Multi-frequency Characterization of Single Cells Through CM Factors and Dielectrophoresis

by Alex T. Jaffe

S.B in Electrical Science and Engineering, MIT, 2016

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

June 2018

Author: _____
Alex T. Jaffe
Department of Electrical Engineering and Computer Science
May 25, 2018

Certified by: _____
Joel Voldman
Professor of Electrical Engineering and Computer Science, Thesis Supervisor
May 25, 2018

Accepted by: _____
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Multi-frequency Characterization of Single Cells Through CM Factors and Dielectrophoresis

Submitted to the Department of Electrical Engineering and Computer Science on May 25, 2018

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis explores the use of dielectrophoresis to discern the electrical properties of single cells by observing them at multiple frequencies. We first simulate experimental conditions to show that as we increase the number of measured frequencies, we are able to better discriminate among different cells. Furthermore, we use the simulation to find the optimal number and value of frequencies to use to best discriminate among different cells in general in a large range of frequencies and a smaller, experimentally feasible, range of frequencies. We then fabricate a microfluidic device to measure balance positions of cells and calibrate it with polystyrene beads. We utilize BA/F3 cells to define balance position equilibration when shifting from one frequency to the next within this device. Finally, we find balance positions for three different activation levels of HL60 cells treated with Cytochalasin D using the optimal frequency sequence obtained in simulation to determine the differences in discrimination abilities depending on the number of frequencies used. We quantify the discrimination abilities of the optimal one, two, and three frequencies by minimizing 0-1 loss.

Thesis Supervisor: Joel Voldman
Professor of Electrical Engineering and Computer Science
Associate Head, Electrical Engineering and Computer Science

Acknowledgements

This thesis work was supported by the NIH (U24 AI118656) and DARPA SPAWAR N66001-11-1-4182. I would most of all like to thank Joel Voldman for being an excellent mentor to me in terms of this research as well as future academic aspirations. Second, I would like to thank Hao-wei Su for providing initial mentorship to me when I first joined the Voldman Group as a UROP. Third, I would like to thank Nicha Apichitsopa for assisting with experiments and corroborating experimental data, and Hao-wei Su and Javier Prieto for providing the foundational research to make this work possible. Finally, I would like to acknowledge help from Dan Wu for initial device fabrication instruction, Sarvesh Varma for outstanding general research advice, and Jaemyon Lee for sharing knowledge from his similar research with DEP fibers.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction to the Electrical Properties of Cells and Dielectrophoresis

Before going into the details of this thesis on multi-frequency dielectrophoresis, we provide a brief introduction on the theory of dielectrophoresis and Clausius-Mossotti (CM) factors preceded by the motivation for this thesis. Parts of this thesis have been submitted for publication.

Significance of Multi-frequency Dielectrophoresis for Cell Identification

Cell-based assays in microfluidics are of significant importance, being employed for basic science as well as the diagnosis of disease[1]. Assays of single cells, as opposed to populations, are of particular interest given the widespread understanding of population heterogeneity and the importance of rare cells[2]. One class of single cell-based assays are those that are label-free, giving them the advantage of being able to measure cellular phenotype or separate cells based on those phenotypes without altering the cell via labeling with dye, antibody, etc.[3] Label-free assays include measurements of cell size, optical properties [4] acoustic properties[5], and mechanical properties[6,7,8]. In particular, one popular class of label-free cell-based assay examines cells' electrical properties.

There currently are three central methods of analyzing single cells by their electrical properties: electrorotation, impedance cytometry, and dielectrophoresis[8,9,10]. Each method has tradeoffs in their throughput and specificity (based on the depth of analysis of each cell). Electrorotation uses a rotating electric field to induce the rotation of a particle as a result of electrical torque, where the torque and thus the rotational velocity depends on the electrical

properties of the particle[11,12]. Measurement of the rotational velocity thus allows estimation of the electrical properties of cells. Electrorotation has been extended to allow for analysis of hundreds of cells at once[13]. However, acquiring a full spectrum for a single cell takes around 30 minutes[10,14,15], which lowers throughput.

In comparison to electrorotation, microfluidic impedance cytometry is generally higher throughput[16,17]. It involves the continuous flow of cells through a channel where electrodes record cell impedance, often at to two frequencies[18]. The utility of impedance cytometry is well exemplified in work by Morgan and colleagues, where impedance cytometry was used to perform a 3-part differential white blood cell count with a throughput of about 1000 cells per second[17]. However, when not combined with other methods, such as optics and fluorescence[19], it is typically limited the two frequencies per single cell[16].

Dielectrophoretic methods for discriminating single cells tend to have throughputs lower than that of impedance cytometry but higher than that of electrorotation[18]. Dielectrophoretic methods apply a non-uniform electric field to induce a translational dielectrophoretic force on a cell. Sometimes the measurement involves a force balance between a dielectrophoretic force and a fluidic drag force, yielding an observable balance position that maps a cell position to its CM factor[20,21,22,23]. In 2013 this lab introduced the DEP spring, in which a dielectrophoretic force induced by coplanar electrodes exerts a force that is balanced by the fluidic drag force, resulting in a well-defined balance position[23]. We used this approach to analyze cells on a single cell basis under continuous-flow. Balance positions were obtained for thousands of single cells at a given frequency and solution conductivity. These balance positions yielded estimates of the CM factors of cells. Unfortunately, the method only allowed for measuring a single frequency for each cell, limiting the depth of analysis.

CM Factor Background

A cell has certain intrinsic electrical properties dependent on the cell itself as well as extrinsic electrical properties dependent on the environment of the cell. The finite electrical properties of a single shell model of a cell are listed with their symbols in Table 1.

Table 1: A key for used parameters in the single shell electrical model of a cell.

Parameter	Symbol
Cytoplasm Conductivity	σ_3
Cytoplasm Permittivity	ϵ_3
Membrane Conductivity	σ_2
Membrane Permittivity	ϵ_2
Medium Conductivity	σ_1
Medium Permittivity	ϵ_1
Outer Radius	R_1
Inner Radius	R_2

These electrical properties can be utilized to derive complex permittivities of each compartment (1). Compartment 1 refers to the media, compartment 2 refers to the cell membrane, and compartment 3 refers to the cytoplasm. The tilde implies that the value the permittivity is complex in that it has a real and an imaginary part. The real part is the permittivity and the imaginary part is the quotient of the conductivity over the radial frequency. These complex permittivities are used to derive the complex permittivity for the cell model itself (2). Finally, we use the complex permittivity of the cell model and that of the medium to derive the CM factor, the effective polarizability of the cell (3).

$$n = 1,2,3;\ \tilde{\epsilon}_n = \epsilon_n - i\frac{\sigma_n}{\omega} \tag{1}$$

$$\tilde{\epsilon}_{2,3} = \tilde{\epsilon}_2 \frac{\left(\frac{R_1}{R_2}\right)^3 + 2\left(\frac{\tilde{\epsilon}_3 - \tilde{\epsilon}_2}{\tilde{\epsilon}_3 + 2\tilde{\epsilon}_2}\right)}{\left(\frac{R_1}{R_2}\right)^3 - \left(\frac{\tilde{\epsilon}_3 - \tilde{\epsilon}_2}{\tilde{\epsilon}_3 + 2\tilde{\epsilon}_2}\right)} \tag{2}$$

$$Re[CM] = Re\left[\frac{\tilde{\epsilon}_{2,3} - \tilde{\epsilon}_1}{\tilde{\epsilon}_{2,3} + 2\tilde{\epsilon}_1}\right] \tag{3}$$

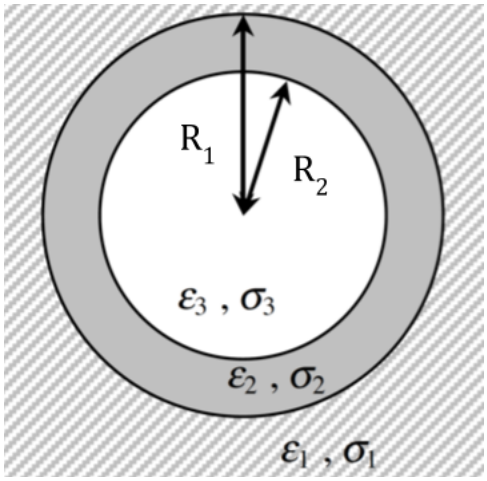

Figure 1: Single shell model of a cell adapted from AC Electrokinetics (ref. 24). The symbols correspond to the parameters in Table 1.

Dielectrophoretic Spring Background

Dielectrophoresis is a force that is exerted on a polarizable particle resulting from an electric field gradient[24]. The dielectrophoretic force can be derived from the CM factor of the cell, the cell radius, and external parameters such as the electric field and the permittivity of the media. Figure 2A shows a microfluidic channel on top of slanted electrodes. This can be considered a high level diagram for our DEP Spring device (Chapter 3). Cells flow through the channel from left to right until they encounter the slanted electrodes. These slanted electrodes produce a non-uniform electric field that polarizes a cell and exerts a dielectrophoretic force on

it. With the x-direction defined to be straight along the electrodes and the y-direction defined to be orthogonal to the electrodes, we have:

$$F_{DEP} = 2\pi {R_1}^3 \epsilon_1 Re[CM] \nabla |E|^2 \quad (4)$$

$$F_{DEP,y} = 2\pi {R_1}^3 \epsilon_1 Re[CM] q_R(\mathrm{y}) V_{RMS}^2 p(f, \sigma_1) \quad (5)$$

Due to the high conductivity of the media and intrinsic electrical properties of the cell, this y-component of the dielectrophoretic force pushes the cells away from the electrodes. Therefore the direction of the cells has been altered given the slanted electrodes applying the electric field. The hydrodynamic drag force will work to try to keep the cells flowing in the center of the channel and from left to right. Since the slanted electrodes are pushing the cell away from the center and away from flowing, the y-component of the dielectrophoretic force is balanced with the y-component of the hydrodynamic drag force to create a force balance and result in a measurable one-dimensional balance position of a particle in a microfluidic channel[23] as illustrated in Figure 2B.

$$v_y = \frac{6Q \sin\theta}{wh^3} R(h - R_1) \quad (6)$$

$$F_{Drag,y} = -6\pi R_1 \eta v_y \quad (7)$$

$$F_{DEP,y} + F_{Drag,y} = 0 \quad (8)$$

$$\delta = q_R^{-1}\left(\frac{3\eta \sin\theta \left[\frac{6Q}{wh^3}(h - R_1)\right]}{R_1 \epsilon_1 Re[CM] V_{RMS}^2 p(f, \sigma_1)}\right) \quad (9)$$

In equations (6), (7), (8), and (9), ${q_R}^{-1}$ reflects the dependence of the DEP force on position, η is the medium viscosity, θ is the angle between the electrodes and the fluid flow in the channel, w is the channel width, h is channel height, R_1 is the cell radius, ε_1 is the medium permittivity, V_{RMS} is the root-mean-square voltage across the electrodes, Q is the flow rate, and $p(f, \sigma_1)$ is a normalization factor that corrects for any drop at the electrode solution interface within the channel that depends on frequency and media conductivity σ_1.

We see in Figure 2B that this force balance causes an equilibration of y-component forces, causing the cell to arrive at a balance position. From this balance position δ, the particle's Clausius-Mossotti factor can be derived[23].

$$Re[CM] = \frac{18Q\eta \sin\theta\,(h - R_1)}{V_{RMS}^2 wh^3 \epsilon_1 p(f, \sigma_1) R_1 q_R(\delta)} \tag{10}$$

From this factor, one can derive several empirical electrical properties of single cells (permittivity and conductivity of membranes and cytoplasm) mapping to a given mathematical model of a cell[24]. This should yield plenty of information that will vary depending on what cell is being identified, which makes dielectrophoresis a viable method for cell identification.

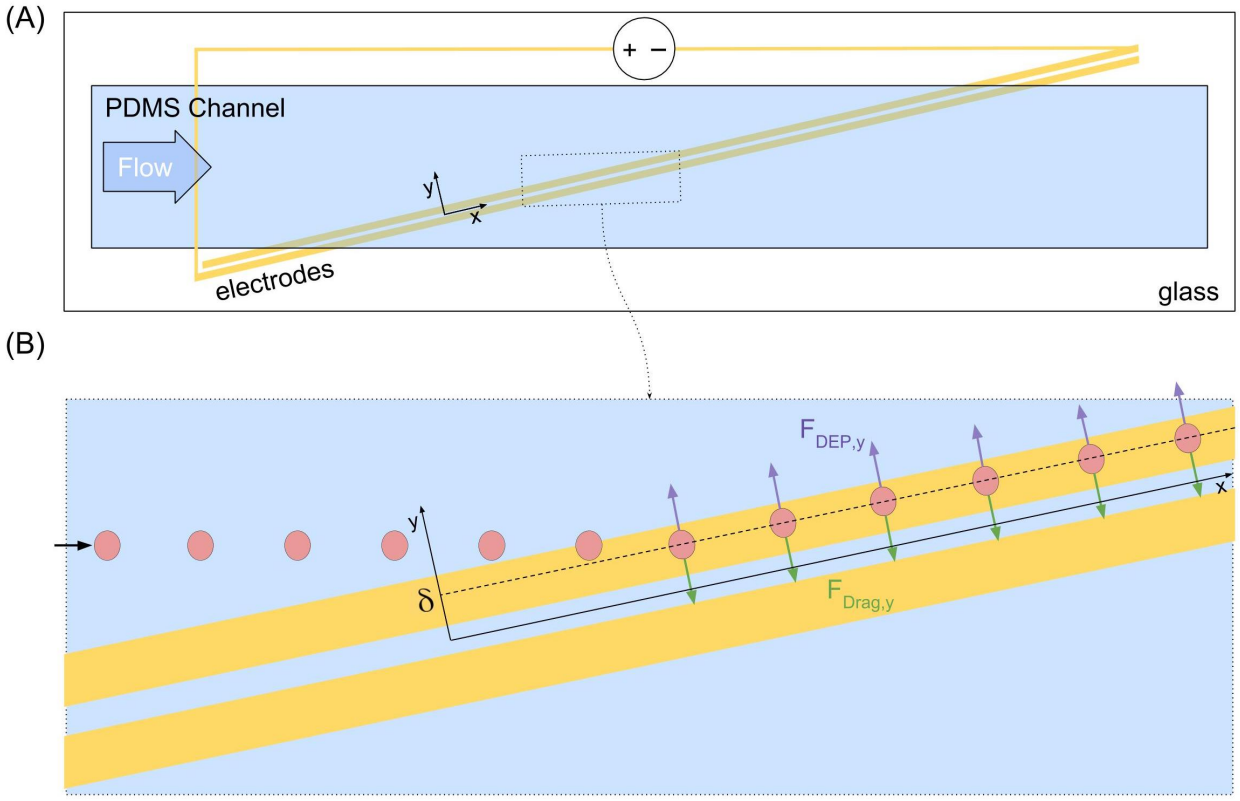

Figure 2: (A) Dimensional diagram of the DEP Spring device. (B) Zooming in on a region in (A) to show the DEP Spring concept of the force balance between the hydrodynamic drag force and the dielectrophoretic force. The red circles signify the path of a single cell within the channel. The black arrow on the left signifies the flow direction of the cell. The purple arrows signify the y-component of dielectrophoretic force acting on the cell and the green arrows represent the drag force acting on the cell. δ is the balance position of equilibration of the y-axis position along the electrodes.

Research Description

This thesis extends the DEP spring to measure multiple frequencies. Measurement at different frequencies allows investigation of the frequency-dependent electrical phenotype of the cells, as the measurements are obtained at frequencies that probe different parts of the cell. We call this new method the multi-frequency DEP spring. We first use stochastic simulations to understand how increasing the number of measured frequencies increases the ability to discriminate cells. We then establish a balance position validation data analysis technique to bridge the gap between multi-frequency CM factor simulation and multi-frequency DEP Spring experimentation. Then, informed by the simulations, we develop and characterize the multi-frequency DEP spring and show its utility in characterizing cells exposed to cytoskeletal inhibitors.

Chapter 2: Simulating Cell Discrimination by Electrical Properties

We first undertook simulations to understand how measuring multiple frequencies affects cell discrimination ability and what frequencies are optimal for discriminating cells. In practical experiments, one could only observe cells at a discrete and limited set of frequencies, providing incomplete information as to the cells' electrical properties. Therefore, we used simulations to test a large set of frequencies to indicate which subset of frequencies we should use during experiments. Given a single-shell model of a mammalian cell, considering the cytoplasm and the membrane as separate compartments, there were five dielectric parameters that identified a cell: cytoplasm permittivity and conductivity, membrane permittivity and conductivity, and radius. We could thus completely describe this model via perfect measurements of the CM factor at five independent frequencies[24]. However, all measurements had associated uncertainty, and high frequencies, which provided information as to the cytoplasmic compartment, are challenging to access. We thus wanted to determine which measurement frequencies (and how many) provided the most information about a cell when measured with some uncertainty.

Simulation Structure

We used a Monte Carlo simulation to create a set of cells whose properties are drawn from a distribution of cell electrical properties using continuous distributions of parameters in the single-shell model of a cell. These parameters were cell radius, cytoplasmic conductivity and permittivity, membrane (shell) conductivity, permittivity and thickness, and medium properties. We used uniform distributions of parameters across a range informed by literature (Table 2). In

our model we fixed the shell thickness, due to the known thickness of the plasma membrane's phospholipid bilayer, as well as the medium conductivity and permittivity.

The underlying method of the simulation first created cells drawn from a distribution of properties and simulates their CM factors based on those properties (Figure 3A). It then determined the optimal frequency that would differentiate the most cells from one another under some assumption of the uncertainty of the measurement (Figure 3A). In our case we estimated the position uncertainty based on experimental balance position measurements. In the simulations, we sought to find the number of cells remaining within a set tolerance in CM factor dependent on predicted balance position (Figure 3E) at a given frequency. The chosen frequency with the fewest cells remaining was chosen as the best first frequency. Then, using that first frequency, the algorithm searched for the best second frequency (Figure 3B). This approach continued until including further frequencies did not further narrow the number of cells (Figure 3C,D).

Simulation Methods

Simulations were performed using MATLAB 2016a. The Monte Carlo simulation of the intrinsic parameters was varied with a uniform distribution across the given ranges (Table 2). The simulation was run 100 times with 1000 simulated cells CM factor spectra per run. For each run, a cell CM factor spectrum was chosen at random, and a tolerance in the CM factor was estimated based on the experimentally measured balance position uncertainty. We used a position uncertainty of 0.5 microns, which was ~2× the standard deviation of the noise in the experimentally measured balance positions. When a given frequency was tested, certain CM factor spectra would be similar enough at that frequency to the chosen CM factor spectrum to remain within the tolerance. However, several other CM factor spectra will not be similar enough. Whichever frequency in the spectrum resulted in the fewest neighboring cell CM factor

spectra within the CM factor tolerance was chosen to be the optimal frequency. That optimal frequency was then held constant and the process repeated to select the second optimal frequency. Once the second optimal frequency is selected, the first two frequencies are held constant and the process repeated to select the third optimal frequency, and so on.

Table 2: The values used for the parameters in the simulations The medium properties are held constant as they experimentally controlled. The inner radius of the cell is constant relative to the outer radius of the cell, as we assume a phospholipid bilayer thickness of 10 nm.

Parameters	**Values**
Cytoplasm Conductivity	$0.2\ S/m\ \leftrightarrow 1.2\ S/m$[25,26]
Cytoplasm Permittivity	$20\ \epsilon_0\ \leftrightarrow\ 80\ \epsilon_0$[25,26,27]
Membrane Conductivity	$10\ nS/m\ \leftrightarrow 1\ \mu S/m$[25,26]
Membrane Permittivity	$2\ \epsilon_0\ \leftrightarrow\ 20\ \epsilon_0$[25,26,28]
Medium Conductivity	$1.5\ S/m$
Medium Permittivity	$78.5\ \epsilon_0$
Outer Radius	$2.0\ \mu m\ \leftrightarrow 8.0\ \mu m$
Inner Radius	$1.99\ \mu m\ \leftrightarrow 7.99\ \mu m$[29]

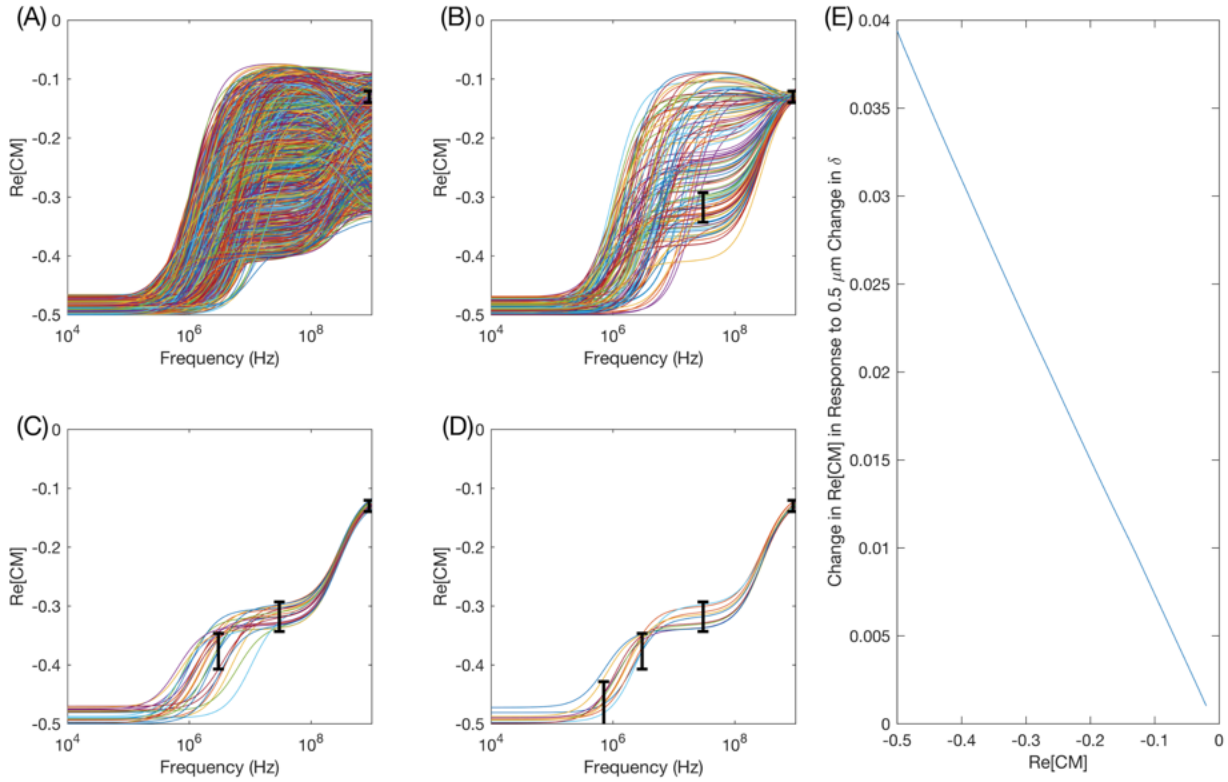

Figure 3: (A) Simulated CM factors for 1000 cells and the first selected optimal frequency (~900 MHz, threshold marker). The balance position uncertainty of 0.5 microns translates into an equivalent tolerance in CM factor measurement. (B) Simulated CM factors for cells still remaining after measurement of the first frequency, along with the second selected optimal frequency. (C-D) Simulated CM factors for cells remaining after measurement of the second (C) and third (D) frequency, along with the optimal third (C) and fourth (D) frequencies. (E) shows the sensitivity of Re[CM] to position uncertainty, i.e., how a 0.5 μm change in balance position δ ($\Delta\delta = 0.5\ \mu m$) changes Re[CM] at different Re[CM].

Simulation Results

As shown in Figure 4, this simulation was executed for two frequency ranges to determine whether experimental limitations affected the optimal choice of frequencies. In one case we simulated a large frequency range (10 kHz to 1 GHz), while in the second case we narrowed to the range explored experimentally (500 kHz to 25 MHz).

Comparing the two ranges (Figure 4A, E), we could see that the Re[CM] spectrum had two dispersions in the wider frequency range (Figure 4A) and only one in the narrower experimentally accessible range (Figure 4E), consistent with the known general location of the higher-frequency dispersion. Fewer dispersions in the narrower range indicated that there was

more independent information in the wider frequency range than in the narrower range. Examining how increasing numbers of test frequencies narrows the fraction of cells remaining in the wide experimental range (Figure 4C-D), we saw that the ability to discern cells improves as we increased from one to four different frequencies in the sequence. In particular, measuring at one frequency (optimally chosen to be 50 MHz, Figure 4D), left 11% of cells on average; increasing to two frequencies left only 3% of cells remaining, while four frequencies narrowed down to 0.5% of cells (Figure 4D). Adjusting the balance position uncertainty affected the number of frequencies above which no further discrimination occurs, as expected (Figure 5, Figure 6).

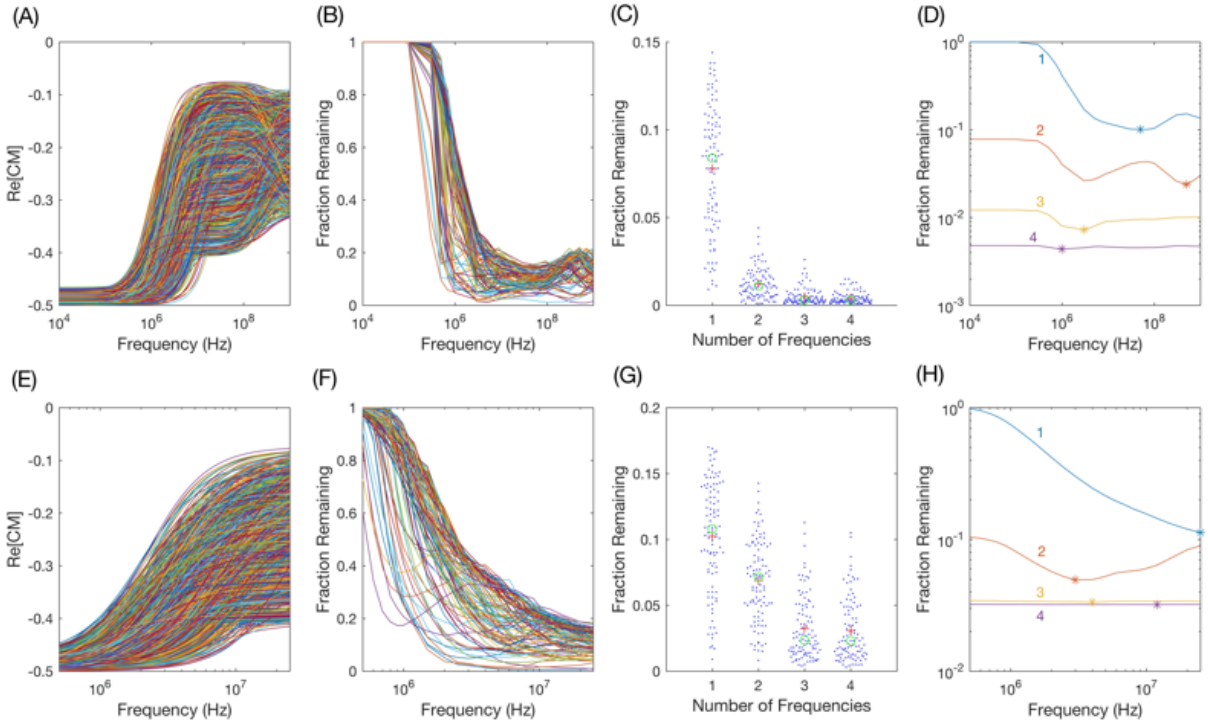

Figure 4: Monte Carlo simulation results for measurements across a wide (A-D) and narrow (E-H) frequency range. (A, E) Re[CM] factors for one simulation of 1000 randomly generated cell models. (B,F) Fraction of cells remaining in the threshold after balance positions at various frequencies are measured across 100 simulations. (C, G) Fraction of cells remaining as balance positions are measured at an increasing number of frequencies (+ = mean, o = median). (D, H) Mean fraction of cells remaining as a function of which frequency is tested, as balance positions at increasing numbers of frequencies are measured (denoted by the number next to each line). Results for each additional frequency (2 to 4) are predicated on choosing the best prior frequency.

Examining the particular optimal frequencies chosen in the wide frequency range, frequencies around the second dispersion were more informative than lower frequencies as the frequencies around the second dispersion had the high variability in CM (Figure 4A) and the balance position was strongly sensitive to Re[CM] when Re[CM] was close to zero (Figure 3E). After that, the optimal frequencies were around the first dispersion as CM factor spectra around the second dispersion were already narrowed down from the first optimal frequency. The fourth frequency only provided 0.3% additional discrimination ability, and the particular choice of frequency value was not very sensitive (0.05% change in fraction of cells remaining across the frequency range).

Turning to the narrower, more easily accessible experimental range (Figure 4E-H), we saw a similar decrease in the fraction of cells remaining as additional frequencies were measured (Figure 4G). Consistent with the wide frequency range, the optimal first frequency was the highest-accessible frequency, in this case 25 MHz. We also saw that the ability to discriminate saturated at ~4% cells remaining after 3 frequencies, worse than in the wider frequency range, because not all cell properties were accessible at the lower frequency range. In particular, the third frequency choice was not critical; there was a 0.1% variation in the fraction of cells remaining across the frequency range.

We chose an optimal frequency sequence to use for experiments by observing which frequency sequence was most commonly found to be the optimal three-frequency combination in our simulations across the experimental range. Based on these results, the optimal frequency combination was 1.2 MHz, 2 MHz, and 25 MHz, which were the frequencies used for experiments. Since the simulation was run 100 times, the optimal three-frequency combination varied for different simulation runs. For this reason, average optimal frequencies for each numbered frequency (1-4) in Figure 4D,H differ from the optimal frequency combination found.

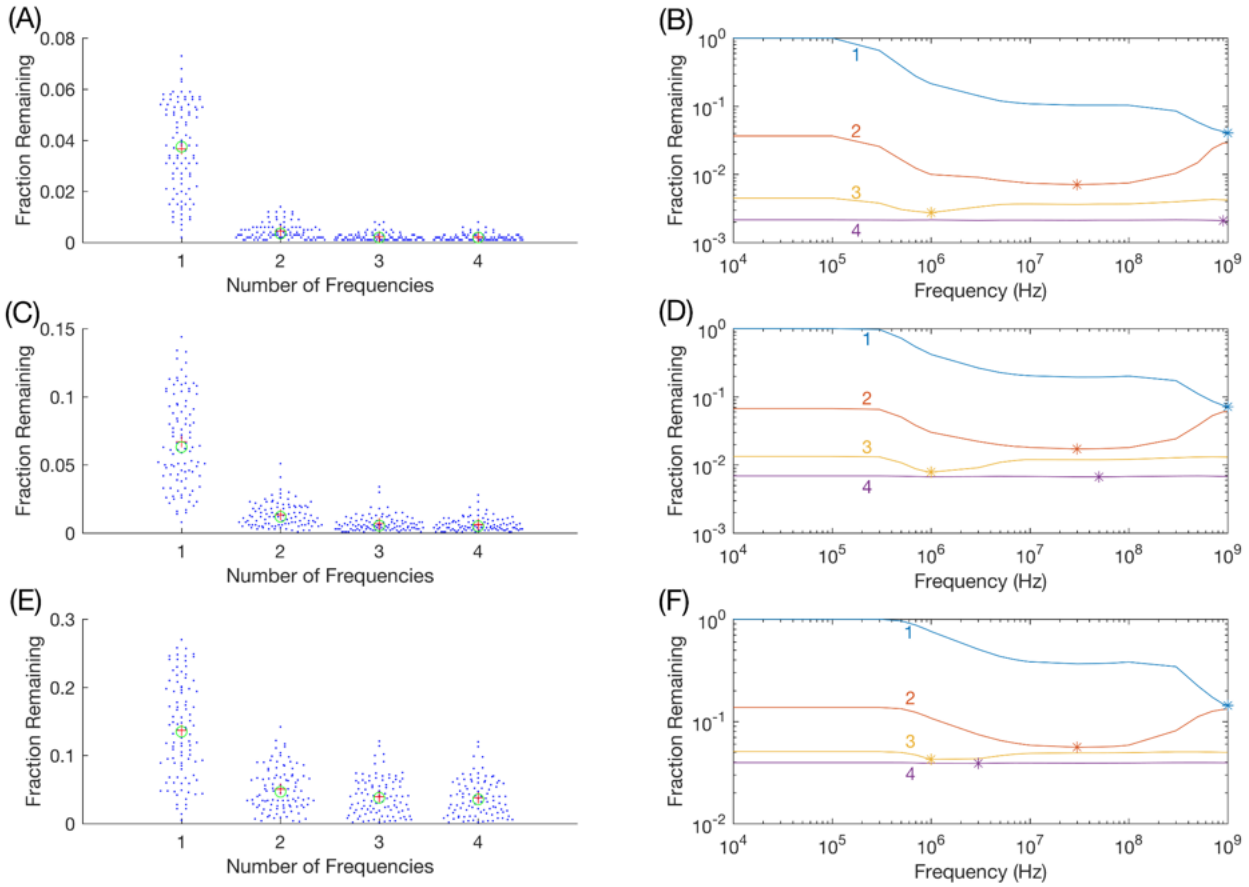

Figure 5: Effect of measurement uncertainty on discrimination ability in the wide frequency range. (A,C,E) Fraction of cells remaining as balance positions are measured at an increasing number of frequencies (+ = mean, o = median) with balance position uncertainty of 0.25 microns (A), 0.5 microns (C), and 1 micron (E). (B,D,F) Mean fraction of cells remaining as a function of which frequency is tested, as balance positions at increasing numbers of frequencies are measured (# of frequencies denoted on the plots), with balance position uncertainty of 0.25 microns (B), 0.5 microns (D), and 1 micron (F). Results for each additional frequency (2 to 4) are predicated on choosing the best prior frequency.

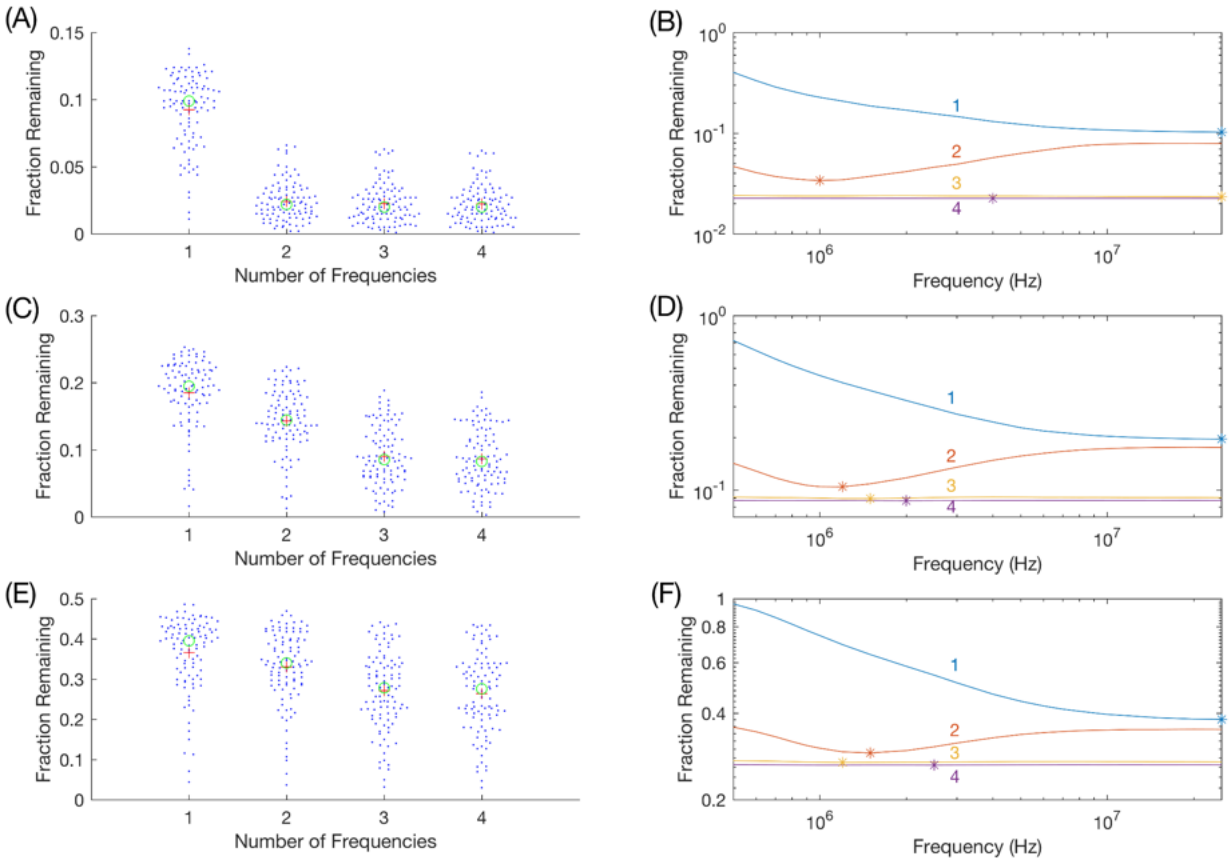

Figure 6: Effect of measurement uncertainty on discrimination ability in the experimental frequency range. (A,C,E) Fraction of cells remaining as balance positions are measured at an increasing number of frequencies (+ = mean, o = median) with balance position uncertainty of 0.25 microns (A), 0.5 microns (C), and 1 micron (E). (B,D,F) Mean fraction of cells remaining as a function of which frequency is tested, as balance positions at increasing numbers of frequencies are measured (# of frequencies denoted on the plots), with balance position uncertainty of 0.25 microns (B), 0.5 microns (D), and 1 micron (F). Results for each additional frequency (2 to 4) are predicated on choosing the best prior frequency.

Chapter 3: Device Fabrication, Operation, and Calibration

The simulations in Chapter 2 establish that there is a theoretical basis for the claim that increasing the number of frequencies a cell model experiences will increase one's ability to discriminate the cell model. The remainder of this thesis focuses on experimentally establishing that when we do increase the amount of frequencies a real cell experiences, we do increase our ability to discriminate that real cell. Chapter 3 speaks to means for fabricating a device to run such experiments on, the core experimental procedures, and using calibration to account for unintended effectors of experimental results.

Device Fabrication

The device used for all experiments of this thesis was the 24th made version of the Iso-Dielectrophoresis System (IDS) Device. This device consists of gold electrodes on glass under a PDMS microfluidic channel with a height of 20 μm, width of 2 mm, and length on the order of centimeters. The electrodes can be connected to an AC signal and ground on either side of them. The device channel was made with Sylgard 184 PDMS, using a cross-linker-to-elastomer ratio of 1:10, and cleaned with isopropanol. Three holes were pushed in on each side of the device, lengthwise. These holes acted as the inlet and outlet holes. The Ti/Au device electrodes were fabricated on a glass substrate using standard microfabrication methods and were cut with a dice saw and cleaned with acetone, methanol, and isopropanol. Three holes for input ports and three output ports were punched into the PDMS channel device. The device channel was attached to the device electrodes using plasma from a Harrick plasma cleaner / sterilizer chamber (model PDC-32G), creating a three dimensional channel with the sides and top being the PDMS and the bottom being the gold electrodes on the glass substrate.

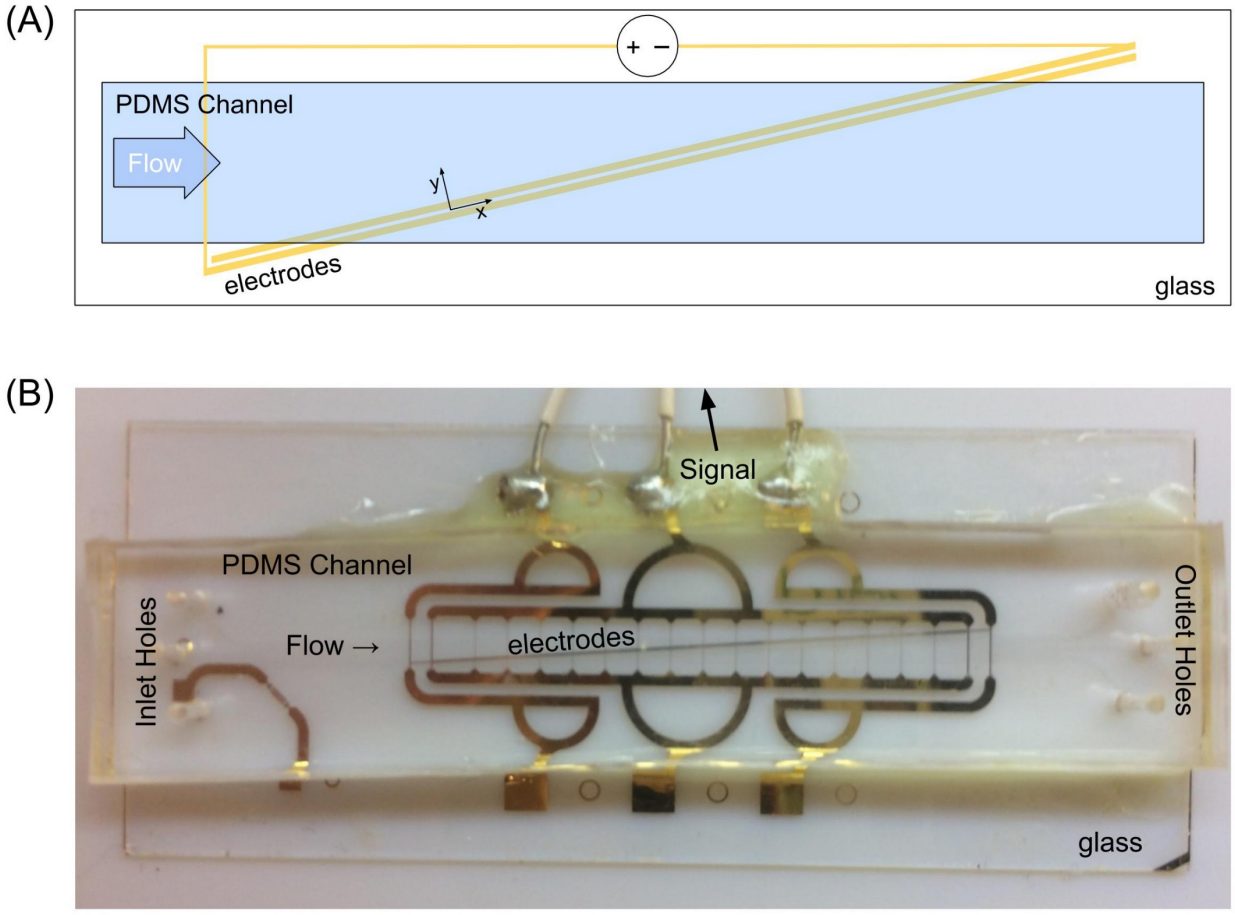

Figure 7: IDS Device Labeled Picture (A) Replicate of the device diagram shown in Figure 2A. (B) Labeled image of the device. The voltage symbol in (A) is analogous to the "Signal" text in (B) and represents the generation of an AC signal across the electrodes. The "electrodes" labels in (A) and (B) correspond to the electrodes over which cells travel and experience DEP forces. The "Flow" labels in (A) and (B) correspond to the opening in the PDMS channel through which cells flow. The "Inlet Holes" and "Outlet Holes" labels in (B) correspond to where cells and buffer enter exit the channel opening. The length of the device is 5 cm, the channel width is 2 mm, and the channel height is 20 μm.

Basic DEP Spring Experimental Procedure

Whether it is beads or cells traveling through the IDS device, the core experimental procedure remains the same. Flow was controlled by two Chemyx Fusion 200 syringe pumps. Before each experiment, the device was primed with 1% BSA in PBS by passing it through the device channel at a flow rate of 20 μL/min for 30 minutes. Meanwhile, 1 mL cells in media were centrifuged down at 1000 RPM for 5 minutes and the supernatant media was replaced with

PBS. PBS was pumped through two of the input ports while PBS with cells at the desired concentration (10^6 cells/ml) was pumped through the third input port with a total flow rate of 0.6 µL/min. This total flow rate is kept constant throughout the entirety of the experiment. Images of cells were obtained by a LAVision Imager QE camera coupled to a Zeiss Imager.M1m microscope with a 10x objective lens and brightfield illumination. Voltages at frequencies less than 15 MHz were applied via an Agilent 33220A function generator, while those greater than 15 MHz were applied by a TGR1040 RF signal generator and amplified by a TVA-R5-13 RF power amplifier. The microscope, the camera, and both function generators had automated control from a MATLAB graphical user interface (GUI), created by Hao-wei Su. After the experiment's end, the device was cleaned with PBS at a flow rate of 40 µL/min for 5 minutes, then distilled water at a flow rate of 100 µL/min for 1 minute. The device was then purged with air for drying for 5 minutes and then stored.

Detailed Software and Electrical Operation Procedure for DEP Spring Setup

What follows is the experimental procedure for a typical DEP Spring experiment. It is quite detailed, therefore numbering is used for organizational purposes out of courtesy to the reader who could be trying to replicate or slightly modify this procedure.

1. Make all the proper electrical connections
    a. Function Generator Output to Relay Switch Input 1
    b. High Frequency Function Generator Output to RF Amplifier Input
    c. RF Amplifier Output to Relay Switch input 2
    d. Relay Switch Output to splitter
        i. One splitter output goes to Oscilloscope

ii. One splitter output goes to alligator clips which attach to device (either permutation is fine)

2. Open MATLAB from the DEP Spring computer
3. Navigate to the FS_GUI_HF folder
4. Type in the command window: FS_GUI
5. A GUI will come up with all the controls one needs to run a multi-frequency DEP spring experiment except for fluidics control (should be controlled manually via the Chemyx Fusion syringe pumps and 1 mL glass syringes) and microscope stage control (should be controlled manually via the Imager.M1m microscope).
6. Given that the microscope and camera (LAVision Imager QE) are on and the camera window through the microscope is open, use the 10x lens of the microscope in the viewing panel taking up all of the left side and some of the right side of the GUI.
7. Click the “Preview” button at the top of the GUI to get a live image of what the camera is looking at. Go to the top right of the GUI and click the “File” and “Settings” buttons to change the field of view to only include the two electrodes generating the electric field with the center of the electrodes in the center of your field of view. Also, use these buttons to name your saved data and increase the number of saved frames to as many frames as memory will allow. (This is important to minimize the amount of times you’ll need to run the frequency sequence in order to get the data you need).
8. In Function Generator tab, click output to make sure that is running remotely. The
9. In High Frequency Function Generator tab, click function test and then output to make sure that is running remotely
10. In the bottom right text window, type in your frequency sequence as the example dictates
11. Click the “Customized Frequency Sequence” button to run. This will execute your frequency sequence and record the field of view of the camera throughout the duration

of the frequency sequence. When the duration of the frequency sequence ends or the maximum amount of frames is reached.

12. A data file of about 120 MB should be saved with frequencies and time synchronized with frames.
13. After all the device experiments are over, pass distilled water through the device (1 mL per minute) and then pass air through the device for 5 minutes. Clean the syringes with water.
14. Finally, store the device away.

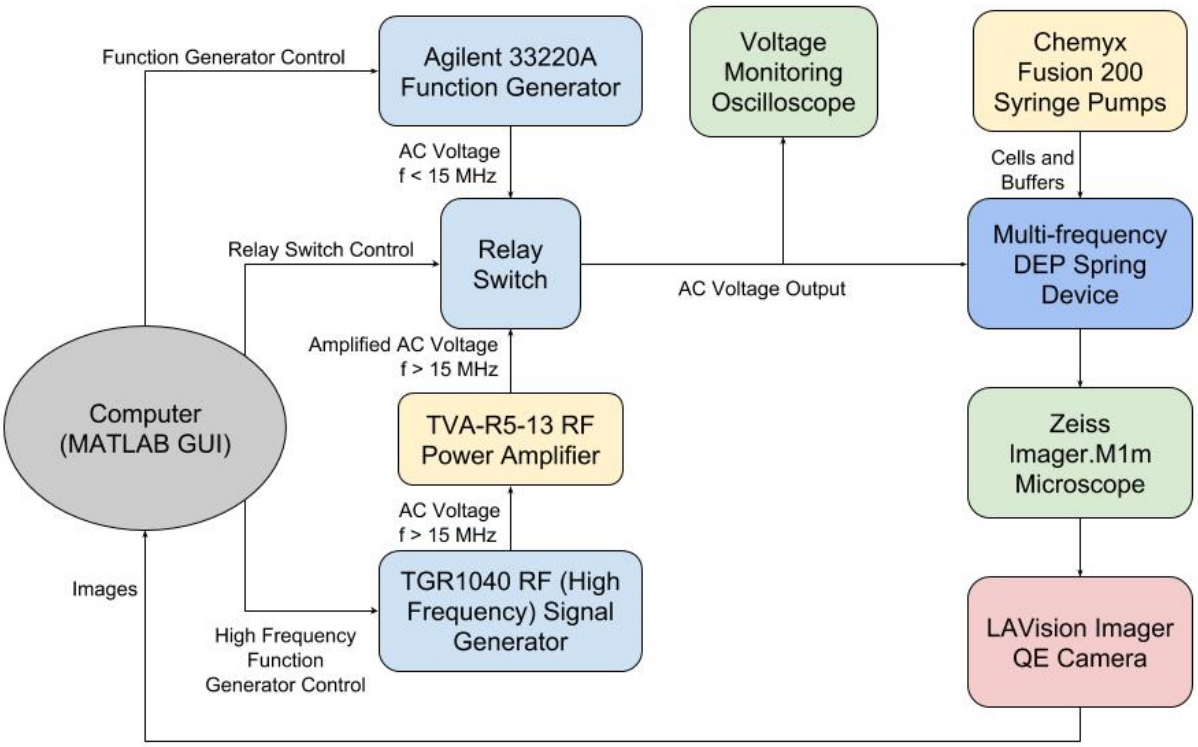

Figure 8: Conceptual diagram of the multi-frequency DEP spring experimental setup. The computer controls all voltage generation and switching through a MATLAB GUI. The high frequency function generator outputs a low voltage which needs to be amplified by the RF power amplifier. The oscilloscope monitors the voltage across the electrodes of the device. The syringe pumps manually control the flow rates of cells and buffers traveling through the device. The camera captures images of cells traveling through the device from the field of view of the microscope and sends the images to the computer.

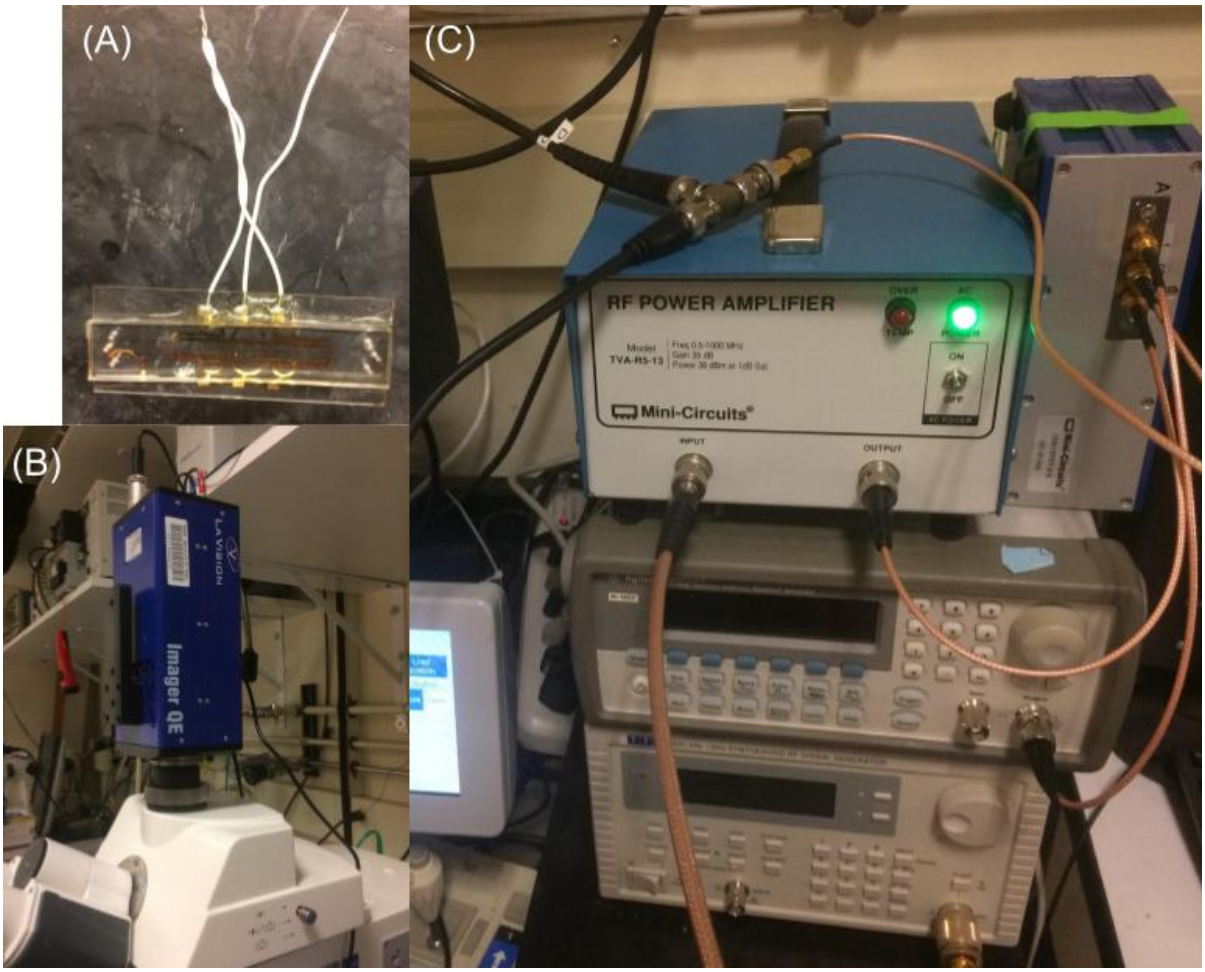

Figure 9: DEP Spring Experimental Setup. (A) shows the device wiring setup to connect to two alligator clips with the two outside electrodes tied together and the middle electrode by itself. (B) shows the camera setup over the microscope as well as the camera window settings. (C) shows the RF power amplifier on top of the function generator on top of the high frequency function generator.

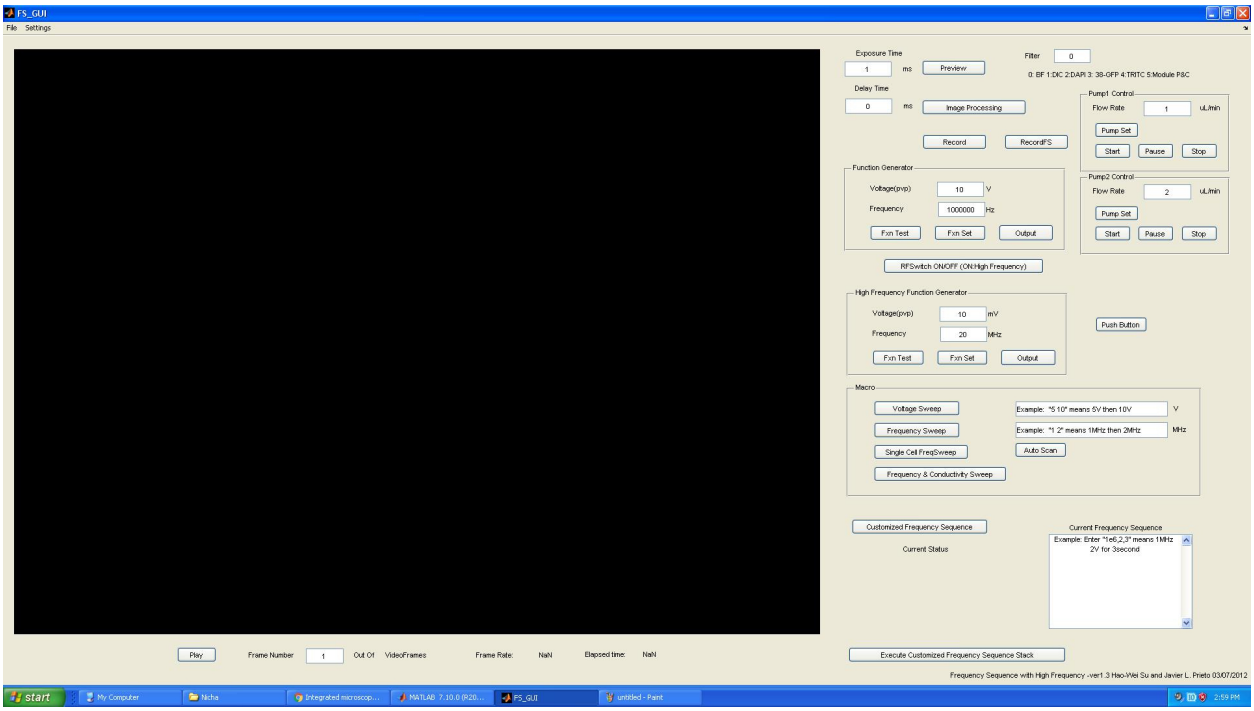

Figure 10: MATLAB FS_GUI graphical user interface. The "Function Generator" pane controls the Agilent Function Generator. The "High Frequency Function Generator" pane controls the TGR1040 1 GHz Synthesized RF Signal Generator.

Basic DEP Spring Data Analysis

During each experiment, image stacks were collected by the MATLAB GUI. The image stacks were used as inputs to a particle detection script which outputted detected particles as data structures by means of a time domain median image filter and pixel intensity thresholding. The detected particles were tracked using a particle tracking script which tracks the particles for the duration of time they are in the field of view of the microscope. The particle detection and tracking portions of the code were created by Hao-wei Su.

After the particle positions are tracked, a four-point moving average time domain filter is applied to each particle's y-position, and defines a balance position δ as the final moving average output y-position of a particle at a certain frequency.

$$\overline{pos}_{y,t} = \frac{pos_{y,t+1} + pos_{y,t} + pos_{y,t-1} + pos_{y,t-2}}{4} \tag{11}$$

Polystyrene Bead Preparation and Calibration

In order to account for frequency dependent variation in potential difference between the two electrodes providing the non-uniform electric field, we performed a calibration with polystyrene beads, which have a constant Re[CM] and thus frequency response across the experimental frequency range. Variations in balance positions across this range could be attributed to noise, voltage variation across in electrodes in the field of view, changes in conductivity in the field of view, changes in channel height across the field of view or any other changes that affect parameters in equation (10). 10 micron $(10.269 \pm 0.502\ \mu\mathrm{m})$ carboxylate-modified polystyrene beads from Polysciences Inc. (Warrington, PA, United States) were prepared for experiments by diluting in PBS at a 1:100 ratio.

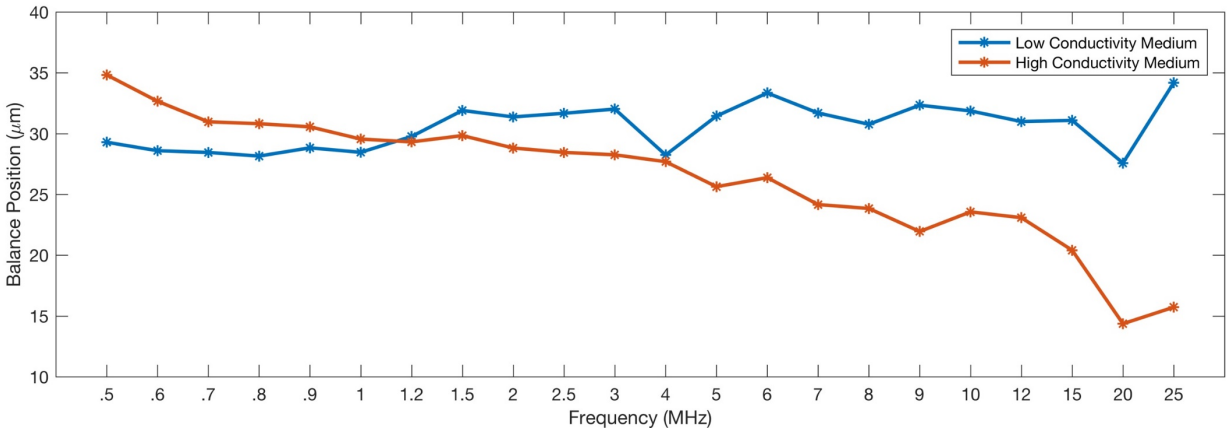

Figure 11: Polystyrene Bead Calibration. Balance positions of 10 micron polystyrene beads in high-conductivity PBS solution (red) and low conductivity sucrose solution (bottom) while varying the input frequency. All tested frequencies were given an amplitude of 8.0 VPP except for 25 MHz which was given 12.0 VPP.

When sending the beads through a high conductivity medium (PBS), we saw a decrease in balance position distance from the center of an electrodes as frequency increased, signifying a voltage attenuation with increased frequency. To confirm our presumption of voltage attenuation, we decreased the conductivity of the medium (sucrose - $\sigma = 10^{-4}\ S/m$) and saw the balance position stabilize across the experimental range. Since the high conductivity

medium was what we used for all further experiments, this observed attenuation would be taken into account when calibrating the balance positions of those further experiments.

Conclusions

In this chapter we fabricated the microfluidic device for experimentation with the multi-frequency DEP spring. We established an experimental procedure utilizing computer (MATLAB) control of electrical signal generation and data collection of images from a microscope of cells traveling through the microfluidic device. We then calibrated our experiment using a control of polystyrene beads to more effectively isolate our relevant observable parameters when experimenting on cells.

Chapter 4: Multi-frequency Balance Position Determination and Validation

This chapter goes through the steps taken to establish balance positions when transiently operating the DEP spring at more than one frequency. In prior research[23] and in the polystyrene beads calibration in Chapter 3, only one frequency would be acting on a given particle. Therefore, the transition period from a balance position of the previous frequency to the balance position of the next frequency must be examined to ensure that each balance position at each of the multiple frequencies would be roughly identical to the scenario where there is only one frequency for the entire duration of the particle's observation.

Multi-frequency DEP Spring

Experiments were run using a microfluidic DEP spring[23] device consisting of a PDMS channel atop a glass substrate containing coplanar electrodes (Figure 12). Cells flowed through the channel and experience a negative DEP force when they encountered two angled electrodes on the floor of the chamber (Figure 12A). This DEP force was counteracted by the fluid drag force. When these forces were of equal magnitude, the cell would reach a balance position, which was dependent on the applied frequency, the Re[CM] factor of the cells, cell size, electric field intensity, etc. The balance position is given by equation (9).

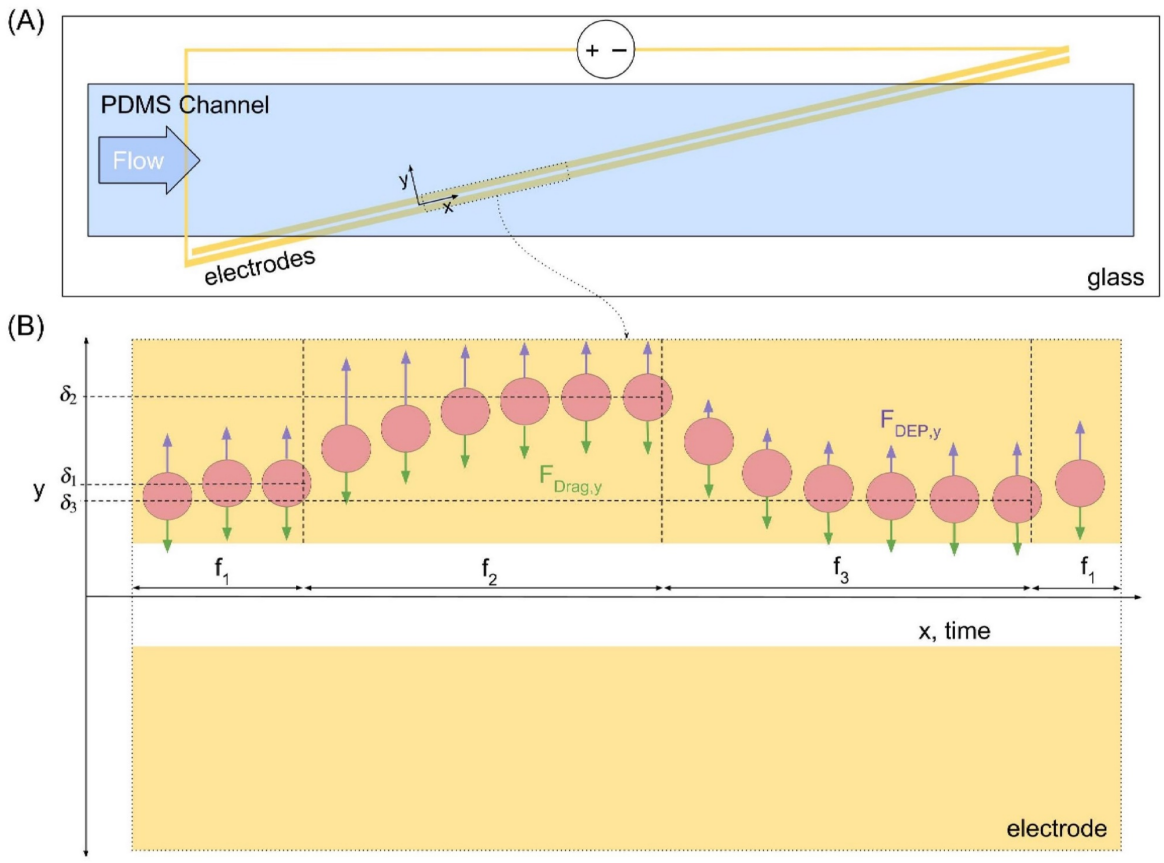

Figure 12: Multi-frequency DEP spring overview. (A) Schematic of the channel with slanted electrodes. (B) Schematic of a single cell experiencing the DEP spring at multiple frequencies (f_1, f_2, and f_3) at different points in time, where they experience a balance between the y-directed DEP force ($F_{DEP,y}$) and the y-directed drag force ($F_{Drag,y}$) and arrive at balance positions δ_1, δ_2, and δ_3, respectively. In this instantiation three frequencies repeat. The center of the electrodes defines the origin of the y-axis.

In the multi-frequency DEP spring, the frequencies changed according to a predetermined sequence (shown as f_1, f_2, f_3, f_1, etc. in Figure 12B), which changed the balance position by changing the Re[CM]. The cells attained the new balance position with some settling time. By visually tracking the cells across the field of view (Figure 12B) and correlating the image stack timestamps to the frequency sequence timestamps, we could correlate balance position to applied frequency.

Cell Culture

Two types of cells were cultured for experiments relevant to this thesis: BA/F3 cells and HL60 cells. BA/F3 cells were cultured from frozen stalk at -80 °C. The media used was RPMI media with penicillin/streptomycin (1x), fetal bovine serum (20%), and L-glutamine (2 mM). The cells were passaged at a 1:10 ratio every 5 days. HL60 cells were similarly cultured in DMEM media with penicillin streptomycin (1x), bovine calf serum (20%), and L-glutamine (2 mM). Both the BA/F3 and HL60 cells typically have over 90% viability.

BA/F3 Balance Position Verification Methods

To validate the ability to measure balance positions at multiple frequencies and understand the limits of the measurements, we undertook experiments with BA/F3 cells. Cells were subjected to sequences of measurement frequencies and their positions were measured (Figure 13). The frequency sequence used was 1.2 MHz, 2 MHz, 25 MHz – the optimal frequency sequence found for the experimental range in simulation (Chapter 2).

On top of the basic data analysis used to obtain cell positions, we added analysis to account for the changing of frequencies acting on a single cell by defining what it means to obtain a valid balance position. We defined a valid balance position if the cell stayed within 10% of the average difference in balance positions between one frequency and the next frequency in the final 250 milliseconds at a given frequency, corresponding to six data points including the final data point that is the balance position. If this criterion was met, the balance position was defined to be the final distance measurement from the center of the electrodes at a given frequency. This validation processed was used for cell discrimination analysis as well.

BA/F3 Balance Position Verification Results

Error! Reference source not found.A shows, among several other cell trajectories, an example of a cell trajectory that properly attained all three balance positions and an example of a cell trajectory that did not do so. The frequency sequence used was 25 MHz for 1.3 seconds, followed by 1.2 MHz for 1.8 seconds, followed by 2 MHz for 0.7 seconds, and then repeated. Due to the similarity in balance positions between 1.2 MHz and 2 MHz, this transition took less time and could afford a shorter duration at 2 MHz. The differences in balance positions between 25 MHz and 1.2 MHz were the largest, causing 1.2 MHz to necessitate the longest duration. We found that ~63% of cells properly attained all three balance positions, while the remainder failed to properly attain at least one of three balance positions. We then computationally examined how decreasing the time at each frequency would affect the fraction of cells that attain valid balance positions (Figure 13B); note that this metric differs from the fraction of cells unable to be discriminated referred to in Figure 1. As expected, decreasing the total frequency sequence duration from 3.8 s to 3.05 s decreased the fraction of valid balance positions to 40%. However, decreasing the duration would allow more balance positions to be measured, assuming camera field of view and flow rate would be kept constant. Overall, then, we found that we were able to reliably measure three frequencies, with the ability to increase the number of measured frequencies, if desired, by altering experimental parameters.

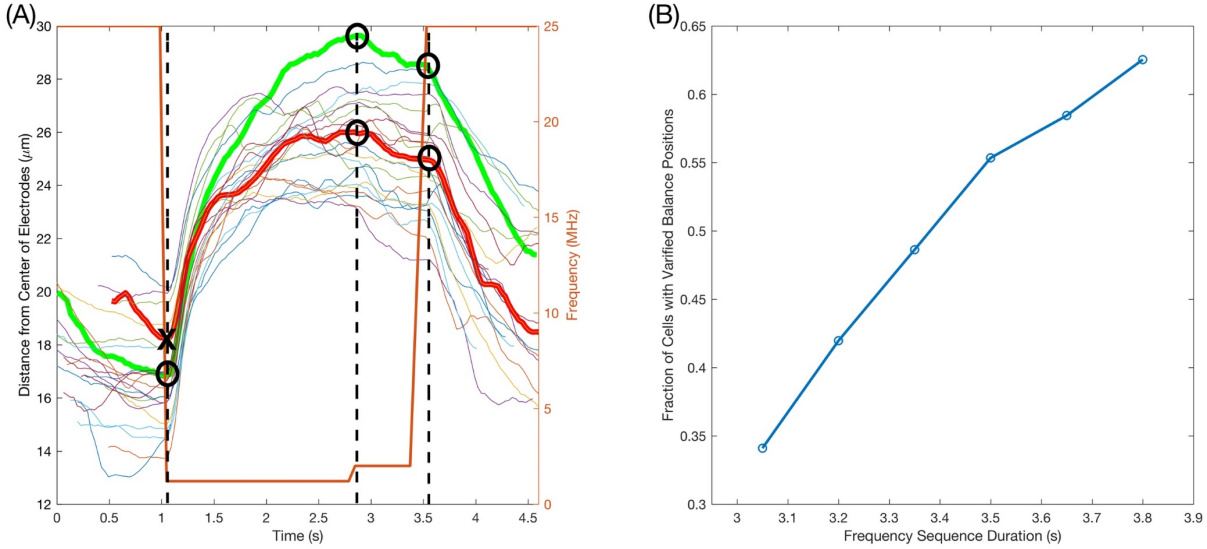

Figure 13: (A) Overlaid measured trajectories of 23 BA/F3 cells, along with the frequency the cells are experiencing as a function in time (bold orange). The bolded green trajectory is an example of a cell that properly attains all three balance positions. The bolded red trajectory is an example of a cell that does not properly attain all three balance positions. The black circles indicate validated balance positions while the black X shows an invalidated balance position. The vertical dashed lines indicate transitions in frequency where balance positions are measured. (B) Fraction of valid balance positions measured as the overall time for the frequency sequence changes.

Conclusions

In this chapter, we established a method for verifying that cells have properly attained balance positions in the multi-frequency DEP spring. We found that a sizeable percentage of

tested BA/F3 cells did not properly attain the three balance positions although most of them did. Furthermore, we found that increasing the number of tested frequencies would greatly decrease the percentage of cells with validated balance positions at all tested frequencies, contributing to the difficulty of experimentally adding more frequencies to the multi-frequency DEP spring.

Chapter 5: Cell Discrimination With Multi-frequency Dielectrophoresis

The most meaningful set of experiments of this thesis is utilizing the multi-frequency DEP spring technique established in Chapter 4 to increase discrimination ability among different cells or similar cells treated with different effectors. The simulation indicated that as we increase the amount of frequencies with which single cell models are tested, our discrimination ability among their CM factors would increase. This experimentation portion intended to validate that simulation through the multi-frequency DEP spring.

HL60 Cell Discrimination Methods

We sought to apply the multifrequency DEP spring to the problem of distinguishing closely related cell states. In particular, we measured the electrical properties of HL60 cells under exposure to different concentrations of Cytochalasin D (CytoD), a drug that blocks actin cytoskeleton polymerization. We chose this treatment to examine whether the known effects of the drug on deformability also translated to any changes in electrical properties.

We measured multifrequency balance positions of 262 cells exposed to two concentrations of CytoD (along with control). To avoid any nonspecific changes due to changes in the cell size, we adjusted for the cell size variations, measuring optically with the microscope used to observe cells going through the DEP spring. In addition, we adjusted for any frequency dependence in the applied field (due to electrode polarization or lead inductance) by making control measurements with polystyrene beads (Figure 11), and using those balance positions to compensate.

In order to tie the experimentation discrimination back to the simulation discrimination and provide more of a complete connection from balance position to CM factor, balance positions, combined with cell radius and other parameters extrinsic to the cell were converted to

CM factors. Since images of the cells were taken, size (radius) measurements were obtained. These radii, however, needed to be corrected in order to account for the lighting inconsistency in the channel causing inconsistent size measurements of the same cell at different points in the channel. The size correction for the HL60 population was done by taking ground truth population radii measurements using a Beckman Z Coulter Counter and adjusting the mean and variance of the relevant radii measured with our imaging software and the microscope to be the same as the mean and variance of the radii sensed by the Coulter Counter. This process is diagramed in Figure 14 and example results for an HI60 control population are shown in Figure 15.

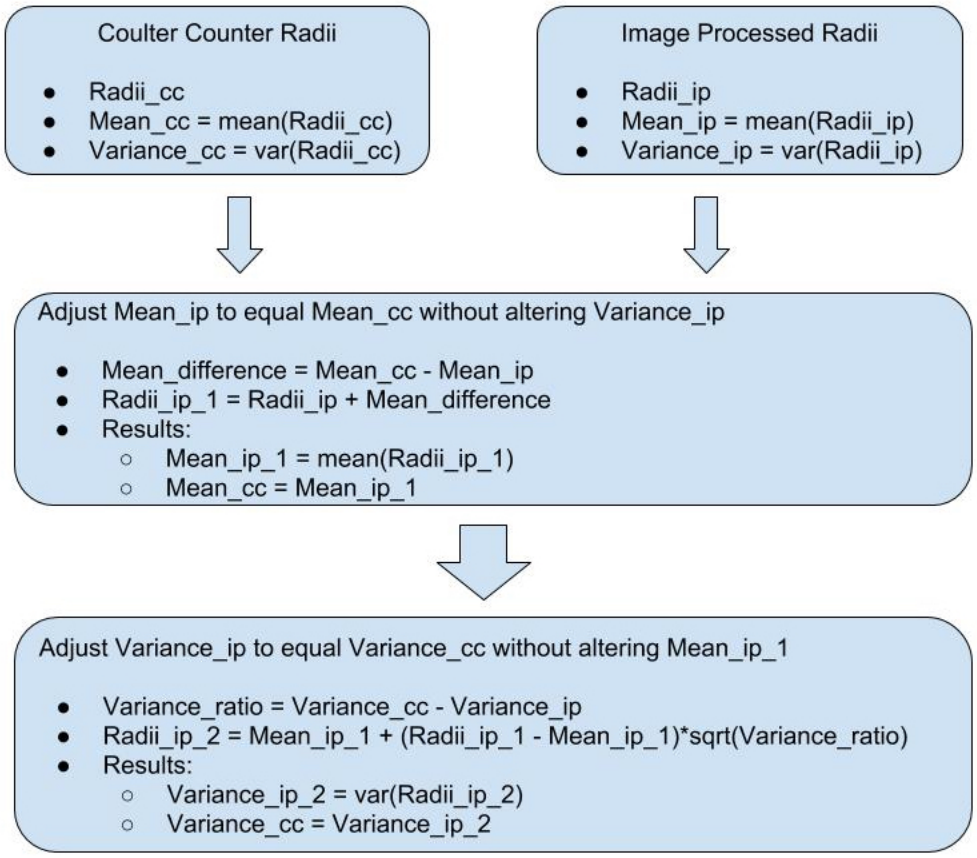

Figure 14: Descriptive flow chart for adjusting radii from imaging software to reflect radii from the coulter counter for the same population

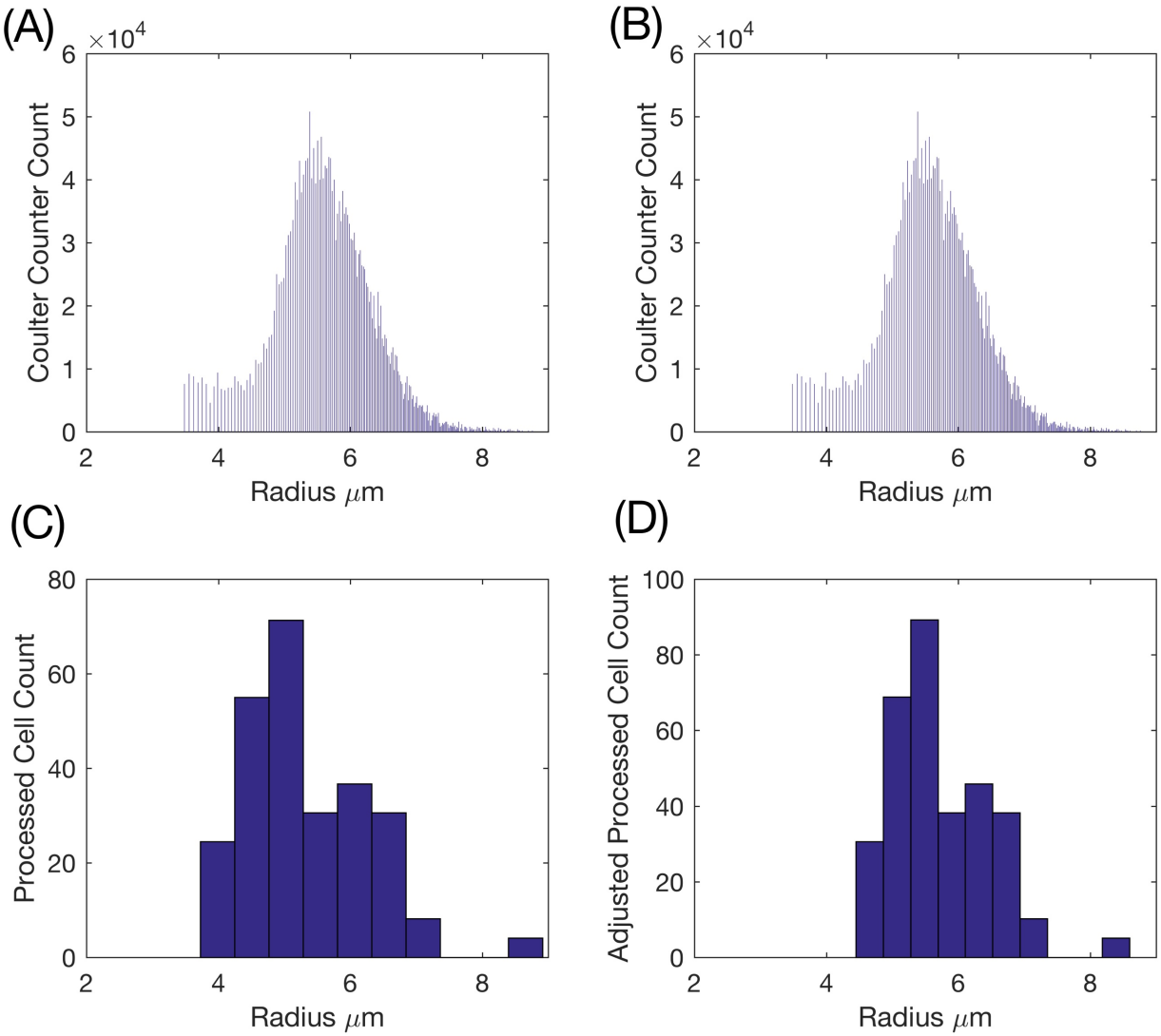

Figure 15: (A): Radii measured from Beckman Coulter Counter. (B): Exact replica of (A) for comparison purposes. (C): Histogram of HL60 control population radii before mean and variance adjustment. (D): Histogram of HL60 control population radii after mean and variance adjustment

Once the HL60 cell balance positions were verified (validated) and converted to CM factors via equation (10), validated cells and their trajectories were then stored as MATLAB data files for analysis. The validated balance positions for different populations were classified by optimizing a weight vector through minimizing 0-1 classification loss. It was imperative that this type of optimization was used instead of a convex, differentiable optimizer, such as Hinge loss or ReLu with least squares or logarithmic objective functions, due to the fact that we were optimizing for discrimination accuracy which has a discrete range of outputs solely dependent

on equation (12). The computation disadvantage to using this classifier was that instead of a gradient descent optimization method, an extremely granular table-lookup approach needed to be utilized for which training took about 4 hours on a MacBook pro with 8 GB of RAM. After this optimization, the discrimination accuracy was calculated via the formula:

$$Acc = \frac{N_{Cells\,Tested} - N_{Errors\,Made}}{N_{Cells\,Tested}} \quad (12)$$

where the number of cells incorrectly classified is subtracted from the total number of cells classified and divided by the total number of cells classified. This would produce a number between 0 and 1, inclusive. Random classification should produce an accuracies of 1/2 for discriminating equal amounts of two different cell types and 1/3 for discriminating equal amounts of three different cell types. Therefore, classifying into more cell categories is more difficult for achieving accuracies close to 1.

HL60 Cell Discrimination Results

Figure 16A presents scatter plots of the measurements. We saw that as cells are treated with increasing concentrations of CytoD, their size-corrected population-average balance positions increased in a dose-dependent manner (B), consistent with an increase in the nDEP force and suggesting a concomitant increase in Re[CM].

To explore the utility of measuring multiple frequencies for distinguishing cells, we trained the 0-1 loss classifier to discriminate between the cells given the three derived Re[CM] factors for each cell. We see in Figure 16**Error! Reference source not found.**C that the discrimination ability when classifying between two populations or among three populations improved when increasing the number of frequencies at which we measured balance positions. We also observe that, as expected, it was easiest to distinguish control from 10 μM CytoD. Furthermore, all twelve discrimination accuracies shown correspond to statistically significant

discriminations with paired sample t-test p-values for each ranging from as high as 0.0012 to as low as 3.8e-14 ($p < 0.05$ is significant).

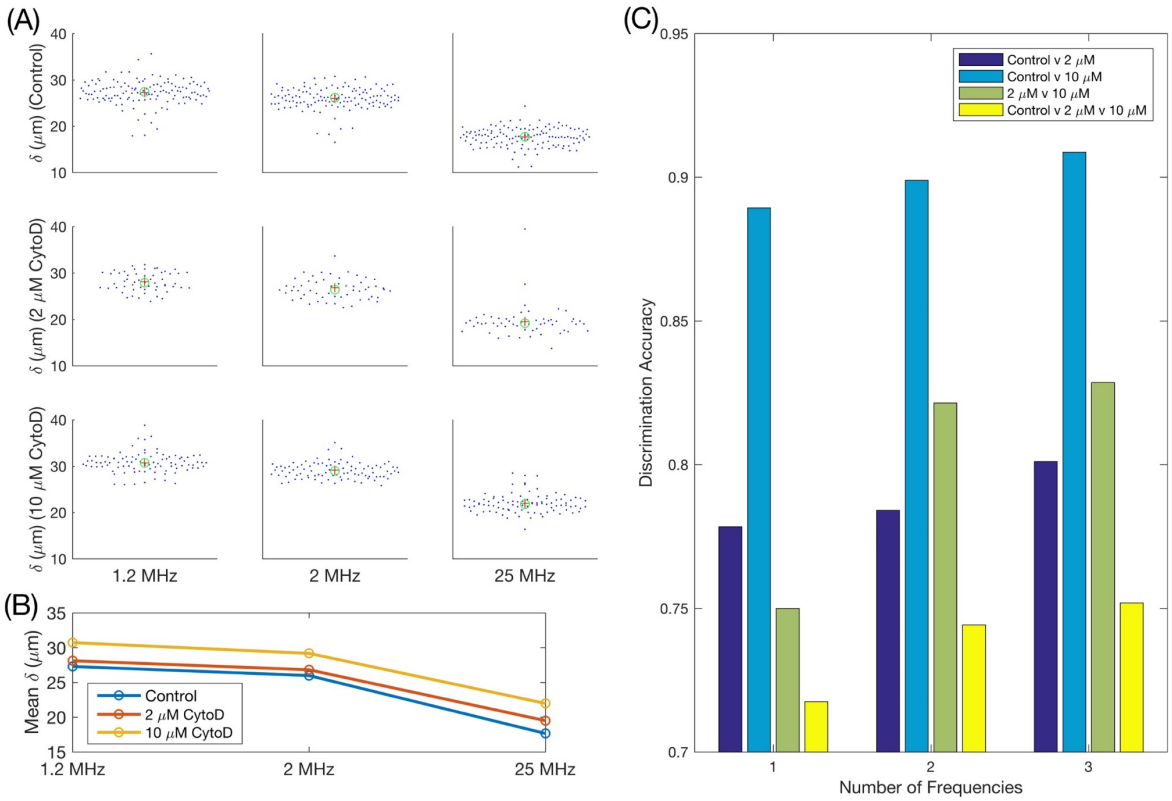

Figure 16: (A): Balance positions for 262 HL60 cells across treatment condition and frequency (+ = mean, o = median). (B): Balance position means at all three frequencies separated by treatment condition. (C): Discrimination accuracies as the number of measured frequencies changes. The yellow bars signify classification into one of three possible classes whereas the other colors signify classification into one of two possible classes.

Conclusions

In this chapter, we experimentally showed that three different populations of HL60 cells are better able to be discriminated when increasing the amount of frequencies in the multi-frequency DEP spring. We empirically proved this claim by attaining validated balance positions with our experimental setup and data validation mentioned in chapters 3 and 4, respectively, and finally with a 0-1 loss classifier used to predict the HL60 cell type via balance position (CM factor) measurements.

Chapter 6: Impact and Future Research Directions

In this final chapter of the thesis, we revisit goals established in the introductory chapter of the thesis, analyze the impact of the results obtained in simulation and experimentation, and consider what work should be done in the future to continue this research.

Furthering the Field of Label-free Single Cell Discrimination

Our goal in introducing the multi-frequency DEP spring was to obtain more information about cells' electrical properties while maintaining throughput. We showed in our simulations that the wider the frequency range is and the more dispersions it has, the more frequencies become useful in discriminating cells. We then defined a metric for assessing the transition from frequency to frequency using BA/F3 cells in order to experimentally ascertain cell balance positions, yielding CM factor values at certain frequencies. Finally, we used three frequencies on single HL60 cells of three different subpopulations to show that as more frequencies are employed to identify the HL60 cells, discrimination accuracy increases.

Considering the simulation we were able to find an optimal number of frequencies to use given both an experimental range and a wider theoretical range. For the former, we observed that three frequencies maximized the information provided, while in the wider range of frequencies, four frequencies substantially increases information. From a theoretical standpoint, we can credit this to being able to access the higher-frequency dispersion in the Re[CM] spectrum with the wider frequency range than the experimental range. From a physical standpoint, we can state that a wider range of frequencies will reach a wider range of compartments and their electrical parameters[24]. In both cases, the precision of the measurements (uncertainty in balance position, in this case) affected the ability to discriminate cells. The two major factors that contribute to this imprecision, in my opinion, are voltage

variations across the field of view to observe balance positions and changes in channel height across the field of view. Regarding voltage variation, the voltage across the slanted electrodes could dip between direct connections to the source due to the fact that the material of the electrodes does have a non-zero amount of resistance. Creating more direct connections from the slanted electrodes to the voltage source should help to mitigate this issue. Regarding channel height, since the channel height is quite small compared to the radius of the cell (often only about two times the radius) small variations in this height could lead to significant changes in balance position via equation (9). Either increasing the channel height or increasing the quality of the PDMS channel mold fabrication to provide a more consistent channel height should help to address this issue.

When we consider which frequencies in the experimental range to use, the results consistently indicate that a frequency high in the range and a frequency low in the range should be included. This is not surprising given that in the single-cell model the high frequencies probe different parameters than the low frequencies. However, the particular optimal frequencies varied with each simulation run, as expected for a stochastic model.

Results with Cytochalasin-treated cells highlight the improved information content derived when measuring cells at multiple frequencies. Here we see increasing accuracy with increasing number of frequencies, regardless of the comparison set. Although we focused experimentally on measuring 3 frequencies, informed by the simulation results, it might be useful experimentally to measure cells at additional frequencies. This is because cells' electrical properties do not necessarily follow the single-shell model exactly. The upper bound on the number of balance positions that one can measure are based on the flowrate and the imaging field-of-view. Decreasing flow rate would increase the residence time and thus the ability to measure additional frequencies, though it could also increase the time needed to achieve a particular balance position. Increasing the microscope field-of-view would also increase the

number of frequencies that could be measured, although typically with a tradeoff in spatial resolution and thus potentially increased uncertainty in the balance position.

On the topic of uncertainty, one can observe that measuring cells of the same population with our device can produce noticeably different balance positions. Although true biophysical differences among cells within the same population would yield different balance positions, measurement error also plays a role in adding uncertainty to the balance position measurement. Causes of measurement error include variations in channel height along the length of the channel within the field of view causing an unintended x-axis dependence of balance position, as well as variations in the electrodes producing the electric field acting on the cells. Improved control of such factors would help to increase discrimination accuracy. In addition, one can leverage repeated observations of the cell as it traverses the field of view to develop a better estimate of the balance position.

The same device was used for all experiments mentioned. This allowed us to maintain relatively constant device properties, such as channel width, channel height, and electric field at a given frequency, across experiments. These device properties are factors in determining the CM factors of the cells traveling through the device channel and can vary from device to device if separate devices are not fabricated in exactly the same manner. Therefore, using separate devices for the same experiment would add uncertainty to our results.

Comparing this method to other methods of discriminating single cells by their electrical properties, we find that our throughput (up to ~1 cell/sec) exceeds that of electrorotation, where the typical maximum throughput is ~5 cells/hr[14,15], which is considerably lower than our maximum throughput, although the depth of analysis is greater (~20 frequencies/cell[11]). Regarding impedance cytometry, we see that the throughput is much higher (~1000 cells/sec), but the typical maximum number of frequencies per cell is two, while we've shown our method yielding three data points per cell in the form of balance positions, and this can be easily increased. Finally, comparing to other DEP-based electrical cell measurement techniques, most

single-cell DEP methods measure cell properties at a single frequency[20]; we've shown utility in measuring more frequencies per cell as a means of increasing cell discrimination ability.

Contributions

There are a few contributions this individual work made to addressing the broader problem of utilizing microfluidics to identify single cells by their electrical properties.

First, it provides a general simulation to discriminate cell models by means of their CM factors by utilizing an arbitrary number of frequencies. This simulation was applied to the multi-frequency dielectrophoretic spring but could easily be adapted to be applied to other cell identification methods, such as electrorotation. Hopefully, this simulation will become a robust stepping stone to evaluate electrical frequency-based cell identification methods in the future.

Second, this thesis created an automated method for validating balance positions, contributing to ensuring that reliable data is used for analysis of multi-frequency DEP spring experiments. Unreliable data is likely to lead to unreliable results and this data validation method specifically defines and separates reliable data (balance positions) for use apart from unreliable data.

Third, it offers a rigorous, albeit slow, method to accurately assess discrimination ability given cell balance position data. Today, 0-1 loss classification is a rarely utilized method of classification due to its lack of convexity contributing to slow processing time and lack of continuity in its objective function. However, in order to properly determine discrimination accuracy differences with utmost precision, 0-1 loss was necessary to use. Therefore, a relevant and necessary application of 0-1 loss has been applied to the field of cell identification by electrical properties.

Finally, the key goal of this thesis was accomplished in that we rigorously and completely answered the question through simulation and experimentation of if increasing the amount of tested frequencies does indeed increase discrimination ability of a single cell.

Furthermore, we showed via simulation which frequencies could be optimal in experimentation to most effectively discriminate arbitrary cells. In experimentation, we showed that as we incrementally increased the number of frequencies, we observed a subsequent increase in discrimination ability.

Future Steps

Despite showing utility with our method, we note that our current throughput is indeed modest. Several approaches exist to increase throughput, such as by increasing cell concentration and using image tracking methods to keep track of all of the cells in the field of view. Furthermore, we could increase flow rate while also increasing the voltage peak-to-peak amplitude between the electrodes to assure balance positions would still be attained. Field strength could also be increased, allowing a higher flow rate, by putting electrodes on top of the channel as well as on the bottom of the channel, as we have done in other work[30].

An important next step in our experimentation would be showing our ability to discriminate cells in a mixture. The experiments presented here have discriminated cells in pure populations. In order to discriminate within mixed populations, we could use a label to provide ground-truth information as to the cell identity. As long as cells are far enough apart such that they do not interact electrically or hydrodynamically, measuring in mixed populations should be straightforward.

Conclusions

This thesis shows through simulation and experiments that measuring electrical properties of cells at multiple frequencies for single cells increases our ability to discriminate them. We first quantify how this ability improves through simulation of testing multiple frequencies per cell, improving by showing fewer and fewer CM factor spectra within tolerance

thresholds as we incrementally increase the amount of frequencies per cell from one to four. We then establish what it means to successfully attain balance positions when transitioning from frequency to frequency within the experimental multi-frequency DEP spring, showing that there is a rough limit to how many frequencies can be tested on a single cell. Finally, we show consistent increases in accuracy when discriminating HL60 cells treated with different concentrations of Cytochalasin D when increasing from one to two and two to three measured frequencies per cell using the optimal frequencies obtained from our simulation.

Appendix

Simulation Code

```
clearvars
close all

load('CM2BP.mat')

%% Monte Carlo Simulation of Re[CM]


% USE equation 3.20 of AC Electrokinetics to find CM Factor
e_0 = 8.85e-12;
m = 1000; % number of different cells (particles)
maximum = m;
n = 1000; % number of different frequencies
f = logspace(4,9,n);

a_1(1:m) = 2.00e-6 + 6e-6*rand([1,1000]); %outer shell radius
a_2 = zeros([1,m]);
for k = 1:m
    a_2(k) = a_1(k) - 1e-8; %2e-8*rand([1,1000]); %inner shell radius
end
e_1(1:m) = 78.5*e_0; %constant, media permittivity
e_2(1:m) = 2*e_0 + 18*e_0*rand([1,1000]); %membrane permittivity
e_3(1:m) = 20*e_0 + 40*e_0*rand([1,1000]); %cytoplasm permittivity
s_1(1:m) = 1.5; %constant, media conductivity should be 1e-4
s_2(1:m) = 1e-8 + (1e-4 - 1e-8)*rand([1,1000]); %membrane conductivity
s_3(1:m) = 0.2 + 1.0*rand([1,1000]); %cytoplasm conductivity

comp_e1 = zeros([m,n]);
comp_e2 = zeros([m,n]);
comp_e3 = zeros([m,n]);
for k = 1:1000
comp_e1(k,1:n) = e_1(k) - 1i*s_1(k)./(2*pi*f);
comp_e2(k,1:n) = e_2(k) - 1i*s_2(k)./(2*pi*f);
comp_e3(k,1:n) = e_3(k) - 1i*s_3(k)./(2*pi*f);
end

gamma = zeros([m,n]);
for k = 1:n
gamma(:,k) = a_1./a_2;
end
perm_diff = (comp_e3 - comp_e2)./(comp_e3 + 2*comp_e2);
complex_perm = comp_e2.*(gamma.^3 + 2*perm_diff)./(gamma.^3 - perm_diff);
CM(1:m,1:n) = (complex_perm - comp_e1)./(complex_perm + 2*comp_e1);

%figure(1)
figure(1)
```

```
subplot(2,4,1)
semilogx(f,real(CM(1,:)))
hold on
for k = 2:m
semilogx(f,real(CM(k,:)))
end
xlabel('Frequency (Hz)','fontsize',12)
ylabel('Re[CM]','fontsize',12)
xlim([1e4 1e9])

text(-0.2,1.2,'(A)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)
hold off



%%
figure(1)
subplot(2,4,2)

total = 100;
m = 1000; % number of different cells (particles)
n = 1000; % number of different frequencies
f = logspace(4,9,n);
f = f';

e_0 = 8.85e-12;

%% Creation of Intelligent Frequency Dependent Noise Thresholds

load('CM2BP.mat') % made by observing Re[CM] mapping to BP in
freqSeq_simulation.m (device simulation)
ReCMrange = -0.5:0.01:-0.01;
ReCMrange = ReCMrange'; % proportional to a monotonic increase in frequency
%BP_noise = 0.25e-6;
BP_noise = 0.5e-6; %in microns, the estimated maximum balance position
difference we see with same CM factor
%BP_noise = 1e-6;
difference = diff(CM2BP(:,2)); % specific to the device simulation
diffend = difference(end);
diff2 = abs([difference; diffend])/0.01; % the difference in balance position
from 0.01 CM increment
% normalizing to account for 0.01 CM increment per balance position
threshold_converter = [ReCMrange diff2];
thresholds = [ReCMrange BP_noise./threshold_converter(:,2)];


%% Establish Frequency Banks

freq_bank_all = [1e4*(1:2:9) 1e5*(1:2:9) 1e6*(1:2:9) 1e7*(1:2:9) 1e8*(1:2:9)
1e9]; %frequency bank that is currently feasible for experiments


%% The loop of all loops
CM_remaining_single_all = zeros([length(freq_bank_all),total]);
CM_remaining_double_all = zeros([length(freq_bank_all),total]);
```

```
CM_remaining_triple_all = zeros([length(freq_bank_all),total]);
CM_remaining_quad_all = zeros([length(freq_bank_all),total]);

for count = 1:total



a_1(1:m) = 2.00e-6 + 6e-6*rand([1,1000]); %outer shell radius
a_2 = zeros([1,m]);
for k = 1:m
    a_2(k) = a_1(k) - 1e-8; %2e-8*rand([1,1000]); %inner shell radius
end
e_1(1:m) = 78.5*e_0; %constant, media permittivity
e_2(1:m) = 2*e_0 + 18*e_0*rand([1,1000]); %membrane permittivity
e_3(1:m) = 20*e_0 + 40*e_0*rand([1,1000]); %cytoplasm permittivity
s_1(1:m) = 1.5; %constant, media conductivity should be 1e-4
s_2(1:m) = 1e-8 + (1e-4 - 1e-8)*rand([1,1000]); %membrane conductivity
s_3(1:m) = 0.2 + 1.0*rand([1,1000]); %cytoplasm conductivity

comp_e1 = zeros([m,n]);
comp_e2 = zeros([m,n]);
comp_e3 = zeros([m,n]);
for k = 1:m
comp_e1(k,1:n) = e_1(k) - 1i*s_1(k)./(2*pi*f);
comp_e2(k,1:n) = e_2(k) - 1i*s_2(k)./(2*pi*f);
comp_e3(k,1:n) = e_3(k) - 1i*s_3(k)./(2*pi*f);
end

gamma = zeros([m,n]);
for k = 1:n
gamma(:,k) = a_1./a_2;
end
perm_diff = (comp_e3 - comp_e2)./(comp_e3 + 2*comp_e2);
complex_perm = comp_e2.*(gamma.^3 + 2*perm_diff)./(gamma.^3 - perm_diff);
CM(1:m,1:n) = (complex_perm - comp_e1)./(complex_perm + 2*comp_e1); % m CM
factors derived at n frequencies


%freq_bank_all = [1e4*(1:2:9) 1e5*(1:2:9) 1e6*(1:2:9) 1e7*(1:2:9) 1e8*(1:2:9)
1e9]; %frequency bank that is currently feasible for experiments
diff_freq_all = zeros([n,length(freq_bank_all)]);
freq_ind_all = zeros([1 length(freq_bank_all)]); % the index for the
frequency
for i = 1:length(freq_bank_all)
    diff_freq_all(:,i) = f - freq_bank_all(i); % to see which frequency in f
is closest to the
    % given frequency in freq_bank_exp
    [~, idx_freq_all] = min(abs(diff_freq_all(:,i)));
    freq_ind_all(i) = idx_freq_all;
    f_applied_all(i) = f(idx_freq_all); % should be the frequencies we want
to probe with that are most similar to the input frequencies
end

CM_applied_all = zeros([1 length(freq_ind_all)]); %the index for the CM
factor
for i = 1:length(freq_ind_all)
```

```
    CM_applied_all(i) = CM(1,freq_ind_all(i)); % value of a random cell's CM
spectrum
    % at the indicated frequency in f_applied_exp
end

%%

%CM_specificity_single_exp = zeros([length(freq_bank_exp),total]);
good_single_frequencies_all = [];
for i = 1:length(f_applied_all)
    frequencies = f_applied_all(i);
    CM_remaining_single_all(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind_all(i), CM_applied_all(i));
    %CM_specificity_single_all(i,count) = min(size((CM_remaining))); % amount
of random CM spectrums within
    % the threshold at this frequency
    if CM_remaining_single_all(i,count) < 50 % add to list of good
frequencies if it narrows it down enough
        good_single_frequencies_all = [good_single_frequencies_all;
freq_bank_all(i)];
    end
end
[~, idx_all1] = min(CM_remaining_single_all(:,count));
best_single_frequency_all = f_applied_all(idx_all1); % identify the single
frequency which narrows down
% the most, and use it as the first of the two double frequencies
best_single_CM_all = CM_applied_all(idx_all1); %


    % Find best two frequencies for experimental frequency bank
%CM_specificity_double_exp = zeros([length(freq_bank_exp),total]);
good_double_frequencies_all = [];
for i = 1:length(freq_bank_all)
    frequencies = [best_single_frequency_all f_applied_all(i)];
    freq_ind = [freq_ind_all(idx_all1) freq_ind_all(i)];
    CM_applied = [best_single_CM_all CM_applied_all(i)];
    CM_remaining_double_all(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind, CM_applied);
    %CM_specificity_double_all(i,count) = min(size((CM_remaining)));
    if CM_remaining_double_all(i,count) < 50
        good_double_frequencies_all = [good_double_frequencies_all;
frequencies];
    end
end
[~, idx_all2] = min(CM_remaining_double_all(:,count));
best_double_frequencies_all = [best_single_frequency_all
f_applied_all(idx_all2)];
best_double_CM_all = [best_single_CM_all CM_applied_all(idx_all2)];

% Find best three frequencies for experimental frequency bank
%CM_specificity_triple_exp = zeros([length(freq_bank_exp),total]);
good_triple_frequencies_all = [];
for i = 1:length(freq_bank_all)
    frequencies = [best_double_frequencies_all f_applied_all(i)];
    freq_ind = [freq_ind_all(idx_all1) freq_ind_all(idx_all2)
freq_ind_all(i)];
    CM_applied = [best_double_CM_all CM_applied_all(i)];
```

```
    CM_remaining_triple_all(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind, CM_applied);
    %CM_specificity_triple_all(i,count) = min(size((CM_remaining)));
    if CM_remaining_triple_all(i,count) < 50
        good_triple_frequencies_all = [good_triple_frequencies_all;
frequencies];
    end
end
[~, idx_all3] = min(CM_remaining_triple_all(:,count));
best_triple_frequencies_all = [best_double_frequencies_all
f_applied_all(idx_all3)];
best_triple_CM_all = [best_double_CM_all CM_applied_all(idx_all3)];

% Find best four frequencies for experimental frequency bank
%CM_specificity_quad_exp = zeros([length(freq_bank_exp),total]);
good_quad_frequencies_all = [];
for i = 1:length(freq_bank_all)
    frequencies = [best_triple_frequencies_all f_applied_all(i)];
    freq_ind = [freq_ind_all(idx_all1) freq_ind_all(idx_all2)
freq_ind_all(idx_all3) freq_ind_all(i)];
    CM_applied = [best_triple_CM_all CM_applied_all(i)];
    CM_remaining_quad_all(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind, CM_applied);
    %CM_specificity_quad_all(i,count) = min(size((CM_remaining)));
    if CM_remaining_quad_all(i,count) < 50
        good_quad_frequencies_all = [good_quad_frequencies_all; frequencies];
    end
end
[~, idx_all4] = min(CM_remaining_quad_all(:,count));
best_quad_frequencies_all = [best_triple_frequencies_all
f_applied_all(idx_all4)];
best_quad_CM_all = [best_triple_CM_all CM_applied_all(idx_all4)];

end

%%
figure(1)
subplot(2,4,2)
semilogx(freq_bank_all,CM_remaining_single_all/maximum)
%title('First Frequency Optimization')
xlabel('Frequency (Hz)','fontsize',12)
ylabel('Fraction Remaining','fontsize',12)
xlim([1e4 1e9])
text(-0.2,1.2,'(B)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)



% plotSpread
CM_remaining = [(min(CM_remaining_single_all)/maximum)',
(min(CM_remaining_double_all)/maximum)',
(min(CM_remaining_triple_all)/maximum)',
(min(CM_remaining_quad_all)/maximum)'];
figure(1); subplot(2,4,3); plotSpread(CM_remaining)
hold on
figure(1); subplot(2,4,3); scatter([1 2 3 4], mean(CM_remaining,1), 'r+')
figure(1); subplot(2,4,3); scatter([1 2 3 4], median(CM_remaining,1), 'go')
```

```
%title('Increasing Number of Test Frequencies')
xlabel('Number of Frequencies','fontsize',12)
ylabel('Fraction Remaining','fontsize',12)
text(-0.2,1.2,'(C)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)
%legend('Cells Remaining','Mean of Cells Remaining', 'Median of Cells
Remaining')

%%
figure(1)
subplot(2,4,4)
loglog(freq_bank_all,mean(CM_remaining_single_all,2)/maximum)
hold on
loglog(freq_bank_all,mean(CM_remaining_double_all,2)/maximum)
loglog(freq_bank_all,mean(CM_remaining_triple_all,2)/maximum)
loglog(freq_bank_all,mean(CM_remaining_quad_all,2)/maximum)
%legend('One Frequency','Two Frequencies','Three Frequencies','Four
Frequencies')
xlabel('Frequency (Hz)','fontsize',12)
ylabel('Fraction Remaining','fontsize',12)
%title('Averaging Across Frequency')
xlim([1e4 1e9])
text(-0.2,1.2,'(D)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)
text(0.3,0.95,'1','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0 0.4470 0.7410])
[f1,idxf1] = min(mean(CM_remaining_single_all,2));
scatter(freq_bank_all(idxf1),f1/maximum,'*','MarkerEdgeColor',[0 0.4470
0.7410])
text(0.3,0.71,'2','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0.8500 0.3250 0.0980])
[f2,idxf2] = min(mean(CM_remaining_double_all,2));
scatter(freq_bank_all(idxf2),f2/maximum,'*','MarkerEdgeColor',[0.8500 0.3250
0.0980])
text(0.3,0.44,'3','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0.9290 0.6940 0.1250])
[f3,idxf3] = min(mean(CM_remaining_triple_all,2));
scatter(freq_bank_all(idxf3),f3/maximum,'*','MarkerEdgeColor',[0.9290 0.6940
0.1250])
text(0.3,0.31,'4','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0.4940 0.1840 0.5560])
[f4,idxf4] = min(mean(CM_remaining_quad_all,2));
scatter(freq_bank_all(idxf4),f4/maximum,'*','MarkerEdgeColor',[0.4940 0.1840
0.5560])

%% Monte Carlo Simulation of Re[CM]


% USE equation 3.20 of AC Electrokinetics to find CM Factor
e_0 = 8.85e-12;
m = 1000; % number of different cells (particles)
maximum = m;
n = 1000; % number of different frequencies
f1 = 5.699; f2 = 7.397940008672038; %correspond to 500kHz and 25MHz
f = logspace(f1,f2,n);
```

```
a_1(1:m) = 2.00e-6 + 6e-6*rand([1,1000]); %outer shell radius
a_2 = zeros([1,m]);
for k = 1:m
    a_2(k) = a_1(k) - 1e-8; %2e-8*rand([1,1000]); %inner shell radius
end
e_1(1:m) = 78.5*e_0; %constant, media permittivity
e_2(1:m) = 2*e_0 + 8*e_0*rand([1,1000]); %membrane permittivity
e_3(1:m) = 20*e_0 + 40*e_0*rand([1,1000]); %cytoplasm permittivity
s_1(1:m) = 1.5; %constant, media conductivity should be 1e-4
s_2(1:m) = 1e-8 + (1e-4 - 1e-8)*rand([1,1000]); %membrane conductivity
s_3(1:m) = 0.2 + 1.0*rand([1,1000]); %cytoplasm conductivity

comp_e1 = zeros([m,n]);
comp_e2 = zeros([m,n]);
comp_e3 = zeros([m,n]);
for k = 1:1000
comp_e1(k,1:n) = e_1(k) - 1i*s_1(k)./(2*pi*f);
comp_e2(k,1:n) = e_2(k) - 1i*s_2(k)./(2*pi*f);
comp_e3(k,1:n) = e_3(k) - 1i*s_3(k)./(2*pi*f);
end

gamma = zeros([m,n]);
for k = 1:n
gamma(:,k) = a_1./a_2;
end
perm_diff = (comp_e3 - comp_e2)./(comp_e3 + 2*comp_e2);
complex_perm = comp_e2.*(gamma.^3 + 2*perm_diff)./(gamma.^3 - perm_diff);
CM(1:m,1:n) = (complex_perm - comp_e1)./(complex_perm + 2*comp_e1);

%figure(1)
figure(1)
subplot(2,4,5)
semilogx(f,real(CM(1,:)))
hold on
for k = 2:m
semilogx(f,real(CM(k,:)))
end
xlim([5e5 2.5e7])
xlabel('Frequency (Hz)','fontsize',12)
ylabel('Re[CM]','fontsize',12)
text(-0.2,1.2,'(E)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)
%title('Monte Carlo Simulation of CM Factor')

% hax=axes;
% SP=5e5; %your point goes here
% line([SP SP],get(hax,'YLim'),'Color',[1 0 0])
% SP=5e5; %your point goes here
% line([SP SP],get(hax,'YLim'),'Color',[1 0 0])
hold off



%%
figure(1)
subplot(2,4,6)
```

```
total = 100;
m = 1000; % number of different cells (particles)
n = 1000; % number of different frequencies
f = logspace(f1,f2,n);
f = f';

e_0 = 8.85e-12;

%% Creation of Intelligent Frequency Dependent Noise Thresholds

load('CM2BP.mat') % made by observing Re[CM] mapping to BP in
freqSeq_simulation.m (device simulation)
ReCMrange = -0.5:0.01:-0.01;
ReCMrange = ReCMrange'; % proportional to a monotonic increase in frequency
%BP_noise = 0.25e-6;
BP_noise = 0.5e-6; %in microns, the estimated maximum balance position
difference we see with same CM factor
%BP_noise = 1e-6;
difference = diff(CM2BP(:,2)); % specific to the device simulation
diffend = difference(end);
diff2 = abs([difference; diffend])/0.01; % the difference in balance position
from 0.01 CM increment
% normalizing to account for 0.01 CM increment per balance position
threshold_converter = [ReCMrange diff2];
thresholds = [ReCMrange BP_noise./threshold_converter(:,2)];


%% Establish Frequency Banks

freq_bank_exp = 1e5*[5 6 7 8 9 10 12 15 20 25 30 40 50 60 70 80 90 100 120
150 200 250];

%% The loop of all loops
CM_remaining_single_exp = zeros([length(freq_bank_exp),total]);
CM_remaining_double_exp = zeros([length(freq_bank_exp),total]);
CM_remaining_triple_exp = zeros([length(freq_bank_exp),total]);
CM_remaining_quad_exp = zeros([length(freq_bank_exp),total]);

for count = 1:total



a_1(1:m) = 2.00e-6 + 6e-6*rand([1,1000]); %outer shell radius
a_2 = zeros([1,m]);
for k = 1:m
    a_2(k) = a_1(k) - 1e-8; %2e-8*rand([1,1000]); %inner shell radius
end
e_1(1:m) = 78.5*e_0; %constant, media permittivity
e_2(1:m) = 2*e_0 + 8*e_0*rand([1,1000]); %membrane permittivity
e_3(1:m) = 20*e_0 + 40*e_0*rand([1,1000]); %cytoplasm permittivity
s_1(1:m) = 1.5; %constant, media conductivity should be 1e-4
s_2(1:m) = 1e-8 + (1e-4 - 1e-8)*rand([1,1000]); %membrane conductivity
s_3(1:m) = 0.2 + 1.0*rand([1,1000]); %cytoplasm conductivity

comp_e1 = zeros([m,n]);
comp_e2 = zeros([m,n]);
```

```
comp_e3 = zeros([m,n]);
for k = 1:m
comp_e1(k,1:n) = e_1(k) - 1i*s_1(k)./(2*pi*f);
comp_e2(k,1:n) = e_2(k) - 1i*s_2(k)./(2*pi*f);
comp_e3(k,1:n) = e_3(k) - 1i*s_3(k)./(2*pi*f);
end

gamma = zeros([m,n]);
for k = 1:n
gamma(:,k) = a_1./a_2;
end
perm_diff = (comp_e3 - comp_e2)./(comp_e3 + 2*comp_e2);
complex_perm = comp_e2.*(gamma.^3 + 2*perm_diff)./(gamma.^3 - perm_diff);
CM(1:m,1:n) = (complex_perm - comp_e1)./(complex_perm + 2*comp_e1); % m CM
factors derived at n frequencies


%freq_bank_all = [1e4*(1:2:9) 1e5*(1:2:9) 1e6*(1:2:9) 1e7*(1:2:9) 1e8*(1:2:9)
1e9*(1:2:9)]; %frequency bank that is currently feasible for experiments
diff_freq_exp = zeros([n,length(freq_bank_exp)]);
freq_ind_exp = zeros([1 length(freq_bank_exp)]); % the index for the
frequency
for i = 1:length(freq_bank_exp)
    diff_freq_exp(:,i) = f - freq_bank_exp(i); % to see which frequency in f
is closest to the
    % given frequency in freq_bank_exp
    [~, idx_freq_exp] = min(abs(diff_freq_exp(:,i)));
    freq_ind_exp(i) = idx_freq_exp;
    f_applied_exp(i) = f(idx_freq_exp); % should be the frequencies we want
to probe with that are most similar to the input frequencies
end

CM_applied_exp = zeros([1 length(freq_ind_exp)]); %the index for the CM
factor
for i = 1:length(freq_ind_exp)
    CM_applied_exp(i) = CM(1,freq_ind_exp(i)); % value of a random cell's CM
spectrum
    % at the indicated frequency in f_applied_exp
end



%CM_specificity_single_exp = zeros([length(freq_bank_exp),total]);
good_single_frequencies_exp = [];
for i = 1:length(f_applied_exp)
    frequencies = f_applied_exp(i);
    CM_remaining_single_exp(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind_exp(i), CM_applied_exp(i));
    %CM_specificity_single_exp(i,count) = min(size((CM_remaining))); % amount
of random CM spectrums within
    % the threshold at this frequency
    if CM_remaining_single_exp(i,count) < 50 % add to list of good
frequencies if it narrows it down enough
        good_single_frequencies_exp = [good_single_frequencies_exp;
freq_bank_exp(i)];
    end
end
```

```
[~, idx_exp1] = min(CM_remaining_single_exp(:,count));
best_single_frequency_exp = f_applied_exp(idx_exp1); % identify the single
frequency which narrows down
% the most, and use it as the first of the two double frequencies
best_single_CM_exp = CM_applied_exp(idx_exp1); %



    % Find best two frequencies for experimental frequency bank
%CM_specificity_double_exp = zeros([length(freq_bank_exp),total]);
good_double_frequencies_exp = [];
for i = 1:length(freq_bank_exp)
    frequencies = [best_single_frequency_exp f_applied_exp(i)];
    freq_ind = [freq_ind_exp(idx_exp1) freq_ind_exp(i)];
    CM_applied = [best_single_CM_exp CM_applied_exp(i)];
    CM_remaining_double_exp(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind, CM_applied);
    %CM_specificity_double_exp(i,count) = min(size((CM_remaining)));
    if CM_remaining_double_exp(i,count) < 50
        good_double_frequencies_exp = [good_double_frequencies_exp;
frequencies];
    end
end
[~, idx_exp2] = min(CM_remaining_double_exp(:,count));
best_double_frequencies_exp = [best_single_frequency_exp
f_applied_exp(idx_exp2)];
best_double_CM_exp = [best_single_CM_exp CM_applied_exp(idx_exp2)];

% Find best three frequencies for experimental frequency bank
%CM_specificity_triple_exp = zeros([length(freq_bank_exp),total]);
good_triple_frequencies_exp = [];
for i = 1:length(freq_bank_exp)
    frequencies = [best_double_frequencies_exp f_applied_exp(i)];
    freq_ind = [freq_ind_exp(idx_exp1) freq_ind_exp(idx_exp2)
freq_ind_exp(i)];
    CM_applied = [best_double_CM_exp CM_applied_exp(i)];
    CM_remaining_triple_exp(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind, CM_applied);
    %CM_specificity_triple_exp(i,count) = min(size((CM_remaining)));
    if CM_remaining_triple_exp(i,count) < 50
        good_triple_frequencies_exp = [good_triple_frequencies_exp;
frequencies];
    end
end
[~, idx_exp3] = min(CM_remaining_triple_exp(:,count));
best_triple_frequencies_exp = [best_double_frequencies_exp
f_applied_exp(idx_exp3)];
best_triple_CM_exp = [best_double_CM_exp CM_applied_exp(idx_exp3)];

% Find best four frequencies for experimental frequency bank
%CM_specificity_quad_exp = zeros([length(freq_bank_exp),total]);
good_quad_frequencies_exp = [];
for i = 1:length(freq_bank_exp)
    frequencies = [best_triple_frequencies_exp f_applied_exp(i)];
    freq_ind = [freq_ind_exp(idx_exp1) freq_ind_exp(idx_exp2)
freq_ind_exp(idx_exp3) freq_ind_exp(i)];
    CM_applied = [best_triple_CM_exp CM_applied_exp(i)];
```

```
    CM_remaining_quad_exp(i,count) = mccm_no_plot_adj(frequencies,
thresholds, ReCMrange, CM, freq_ind, CM_applied);
    %CM_specificity_quad_exp(i,count) = min(size((CM_remaining)));
    if CM_remaining_quad_exp(i,count) < 50
        good_quad_frequencies_exp = [good_quad_frequencies_exp; frequencies];
    end
end
[~, idx_exp4] = min(CM_remaining_quad_exp(:,count));
best_quad_frequencies_exp = [best_triple_frequencies_exp
f_applied_exp(idx_exp4)];
best_quad_CM_exp = [best_triple_CM_exp CM_applied_exp(idx_exp4)];

end

%%
figure(1)
subplot(2,4,6)
semilogx(freq_bank_exp,CM_remaining_single_exp/maximum)
xlim([5e5 2.5e7])
%title('First Frequency Optimization')
xlabel('Frequency (Hz)','fontsize',12)
ylabel('Fraction Remaining','fontsize',12)
text(-0.2,1.2,'(F)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)


%%

% plotSpread
CM_remaining = [(min(CM_remaining_single_exp)/maximum)',
(mean(CM_remaining_double_exp)/maximum)',
(mean(CM_remaining_triple_exp)/maximum)',
(mean(CM_remaining_quad_exp)/maximum)'];
figure(1); subplot(2,4,7); plotSpread(CM_remaining)
hold on
figure(1); subplot(2,4,7); scatter([1 2 3 4], mean(CM_remaining,1), 'r+')
figure(1); subplot(2,4,7); scatter([1 2 3 4], median(CM_remaining,1), 'go')


%title('Increasing Number of Test Frequencies')
xlabel('Number of Frequencies','fontsize',12)
ylabel('Fraction Remaining','fontsize',12)
text(-0.2,1.2,'(G)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)
%legend('Cells Remaining','Mean of Cells Remaining', 'Median of Cells
Remaining')
%title('Averaging Across Number of Test Frequencies')


%%
figure(1)
subplot(2,4,8)
loglog(freq_bank_exp,mean(CM_remaining_single_exp,2)/maximum)
hold on
loglog(freq_bank_exp,mean(CM_remaining_double_exp,2)/maximum)
loglog(freq_bank_exp,mean(CM_remaining_triple_exp,2)/maximum)
```

```
loglog(freq_bank_exp,mean(CM_remaining_quad_exp,2)/maximum)
xlim([5e5 2.5e7])
%ylim([7e-2 1])
%legend('One Frequency','Two Frequencies','Three Frequencies','Four
Frequencies')
xlabel('Frequency (Hz)','fontsize',12)
ylabel('Fraction Remaining','fontsize',12)
text(-0.2,1.2,'(H)','Units', 'Normalized', 'VerticalAlignment', 'Top',
'fontsize', 16)
text(0.3,0.95,'1','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0 0.4470 0.7410])
[f1,idxf1] = min(mean(CM_remaining_single_exp,2));
scatter(freq_bank_exp(idxf1),f1/maximum,'*','MarkerEdgeColor',[0 0.4470
0.7410])
text(0.3,0.51,'2','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0.8500 0.3250 0.0980])
[f2,idxf2] = min(mean(CM_remaining_double_exp,2));
scatter(freq_bank_exp(idxf2),f2/maximum,'*','MarkerEdgeColor',[0.8500 0.3250
0.0980])
text(0.3,0.37,'3','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0.9290 0.6940 0.1250])
[f3,idxf3] = min(mean(CM_remaining_triple_exp,2));
scatter(freq_bank_exp(idxf3),f3/maximum,'*','MarkerEdgeColor',[0.9290 0.6940
0.1250])
text(0.3,0.27,'4','Units', 'Normalized', 'VerticalAlignment', 'Top',
'color',[0.4940 0.1840 0.5560])
[f4,idxf4] = min(mean(CM_remaining_quad_exp,2));
scatter(freq_bank_exp(idxf4),f4/maximum,'*','MarkerEdgeColor',[0.4940 0.1840
0.5560])
```

Particle Detection Code:

Input: Image stack
Output: Detected particles per image

```
close all
clearvars
format compact
%% Read files
name_list=dir('Dev24BAF3_1_0005.mat');

%% Load matlab
for i_name= 1:length(name_list)
matName = name_list(i_name).name;
load(matName)
mkdir(matName(1:end-4))

figure(1)
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
% Image Cropping
% stack = stack(:,end-200:end,:);
```

```
% Time Domain Median filter to extract background
background = median(stack(:,:,1:100),3);
%
figure(1),subplot(4,1,1),imshow(background,[0 256])


backgroundEdge=edge(background);
gapCenter = zeros(1,size(background,2));
for i = 1:size(background,2)
    gapCenter(i) = mean(find(backgroundEdge(:,i)==1,2));
end

gapCenter = mean(gapCenter((gapCenter-mean(gapCenter))<std(gapCenter)*2));

% Define electrodes centerline
x1 = 1;
x2 = 1280;
y1 = 50;
y2 = 50;

% pixel size
pitch_pixel = 2*(39-11) ; % pixels
pixel_size = 60/pitch_pixel; %um

% Define features array
Distance_Array = [];
Area_Array = [];
Centroid_Array=[];
Eccentricity_Array =[];
Frame_Array=[];
MeanIntensity_Array = [];
VarIntensity_Array = [];

cellCounter = 1;

for frame_num = 1  :size(stack,3)
    if mod(frame_num,10) ==0
        background = median(stack(:,:,frame_num-9:frame_num),3);
    end

% Display frame
I = stack(:,:,frame_num);
figure(1),subplot(4,1,2), imshow(I,[0 255])
line([x1 x2], [y1 y2],'color','r')

% Substract background
I2 = abs(double(I)-double(background(:,:)));
figure(1),subplot(4,1,3), imshow(I2,[0 255])
line([x1 x2], [y1 y2],'color','r')

% Threshold the image
bw = I2>10;
figure(1),subplot(4,1,4), imagesc(bw,[0 1])

% Identify objects
cellObject = bwconncomp(bw, 4);
```

```
% Extract information
cellArea = regionprops(cellObject, 'FilledArea');
cellCentroid = regionprops(cellObject, 'Centroid');
cellEccentricity = regionprops(cellObject, 'Eccentricity');


% Calculating distance from Line: ax+by+c = 0
for i = 1:cellObject.NumObjects
    pos = cellCentroid(i).Centroid;
    slope = (y2-y1)/(x2-x1);
    a = slope;
    b = -1;
    c = -slope*x1+y1;
    distance = -(a*pos(1)+b*pos(2)+c)/(a^2+b^2)^0.5;
    cellXStart=round(pos(2)-distance/cos(atan((y2-y1)/(x2-x1))));
    cellXEnd=cellXStart+60;
    cellYStart=round(pos(1))-10;
    cellYEnd=round(pos(1))+10;

    if (cellYStart>0 && cellYEnd<=size(I,2) && cellArea(i).FilledArea>5)
    Distance_Array = [Distance_Array; distance;];
    Area_Array = [Area_Array; cellArea(i).FilledArea;];
    Centroid_Array = [Centroid_Array; cellCentroid(i).Centroid];
    Eccentricity_Array = [Eccentricity_Array;
cellEccentricity(i).Eccentricity];
    Frame_Array = [Frame_Array; frame_num;];


    Intensity = double(I2(cellObject.PixelIdxList{i}));
    MeanIntensity_Array = [MeanIntensity_Array; mean(Intensity)];
    VarIntensity_Array = [VarIntensity_Array; var(Intensity)];

    cellCounter = cellCounter+1;
    end
end
%cc
%pause
end

% Unit conversion
Area_Array = Area_Array*pixel_size^2;
Distance_Array = Distance_Array*pixel_size;
Centroid_Array = Centroid_Array*pixel_size;

% Save feature arrays
save(sprintf('data%s.mat',matName(1:end-
4)),'Area_Array','Distance_Array','Centroid_Array','Eccentricity_Array','Fram
e_Array','MeanIntensity_Array','VarIntensity_Array','paramsStack')

%% Display Data
% Filter data with selection
Distance_upper =  100;
Distance_lower =   0;
Area_upper     = 200;
Area_lower     =   0;
```

```
selection = Distance_Array<60 & Distance_Array>0 & Area_Array<200  &
Area_Array >pi*2^2;

figure(3)
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
% Plot Histogram of Distance
nbins = 100;
figure(3),subplot(1,3,1), hist(Distance_Array(selection),
linspace(Distance_lower,Distance_upper,nbins))
title('Histogram of Distance from the center of the electrodes \delta');
xlabel('\delta (\mum)')
ylabel('Number of cells')
xlim([Distance_lower Distance_upper])
%ylim([0 400])

% Plot Histogram of Size
nbins = 100;
figure(3),subplot(1,3,2),  hist(Area_Array(selection),
linspace(Area_lower,Area_upper,nbins))
title('Histogram of Area');
xlabel('Area \mum^2')
ylabel('Number of cells')
xlim([Area_lower Area_upper])
%ylim([0 250])

% Plot Size and Distance Scatter Plot
figure(3),subplot(1,3,3),
plot(Area_Array(selection),Distance_Array(selection),'.')
title('Scatter Plot for \delta');
xlabel('Area \mum^2')
ylabel('Distance from the center of the electrodes\mum')
xlim([Area_lower Area_upper])
ylim([Distance_lower Distance_upper])

saveas(gcf, sprintf('Hist_%s.tif',matName(1:end-4)),'tif')

%% Display cell
Time_Array = zeros(length(Frame_Array),1);

for i = 1: length(Frame_Array)
    Time_Array(i) = paramsStack.time(Frame_Array(i));
end

figure, plot(Time_Array, Distance_Array,'b.')
hold on
plot(paramsStack.time,paramsStack.frequency/1e6,'r')
grid on
%xlim([0 10])
title('Dancing Cell')
xlabel('time(sec)')
ylabel('balance position(\mum)')

end
```

Particle Tracking Code

Input: Detected particles per image
Output: Selected list of tracked particles per image stack

```
% added moving average filter to y-coordinates of cells

close all
clear tracks Area_Array Centroid_Array Distance_Array Eccentricity_Array
Frame_Array Freq_Array Freqeuncy_Array i i_name ID_Array img img2 Index_Array
j matName MeanIntensity myparticle name_list param paramsStack pos
SelectedIndex selection Time_Array tracks VarIntensity X_Array Y_Array
%clear tracks
%clear all
%clc
format compact
%% Read files
name_list=dir('dataDev24BAF3_1_0019.mat');
%% Load Matlab
for i_name= 1:length(name_list)
% Filename
matName = name_list(i_name).name;
load(matName)

Time_Array = paramsStack.time(Frame_Array)';

%selection = Distance_Array < 55 & Area_Array>120 &Time_Array>0; %beads
%selection = Distance_Array < 100 & Area_Array>10 &Time_Array>0;
selection = Distance_Array < 55 & Distance_Array > 0 & Area_Array > 30 &
Area_Array < 500 & Time_Array >0 & Time_Array < 65; %cells


% Display cell
figure(1), plot(Time_Array(selection), Distance_Array(selection),'b.')
hold on
plot(paramsStack.time,paramsStack.frequency/1e6,'r')
grid on
%xlim([0 10])
%title(regexprep(sprintf('Dancing Cell: %s',matName),'_','-'))
title('Multiple Cell Trajectories')
xlabel('time(sec)')
ylabel('Distance (\mum), Frequency (MHz)')


end

%% Track particles

% Define array
X_Array = Centroid_Array(:,1);
Y_Array = Distance_Array; %Y_Array = Centroid_Array(:,2);
Time_Array = paramsStack.time(Frame_Array)';
Frequency_Array = paramsStack.frequency(Frame_Array)'/1e6;
Index_Array = [1:length(X_Array)]';
```

```
pos = [X_Array Y_Array Frequency_Array Index_Array Area_Array
MeanIntensity_Array VarIntensity_Array Time_Array];
pos = pos(selection,:);
% Define tracking parameters
param.mem = 0 ;
param.good = 10;
param.dim = 2;
param.quiet = 1;

tracks = track_Dec19(pos,200,param);

%%

X_Array = tracks(:,1);
Y_Array = tracks(:,2);
Freq_Array = tracks(:,3);
Index_Array = tracks(:,4);
Area_Array =  tracks(:,5);
MeanIntensity_Array = tracks(:,6);
VarIntensity_Array = tracks(:,7);
Time_Array = tracks(:,end-1);
ID_Array = tracks(:,end);

%% Showing every possible cell trajectories
ceil(ID_Array(end)^0.5)

 for i = 1:ID_Array(end)
    selection = ID_Array==i;
    figure(1)
    plot(Time_Array(selection),Y_Array(selection),'r')
    hold on
    if length(Time_Array(selection)) > 100
        figure(2)
        subplot(ceil(ID_Array(end)^0.5),ceil(ID_Array(end)^0.5),i)


        plot(Time_Array(selection),Y_Array(selection),'r')
        hold on
        plot(Time_Array(selection),Freq_Array(selection),'b')
        %plot(Time_Array(selection),MeanIntensity_Array(selection)/10,'k')
        %plot(Time_Array(selection),VarIntensity_Array(selection)/100,'g')
        %plot(Time_Array(selection),Area_Array(selection),'k')
        %ylim([0 30])
%         title('Single Cell Tracking')
%         ylabel('Distance from Centroid Line , f (MHz)')
%         xlabel('Time (s)')
%         legend('Distance from Centroid Line (\mum)','Frequency (MHz)')

    end

    figure(3)
    subplot(ceil(ID_Array(end)^0.5),ceil(ID_Array(end)^0.5),i)
    SelectedIndex = Index_Array(selection);
    ylabel(i)
 end

%pause
```

```
 %% Only Showing Good Trajectories and Getting s
ceil(ID_Array(end)^0.5)

 for i = 1:ID_Array(end)          %trying to make automated

    selection = ID_Array==i;
    if length(Time_Array(selection)) > 60  % max time w/o getting all bp's

        myparticle.Time_Array = Time_Array(selection);
        % moving average (4 point)
        cell_y = Y_Array(selection);
        myparticle.Y_Array = zeros(length(cell_y),1);
        for k = 1:4
            myparticle.Y_Array(k) = 1/k*sum(cell_y(1:k));
        end
        for j = 5:length(cell_y)-1
            myparticle.Y_Array(j) = 1/k*sum(cell_y(j-2:j+1));
        end
        myparticle.Y_Array(end) = 1/3*sum(cell_y(end-2:end));
        %myparticle.Y_Array = Y_Array(selection);
        myparticle.X_Array = X_Array(selection);
        myparticle.Freq_Array = Freq_Array(selection);
        myparticle.Area_Array = Area_Array(selection);
        myparticle.Area = mean(myparticle.Area_Array(1:20));

        %balance positions specific to each frequency for each particle
        myparticle.Diff_Freq = diff(myparticle.Freq_Array);
        idxDiff = find(myparticle.Diff_Freq ~= 0);

        % The rule to determine what constitutes a balance position is the
        % difference of the y-tracking from the balance position to 5 points
before
        % the balance position needing to be a small enough fraction of the
        % average balance position difference between the frequencies

        myparticle.balance_points = [myparticle.Freq_Array(idxDiff)
myparticle.Y_Array(idxDiff+1)];
        myparticle.bp_var = zeros(length(idxDiff),2);
        for l = 1:length(idxDiff)
            if min(idxDiff) > 3
                myparticle.bp_var(l,:) = [myparticle.Freq_Array(idxDiff(l))
var(cell_y(idxDiff(l)-3:idxDiff(l)))];
            end
        end
        myparticle.bp_std = zeros(length(idxDiff),2);
        for l = 1:length(idxDiff)
            if min(idxDiff) > 3
                myparticle.bp_std(l,:) = [myparticle.Freq_Array(idxDiff(l))
std(cell_y(idxDiff(l)-3:idxDiff(l)))];
            end
        end
        % the +1 in the Y_Array accounts for the large transition time we
        % were neglecting beforehand from one frequency to the next: 1.2 to
        % 2 and 25 to 1.2 are about 2 data points to transition, 2 to 25 is
        % about 5 data points to transition
        % Additionally, including transition time makes the time at each
```

```
        % frequency longer

        ListOfParticles = [ListOfParticles myparticle];
    end

 end

 save(sprintf('ListOfParticle_%s_SingleCell_movavg1.mat',matName(1:end-
4)),'ListOfParticles')
```

Balance Position Validation Code

Input: Selected list of tracked particles
Output: Selected list of balance position validated tracked particles

```
%  verification and cell processing

        % The rule to determine what constitutes a balance position is the
        % difference of the y-tracking from the balance position to 5 points
before
        % the balance position needing to be a small enough fraction of the
        % average balance position difference between the frequencies

% get balance position of each cell that matters
load('ListOfParticles_HL60.mat')
%ListOfParticles_control = ListOfParticles;
ListOfParticles = ListOfParticles_CytoD2uM;

%% Get Differences in Balance position Means

bp1_2 = []; %zeros(length(ListOfParticles),1);
bp2 = []; %zeros(length(ListOfParticles),1);
bp25 = []; %zeros(length(ListOfParticles),1);

% add to arrays
    for i = 1:length(ListOfParticles)
        if sum(ListOfParticles(i).balance_points(:,1)) > 28
            for j = 1:3 % because there are 3 frequencies, disregarding later
balance positions
                if ListOfParticles(i).balance_points(j,1) == 1.2;
                    bp1_2 = [bp1_2; ListOfParticles(i).balance_points(j,2)];
                elseif ListOfParticles(i).balance_points(j,1) == 2;
                    bp2 = [bp2; ListOfParticles(i).balance_points(j,2)];
                elseif ListOfParticles(i).balance_points(j,1) == 25;
                    bp25 = [bp25; ListOfParticles(i).balance_points(j,2)];
                end
            end
        end
    end

bp25bp1_2diff = mean(bp1_2) - mean(bp25);
bp1_2bp2diff = mean(bp2) - mean(bp1_2);
bp2bp25diff = mean(bp25) - mean(bp2);
```

```
%% Balance position Processing

% get rid of all invalid balance positions, weed out valid balance positions
that
% aren't stable enough, get rid of all cells with less than three balance
% points, and only keep first three balance positions

rows2delete = [];
for i = 1:length(ListOfParticles)
    bp2delete = [];
    for j = 1:size(ListOfParticles(i).balance_points,1)
        if ListOfParticles(i).balance_points(j,1) == 0
            bp2delete = [bp2delete; j];
        end
    end
    ListOfParticles(i).balance_points(bp2delete,:) = [];
    % finding indices while accounting for error from going from 0 MHz at
    % turn on
    idxDiff = find(ListOfParticles(i).Diff_Freq ~= 0 &
ListOfParticles(i).Diff_Freq ~= 1.2 & ListOfParticles(i).Diff_Freq ~= 2 &
ListOfParticles(i).Diff_Freq ~= 25);
    if min(idxDiff) > 5
        bp_y = ListOfParticles(i).Y_Array(idxDiff-5);
    else
        bp_y = zeros(size(ListOfParticles(i).balance_points,1),1);
        bp_y(1) = ListOfParticles(i).Y_Array(1);
        if size(bp_y,1) > 1
            bp_y(2:end) = ListOfParticles(i).Y_Array(idxDiff(2:end)-5);
        end
    end
    bp2delete2 = [];
    for j = 1:size(ListOfParticles(i).balance_points,1)
        if ListOfParticles(i).balance_points(j,1) == 25
            if bp2bp25diff*0.135 > bp_y(j) -
ListOfParticles(i).balance_points(j,2)
                % because reference is negative, two time constants for
                % first order
                % about 300 ms out of 1276 ms
                bp2delete2 = [bp2delete2; j];
            end
        elseif ListOfParticles(i).balance_points(j,1) == 2
            if bp1_2bp2diff*0.135 > bp_y(j) -
ListOfParticles(i).balance_points(j,2)
                % because reference is negative, two time constants for
                % first order
                % about 220 ms out of 766 ms
                bp2delete2 = [bp2delete2; j];
            end
        elseif ListOfParticles(i).balance_points(j,1) == 1.2
            if bp25bp1_2diff*0.135 < ListOfParticles(i).balance_points(j,2) -
bp_y(j)
                % because reference is positive, two time constants for
                % first order
                % about 220 ms out of 1785 ms
                bp2delete = [bp2delete; j];
```

```
            end
        end
    end
    ListOfParticles(i).balance_points(bp2delete2,:) = [];

    one_point_two = find(ListOfParticles(i).balance_points(:,1) == 1.2);
    two = find(ListOfParticles(i).balance_points(:,1) == 2);
    twenty_five = find(ListOfParticles(i).balance_points(:,1) == 25);
    bp_logic = [not(isempty(one_point_two)); not(isempty(two));
not(isempty(twenty_five))];
    bp_new = []; %only account for one balance position of each frequency
    if bp_logic(1) ~= 0
        bp_new = [bp_new;
ListOfParticles(i).balance_points(one_point_two(1),:)];
    end
    if bp_logic(2) ~= 0
        bp_new = [bp_new; ListOfParticles(i).balance_points(two(1),:)];
    end
    if bp_logic(3) ~= 0
        bp_new = [bp_new;
ListOfParticles(i).balance_points(twenty_five(1),:)];
    end
    ListOfParticles(i).balance_points = bp_new;
    if length(ListOfParticles(i).balance_points) < 3
        rows2delete = [rows2delete; i];
    end
    ListOfParticles(i).Radius = sqrt(ListOfParticles(i).Area/pi);
end
ListOfParticles(rows2delete) = [];
```

Radii Adjustment Code

```
clearvars
close all

%% Get true radii mean for 3 populations

load('coultercounter_size.mat')

radii = zeros(length(range)-1,1);
for i = 1:length(range)-1
    radii(i) = (range(i)+range(i+1))/4; % getting average radius from
diameter range
end

radii_control = []; radii_cd2uM = []; radii_cd10uM = [];
for i = 1:length(count_control)-1
    radii_control = [radii_control; radii(i)*ones(count_control(i),1)];
end
for i = 1:length(count_cd2uM)-1
    radii_cd2uM = [radii_cd2uM; radii(i)*ones(count_cd2uM(i),1)];
end
for i = 1:length(count_cd10uM)-1
    radii_cd10uM = [radii_cd10uM; radii(i)*ones(count_cd10uM(i),1)];
```

```
end

mean_control = mean(radii_control);
mean_cd2uM = mean(radii_cd2uM);
mean_cd10uM = mean(radii_cd10uM);
var_control = var(radii_control);
var_cd2uM = var(radii_cd2uM);
var_cd10uM = var(radii_cd10uM);

%% Get flawed radii mean for 3 populations

load('ListOfParticles_HL60_processed.mat')


flawed_mean_control = mean([ListOfParticles_control.Radius]);
flawed_mean_CytoD2uM = mean([ListOfParticles_CytoD2uM.Radius]);
flawed_mean_CytoD10uM = mean([ListOfParticles_CytoD10uM.Radius]);

flawed_var_control = var([ListOfParticles_control.Radius]);
flawed_var_CytoD2uM = var([ListOfParticles_CytoD2uM.Radius]);
flawed_var_CytoD10uM = var([ListOfParticles_CytoD10uM.Radius]);

%% Adjust flawed radii by differences in means

meandiff_control = mean_control - flawed_mean_control;
meandiff_CytoD2uM = mean_cd2uM - flawed_mean_CytoD2uM;
meandiff_CytoD10uM = mean_cd10uM - flawed_mean_CytoD10uM;

for i = 1:length(ListOfParticles_control)
    ListOfParticles_control(i).Radius = ListOfParticles_control(i).Radius +
meandiff_control;
end
for i = 1:length(ListOfParticles_CytoD2uM)
    ListOfParticles_CytoD2uM(i).Radius = ListOfParticles_CytoD2uM(i).Radius +
meandiff_CytoD2uM;
end
for i = 1:length(ListOfParticles_CytoD10uM)
    ListOfParticles_CytoD10uM(i).Radius = ListOfParticles_CytoD10uM(i).Radius
+ meandiff_CytoD10uM;
end

%% Adjusting flawed radii by differences in variance

% How do I do this. I know the formula for variance is E[(X-u)^2]. So we
% need to (slightly) amplify the squared difference to reflect the slightly
% larger variance

var_control_ratio = var_control/flawed_var_control;
var_cd2uM_ratio = var_cd2uM/flawed_var_CytoD2uM;
var_cd10uM_ratio = var_cd10uM/flawed_var_CytoD10uM;

for i = 1:length(ListOfParticles_control)
    ListOfParticles_control(i).Radius = mean_control +
(ListOfParticles_control(i).Radius - mean_control)*sqrt(var_control_ratio);
end
for i = 1:length(ListOfParticles_CytoD2uM)
```

```
    ListOfParticles_CytoD2uM(i).Radius = mean_cd2uM +
(ListOfParticles_CytoD2uM(i).Radius - mean_cd2uM)*sqrt(var_cd2uM_ratio);
end
for i = 1:length(ListOfParticles_CytoD10uM)
    ListOfParticles_CytoD10uM(i).Radius = mean_cd10uM +
(ListOfParticles_CytoD10uM(i).Radius - mean_cd10uM)*sqrt(var_cd10uM_ratio);
end

%% Checking

fixed_mean_control = mean([ListOfParticles_control.Radius]);
fixed_mean_CytoD2uM = mean([ListOfParticles_CytoD2uM.Radius]);
fixed_mean_CytoD10uM = mean([ListOfParticles_CytoD10uM.Radius]);

fixed_var_control = var([ListOfParticles_control.Radius]);
fixed_var_CytoD2uM = var([ListOfParticles_CytoD2uM.Radius]);
fixed_var_CytoD10uM = var([ListOfParticles_CytoD10uM.Radius]);
```

Balance Position Calibration and Conversion to Re[CM] Code

Input: Selected list of balance position validated tracked particles
Output: Re[CM]'s of each validated particle for each balance position at each frequency

```
clearvars
close all


%% Swarm Plot Beads Script

swarm_plot_beads_high
bp_all = [bp1; bp10; bp12; bp15; bp2; bp25; bp3; bp4; bp5; bp6; bp7; bp8;
    bp9; bp1_2; bp1_5; bp2_5];

meanbp_all = mean(bp_all);
meanbp1_2 = mean(bp1_2); meanbp2 = mean(bp2); meanbp25 = mean(bp25);
adjbp1_2 = meanbp_all - meanbp1_2;
adjbp2 = meanbp_all - meanbp2;
adjbp25 = meanbp_all - meanbp25;

close all
%% Load lists of particles with every cell having at least 3 balance
positions

load('ListOfParticles_HL60_radius_corrected.mat')

%% control

% initialize arrays
bp1_2_C = []; %zeros(length(ListOfParticles),1);
bp2_C = []; %zeros(length(ListOfParticles),1);
bp25_C = []; %zeros(length(ListOfParticles),1);
```

```
% add to arrays
for i = 1:length(ListOfParticles_control )
    if sum(ListOfParticles_control (i).balance_points(:,1)) > 28
        for j = 1:3 % because there are 3 frequencies, disregarding later
balance positions
            if ListOfParticles_control (i).balance_points(j,1) == 1.2;
                bp1_2_C = [bp1_2_C;
ListOfParticles_control(i).balance_points(j,2)];
            elseif ListOfParticles_control (i).balance_points(j,1) == 2;
                bp2_C = [bp2_C;
ListOfParticles_control(i).balance_points(j,2)];
            elseif ListOfParticles_control (i).balance_points(j,1) == 25;
                bp25_C = [bp25_C;
ListOfParticles_control(i).balance_points(j,2)];
            end
        end
    end
end

bp1_2_C = bp1_2_C + adjbp1_2;
bp2_C = bp2_C + adjbp2;
bp25_C = bp25_C + adjbp25;

%% 2um CytoD


% initialize arrays
bp1_2_2 = []; %zeros(length(ListOfParticles),1);
bp2_2 = []; %zeros(length(ListOfParticles),1);
bp25_2 = []; %zeros(length(ListOfParticles),1);

% add to arrays
    for i = 1:length(ListOfParticles_CytoD2uM)
        if sum(ListOfParticles_CytoD2uM(i).balance_points(:,1)) > 28
            for j = 1:3 % because there are 3 frequencies, disregarding later
balance positions
                if ListOfParticles_CytoD2uM(i).balance_points(j,1) == 1.2;
                    bp1_2_2 = [bp1_2_2;
ListOfParticles_CytoD2uM(i).balance_points(j,2)];
                elseif ListOfParticles_CytoD2uM(i).balance_points(j,1) == 2;
                    bp2_2 = [bp2_2;
ListOfParticles_CytoD2uM(i).balance_points(j,2)];
                elseif ListOfParticles_CytoD2uM(i).balance_points(j,1) == 25;
                    bp25_2 = [bp25_2;
ListOfParticles_CytoD2uM(i).balance_points(j,2)];
                end
            end
        end
    end

bp1_2_2 = bp1_2_2 + adjbp1_2;
bp2_2 = bp2_2 + adjbp2;
bp25_2 = bp25_2 + adjbp25;

%% 10um CytoD

% initialize arrays
```

```
bp1_2_10 = []; %zeros(length(ListOfParticles),1);
bp2_10 = []; %zeros(length(ListOfParticles),1);
bp25_10 = []; %zeros(length(ListOfParticles),1);

% add to arrays
    for i = 1:length(ListOfParticles_CytoD10uM)
        if sum(ListOfParticles_CytoD10uM(i).balance_points(:,1)) > 28
            for j = 1:3 % because there are 3 frequencies, disregarding later
balance positions
                if ListOfParticles_CytoD10uM(i).balance_points(j,1) == 1.2;
                    bp1_2_10 = [bp1_2_10;
ListOfParticles_CytoD10uM(i).balance_points(j,2)];
                elseif ListOfParticles_CytoD10uM(i).balance_points(j,1) == 2;
                    bp2_10 = [bp2_10;
ListOfParticles_CytoD10uM(i).balance_points(j,2)];
                elseif ListOfParticles_CytoD10uM(i).balance_points(j,1) ==
25;
                    bp25_10 = [bp25_10;
ListOfParticles_CytoD10uM(i).balance_points(j,2)];
                end
            end
        end
    end

bp1_2_10 = bp1_2_10 + adjbp1_2;
bp2_10 = bp2_10 + adjbp2;
bp25_10 = bp25_10 + adjbp25;

%% P values

[h25_c10,p25_c10] = ttest(bp25_C(1:54), bp25_10(1:54));
[h25_c2,p25_c2] = ttest(bp25_C(1:54), bp25_2(1:54));
[h25_210,p25_210] = ttest(bp25_2(1:54), bp25_10(1:54));

[h251_2_c10,p251_2_c10] = ttest([bp25_C(1:54), bp1_2_C(1:54)],
[bp25_10(1:54), bp1_2_10(1:54)]);
[h251_2_c2,p251_2_c2] = ttest([bp25_C(1:54), bp1_2_C(1:54)], [bp25_2(1:54),
bp1_2_2(1:54)]);
[h251_2_210,p251_2_210] = ttest([bp25_2(1:54), bp1_2_2(1:54)],
[bp25_10(1:54), bp1_2_10(1:54)]);

[h251_22_c10,p251_22_c10] = ttest([bp25_C(1:54), bp1_2_C(1:54), bp2_C(1:54)],
[bp25_10(1:54), bp1_2_10(1:54), bp2_10(1:54)]);
[h251_22_c2,p251_22_c2] = ttest([bp25_C(1:54), bp1_2_C(1:54), bp2_C(1:54)],
[bp25_2(1:54), bp1_2_2(1:54), bp2_2(1:54)]);
[h251_22_210,p251_22_210] = ttest([bp25_2(1:54), bp1_2_2(1:54), bp2_2(1:54)],
[bp25_10(1:54), bp1_2_10(1:54), bp2_10(1:54)]);



%% Convert balance positions and radii to CM
load('CM_Radius_2BP.mat')
ReCM = CM; % symantics
bp = bp - 1.8e-5; % simulation range adjustment for actuals (ballpark
estimate)
for k = 1:length(bp1_2_C)
```

```
    [~,j] = min(abs(radius - ListOfParticles_control(k).Radius/1e6));
%interpolate radius
    [~,i] = min(abs(bp1_2_C(k)/1e6 - bp(:,j))); % interpolate balance
position
    ListOfParticles_control(k).ReCM1_2 = ReCM(i); % assign Re[CM]
    [~,i] = min(abs(bp2_C(k)/1e6 - bp(:,j))); % interpolate balance position
    ListOfParticles_control(k).ReCM2 = ReCM(i); % assign Re[CM]
    [~,i] = min(abs(bp25_C(k)/1e6 - bp(:,j))); % interpolate balance position
    ListOfParticles_control(k).ReCM25 = ReCM(i); % assign Re[CM]
end
for k = 1:length(bp1_2_2)
    [~,j] = min(abs(radius - ListOfParticles_CytoD2uM(k).Radius/1e6));
%interpolate radius
    [~,i] = min(abs(bp1_2_2(k)/1e6 - bp(:,j))); % interpolate balance
position
    ListOfParticles_CytoD2uM(k).ReCM1_2 = ReCM(i); % assign Re[CM]
    [~,i] = min(abs(bp2_2(k)/1e6 - bp(:,j))); % interpolate balance position
    ListOfParticles_CytoD2uM(k).ReCM2 = ReCM(i); % assign Re[CM]
    [~,i] = min(abs(bp25_2(k)/1e6 - bp(:,j))); % interpolate balance position
    ListOfParticles_CytoD2uM(k).ReCM25 = ReCM(i); % assign Re[CM]
end
for k = 1:length(bp1_2_10)
    [~,j] = min(abs(radius - ListOfParticles_CytoD10uM(k).Radius/1e6));
%interpolate radius
    [~,i] = min(abs(bp1_2_10(k)/1e6 - bp(:,j))); % interpolate balance
position
    ListOfParticles_CytoD10uM(k).ReCM1_2 = ReCM(i); % assign Re[CM]
    [~,i] = min(abs(bp2_10(k)/1e6 - bp(:,j))); % interpolate balance position
    ListOfParticles_CytoD10uM(k).ReCM2 = ReCM(i); % assign Re[CM]
    [~,i] = min(abs(bp25_10(k)/1e6 - bp(:,j))); % interpolate balance
position
    ListOfParticles_CytoD10uM(k).ReCM25 = ReCM(i); % assign Re[CM]
end

ReCM1_2_C = [ListOfParticles_control.ReCM1_2]';
ReCM2_C = [ListOfParticles_control.ReCM2]';
ReCM25_C = [ListOfParticles_control.ReCM25]';

ReCM1_2_2 = [ListOfParticles_CytoD2uM.ReCM1_2]';
ReCM2_2 = [ListOfParticles_CytoD2uM.ReCM2]';
ReCM25_2 = [ListOfParticles_CytoD2uM.ReCM25]';

ReCM1_2_10 = [ListOfParticles_CytoD10uM.ReCM1_2]';
ReCM2_10 = [ListOfParticles_CytoD10uM.ReCM2]';
ReCM25_10 = [ListOfParticles_CytoD10uM.ReCM25]';
```

Discrimination Accuracy Code

```
function [acc_0, acc_1, acc_2, acc_3] = acc_ReCM3(ReCM)
% acc_ReCM gets the accuracy in discriminating observed ReCM's of different
% cell types
X0 = ReCM(:,1); % just bias
X1 = ReCM(:,[1 4]); % add 25 MHz ReCMs
X2 = ReCM(:,[1 2 4]); % add 1.2 MHz ReCMs
```

```
X3 = ReCM(:,[1 2 3 4]); % add 2 MHz ReCMs
Y = ReCM(:,5); % cell type (0, 1, or 2)

w = [-logspace(2,-3,50)'; logspace(-3,2,50)']; % number of possibilities for
each dimension of weight vector
% w = linspace(-1,1,100)';
% W = [wparam wparam wparam wparam]';
% w0 = W(1,:); w1 = W(1:2,:); w2 = W(1:3,:); w3 = W;

L0 = zeros(length(w),1);
L1 = zeros(length(w),length(w));
L2 = zeros(length(w),length(w),length(w));
L3 = zeros(length(w),length(w),length(w),length(w));
for i = 1:length(w);
    L0(i) = sum(Y ~= min(max(round(X0*w(i)),0),2));
    for j = 1:length(w)
        L1(i,j) = sum(Y ~= min(max(round(X1*[w(i); w(j)]),0),2));
        for k = 1:length(w)
            L2(i,j,k) = sum(Y ~= min(max(round(X2*[w(i); w(j); w(k)]),0),2));
            for l = 1:length(w)
                L3(i,j,k,l) = sum(Y ~= min(max(round(X3*[w(i); w(j); w(k);
w(l)]),0),2));
            end
        end
    end
end

% get accuracies

% 0 frequencies
[L0min,idx0] = min(L0);
w0 = w(idx0);
acc_0 = (length(X0) - L0min)/length(X0);

% 1 frequency
[L1min,idx1] = min(L1(:));
% [d0, d1] = find(L1 == L1min);
% w1 = [w(d0); w(d1)];
acc_1 = (length(X1) - L1min)/length(X1);

% 2 frequencies
[L2min,idx2] = min(L2(:));
% [d0, d1, d2] = find(L2 == L2min);
% w2 = [w(d0(1)); w(d1(1)); w(d2(1))];
acc_2 = (length(X2) - L2min)/length(X2);

% 3 frequencies
[L3min,idx3] = min(L3(:));
% [d0, d1, d2, d3] = find(L3 == L3min);
% w3 = [w(d0); w(d1); w(d2); w(d3)];
acc_3 = (length(X3) - L3min)/length(X3);

%w4 = 5; %dummy to not break out
end
```

Bibliography

1. Sackmann EK, Fulton AL, Beebe DJ. The present and future role of microfluidics in biomedical research. Nature. 2014;507(7491):181.
2. Zare RN, Kim S. Microfluidic platforms for single-cell analysis. Annual review of biomedical engineering. 2010;12:187-201.
3. Minor LK. Label-free cell-based functional assays. Combinatorial chemistry & high throughput screening. 2008;11(7):573-80.
4. Fang Y. Label-free cell-based assays with optical biosensors in drug discovery. Assay and drug development technologies. 2006;4(5):583-95.
5. Petersson F, Åberg L, Swärd-Nilsson A-M, Laurell T. Free flow acoustophoresis: microfluidic-based mode of particle and cell separation. Analytical chemistry. 2007;79(14):5117-23.
6. Cross SE, Jin Y-S, Rao J, Gimzewski JK. Nanomechanical analysis of cells from cancer patients. Nature nanotechnology. 2007;2(12):780.
7. Hur SC, Henderson-MacLennan NK, McCabe ER, Di Carlo D. Deformability-based cell classification and enrichment using inertial microfluidics. Lab on a Chip. 2011;11(5):912-20.
8. Morgan H, Sun T, Holmes D, Gawad S, Green NG. Single cell dielectric spectroscopy. Journal of Physics D: Applied Physics. 2006;40(1):61.
9. Voldman J. Electrical forces for microscale cell manipulation. Annu Rev Biomed Eng. 2006;8:425-54.
10. Mansor MA, Ahmad MR. Single cell electrical characterization techniques. International journal of molecular sciences. 2015;16(6):12686-712.
11. Yang J, Huang Y, Wang X, Wang X-B, Becker FF, Gascoyne PR. Dielectric properties of human leukocyte subpopulations determined by electrorotation as a cell separation criterion. Biophysical journal. 1999;76(6):3307-14.
12. Eppmann P, Gimsa J, Prüger B, Donath E. Dynamic light scattering from oriented, rotating particles: a theoretical study and comparison to electrorotation data. Journal de Physique III. 1996;6(3):421-32.
13. Hölzel R. Electrorotation of single yeast cells at frequencies between 100 Hz and 1.6 GHz. Biophysical journal. 1997;73(2):1103-9.
14. De Gasperis G, Wang X, Yang J, Becker FF, Gascoyne PR. Automated electrorotation: dielectric characterization of living cells by real-time motion estimation. Measurement Science and Technology. 1998;9(3):518.
15. Cristofanilli M, De Gasperis G, Zhang L, Hung M-C, Gascoyne PR, Hortobagyi GN. Automated electrorotation to reveal dielectric variations related to HER-2/neu overexpression in MCF-7 sublines. Clinical Cancer Research. 2002;8(2):615-9.
16. Sun T, Morgan H. Single-cell microfluidic impedance cytometry: a review. Microfluidics and Nanofluidics. 2010;8(4):423-43.
17. Holmes D, Pettigrew D, Reccius CH, Gwyer JD, van Berkel C, Holloway J, et al. Leukocyte analysis and differentiation using high speed microfluidic single cell impedance cytometry. Lab on a Chip. 2009;9(20):2881-9.
18. Chen J, Xue C, Zhao Y, Chen D, Wu M-H, Wang J. Microfluidic impedance flow cytometry enabling high-throughput single-cell electrical property characterization. International journal of molecular sciences. 2015;16(5):9804-30.
19. Spencer D, Elliott G, Morgan H. A sheath-less combined optical and impedance micro-cytometer. Lab on a Chip. 2014;14(16):3064-73.

20. Pethig R. Dielectrophoresis: Status of the theory, technology, and applications. Biomicrofluidics. 2010;4(2):022811.
21. Huang Y, Wang X-B, Becker FF, Gascoyne P. Introducing dielectrophoresis as a new force field for field-flow fractionation. Biophysical journal. 1997;73(2):1118-29.
22. Davis JM, Giddings JC. Feasibility study of dielectrical field-flow fractionation. Separation Science and Technology. 1986;21(9):969-89.
23. Su H-W, Prieto JL, Voldman J. Rapid dielectrophoretic characterization of single cells using the dielectrophoretic spring. Lab on a Chip. 2013;13(20):4109-17.
24. Morgan H, Green NG. AC electrokinetics: Research Studies Press; 2003.
25. Markx GH, Davey CL. The dielectric properties of biological cells at radiofrequencies: applications in biotechnology. Enzyme and Microbial Technology. 1999;25(3-5):161-71.
26. Davey CL, Kell DB. The low-frequency dielectric properties of biological cells. Bioelectrochemistry of cells and tissues: Springer; 1995. p. 159-207.
27. Gimsa J, Müller T, Schnelle T, Fuhr G. Dielectric spectroscopy of single human erythrocytes at physiological ionic strength: dispersion of the cytoplasm. Biophysical journal. 1996;71(1):495-506.
28. Cevc G. Membrane electrostatics. Biochimica et Biophysica Acta (BBA)-Reviews on Biomembranes. 1990;1031(3):311-82.
29. Andersen OS, Koeppe RE. Bilayer thickness and membrane protein function: an energetic perspective. Annu Rev Biophys Biomol Struct. 2007;36:107-30.
30. Prieto JL, Su HW, Hou HW, Vera MP, Levy BD, Baron RM, Han J, Voldman J. Monitoring sepsis using electrical cell profiling. Lab on a Chip. 2016;16(22):4333-40.