

BME Measurement and Analysis Laboratory

Image Capture and Analysis:

Fluorescence Microscopy

Ashley Mathew, Jahnavi Shah, Kyle Shyy, Muhammad Zahid, and Matthew Dailey

Spring 2025

Introduction

Microscopy techniques are indispensable to biomedical engineering, merging visualization and quantitative analysis of biological structures and processes into one. This lab focuses specifically on fluorescence microscopy, a tool that enables researchers to examine cell morphology, molecular interactions, and dynamic events at high spatial resolution. The two key applications employed for this lab are live-dead cell viability assessment and Fluorescent Recovery after Photobleaching (FRAP) analysis.

Fluorescent microscopy utilizes fluorophores—molecules that absorb light at a wavelength and emit it at a lower wavelength—providing high-quality contrast between different cellular components. This is especially prominent in live-dead cell assay—a common technique to differentiate between viable vs non-viable cells based on membrane integrity and metabolic activity, using different fluorescent stains. This can provide valuable information on cell proliferation, toxicity screening, and therapeutic efficacy.

On the other hand, FRAP is a method to assess molecular mobility within cells. By irreversibly photobleaching a defined region of interest (ROI) and tracking fluorescent recovery over time, researchers can extract information like diffusion coefficients and evaluate intracellular viscosity. These parameters provide information on protein dynamics, membrane fluidity, and biomaterial properties, which are essential for understanding cellular behavior and developing biomedical applications.

This lab explores the fundamentals of fluorescence imaging, including microscope operation, image acquisition, and digital image processing techniques. This would provide hands-on experience with quantitative data and how fluorescence-based methods contribute to biomedical research.

Methods

Experiment 1:

Using ImageJ, the gray value of ten pixels in 3 separate images was obtained at the same coordinates. These values were collected in the background as well as in the cell. Using these grey values, calculations were made for the SNR of each individual pixel as well as the SNR of all ten pixels selected together. The SNR was calculated using the equation, $\text{SNR} = \log(S/N)$ where 'S' is the value of pixels in the cell, and 'N' is the value of pixels in the background.

Experiment 2:

Using a microscope, acquire images of a prepared slide with cells. Use the eyepiece to see the slide and focus the image using the knob and adjust the brightness as needed using the dial. To change between different filters, use the knob on the right side of the sample holder with the labeled excitation wavelengths. Blue(DAPI) is about 405nm, red is about 594 nm, and green is about 488 nm. Make sure to adjust the exposure time after switching through different filters as needed. Keep the exposure time between 10-200ms. After all group members are satisfied with the image on the computer screen, save the image (make sure to title the correct filter to know which color to apply in ImageJ) to a flash drive to be accessed later. Lastly, apply the pseudo coloring in ImageJ by going to Image>Lookup Tables>(then choose the corresponding color the desired filter). Save the new images from ImageJ.

Experiment 3:

Fluorescence microscopy was used to capture images of live (green) and dead (red) stained cells. The microscope stage was adjusted for a flat orientation, and focus was optimized

using coarse and fine adjustment knobs. Different excitation wavelengths were used, and exposure time was set between 10-200 ms to maximize contrast while preventing saturation. Images were saved for further processing.

Live and dead cell images were opened in ImageJ, and thresholding was adjusted to highlight the nuclei while minimizing background noise. The images were converted to binary, cleaned using erosion, dilation, and watershed segmentation, and saved for further analysis. The "Analyze Particles" function was used to count segmented cells, and the results were recorded. A MATLAB script automated cell counting by loading images with `uigetfile()`, converting them to grayscale using `rgb2gray()`, and applying adaptive thresholding via `imbinarize()`. Small noise was removed using `bwareaopen()`, and `regionprops()` identified and counted cell regions. The processed images were displayed with overlays, and MATLAB results were compared with ImageJ counts. Cell counts from ImageJ and MATLAB were recorded and analyzed for discrepancies. The average cell size was measured in pixels, and images at different processing stages were saved. Differences due to segmentation techniques and thresholding methods were discussed to ensure accurate live/dead cell quantification

Experiment 4:

To assess the molecular dynamics using FRAP, data was collected and analyzed to determine the diffusion coefficient and viscosity of protein-dense droplets. The pre-collected FRAP data was retrieved from Canvas, with pre-bleach fluorescence set to 1 and minimum fluorescence post-bleaching set to 0 to normalize the data. The fluorescence recovery curve was fitted to the exponential equation provided $f(t) = A(1 - e^{-t/\tau^2})$ (where A is plateau fluorescence intensity and τ is characteristic recovery time) using MATLAB. The diffusion coefficient D was calculated using $D = r^2/\tau$, where r represents the estimated radius of the bleached region based on the provided scale bar. Stokes Einstein equation

$\eta = (KE * T) / (6 * \pi * RH * D)$ (η is viscosity, KE is boltzmann constant, T is temperature in Kelvin, RH is hydrodynamic radius of the molecule, and D is the diffusion constant) was used to determine the viscosity of the protein-dense droplets.

Results

Experiment 1 (SNR):

10 ms	
Background	
Pixel	Value
(1455,743)	118
(1456,743)	131
(1457,743)	135
(1458,743)	135
(1459,743)	143
(1460,743)	138
(1461,743)	135
(1462,743)	123
(1463,743)	131
(1464,743)	128
All 10 of them	131.7

10 ms Cell	
Pixel	Value
(1203,797)	239
(1204,797)	222
(1205,797)	220
(1206,797)	265
(1207,797)	226
(1208,797)	243
(1209,797)	260
(1210,797)	234
(1211,797)	236
(1212,797)	233
All 10 of them	237.8

SNR
0.3065
0.2291
0.2121
0.2929
0.1988
0.2457
0.2846
0.2793
0.2556
0.2601
0.2566

Table 1.1 - 10 ms exposure time *Leftmost table is pixels selected in the background, middle table is pixels in the cell, rightmost table is the SNR corresponding to that pixel/value*

500 ms	
Background	
Pixel	Value
(1455,743)	1988
(1456,743)	2188
(1457,743)	2174
(1458,743)	2129
(1459,743)	2034
(1460,743)	2133
(1461,743)	2079
(1462,743)	2229
(1463,743)	1943
(1464,743)	2162
All 10 of them	2105.9

500 ms Cell	
Pixel	Value
(1203,797)	6915
(1204,797)	7557
(1205,797)	7629
(1206,797)	8135
(1207,797)	7565
(1208,797)	7885
(1209,797)	7931
(1210,797)	7573
(1211,797)	7769
(1212,797)	7464
All 10 of them	7642.3

SNR
0.5414
0.5383
0.5452
0.5822
0.5705
0.5678
0.5815
0.5312
0.6019
0.5381
0.5598

Table 1.2 - 500 ms exposure time *Leftmost table is pixels selected in the background, middle table is pixels in the cell, rightmost table is the SNR corresponding to that pixel/value*

5000 ms	
Background	
Pixel	Value
(1455,743)	17143

5000 ms Cell	
Pixel	Value
(1203,797)	59938

SNR
0.5436

(1456,743)	16856
(1457,743)	17260
(1458,743)	17105
(1459,743)	16844
(1460,743)	17327
(1461,743)	16877
(1462,743)	17114
(1463,743)	16674
(1464,743)	17744
All 10 of them	17094.4

(1204,797)	62476
(1205,797)	63684
(1206,797)	65473
(1207,797)	65535
(1208,797)	65535
(1209,797)	65535
(1210,797)	65508
(1211,797)	65535
(1212,797)	63259
All 10 of them	64247.8

0.5690
0.5670
0.5829
0.5900
0.5777
0.5892
0.5829
0.5944
0.5521
0.5750

Table 1.3 - 5000 ms exposure time *Leftmost table is pixels selected in the background,

middle table is pixels in the cell, rightmost table is the SNR corresponding to that pixel/value*

Experiment 2:

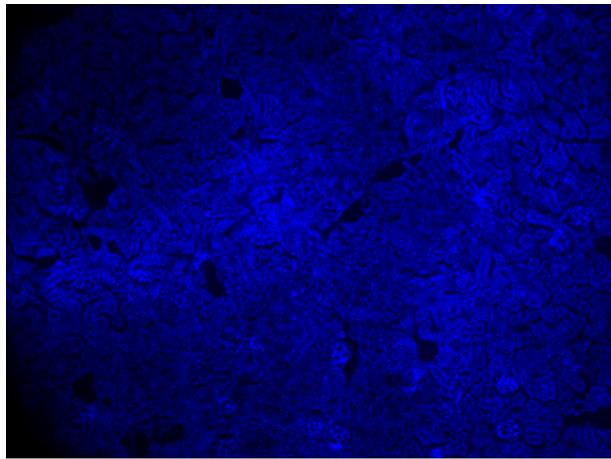


Figure 2.1: Dapi Filter

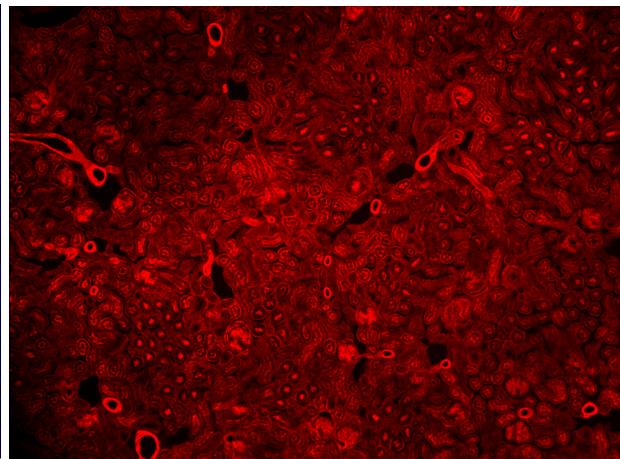


Figure 2.2: Red Filter

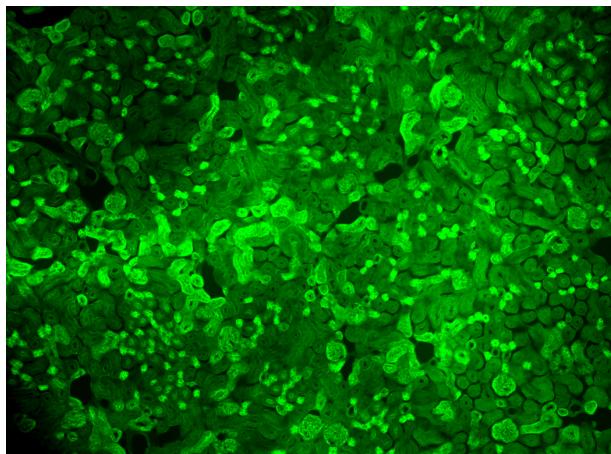


Figure 2.3: Green Filter

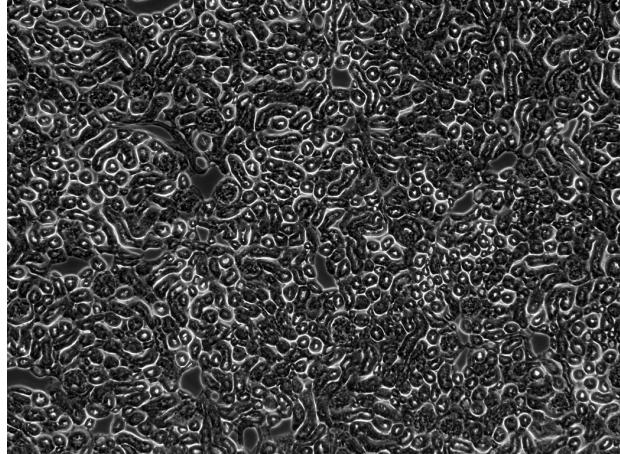


Figure 2.4: Phase Contrast

The slide was securely placed in the microscope holder but also adjusted to find different parts of the sample to acquire images. We looked through the eyepiece while adjusting some of the physical parameters, such as the focus and the brightness/intensity. Afterwards, the exposure time was adjusted to make sure the image had great contrast; however, increasing the exposure time too much would result in oversaturation. We wouldn't be able to see any cells in the picture due to the pixels maxing out in saturation, making the image completely white. The exposure times for the images were roughly between 40ms - 48ms. We then saved the images into a flash drive and titled them after the filter we were using to be able to know what color to

apply to what image when analyzing them in ImageJ. We took images under different filters to highlight different aspects of the cell.

Experiment 3:

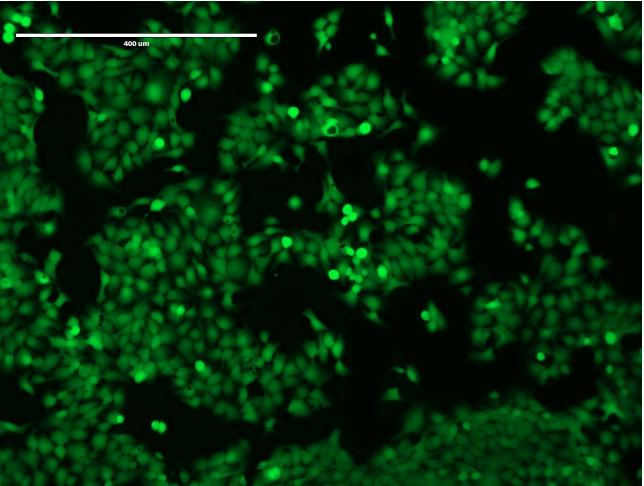
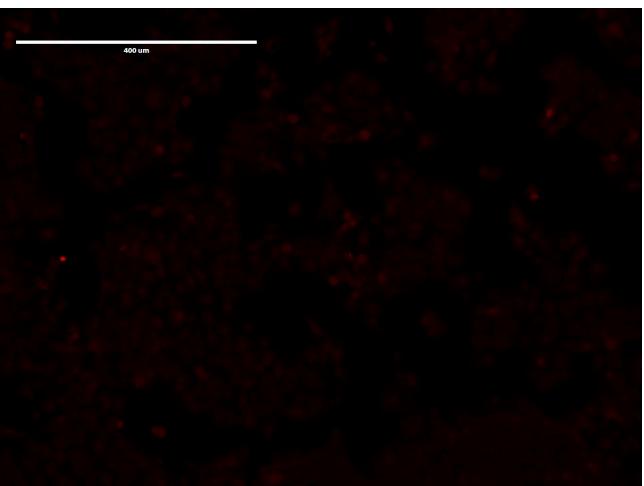
Cell Type	Images Taken in Lab
Live Cells	
Dead Cells	

Table 3.1
Images Taken in Lab of Live/Dead Cells

In order to complete the cell counts, we utilized ImageJ. Our process for analyzing the data was to upload the image, convert it into 8-bit, and then apply a threshold filter. Then, we applied the binary, erode, and watershed functions in order to create adequate separation between cells and properly analyze cell counts. Additionally, we only completed imaging analysis on the total and dead counts and subtracted the dead count from the total count in order to get the live count. The resultant modified images and cell counts are displayed below.

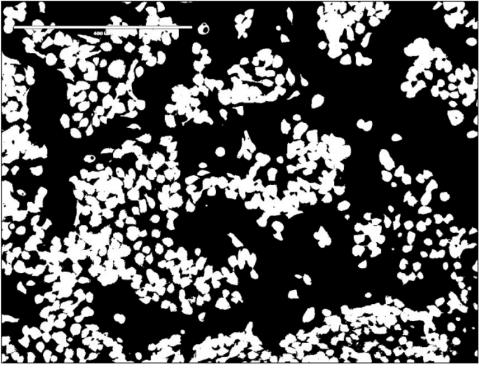
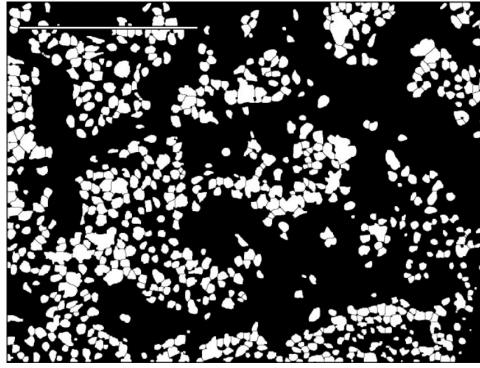
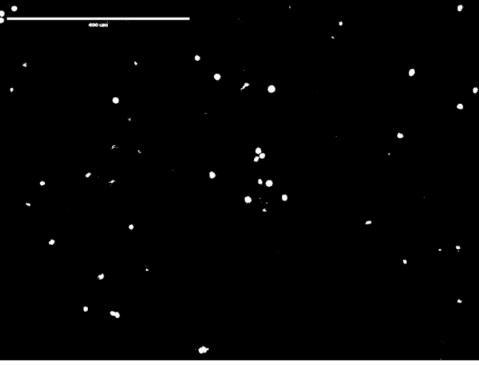
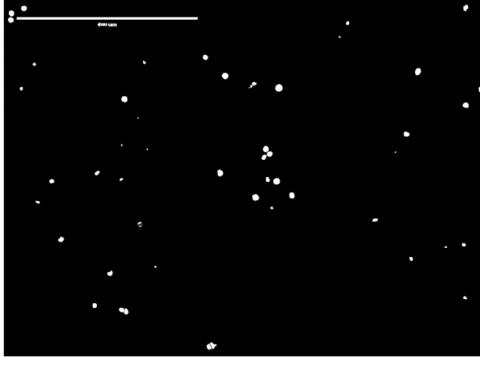
Cell Type	Threshold Image	Watershed Image
Live Cells		
Dead Cells		

Table 3.2

Image Comparison Between Threshold and Watershed Images of Live/Dead Cells

Utilizing our filtered images, we ended with the following counts for live, dead, and total cells according to the analysis done by ImageJ. Utilizing our filtered images, we ended up with the following counts for live, dead, and total cells according to the analysis done by ImageJ. We then compared these counts with those we calculated utilizing MATLAB. The resultant code and cell counts per group are shown below:

	ImageJ Cell Count	MATLAB Cell Count
Live Cells	717	263
Dead Cells	59	261

Table 3.3
ImageJ vs. MATLAB Cell Count on Live/Dead Cells

The code we compiled in order to obtain these results is shown below:

```

clc; clear; close all;

% Read images
[filename, pathname] = uigetfile({'*.jpg;*.png;*.tif'}, 'Select the live cell image');
live_img = imread(fullfile(pathname, filename));

[filename, pathname] = uigetfile({'*.jpg;*.png;*.tif'}, 'Select the dead cell image');
dead_img = imread(fullfile(pathname, filename));

% Convert to grayscale
live_gray = rgb2gray(live_img);
dead_gray = rgb2gray(dead_img);

% Apply thresholding
live_bw = imbinarize(live_gray, 'adaptive');
dead_bw = imbinarize(dead_gray, 'adaptive');

% Remove noise (small objects)
live_bw = bwareopen(live_bw, 20);
dead_bw = bwareopen(dead_bw, 20);

% Label connected components
live_stats = regionprops(live_bw, 'Area');
dead_stats = regionprops(dead_bw, 'Area');

% Count cells
num_live_cells = numel(live_stats);
num_dead_cells = numel(dead_stats);

disp(['Live Cells: ', num2str(num_live_cells)]);
disp(['Dead Cells: ', num2str(num_dead_cells)]);

% Display results
figure;
subplot(1,2,1);
imshow(live_bw);
title(['Live Cells: ', num2str(num_live_cells)]);

subplot(1,2,2);
imshow(dead_bw);
title(['Dead Cells: ', num2str(num_dead_cells)]);

```

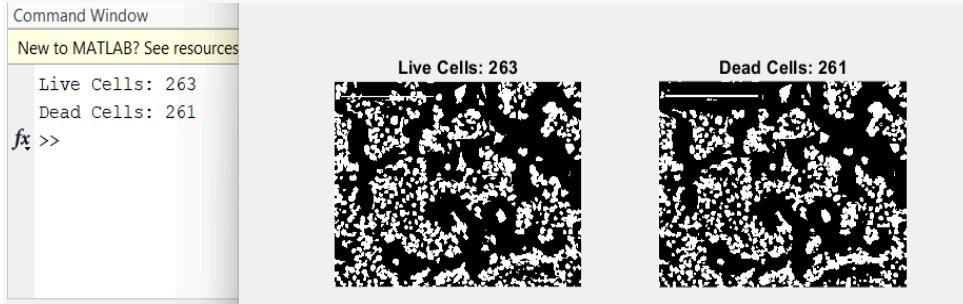


Figure 3.1

Image of Matlab code along with the results

Type of cell	Original	Matlab	Image J
Live Cells			
Dead Cells			

Table 3.4
Comparisons of Live/Dead Cells: Original, Matlab & Image J

Experiment 4 (FRAP)

```
% Load data from the Excel file
data = readtable('FRAP data (1).xlsx');

% Assuming the first column is time (t) and the second column is intensity (y)
t = data(:,1);
y = data(:,2);

% Define the exponential model function
exp_model = @(params, t) params(1) * (1 - exp(-t / params(2)));

% Initial guesses for A and tau
A0 = max(y);
tau0 = mean(t);
params0 = [A0, tau0];

% Perform nonlinear curve fitting
params_fit = lsqcurvefit(exp_model, params0, t, y);

% Extract the fitted parameters
A_fit = params_fit(1);
tau_fit = params_fit(2);

% Plot the original data and the fitted curve
figure;
scatter(t, y, 'bo'); % Original data
hold on;
t_fine = linspace(min(t), max(t), 100); % Smooth curve
y_fit = exp_model(params_fit, t_fine);
plot(t_fine, y_fit, 'r-', 'LineWidth', 2); % Fitted curve
hold off;
xlabel('Time');
ylabel('Fluorescence Intensity');
legend('Data', 'Exponential Fit');
title(['Exponential Fit: A = ', num2str(A_fit), ', \tau = ', num2str(tau_fit)]);
```

Figure 4.1: Code to fit the FRAP data to the exponential model

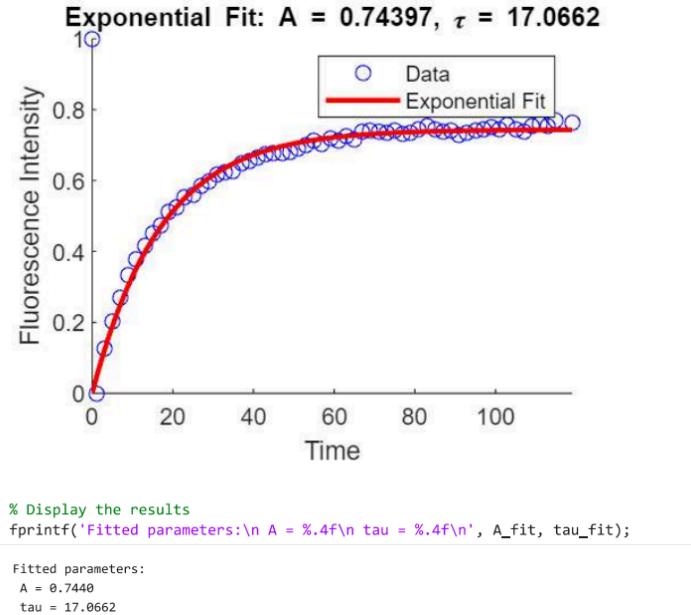


Figure 4.2: Shows the fitted exponential model with FRAP data with plateau value A and Tau (half-life) value.

FRAP is an imaging technique used in biomedical engineering (BME) to study the protein mobility and dynamics of molecular diffusion within living cells. Using FRAP, the region of interest is selectively bleached, and the recovery of fluorescence is monitored over time. This allows the researchers to quantify the rate of molecular movement and interaction with cellular compartments. This technique has proven important in understanding cellular processes like cellular signaling pathways, protein trafficking, and membrane fluidity. Within BME, its use extends to understanding the behavior of molecules within biomaterials in complex biological systems. Using the data acquired for FRAP from Canvas, MATLAB code was generated to fit the exponential data. The value of A (where the fluorescence plateaus) was found to be 0.74397. The Tau value displayed in Figure 4.2 is Tau 1/2 which was then used to find the Tau

value $\tau = \tau_1/2/\ln(2) = 24.62$. Assuming $r=1\text{ }\mu\text{m}$ (as suggested by the TA), the diffusion coefficient would be $D= 4.06*10^{-14}\text{ m}^2/\text{s}$. Using the diffusion coefficient value, the viscosity was calculated using $\eta = (KE * T)/(6 * \pi * RH * D)$ the Boltzmann constant of $1.38*10^{-23}\text{J/K}$, temperature of 288.15K , and hydrodynamic radius of 6.5nm . The viscosity was found to be $\eta=0.799\text{ Pa*s}$.

Discussion

Experiment 1:

Increasing the exposure time can help out in improving the SNR by creating a difference in the sample and background, which helps differentiate the two. However, there are also drawbacks to increasing the exposure time. The image may become too saturated with long exposure times, and it may become too bright to collect data. Also, increasing exposure time can lead to an increase in background noise, which may make the background seem brighter than it actually is and cause an error in the SNR. Depending on the life of the cells and the dye, the exposure time may also damage or bleach the cells with the dye. The signal in the background comes from noise in the camera and can also be due to scattered light from the microscope. The light may bounce off the slide and show up in the picture, which can contribute to the background signal. Also, the long exposure times give more time for noise and photons to illuminate the background, however, it also brightens the cells, which helps determine the SNR. In order to determine the SNR for the whole image, we could take more values, but this time for each cell and also more background cells. We could average these values and then use those values to compute a general SNR for the whole image. Another way may be to use a coding platform like MATLAB to select and average all the values in the cells and all the values in the background. Using these averages, we could obtain an SNR for the whole image.

Experiment 2:

Different filters served different purposes. We used DAPI (a blue filter) to highlight the nucleic acids of the cell, which will indicate the nucleus. The green filter is used to highlight structures and membranes, and the red filter is used to highlight actin filaments. The most successful filter was the green filter because we could see the separate cell membranes outlined clearly. Using the blue filter, we can see the nuclei inside the cells, but it is not as clear as the green filter distinguishing the cell membranes. With the red filter, we can only see one of the actin filaments clearly. The types of cells displayed in the image to prepare the slides are a type of eukaryotic cells. This is because the DAPI filter highlights the nuclei of the cells, which are only present in eukaryotic cells. More specifically, *Figure 1.0* shows that the nuclei of these cells are round, indicating they are most likely epithelial cells. There weren't many issues with acquiring clear images of the epithelial cells; however, finding a good spot to take an image was a little challenging because maybe there is a better section to capture an image.

Experiment 3:

The live and dead cell count from the Matlab code was 263 and 261, respectively. The cell count achieved from Image J was 717 and 59, respectively. The difference in values is significant due to variations in image processing techniques, thresholding methods, and object detection algorithms. MATLAB's counting function, especially with adaptive thresholding, may interpret intensity levels differently than ImageJ's thresholding methods (e.g., Otsu, Mean, or Minimum). Additionally, ImageJ includes built-in noise filtering, watershed separation for touching cells, and adjustable parameters that impact detection. MATLAB may remove small objects differently than ImageJ's "Analyze Particles" function, leading to discrepancies in counted cells. MATLAB could also pick up bright spots that may not be cells but still recognize them as one. Differences in preprocessing steps, edge detection, and segmentation approaches

between the two software tools can also contribute to result variations. This is why it is detrimental that clear pictures should be taken and proper lighting is achieved to minimize error.

Experiment 4:

In experiment 4, the FRAP data was analyzed to identify the diffusion coefficient and viscosity of the droplets composed of RGG-GFP-RGG protein. The calculated viscosity of the protein droplet was $\eta=0.799 \text{ Pa}^*\text{s}$ or 799 cPs. This viscosity is considered high relative to some commonly known material, however glycerol has 0.934 Pa*s at 25°C, Alkyd resin is 0.5-3 Pa*s, Castor oil 1-5 Pa*s, Gelatine is 1.2 Pa*s, Glycerine is 0.95 Pa*s, hand cream is 0.78 Pa*s, Milk Whey is 0.8-1.5 Pa*s, resin solution is 0.880 Pa*s. Hence, it can be assumed that the protein droplet has a viscosity close to hand cream, milk whey, or resin solution. Possible sources of error were considered, including variations in bleaching region size, background fluorescence interference, and curve fitting accuracy. To further validate the results, the obtained viscosity was compared to literature-reported viscosities of similar biomaterial.

Conclusion

Experiment 1 focuses on exposure time and signal-to-noise ratio (SNR). By using ImageJ to collect data on three of the same samples with different exposure times, we were able to collect grey values for ten pixels inside the cell and in the background. By making sure we got this data at the same location in each image, we limited the error of the results. After collecting the data, the SNR of each pixel was calculated for each of the three images. The SNR was also calculated for all ten pixels selected together. With increasing the exposure time, we saw that the SNR also increased, most noticeably from changing the exposure time from 10 ms to 500 ms. The SNR slightly increased from the 500 ms to 5000 ms image, signifying the importance of exposure time when collecting data on dyed samples.

Experiment 2 indicates different parts of the cell using different excitation wavelengths. The DAPI filter is used to see the nucleic acids, which would indicate where the nuclei is. The green filter outlines the cell membrane, and the red filter indicates the actin filaments. Images are acquired by finding a specific area on the slide to acquire an image. The brightness/intensity of the light and the focus are adjusted using their respective knob/dial. After the focus and the intensity are adjusted, the exposure time is adjusted to acquire an image with a reasonable contrast. Increasing the exposure time too much would result in an all white image, which we wouldn't be able to gather any results from. Our exposure time for each of the images was between 40-48ms.

Experiment 3 deals with the counting of Live/Dead cells using fluorescence. We successfully utilized fluorescence microscopy, ImageJ, and MATLAB to quantify live and dead cells, demonstrating the effectiveness of image processing techniques. While both methods provided reliable counts, differences in thresholding and segmentation led to slight variations in results. Automating the process with MATLAB improved efficiency and reproducibility, highlighting the importance of computational tools in biomedical image analysis.

Experiment 4 utilizes FRAP, which is usually used to study the movement of molecules within a cell. It provided critical insight into molecular diffusion and the viscosity of protein-dense droplets, demonstrating its applicability in quantifying biomolecular dynamics. By fitting fluorescence recovery data to an exponential model, we were able to extract information on parameters such as diffusion coefficient and viscosity to understand the physical properties of intracellular environments. The viscosity values were comparable to a number of known materials like hand cream, resin solution, and milk whey, reinforcing the relevance of this technique in creating biomaterials with known viscosities. Although sources of error such as measurement viability and background fluorescence, could have influenced the results, the overall findings support the effectiveness of FRAP in characterizing protein mobility and phase-separated condensates.