays. *Biomed. Microdev.* **2007,** *9*, 25.

31. King, K. R.; Wang, S.; Irimia, D.; Jayaraman, A.; Toner, M.; Yarmush, M. L., A high-throughput microfluidic real-time gene expression living cell array. *Lab Chip* **2007,** *7*, 77.

32. Cornish, T.; Branch, D. W.; Wheeler, B. C.; Campanelli, J. T., Microcontact printing: A versatile technique for the study of synaptogenic molecules. *Molecular and Cellular Neuroscience* **2002,** *20* (1), 140.

33. Anderson, H.; van den Berg, A., Microfluidic devices for cellomics: a review. *Sens. Actuators B* **2003,** *92*, 315.

34. Chen, C. S.; Mrksich, M.; Huang, S.; Whitesides, G. M.; Ingber, D. E., Geometric control of cell life and death. *Science* **1997,** *276*, 1425.

35. Das, M.; Gregory, C. A.; Molnar, P.; Riedel, L. M.; Wilson, K.; Hickman, J. J., A defined system to allow skeletal muscle differentiation and subsequent integration with silicon microstructures. *Biomaterials* **2006,** *27* (24), 4374-4380.

36. Das, M.; Molnar, P.; Devaraj, H.; Poeta, M.; Hickman, J. J., Electrophysiological and morphological characterization of rat embryonic motoneurons in a defined system. *Biotechnol. Prog.* **2003,** *19* (6), 1756-61.

37. Das, M.; Wilson, K.; Molnar, P.; Hickman, J. J., Differentiation of skeletal muscle and integration of myotubes with silicon microstructures using serum-free medium and a synthetic silane substrate. *Nat. Protocols* **2007,** *2* (7), 1795-1801.

38. Wilson, K.; Molnar, P.; Hickman, J., Integration of functional myotubes with a Bio-MEMS device for non-invasive interrogation. *Lab on a Chip* **2007,** *7* (7), 920-922.

39. Yu, H.; Alexander, C. M.; Beebe, D. J., Understanding microchannel culture: parameters involved in soluble factor signaling. *Lab Chip* **2007,** *7*, 726.

40. Gottwald, E.; Giselbrecht, S.; Augspurger, C.; Lahni, B.; Dambrowsky, N., A chip-based platform for the in vitro generation of tissues in three-dimensional organization. *Lab Chip* **2007,** *7*, 777.

41. Kosar, T. F.; Tourovkaia, A.; Figueroa-Masot, X. A., M.E.; Folch, A., A nanofabricated planar aperture as a mimic of the nerve–muscle contact during synaptogenesis. *Lab on a chip* **2006,** *6* (5), 632.

42. Meyvantsson, I.; Beebe, D. J., Cell Culture Models in Microfluidic Systems. *Annual Review of Analytical Chemistry* **2008,** *1* (1), 423-449.

43. Park, T. H.; Shuler, M. L., Integration of Cell Culture and Microfabrication Technology. *Biotechnol. Prog.* **2003,** *19* (2), 243-253.

44. Sin, A.; Chin, K. C.; Jamil, M. F.; Kostov, Y.; Rao, G.; Shuler, M. L., The Design and Fabrication of Three-Chamber Microscale Cell Culture Analog Devices with Integrated Dissolved Oxygen Sensors. *Biotechnol. Prog.* **2004,** *20* (1), 338-345.

45. Tan, W.; Desai, T. A., Microscale multilayer cocultures for biomimetic blood vessels. *J. Biomed. Mater. Res.* **2004,** *72A*, 146.

46. Tao, S.; Young, C.; Redenti, S.; Zhang, Y.; Klassen, H., Survival, migration and differentiation of retinal progenitor cells transplanted on micro-machined poly(methyl methacrylate) scaffolds to the subretinal space. *Lab Chip* **2007,** *7*, 695.

47. Taylor, A. M.; Blurton-Jones, M.; Rhee, S. W.; Cribbs, D. H.; Cotman, C. W.; Jeon, N. L., A microfluidic culture platform for CNS axonal injury, regeneration and transport. *Nat. Methods* **2005,** *2*, 599.

48. Tsang, V. L.; Chen, A. A.; Cho, L. M.; Jadin, K. D.; Sah, R. L., Fabrication of 3D hepatic tissues by additive photopatterning of cellular hydrogels. *FASEB J.* **2007,** *21*, 790.

49. Ratner, B.; Hoffman, A.; Schoen, F.; Lemons, J., *Biomaterials science: an introduction to materials in medicine.* 2 ed.; Elsevier Academic Press: Amsterdam London, 2004.

50. Benjamin G. Keselowsky, D. M. C. A. J. G., Surface chemistry modulates fibronectin conformation and directs integrin binding and specificity to control cell adhesion. *Journal of Biomedical Materials Research Part A* **2003,** *66A* (2), 247-259.

51. Capadona, J. R.; Collard, D. M.; Garcia, A. J., Fibronectin Adsorption and Cell Adhesion to Mixed Monolayers of Tri(ethylene glycol)- and Methyl-Terminated Alkanethiols. *Langmuir* **2003,** *19* (5), 1847-1852.

52. Michael, K. E.; Vernekar, V. N.; Keselowsky, B. G.; Meredith, J. C.; Latour, R. A.; Garcia, A. J., Adsorption-Induced Conformational Changes in Fibronectin Due to Interactions with Well-Defined Surface Chemistries. *Langmuir* **2003,** *19* (19), 8033-8040.

53. Keselowsky, B. G.; Collard, D. M.; Garcia, A. J., Integrin binding specificity regulates biomaterial surface chemistry effects on cell differentiation. *Proceedings of the National Academy of Sciences of the United States of America* **2005,** *102* (17), 5953-5957.

54. Liu, L.; Chen, S.; Giachelli, C. M.; Ratner, B. D.; Jiang, S., Controlling osteopontin orientation on surfaces to modulate endothelial adhesion. *Journal of Biomedical Materials Research Part A* **2005,** *74A*, 23-31.

55. Blawas, A. S.; Reichert, W. M., Protein patterning. *Biomaterials* **1998,** *19* (7-9), 595-609.

56. Stenger, D. A.; Georger, J. H.; Dulcey, C. S.; Hickman, J. J.; Rudolph, A. S.; Nielsen, T. B.; McCort, S. M.; Calvert, J. M., Coplanar molecular assemblies of amino- and perfluorinated alkylsilanes: characterization and geometric definition of mammalian cell adhesion and growth. *J. Am. Chem. Soc.* **1992,** *114* (22), 8435-8442.

57. Stenger, D. A.; Hickman, J. J.; Bateman, K. E.; Ravenscroft, M. S.; Ma, W.; Pancrazio, J. J.; Shaffer, K.; Schaffner, A. E.; Cribbs, D. H.; Cotman, C. W., Microlithographic determination of axonal/dendritic polarity in cultured hippocampal neurons. *Journal of Neuroscience Methods* **1998,** *82* (2), 167-173.

58. Papra, A.; Gadegaard, N.; Larsen, N. B., Characterization of Ultrathin Poly(ethylene glycol) Monolayers on Silicon Substrates. *Langmuir* **2001,** *17* (5), 1457-1460.

59. Kind, H.; Bittner, A. M.; Cavalleri, O.; Kern, K.; Greber, T., Electroless Deposition of Metal Nanoislands on Aminothiolate-Functionalized Au(111) Electrodes. *J. Phys. Chem. B* **1998,** *102* (39), 7582-7589.

60. Das, M.; Bharqava, N.; Gregory, C.; Riedel, L.; Molnar, P.; Hickman, J. J., Adult rat spinal cord culture on an organosilane surface in a novel serum-free medium. *In Vitro Cellular Developmental Biology - Animal* **2005,** *41* (10), 343-348.

61. Schnaar, R. I.; Schaffner, A. E., Separation of cell types from embryonic chicken and rat spinal cord: characterization of motoneuron-enriched fractions. *J. Neurosci.* **1981,** *1* (2), 204-217.

62. Mettling, C.; Gouin, A.; Robinson, M.; el M'Hamdi, H.; Camu, W.; Bloch-Gallego, E.; Buisson, B.; Tanaka, H.; Davies, A. M.; Henderson, C. E., Survival of newly postmitotic motoneurons is transiently independent of exogenous trophic support. *J. Neurosci.* **1995,** *15* (4), 3128-3137.

63. Yan, Q.; Johnson, E. M., Jr., An immunohistochemical study of the nerve growth factor receptor in developing rats. *J. Neurosci.* **1988,** *8* (9), 3481-3498.

64. Schaffner, A. E.; Barker, J. L.; Stenger, D. A.; Hickman, J. J., Investigation of the factors necessary for growth of hippocampal neurons in a defined system. *Journal of Neuroscience Methods* **1995,** *62* (1-2), 111-119.

65. Brewer, G. J., Serum-free B27/neurobasal medium supports differentiated growth of neurons from the striatum, substantia nigra, septum, cerebral cortex, cerebellum, and dentate gyrus. *Journal of Neuroscience Research* **1995,** *42* (5), 674-683.

66. Brewer, G. J., Regeneration and Proliferation of Embryonic and Adult Rat Hippocampal Neurons in Culture. *Experimental Neurology* **1999,** *159* (1), 237-247.

67. G. J. Brewer, J. R. T. E. K. E. P. J. P., Optimized survival of hippocampal neurons in B27-supplemented neurobasal^{<FONT SIZE='-1'>TM</FONT>}, a new serum-free medium combination. *Journal of Neuroscience Research* **1993,** *35* (5), 567-576.

68. Stenger, D. A.; Pike, C. J.; Hickman, J. J.; Cotman, C. W., Surface determinants of neuronal survival and growth on self-assembled monolayers in culture. *Brain Research* **1993,** *630* (1-2), 136-147.

69. Andrade, J. D.; Hlady, V., Protein adsorption and materials compatibilti: A tutorial review and suggested hypotheses. *Advances in Polymer Science* **1986,** *79*, 1-63.

70. Jonsson, U.; Fagerstam, L.; Ivarsson, B.; Johnsson, B.; Karlsson, R.; Lundh, K.; Lofas, s.; Persson, B.; Roos, H.; Ronnberg, I.; al., e., Real-time biospecific interaction

analysis using surface plasmon resonance and a sensor chip technology. *Biotechniques* **1991,** *11* (5), 620-7.

71. Garbassi, F.; Morra, M.; Occhiello, E., *Polymer Surfaces: From Physics to Technology*. John Wiley and Sons Ltd.: West Sussex, 1998.

72. Balamurugan, S.; Ista, L. K.; Yan, J.; Lopez, G. P.; Fick, J.; Himmelhaus, M.; Grunze, M., Reversible Protein Adsorption and Bioadhesion on Monolayers Terminated with Mixtures of Oligo(ethylene glycol) and Methyl Groups. *J. Am. Chem. Soc.* **2005,** *127* (42), 14548-14549.

73. Vroman, L.; Adams, A. L., Adsorption of proteins out of plasma and solutions in narrow spaces. *Journal of Colloid and Interface Science* **1986,** *111* (2), 391-402.

74. Keselowsky, B. G.; Collard, D. M.; Garcia, A., Surface chemistry modulates fibronectin conformation and directs integrin binding and specificity to control cell adhesion. *Journal of Biomedical Materials Research Part A* **2003,** *66A*, 247-59.

75. Keselowsky, B. G.; Collard, D. M.; Garcia, A. J., Surface chemistry modulates focal adhesion composition and signaling through changes in integrin binding. *Biomaterials* **2004,** *25* (28), 5947-5954.

76. Keng, D.; McAnanama, S. R.; Teraoka, I.; Arnold, S., Resonance fluctuations of a whispering gallery mode biosensor by particles undergoing Brownian motion. *Applied Physics Letters* **2007,** *91* (10), 103902-3.

77. Vollmer, F.; Arnold, S.; Keng, D., Single virus detection from the reactive shift of a whispering-gallery mode. *Proc. Natl. Acad. Sci. USA* **2008,** *105* (52), 20701-20704.

78. Vollmer, F., Taking detection to the limit. *B.I.F. Futura* **2005,** *20*, 239-244.

79. Vollmer, F.; Braun, D.; Libchaber, A.; Khoshsima, M.; Teraoka, I.; Arnold, S., Protein detection by optical shift of a resonant microcavity. *Appl. Phys. Lett.* **2002,** *80* (21), 4057-4059.

80. Gorodetsky, M. L.; Savchenkov, A. A.; Ilchenko, V. S., Ultimate Q of optical microsphere resonators. *Optics Letters* **1996,** *21* (7), 453-455.

81. Quan, H.; Guo, Z.; Xu, L.; Pau, S. In *Design, fabrication, and characterization of whispering-gallery mode miniature sensors*, Nanofabrication: Technologies, Devices, and Applications, Philadelphia, PA, USA, SPIE: Philadelphia, PA, USA, 2005; pp 373-381.

82. Armani, D. K.; Kippenberg, T. J.; Spillane, S. M.; Vahala, K. J., Ultra-high-Q toroid microcavity on a chip. **2003,** *421* (6926), 925-928.

83. Topolancik, J.; Vollmer, F.; Ilic, B., Random high-Q cavities in disordered photonic crystal waveguides. *Applied Physics Letters* **2007,** *91* (20), 201102-3.

84. Vollmer, F.; Arnold, S.; Braun, D.; Teraoka, I.; Libchaber, A., Multiplexed DNA Quantification by Spectroscopic Shift of Two Microsphere Cavities. *Biophys. J.* **2003,** *85* (3), 1974-1979.

85. Vollmer, F.; Arnold, S., Whispering-gallery-mode biosensing: label-free detection down to single molecules. *Nature Methods* **2008,** *5* (7).

86. Garcia, A. J.; Huber, F.; Boettiger, D., Force Required to Break alpha 5beta 1 Integrin-Fibronectin Bonds in Intact Adherent Cells Is Sensitive to Integrin Activation State. *J. Biol. Chem.* **1998,** *273* (18), 10988-10993.

87. Lan, M. A.; Gersbach, C. A.; Michael, K. E.; Keselowsky, B. G.; García, A. J., Myoblast proliferation and differentiation on fibronectin-coated self assembled monolayers presenting different surface chemistries. *Biomaterials* **2005,** *26* (22), 4523-4531.

88. McGrath, M. J.; Mitchell, C. A.; Coghill, I. D.; Robinson, P. A.; Brown, S., Skeletal muscle LIM protein 1 (SLIM1/FHL1) induces {alpha}5{beta}1-integrin-dependent myocyte elongation. *Am J Physiol Cell Physiol* **2003,** *285* (6), C1513-1526.

89. Mohanty, J. G.; Jaffe, J. S.; Schulman, E. S.; Raible, D. G., A highly sensitive fluorescent micro-assay of H2O2 release from activated human leukocytes using a dihydroxyphenoxazine derivative. *Journal of Immunological Methods* **1997,** *202* (2), 133-141.

90. van der Veen, M.; Stuart, M. C.; Norde, W., Spreading of proteins and its effect on adsorption and desorption kinetics. *Colloids and Surfaces B: Biointerfaces* **2007,** *54* (2), 136-142.

91. Sivaraman, A.; Leach, J. K.; Townsend, S.; Iida, T.; Hogan, B. J.; Stolz, D. B.; Fry, R.; Samson, L. D.; Tannenbaum, S. R.; Griffith, L. G., A Microscale In Vitro Physiological Model of the Liver: Predictive Screens for Drug Metabolism and Enzyme Induction. *Current Drug Metabolism* **2005,** *6* (6), 569-591.

92. Mohan, D. K.; Molnar, P.; Hickman, J. J., Toxin detection based on action potential shape analysis using a realistic mathematical model of differentiated NG108-15 cells. *Biosensors and Bioelectronics* **2006,** *21* (9), 1804-1811.

93. Bousse, L., Whole cell biosensors. *Sensors and Actuators B: Chemical* **1996,** *34* (1-3), 270-275.

94. Xi, J.; Schmidt, J. J.; Montemagno, C. D., Self-assembled microdevices driven by muscle. *Nature Materials* **2005,** *4*, 180-184.

95. Tanaka, Y.; Sato, K.; Shimizu, T.; Yamato, M.; Okano, T.; Kitamori, T., A micro-spherical heart pump powered by cultured cardiomyocytes. *Lab on a Chip* **2007,** *7* (2), 207-212.

96. Boillee, S.; Vande Velde, C.; Cleveland, D. W., ALS: A Disease of Motor Neurons and Their Nonneuronal Neighbors. *Neuron* **2006,** *52* (1), 39-59.

97. Glass, D. J., Two tales concerning skeletal muscle. *J. Clin. Invest.* **2007,** *117* (9), 2388-2391.

98. Close, R., Dynamic properties of fast and slow skeletal muscles of the rat during development. *J Physiol* **1964,** *173* (1), 74-95.

99. Gordon, A. M.; Huxley, A. F.; Julian, F. J., The variation in isometric tension with sarcomere length in vertebrate muscle fibres. *J Physiol* **1966,** *184* (1), 170-192.

100. Stoney, G. G., The Tension of Metallic Films Deposited by Electrolysis. *Proceedings of the Royal Society of London* **1909,** *82*, 172-175.

101. Evans, D. R.; Craig, V. S. J., Sensing Cantilever Beam Bending by the Optical Lever Technique and Its Application to Surface Stress. *J. Phys. Chem. B* **2006,** *110* (11), 5450-5461.

102. Sander, D.; Enders, A.; Kirschner, J., A simple technique to measure stress in ultrathin films during growth. *Review of Scientific Instruments* **1995,** *66* (9), 4734.

103. Butt, H.-J., A Sensitive Method to Measure Changes in the Surface Stress of Solids. *Journal of Colloid and Interface Science* **1996,** *180* (1), 251-260.

104. Moulard, G.; Contoux, G.; Motyl, G.; Gardet, G.; Courbon, M., Improvement of the cantilever beam technique for stress measurement during the physical vapor deposition process. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **1998,** *16* (2), 736-742.

105. Peterson, C. A.; Workman, R. K.; Sarid, D.; Vermeire, B.; Parks, H. G.; Adderton, D.; Maivald, P., Effects of moisture on Fowler--Nordheim characterization of thin silicon-oxide

films. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* **1999,** *17* (5), 2753-2758.

106. Alexander, S.; Hellemans, L.; Marti, O.; Schneir, J.; Elings, V.; Hansma, P. K.; Longmire, M.; Gurley, J., An atomic-resolution atomic-force microscope implemented using an optical lever. *Journal of Applied Physics* **1989,** *65* (1), 164.

107. Meyer, G.; Nabil, M. A., Novel optical approach to atomic force microscopy. *Applied Physics Letters* **1988,** *53* (12), 1045-1047.

108. Raiteri, R.; Butt, H. J.; Grattarola, M., Changes in surface stress at the liquid/solid interface measured with a microcantilever. *Electrochimica Acta* **2000,** *46* (2-3), 157-163.

109. Müller, P.; Saúl, A., Elastic effects on surface physics. *Surface Science Reports* **2004,** *54* (5-8), 157-258.

110. Sader, J. E.; Larson, I.; Mulvaney, P.; White, L. R., Method for the calibration of atomic force microscope cantilevers. *Review of Scientific Instruments* **1995,** *66* (7), 3789.

111. Dennis, R. G.; Kosnik, I. P. E., Excitability and isometric contractile properties of mammalian skeletal muscle constructs engineered in vitro. *In Vitro Cellular Developmental Biology - Animal* **2000,** *36* (5), 327-335.

112. Brewer, G. J.; Boehler, M. D.; Jones, T. T.; Wheeler, B. C., NbActiv4 medium improvement to Neurobasal/B27 increases neuron synapse densities and network spike rates on multielectrode arrays. *Journal of Neuroscience Methods* **2008,** *170* (2), 181-187.

113. Pugh, C. W.; Radcliffe, P. J., Regulation of angiogenesis by hypoxia: role of the HIF system. *Nat. Med.* **2003,** *9*, 677.