University of South Carolina

Scholar Commons

Theses and Dissertations

2016

Automated Image Analysis And Spatial Computational Modeling Of NF-kB In Cerebrovascular Endothelial Cells

Kasey Catalfomo
University of South Carolina

Follow this and additional works at: https://scholarcommons.sc.edu/etd

Part of the Biomedical Engineering and Bioengineering Commons

---

Recommended Citation

Catalfomo, K.(2016). *Automated Image Analysis And Spatial Computational Modeling Of NF-kB In Cerebrovascular Endothelial Cells.* (Master's thesis). Retrieved from https://scholarcommons.sc.edu/etd/3761

AUTOMATED IMAGE ANALYSIS AND SPATIAL COMPUTATIONAL MODELING OF
NF-κB IN CEREBROVASCULAR ENDOTHELIAL CELLS

by

Kasey Catalfomo

Bachelor of Science
University of South Carolina, 2015

_____

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Biomedical Engineering

College of Engineering and Computing

University of South Carolina

2016

Accepted by:

Edward Gatzke, Director of Thesis

Melissa Moss, Reader

Mark Uline, Reader

Michael Gower, Reader

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

ACKNOWLEDGEMENTS

I would like to thank all the people who contributed to the work described in this thesis. First, I would like to express my sincere gratitude to my advisor Dr. Ed Gatzke for the continuous support of my M.S. study and research. His patient, motivated, experienced guidance helped me throughout my research. He contributed predominantly to a rewarding graduate school experience by giving me intellectual freedom in my work, engaging me in new ideas, and expecting a high quality of work in all my endeavors. I'm incredibly grateful to have had such a great advisor and mentor.

Second, I would like to thank Dr. Melissa Moss for welcoming collaboration with her research efforts. Her expertise, insight, and encouragement motivated me to widen my research. Her example has inspired me to strive for extraordinary success. Through her encouragement and guidance, I have been able to complete the accelerated B.S./M.S. degree program on time.

My sincere thanks also goes to the rest of my readers: Dr. Mark Uline and Dr. Michael Gower, for their encouragement and interest in my work.

The results described in this thesis were accomplished due to the work of Lauren Wolf, Dr. Francisco Gonzalez, and those working in Dr. Melissa Moss's lab. The work presented in this thesis is result of close collaboration.

Lastly, I would like to thank my family and friends from the University of South Carolina for their continuing support throughout all my endeavors.

ABSTRACT

Cerebrovascular endothelial cells play a key part in the inflammatory response of the blood-brain barrier in pathological conditions such as Alzheimer's disease. Specifically, the NF-κB signaling pathway plays a central role. Better understanding of the factors in inflammatory disease progression can lead to more effective treatments for such devastating illnesses like Alzheimer's, asthma, arthritis, cancer, diabetes and many more inflammatory diseases. The proposed approach analyzes spatial NF-κB distribution contained in multispectral stacked micrograph images of cerebrovascular endothelial cells indexed based on dose of the activating protein and the length of activation. Image analysis code identifies the location of nuclear boundaries and quantifies NF-κB in relation to the closest nuclear boundary. This information is used to develop a mathematical model that describes the time and concentration dependence of NF-κB in response to the activating proteins. The proposed method allows for analysis and modeling of previously unexplored spatial behavior of NF-κB.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

A   Constant of the quantity of NF-κB stain intensity at a distance equal to zero

I   The intensity of the NF-κB which corresponds to the presence of NF-κB

k   The rate at which the NF-κB decreases is quantified by scalar constant

B   Constant serves as an offset to account for non-zero minimum of stain intensity within the set distance threshold

x   Distance from the nuclear wall

LIST OF ABBREVIATIONS

Aβ..Amyloid-Beta

AD.. Alzheimer's Disease

BBB.. Blood-Brain Barrier

DAPI ....4',6-Diamidino-2-Phenylindole

DMSO ....Dimethyl Sulfoxide

HBMVEC .... Human Brain Microvascular Endothelial Cells

IκB..Inhibitor of NF-κB

IKK .... IκB kinase

CEMSA.................................... Cerebrovascular Endothelial Micrograph Spatial Analyzer

NF-κB ....Nuclear Factor Kappa-Light-Chain-Enhancer of Activated B cells

TIFF ....Tagged Image File Format

TNF-α...Tumor Necrosis Factor Alpha

CHAPTER 1

INTRODUCTION

1.1 NF-κB IN ALZHEIMER'S AND INFLAMMATION

Alzheimer's disease (AD) is a global public health burden as it is a major cause of mortality.[1] Vascular dysfunction plays a significant role in the disease progression of AD. Impaired clearance of amyloid β-peptide (Aβ) and the accumulation of Aβ plaques on vessels of the blood-brain barrier (BBB) is a hallmark of AD pathology.[2] Because of this phenomena, decades of research has focused on Aβ as a therapeutic target for the treatment of AD.

Neuroinflammation occurs in Alzheimer's patients, but was previously thought to be a response to pathophysiological events.[3] However, recent studies show that the impaired clearance of Aβ can cause inflammation by potentially more than one mechanism, and could contribute largely to AD pathology.[2–4] The inflammatory reaction observed in AD is thought to be caused mostly by the over-activity of microglia, which serve as immune cells that clear Aβ plaques.[3] This activity leads to the release of inflammatory cytokines, including tumor necrosis factor α (TNF-α).[2,3] Inflammatory cytokines activate signaling pathways with nuclear factor-κB (NF-κB), a dimeric transcription factor in the inflammatory response.[5–7] In addition to inflammatory cytokines, Aβ aggregates have been shown to activate these same NF-κB pathways.[4] In unstimulated cells, NF-κB exists in the cytoplasm, bound to the inhibitor of NF-κB, IκB. When a cell is stimulated, IκB is phosphorylated by the IKK complex and then ubiquitinated and degraded. This leads to

NF-κB nuclear translocation where it binds DNA fragments and induces expression of target genes.[4,8] The presence of activated NF-κB leads to production of more IκB, therefore, this process is controlled by a negative feedback loop.[9]

Activation of the NF-κB signaling pathways proves to be a major factor in the cerebrovascular endothelial dysfunction associated with AD.[1–4] Unregulated and sustained activation of NF-κB plays a major role in sustained inflammation.[8] In addition to AD pathology, NF-κB has also been associated with other disease states including asthma, arthritis, cancer, diabetes and others.[3,7] NF-κB is also known to play a vital role in the development of innate and adaptive immunity. A therapeutic that modulates NF-κB signaling could prove to be effective at reducing the prolonged inflammation that influences AD.[8,10] Understanding the dynamics of the transcription factor NF-κB in response to stimulus can lead to the development of better therapeutics for Alzheimer's and other diseases.

1.2 NF-κB MODELS

Up to this point in time, various approaches have been taken to mathematically model the NF-κB signaling pathway.[6,9,11,12] Because TNF-α is a primary inflammatory cytokine, the TNF-α mediated signaling pathway has been thoroughly investigated.[9,13] Kinetic models to date focus primarily on ordinary differential equations to describe the chemical reactions.[6,9,11,12,14] Dynamic behavior of NF-κB signaling has been more recently explored using theoretical models. Some models have concluded that NF-κB exhibits an oscillatory response.[14–18] This proposed oscillatory response is thought to be a result of the negative feedback response combined with activation and nuclear translocation; however, this response has been difficult to reproduce experimentally.[14–18]

In modeling NF-κB kinetics, rate constants are dependent on the stimulus, species, tissue, and physiological state of the organism or cells, therefore, kinetic parameters are complicated to fully define.[5] This leaves much more work to be done considering pathological conditions for the development of therapeutics. Also, there are few quantitative mathematical models that describe the spatial distribution of NF-κB.[6,9,17] Nuclear translocation is a known and significant phenomena in NF-κB signaling, so the spatial relationship between the NF-κB complex and the nucleus and cytosol is necessary to study and quantify in order to develop thorough models.[5,18] Integrating spatial models from *in vitro* studies of NF-κB with kinetic models can provide additional information for the development of a mathematical model based on experimental data. By expressing the complex NF-κB signaling pathway as a mathematical model, preclinical development of therapeutics can be improved and potentially optimized through quantitative understanding of the signaling mechanisms.[19]

1.3 BIOIMAGE INFORMATICS

Bioimage informatics is a developing an area of research, as many wet-lab techniques for analyzing molecular data may be limited due to the lack of spatial information. The relationship between tissue/cellular structures and proteins cannot be captured when many wet-lab techniques require cell lysing. Imaging biological samples allows for additional understanding of proteins and the relationship between cell structure and function.[5,20,21] For example, immunofluorescence labeling of multiple proteins in a single sample can be used to study localization patterns of proteins. Bioimaging techniques allow for quantitative analysis that includes spatial data.[21–24]

Currently, the standard analysis of biomedical images is manual evaluation combined with a basic computational method using a software such as ImageJ.[4,22,24] However, manual evaluation becomes less desirable due to increasing data volume, which has been enabled by faster preparation methods and higher resolution images.[22,25] Fully automated modular image analysis techniques, using readily available software, can address this explosion of imaging data.[23–25] In previous applications, automated techniques have proven to be more efficient and equally accurate when compared to manual evaluation methods.[22,25,26]

While some computational methods have been used to analyze NF-κB, even the most recent software does not have a capabilities to study spatial distributions, and is not fully automated.[23] Integrating automated image analysis into modeling of NF-κB signaling responses will allow for thorough analysis of the transcription factor's spatial behavior.

1.4 DEVELOPING A BIOIMAGE ANALYSIS TOOL TO STUDY NF-KB

This work demonstrates a technique for the analysis of NF-κB in cerebrovascular endothelial cells using commercially available software, MATLAB™. MATLAB™ can efficiently store and read large sets of images and also contains toolboxes to allow for a full range of modeling capabilities. MATLAB™ also can be used in conjunction with other imaging software.

The proposed approach, termed Cerebrovascular Endothelial Micrograph Spatial Analyzer (**CEMSA**), analyzes information contained in multispectral stacked micrograph images of cerebrovascular endothelial cells. Images are acquired as a z-stack using confocal microscopy and complied into a single z projection. The images contain two channels: a DAPI stain layer and a fluorescently tagged NF-κB antibody layer. The images

are then indexed based on concentration of the activating protein and the time of treatment. MATLAB™ image analysis and optimization toolboxes are used to evaluate the transcription factor, quantified from the fluorescently tagged NF-κB antibody, in relation to the wall of the cell nucleus, identified by the DAPI stain. First, this image analysis method identifies the location of nuclear boundaries. Next, the method quantifies NF-κB in relation to the closest nuclear boundary. Finally, this quantitative information is used to develop a mathematical model that describes the time and concentration dependence of NF-κB cellular position in response to two activating proteins.

CHAPTER 2

METHODS

This chapter presents the method of experimental preparation and the development

of the automated **CEMSA** software for computational analysis of NF-κB in endothelial

cells. Three experimental cases (A,B,C) with varying treatments were gathered and used

in conjunction with **CEMSA** software in order to generate mathematical models of the

system. Case A images were gathered from a previous publication.[4] Case B and C images

were gathered from cell culture performed by Lauren Wolf in the Moss Lab at the

University of South Carolina.

2.1 EXPERIMENTAL TREATMENT OF CELLS

2.1.1 Cell culture and preparation of endothelial monolayers: Endothelial

monolayers were comprised of ACBRI 376 primary human brain microvascular

endothelial cells (HBMVECs) (Cell Systems, Kirkland, WA) were maintained in CSC

Complete Medium with CultureBoost and Bac Off® on surfaces coated with Attachment

Factor™ (Cell Systems, Kirkland, WA). For each experiment, HBMVECs within passages

6 through 9 were seeded onto Attachment Factor™ coated glass coverslips (Corning, Inc.)

in 6 well plates with CSC Complete Medium with CultureBoost and Bac Off® at a density

of 5×105 cells/mL. Cells were fed after 24 hours with daily growth medium, and

transitioned to Complete Serum Free Media with RocketFuel (Cell Systems, Kirkland,

WA) at 48 hours post seeding. Cells remained in serum free media for 24 hours until

treatment; this total time of 72 hours post seeding allowed cells to reach confluence (37ºC, 5% CO_2, 95% air) prior to treatment.

     2.1.2. Cell treatment, immunocytochemistry, and image acquisition: HBMVEC monolayers were treated with activating proteins. In Case B, cells were treated with Aβ aggregates in µM concentrations of: 2 µM, 1 µM, 0.5 µM, and 0.1 µM. Case C cells were treated with TNF-α (20 units/mL). For the 30 minute time interval concentration study, endothelial monolayers were also exposed to reduced strength TNF-α at 10% or 50% of full strength (i.e. 2 units/mL and 10 units/mL, respectively). Additionally in Case C, the impact of repeat exposure to TNF-α on NF-κB dispersion was investigated through a second treatment of HBMVECs initially exposed to TNF-α at 240 and 60 minutes with TNF-α and fixated 30 minutes following the second treatment. Untreated HBMVECs were used as the negative control.

     Post treatment, cells were quickly washed with Complete Serum Free Media with RocketFuel (Cell Systems, Kirkland, WA) and fixed to the coverslips with 4% paraformaldehyde for 10 minutes at 25ºC. Fixed cells were permeabilized (0.1% Triton X 100, 0.01 M glycine in PBS) and rinsed again in PBS. Fixed monolayers were subsequently blocked with 5% BSA in DPBS, followed by 5% normal donkey serum in 1% BSA/DPBS to prevent non specific binding. The cells were then incubated overnight at 4ºC with the primary antibody, rabbit polyclonal anti NF-κB p65 (1:500) (Santa Cruz Biotechnology, Dallas, TX) in DPBS. After rinsing the next day with 1% BSA/DPBS to remove the primary antibody, monolayers were again blocked with 5% normal donkey serum in 1% BSA/DPBS and incubated with goat anti rabbit Alexa Fluor®488 secondary antibody (1:1000) (Life Technologies, Carlsbad, CA) for 2 hours at 25ºC to detect bound primary

antibody. The prepared coverslips were subsequently rinsed 3X with 1% BSA, twice with DPBS, and were then mounted onto slides using Fluorshield with DAPI (Sigma Aldrich, St. Louis, MO) for visualization of nuclei.

Slides were visualized under a Zeiss Confocal Laser Scanning Microscope (Carl Zeiss) through a Plan Apochromat 63X/1.4 oil DIC immersion objective (Carl Zeiss). For each slide, 4 different fields containing a minimum of 5 nuclei per field were sampled, and 3 slice z stacks used to envisage the entire cell layer for each field. Each image was comprised of two channels, corresponding to the DAPI labeled nuclei and the Alexa Fluor®488 labeled NF-κB. ImageJ64 software (Schneider et al., 2012) was used to convert each z stack into a Z projection based on the maximum intensity within each channel to create the final images used for image analysis.

2.1.3 Case A: In order to analyze the NF-κB within the endothelial cells over time, monolayers endothelial cells are treated with TNF-α and fixed at three different time points: 10 minutes, 15 minutes, and 30 minutes.

2.1.4 Case B: In order to analyze the concentration dependency and time dependency of NF-κB activation within the endothelial cells in association with AD pathology, monolayers are treated with varying concentrations of Aβ aggregates: 2 µM, 1 µM, 0.5 µM, and 0.1 µM and fixed at three different time points: 30 minutes, 45 minutes, and 60 minutes. Untreated cells and cells treated with buffer equivalent are used as negative controls. TNF-α, a known activator of NF-κB, is used as a positive control.

2.1.5. Case C: In order to analyze the concentration dependency and time dependency of NF-κB activation within the endothelial cells, monolayers are treated with varying concentrations of TNF-α: a maximum concentration, half maximum concentration,

and 0.1 times the maximum concentration (20 units/mL, 10 units/mL and 2 units/mL, respectively) and fixed at 7 time points: 0 minutes, 10 minutes, 20 minutes, 30 minutes, 45 minutes, 60 minutes, 120 minutes, and 240 minutes. Untreated cells fixed at 30 minutes serve as the negative control. An additional experiment evaluates the response after a repeat exposure of TNF-α. Monolayers are initially treated 60 minutes and 240 minutes prior to a second exposure and then fixed 30 minutes after the second exposure. This second exposure is then compared to the single exposure results.

The final TIFF images contain two channels: one with the DAPI nuclear stain and another with the antibody stain, which are shown in Figure 2.1 A and B respectively.
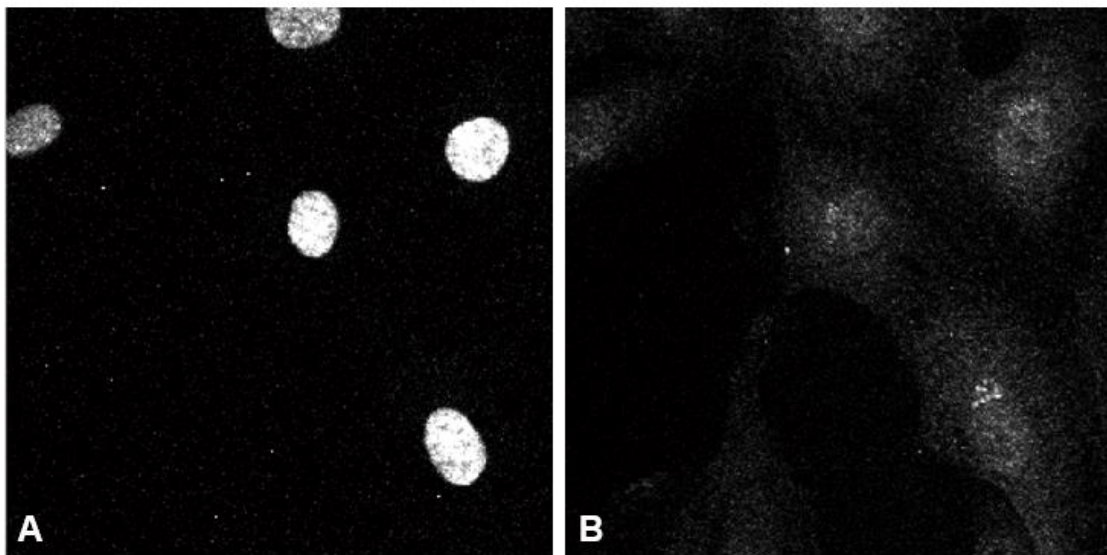
Figure 2.1. *Multispectral Staining of Endothelial Cells.* A) The DAPI stain shows the location of cell nuclei. B) The antibody stain shows NF-κB.

2.1.6. Image organization: To obtain data from the treated endothelial cells, the micrograph TIFFs are analyzed in MATLAB™. The number of TIFF images per sample, the number of different concentrations, and the number of time intervals are used for future sorting the images. Micrograph TIFF images are imported in order of time and

concentration. The imported images are compiled into a single collection of TIFF images. The program iterates through each time interval, each concentration, and each image. For each image iteration, the image channels are read from the original graphics file and two matrices are returned: one for the DAPI stain and the other for the antibody labeled NF-κB. The image matrix from the DAPI stain and antibody labeled NF-κB are then assigned to 2D arrays containing the values of intensity.

2.2 IMAGE PROCESSING USING MATLAB™

2.2.1 Identifying Nuclear Boundary: The DAPI stained channel is used to identify the location of the endothelial cell nuclei. First, the edges in the image are identified using the Sobel method, an edge detection algorithm used in computer vision. This method returns edges at the points where the intensity gradient is at a maximum. The Sobel method is used first to identify a threshold and then used with the adjusted threshold to define the edges of the nuclei. The result of this is shown in Figure 2.2 A. Due to the large amount of noise in the data, several steps are taken in order to reduce the effect of uncertainty when working to identify the nuclei within the image. After using the Sobel method to identify edges, the image is dilated and holes are filled in. Since MATLAB™ defines a hole as a set of pixels separated from the background of the image, filling in the holes differentiates the regions of interest within the image without including noise, as shown in Figure 2.2 B. Next, as seen in Figure 2.2 C, regions of interest that intersect with the border of the image are removed so only whole nuclei can be analyzed. In order to account for pixels added during previous morphological dilation, edges are eroded by a diamond structuring element the size of one pixel. The result can be seen in Figure 2.2 D. The specific MATLAB™ functions used are shown in Appendix A.

10
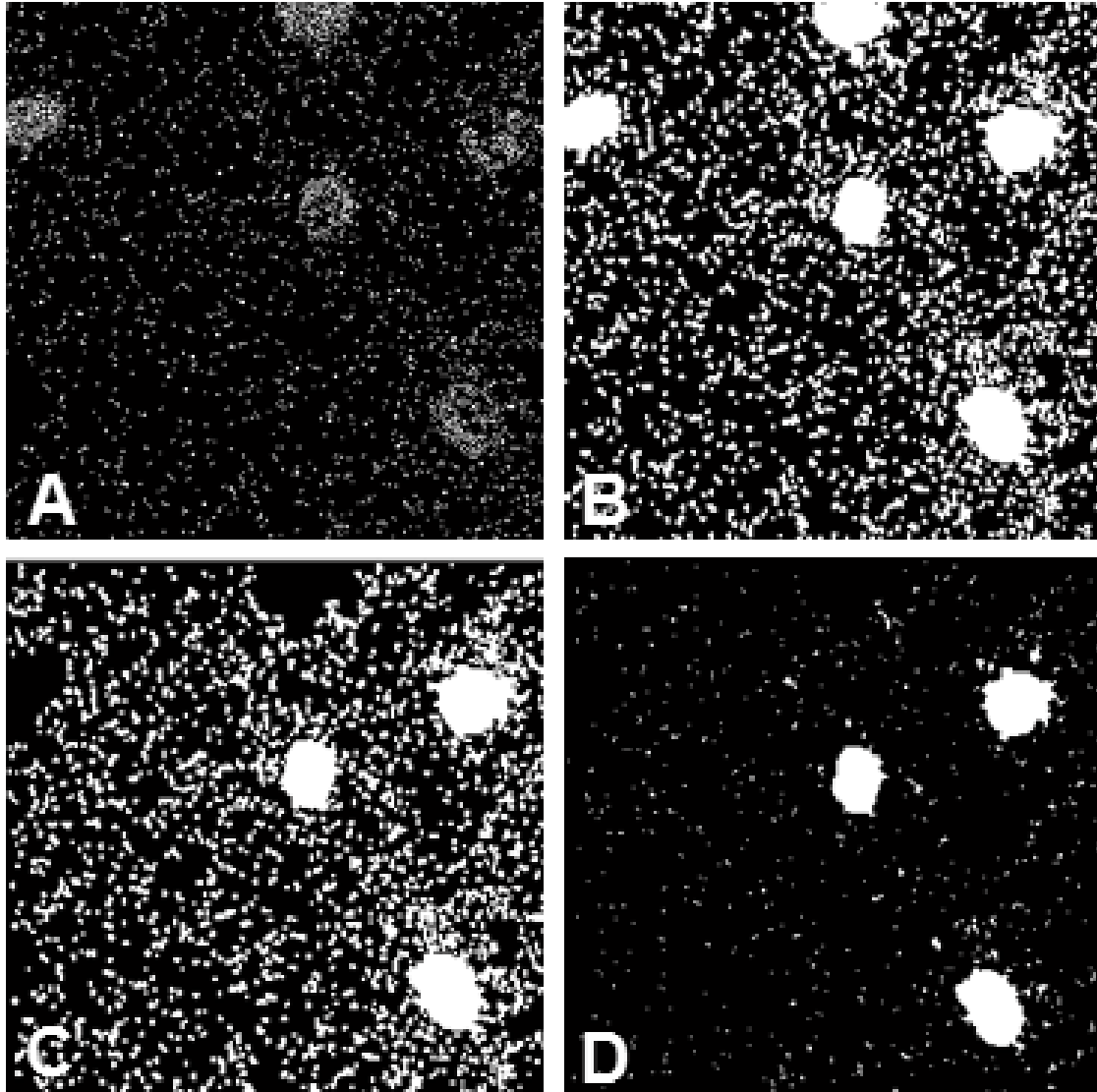
Figure 2.2. *Image Processing of Nuclei.* The DAPI stain is filtered to differentiate cell nuclei. A) The Sobel method is used first to identify a threshold and then used with the adjusted threshold to define the edges of the nuclei. B) The image is dilated and holes are filled in. C) Cells that intersect with the border of the image are removed. D) Edges are eroded by a diamond structuring element the size of one pixel.

While much of the noise is removed by the previous steps, a size exclusion must be

performed to completely differentiate the nuclei of the cells from the background noise.

First, the indices of the regions of interest, shown in white and containing both nuclei and

noise, are identified and labeled. For each labeled region of interest, the number of pixels

is determined and is considered to be the size of the region of interest. If the number of pixels is greater than a set threshold (determined initially by the user), the average intensity of the antibody labeled NF-κB within each region of interest, the size of the region, and the region label is determined. The mean size of all regions are calculated. A new minimum threshold is set at one standard deviation smaller than the mean element size, and a max threshold is set at four times the standard deviation of each element size. The thresholds are determined using this independent statistical algorithm so final regions of interest are whole cell nuclei, and any potential noise or dividing cells are removed from analysis. A final matrix of labeled regions is saved for future use. For visual assessment, returned regions of interested are identified by denoting the maximum DAPI stain intensity within each region of interest and shown in Figure 2.3. Using statistical methods to determine thresholds allows this program to be used with images containing varying size nuclei.

2.2.2 Analyzing NF-κB intensity Inside and Outside Nuclei: Once the elements that are nuclei of cells are labelled, the antibody labeled NF-κB channel can be analyzed. By labeling the nuclei, the location of the nucleus can be used to differentiate cells within the antibody labeled NF-κB channel. The perimeter of the nucleus is defined and can be overlaid on the antibody stain for visual purposes, as seen in Figure 2.4.

Using the perimeter, the antibody labeled NF-κB intensity inside the nuclei is compiled and averaged per sample. The antibody labeled NF-κB intensity outside the nuclei is also be compiled and averaged per sample. The stain intensity of the fluorescently labeled NF-κB is compared to a TNF-α control and correlates to the level of NF-κB activation. The data collected provides a platform for statistical analysis by to identifying cell count, nuclei location, and antibody labeled NF-κB intensity within the image.
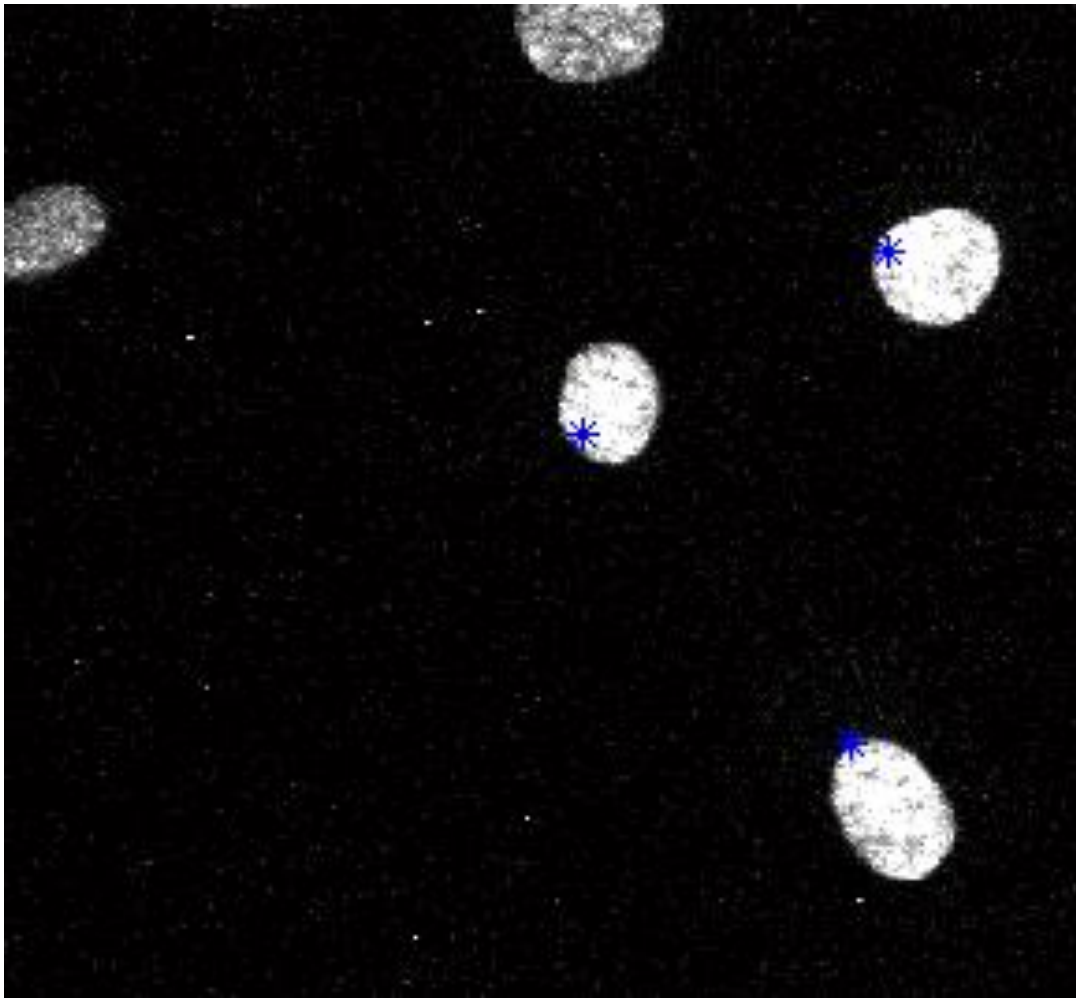
Figure 2.3. *Nuclei Differentiated.* The blue asterisk indicates the pixel with the maximum DAPI stain intensity within the differentiated nuclei.

2.2.3 Spatial Analysis: Because the cell edges are not stained, and NF-κB is known to exhibit nuclear translocation upon activation, quantifying the stain intensity as function of distance from the nuclear boundary can provide a new platform for analysis. To begin analyzing antibody labeled NF-κB intensity as a function of distance from the nuclear boundary, a range of distance (set by user as number of pixels) from the nuclear wall was set. Next, pixels were analyzed iteratively and data for each pixel was stored with the distance from the nearest cell wall and the corresponding antibody labeled NF-κB intensity. For each image set, the function of NF-κB versus the distance from the nuclear wall was

13

fit to an exponential model using optimization techniques. In order to account for varying

pixel counts, the average antibody labeled NF-κB intensity within distance intervals for all

images in a sample was then fit to an exponential model using optimization techniques.
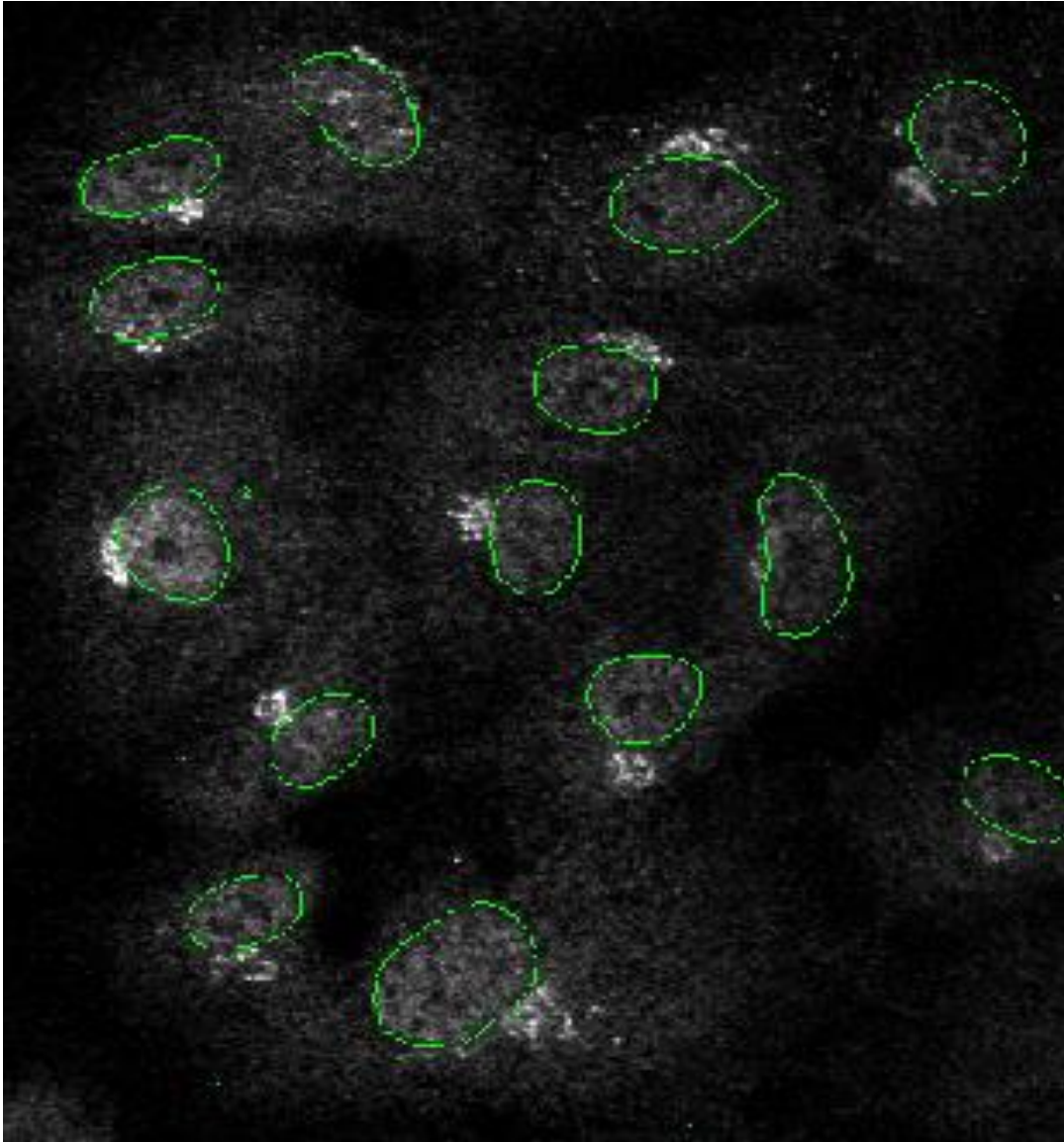
Figure 2.4. *Nuclear Border Overlay.* The border of nuclei can be overlaid on the antibody stained image to show NF-κB inside nucleus and outside nucleus. Areas of high NF-κB concentration close to the nuclear border are evident.

2.2.4 Parameter Concentration Dependency Analysis: It was hypothesized that the

concentration of TNF-α or Aβ aggregates affects the cellular localization of NF-κB,

therefore, a model with concentration dependent parameters can be fit to the experimental data. This model could potentially be used to predict future responses. The function of NF-κB versus the distance from the nuclear wall can be fit to a modified exponential model (Equation 1).

$$(1) \qquad I(x) = Ae^{-kx} + B$$

The modified exponential model contains three constants: A, k, and B. The intensity, *I*, is modeled as a function of distance from the nuclear wall is defined as *x*. *A* is the quantity of NF-κB stain intensity at the nuclear border. The rate at which the antibody labeled NF-κB intensity decreases is quantified by scalar constant *k*. The constant *B* serves as an offset to account for non-zero minimum of stain intensity within the set distance threshold. Congruent with the hypothesized concentration dependent response, the constant parameters are anticipated to vary as a function of the concentration of TNF-α or Aβ aggregates. As a result of the oscillatory response of NF-κB, it is expected that parameters will also vary as a function of time.

CHAPTER 3

RESULTS

3.1 APPLYING CEMSA TO CASE A

Case A consisted of endothelial cell monolayers treated with a single concentration of TNF-α. In order to analyze the NF-κB within the endothelial cells over time, micrographs are taken at three different time points: 10 minutes, 15 minutes, and 30 minutes.

These high resolution images yield a large amount of data points that are difficult to distinguish. Figure 3.1 shows a plot of data from two images taken from the 15 minute time point in Case A.
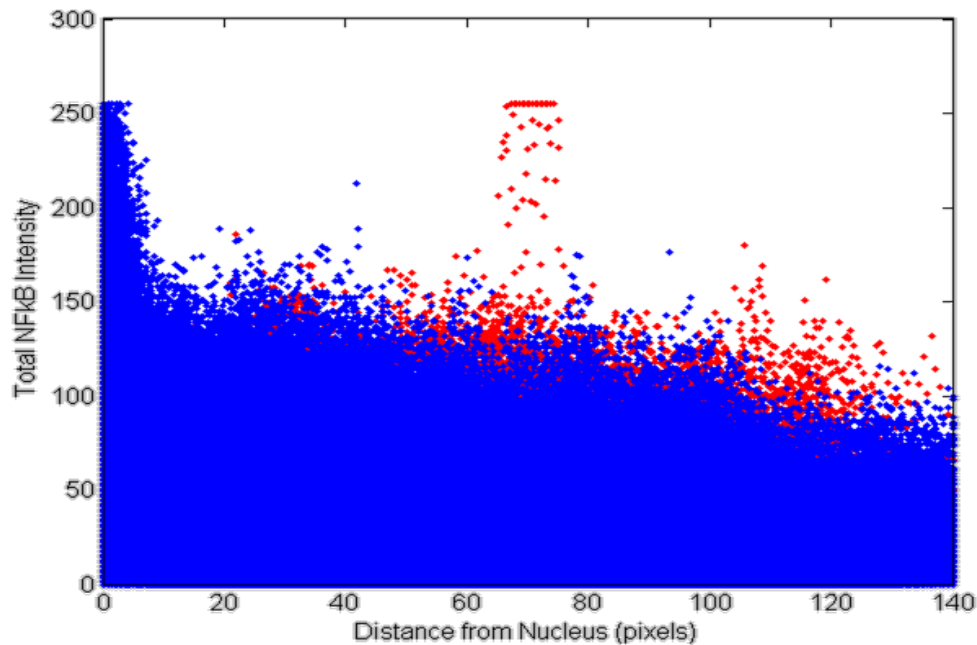
Figure 3.1. *Case A: Raw Data from 15 minute Image Set.* The NF-κB stain intensity is plotted versus the calculated distance from the nucleus. This set consisted of two images with 1.048×10^6 pixels in each image.

The average antibody labeled NF-κB intensity outside the nucleus and inside the nucleus for each image set is averaged per sample set for comparison to previous analyses, seen in Figure 3.2. While previous analysis reports nuclear optic density of NF-κB, the total average NF-κB intensity is shown to compare the inside an outside of the nucleus, since the optical density of NF-κB outside the nucleus cannot be calculated with the image sets that do not have a stained cellular membrane.[4]

Figure 3.2. *Case A: Average NF-κB Intensity Outside and Inside Nucleus.* The average intensity of the NF-κB stain outside the nucleus (black) and inside the nucleus (blue) is plotted at each time point with standard deviation.

After averaging data within distance interval sections to normalize and fitting the exponential model, trends in the model and parameters can be observed. Figure 3.3 shows the model on the left with the plotted parameters on the right. There is a clear trend with the total NF-κB at the border of the nucleus (parameter *A*) that decreases over time up to

30 minutes. This decrease of antibody labeled NF-κB intensity at the border is expected to correlate to an increase in NF-κB transport in the nucleus. This clear decrease in *A* from 15 minutes to 30 minutes is consistent with previous hypotheses that show NF-κB reaching a maximum activation at 45 minutes post treatment. Parameter *k* increases and plateaus, while there is no apparent trend in parameter *B*.
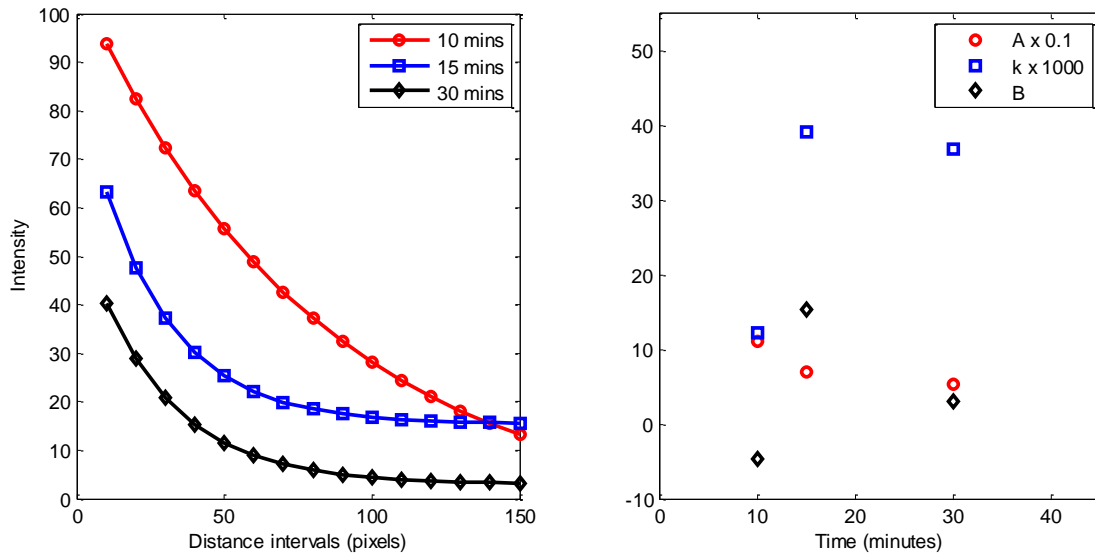
Figure 3.3. *Case A: Model fit of NF-κB as function of Distance from Nuclear Border.* The exponential model is fit for the TNF-α treatment over time (left). Parameters are shown as function of time (right).

3.2. APPLYING CEMSA TO CASE B

Case B consisted of monolayers treated with varying concentrations of Aβ aggregates: 2 μM, 1 μM, 0.5 μM, and 0.1 μM and fixed at three different time points: 30 minutes, 45 minutes, and 60 minutes. Untreated cells and cells treated with buffer equivalent are used as negative controls. TNF-α, a known activator of NF-κB, is used as a positive control. Because 45 minutes has been hypothesized as the point of peak NF-κB, it is the only time point shown.

The average antibody labeled NF-κB intensity outside the nucleus and inside the nucleus for each image set is averaged for comparison, shown in Figure 3.4. The TNF-α and control group show more total NF-κB intensity.
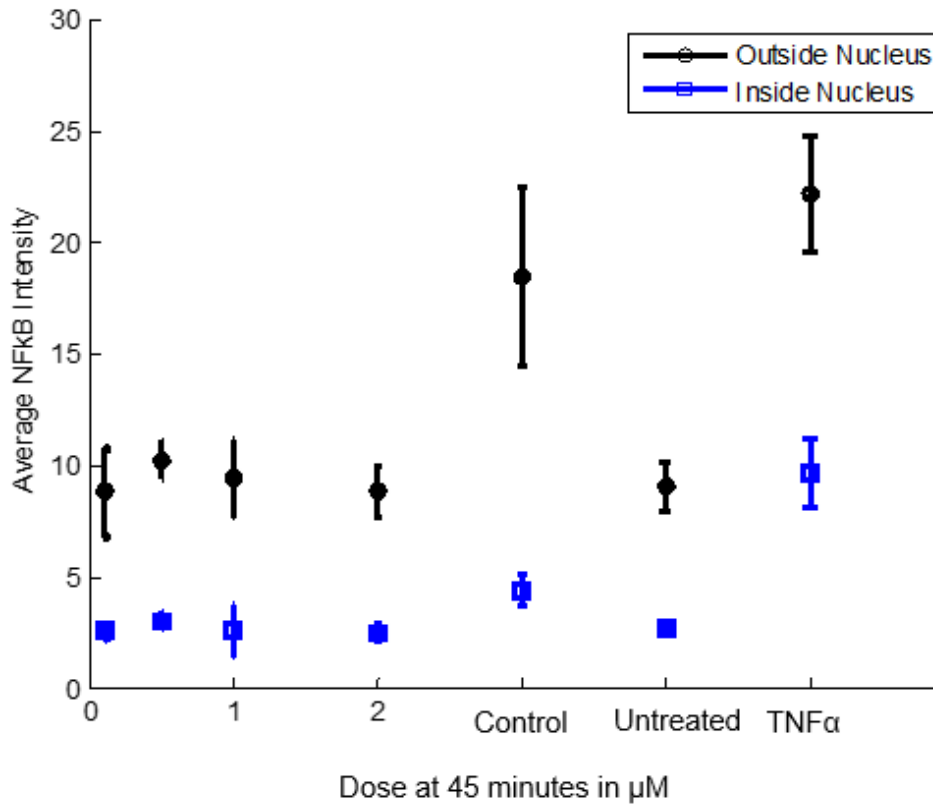
Figure 3.4. *Case B: Average NF-κB Intensity Outside and Inside Nucleus.* The average intensity of the NF-κB stain outside the nucleus (black) and inside the nucleus (blue) is plotted at each concentration point with standard deviation.

After averaging data within distance interval sections to normalize and fitting the exponential model, trends in the model and parameters can be observed in Figure 3.5. This decrease of antibody labeled NF-κB intensity at the border is expected to correlate to an increase in NF-κB transport in the nucleus. There is a trend with the activated NF-κB at the border of the nucleus (parameter *A*) that decreases with increasing Aβ aggregate concentration up to 1 μM, which is congruent with previous hypotheses correlating increasing Aβ aggregate concentration with increased NF-κB translocation at low

concentrations. There is no apparent trend with parameters *k* and *B* and changing Aβ aggregate concentration.
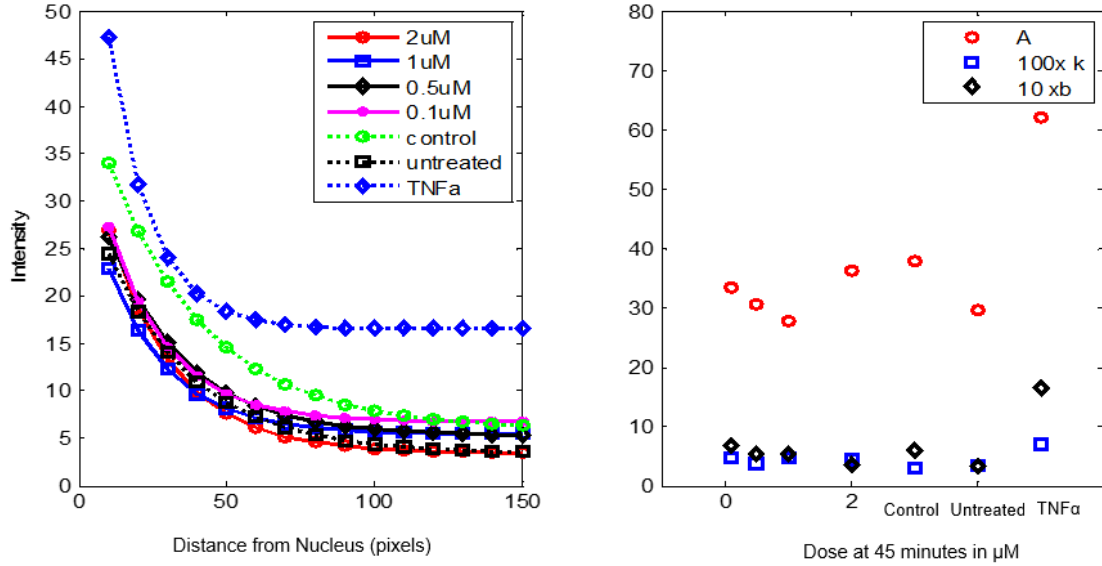
Figure 3.5. *Case B: Model fit of NF-κB intensity as function of Distance from Nuclear Border at 45 minutes.* The exponential model is fit for the Aβ aggregate treatment as a function of concentration (left). Parameters are shown as function of concentration (right).

3.3 APPLYING CEMSA TO CASE C

Case C consisted of monolayers are treated with varying concentrations of TNF-α: a maximum concentration, half maximum concentration, and 0.1 times the maximum concentration (20 units/mL, 10 units/mL and 2 units/mL, respectively) and fixed at 7 time points: 0 minutes, 10 minutes, 20 minutes, 30 minutes, 45 minutes, 60 minutes, 120 minutes, and 240 minutes. Untreated cells fixed at 30 minutes serve as the negative control. An additional experiment evaluates the response after a repeat exposure of TNF-α. Monolayers are initially treated 60 minutes and 240 minutes prior to a second exposure, fixed 30 minutes after the second exposure, and then compared to the single exposure results.

The average antibody labeled NF-κB intensity outside the nucleus and inside the nucleus for each image set is averaged for comparison, shown in Figure 3.6. There is no apparent change with changing TNF-α concentration.
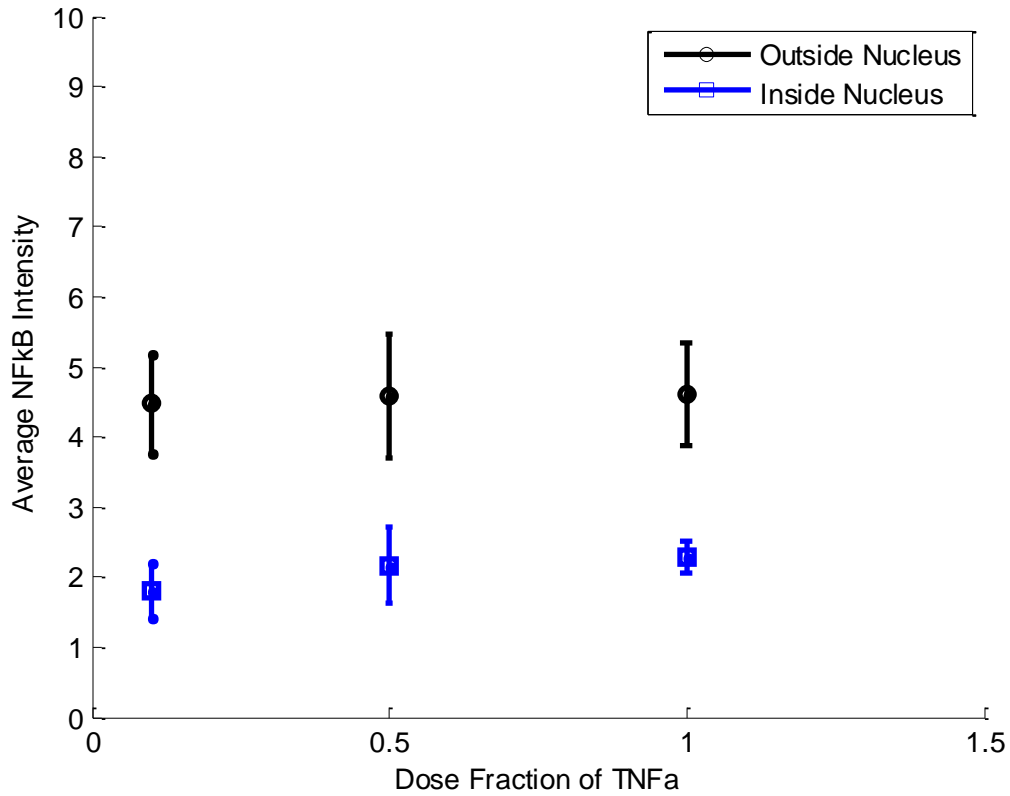
Figure 3.6. *Case C: Average NF-κB Intensity Outside and Inside Nucleus for Concentration Study.* The average intensity of the NF-κB stain outside the nucleus (black) and inside the nucleus (blue) is plotted at each concentration point with standard deviation.

For the concentration dependency study, the model shown in Figure 3.7 shows the total NF-κB at the border of the nucleus (parameter *A*) decreases with decreasing concentration of TNF-α. This indicates increased NF-κB translocation at lower concentrations of TNF-α. There is an evident increase in *B* with increasing TNF-α concentration, however, there is no apparent trend with *k*.

The average NF-κB intensity outside the nucleus and inside the nucleus for each time point in the time-course study is averaged for comparison, shown in Figure 3.8.
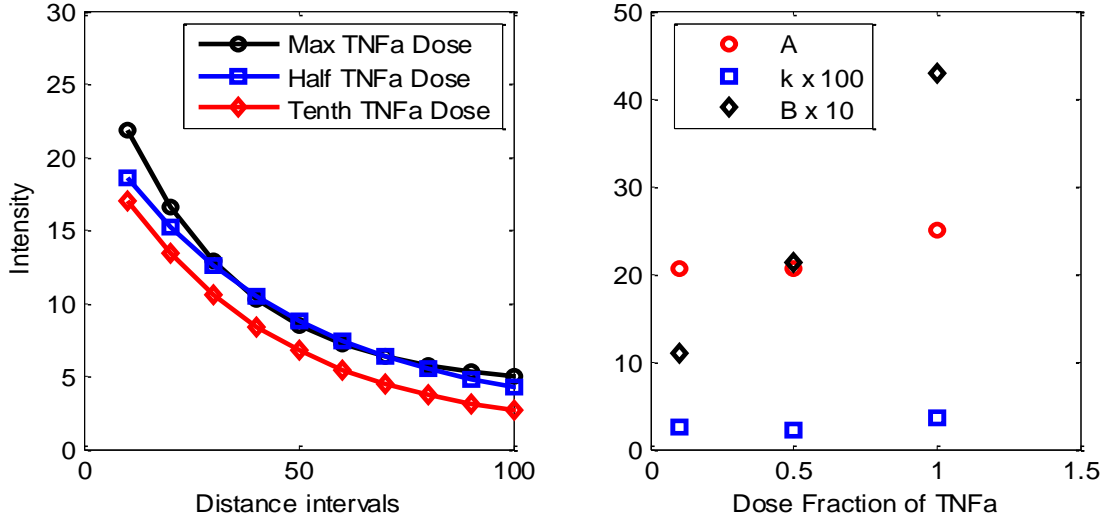
Figure 3.7. *Case C: Model fit of NF-κB intensity as function of Distance from Nuclear Border for Concentration Study.* The exponential model is fit for the varying TNF-α concentration cell treatment. Parameters are shown as function of concentration fraction of TNF-α.
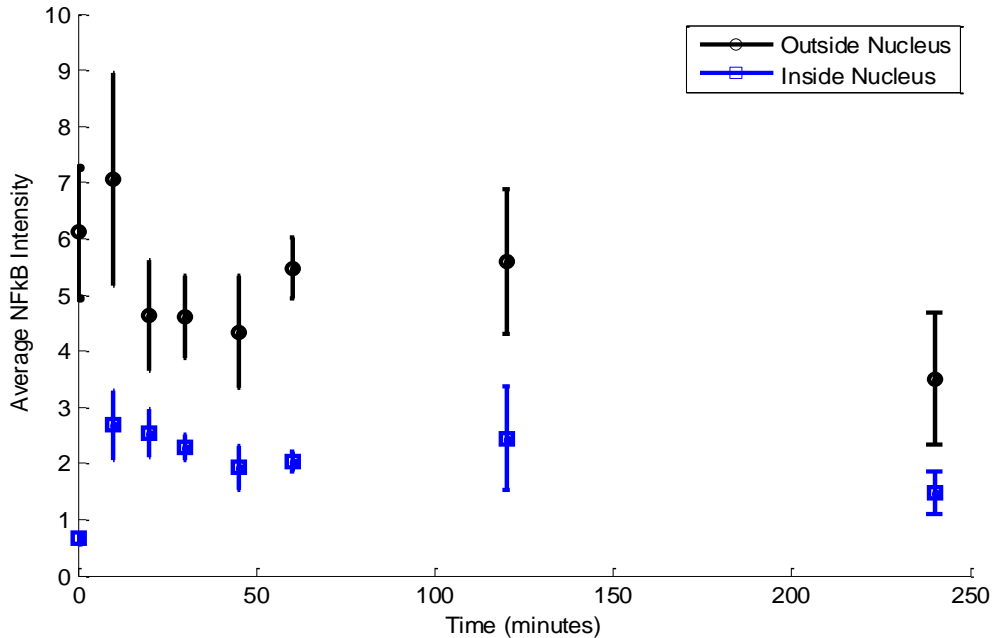
Figure 3.8. *Case C: Average NF-κB Intensity Outside and Inside Nucleus for Time-Course Study.* The average intensity of the NF-κB stain outside the nucleus (black) and inside the nucleus (blue) is plotted at each time point with standard deviation.
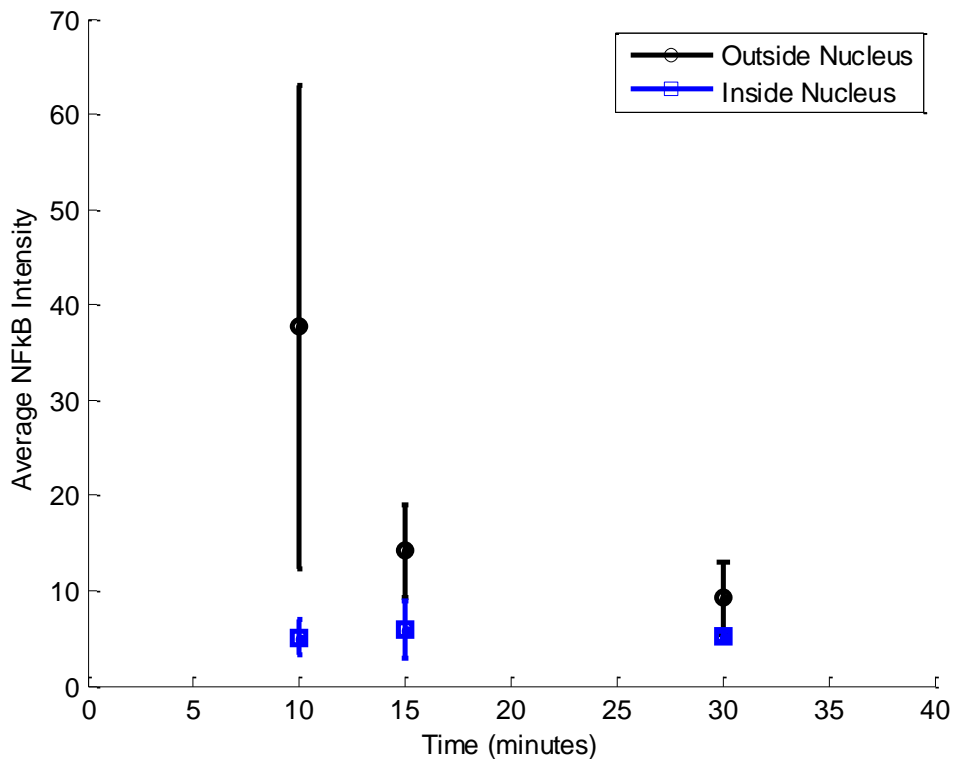
After averaging data within distance interval sections to normalize and fitting the exponential model, trends in the model and parameters can be observed in Figure 3.9. For the time course study treated with 20 units/mL TNF-α, a possible oscillating trend can be observed over time. The minimum value of the total NF-κB at the border of the nucleus (parameter *A*) occurs at 45 minutes, consistent with previous hypotheses that show NF-κB reaching a maximum activation at 45 minutes post treatment. Potential oscillations in parameters *B* and *k* are observed.
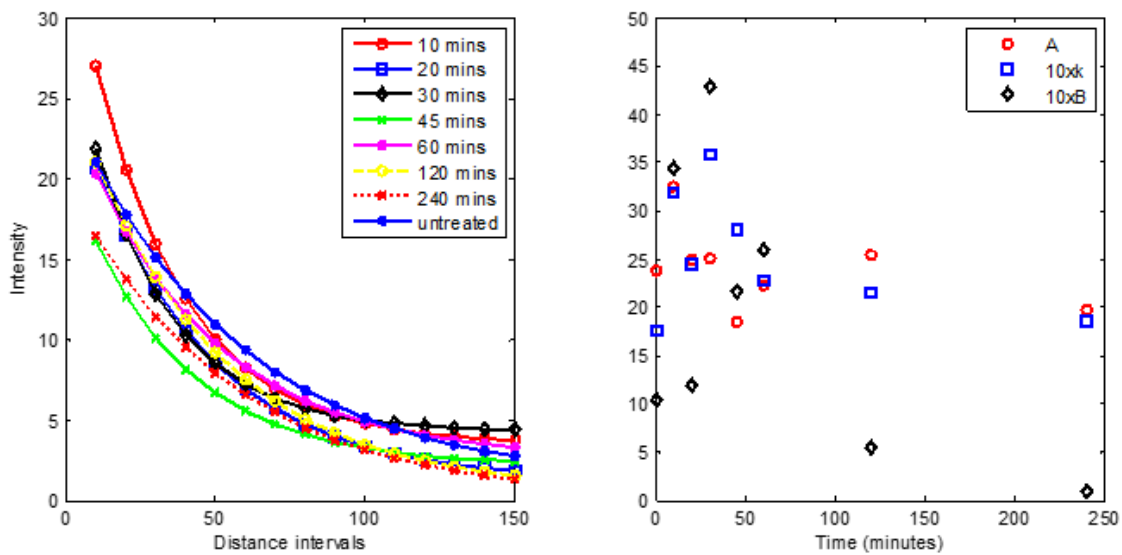
Figure 3.9. *Case C: Model fit of NF-κB intensity as function of Distance from Nuclear Border for Time-Course Study.* The exponential model is fit for the 20 units/mL TNF-α treated response over time. Parameters are shown as function of time.

The average NF-κB intensity outside the nucleus and inside the nucleus for each time point in the repeat exposure study is averaged for comparison, shown in Figure 3.10.

The model of the repeat exposure study is shown in Figure 3.11. The untreated cells showed the highest NF-κB intensity at the border of the nucleus (parameter *A*). The total NF-κB at the border of the nucleus is also higher in the second exposure than in the first

exposure, possibly due to a dampened cellular response. There are no apparent trends with parameters *k* and *B* when analyzing repeat exposure.
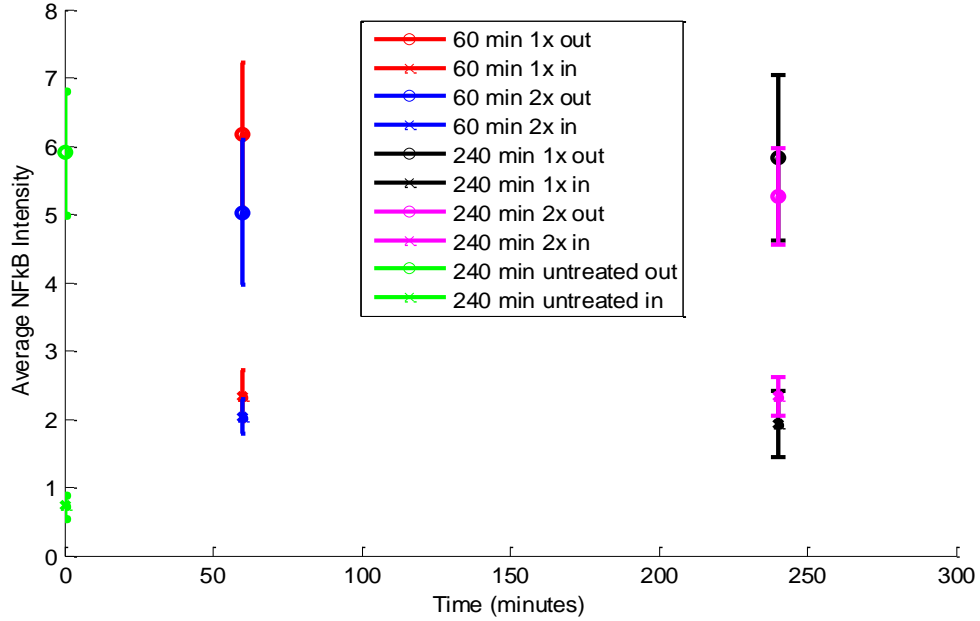
Figure 3.10. *Case C: Average NF-κB Intensity Outside and Inside Nucleus for Repeat Exposure Study.* After being treated with 20 units/mL of TNF-α, the average intensity of the NF-κB outside the nucleus and inside the nucleus is plotted at each point with standard deviation.
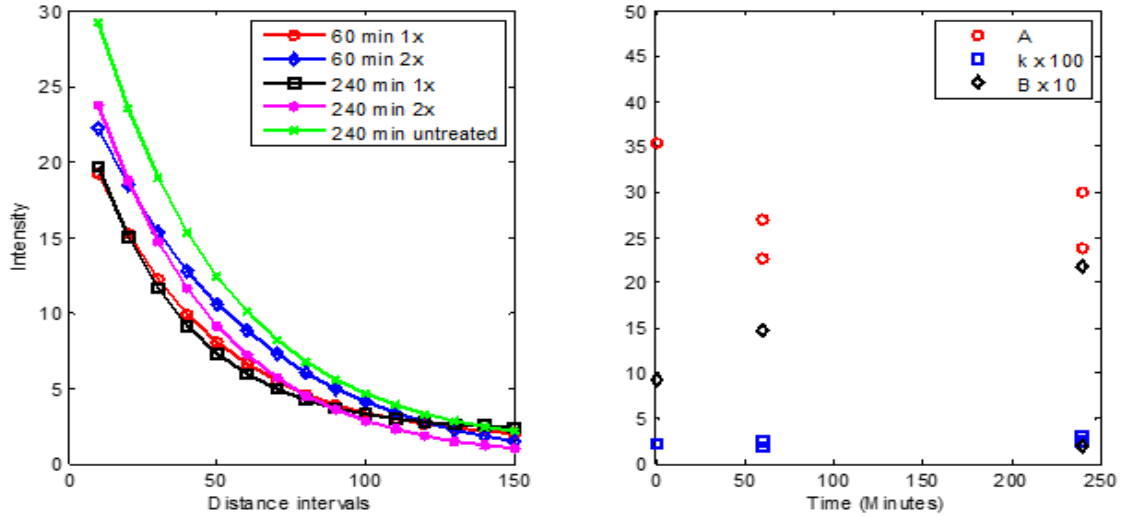
Figure 3.11. *Case C: Model fit of NF-κB intensity as function of Distance from Nuclear Border for Repeat Exposure Study.* The exponential model is fit for the repeat exposure cell treatment. Parameters are shown as function of time.

24

CHAPTER 4

CONCLUSIONS

To fully understand the molecular function of a cell, spatial data must be analyzed. The proposed technique, **CEMSA**, can be used in order to gather relevant spatial distribution data from cellular images. This automated process allows for rapid extraction of data from large image sets, making studies more efficient. Because **CEMSA** requires only a few inputs to be adapted for each experimental set, it can easily be used by researchers with little computational experience. **CEMSA** also stores image data in easily accessible data structures for easy mathematical model generation.

In summary, **CEMSA** identifies and stores the nuclear boundaries, and the NF-κB intensity in relation to the nearest nuclear boundary. By capturing this data, it is possible to analyze the transcription factor behavior in relation to its location over time and in response to various stimulus concentrations. This technique also utilizes readily available software to circumvent manual evaluation, while also allowing the user to customize the analysis depending on the application.

Various hypotheses regarding NF-κB predict its behavior to partition into the nucleus. These can be analyzed further by using **CEMSA** to study spatial distributions within the cytoplasm. Trends in NF-κB behavior as a function of concentration and time were observed in the test sets used. NF-κB at the border of the nucleus (parameter *A*) was shown to decrease with decreasing concentration of TNF-α, indicating increased NF-κB translocation at lower concentrations of TNF-α. Also, the oscillating behavior was

potentially observed. The dampened cellular response of NF-κB with repeat exposure to inflammatory cytokines was observed in Case C.

Further testing of **CEMSA** and further enhancements can be done to ensure reliability and efficiency of the program. Quantitative comparisons to manual methods can be done in order to ensure the consistency of gathered data. Run-time optimization will accelerate the program and allow for more efficient analysis of larger data sets. **CEMSA** can also be used to analyze more test cases with varying treatment methods and more time points. Additionally, **CEMSA** analysis can be used in conjunction with kinetic modeling methods to develop a thorough model of NF-κB based on experimental data.

CEMSA is the one of the first mathematical models to quantify the spatial distribution of NF-κB. It is also one of few models based on spatial experimental data. By modeling the NF-κB response, this work may provide a better understanding of the physiology of inflammation and its role in diseases like AD. This overall method of generating a mathematical model from experimental data can be used to generate mathematical models of various biological systems. This research can open doors for computational modeling of cell signals and the integration of computer models with therapeutic research. Computational models can allow researchers to predict responses within the cell signaling network while also expediting cell image analysis, data analysis, and optimization in therapeutic techniques. Understanding of the factors in inflammatory disease progression can lead to more effective treatments for such devastating illnesses like AD, asthma, arthritis, cancer, diabetes and many other inflammatory diseases.

REFERENCES

1.   Gupta A, Iadecola C. Impaired AB clearance: a potential link between
      atherosclerosis and Alzheimer's disease. *Front Aging Neurosci*. 2015;7(June):1-8.
      doi:10.3389/fnagi.2015.00115.

2.   Bell RD, Zlokovic B V. Neurovascular mechanisms and blood-brain barrier
      disorder in Alzheimer's disease. *Acta Neuropathol*. 2009;118(1):103-113.
      doi:10.1007/s00401-009-0522-3.

3.   Heppner FL, Ransohoff RM, Becher B. Immune attack: the role of inflammation
      in Alzheimer disease. *Nat Rev Neurosci*. 2015;16(6):358-372.
      doi:10.1038/nrn3880.

4.   Gonzalez-Velasquez FJ, Reed JW, Fuseler JW, et al. Activation of brain
      endothelium by soluble aggregates of the amyloid-β protein involves nuclear
      factor-κB. *Curr Alzheimer Res*. 2011;8(1):81-94.
      doi:10.2174/156720511794604606.

5.   Di Z, Herpers B, Fredriksson L, et al. Automated Analysis of NF-kB Nuclear
      Translocation Kinetics in High-Throughput Screening. *PLoS One*. 2012;7(12).
      doi:10.1371/journal.pone.0052337.

6.   Cho K, Shin S, Lee H, Wolkenhauer O. Investigations Into the Analysis and

Modeling of the TNFa-Mediated NF-kB-Signaling Pathway. *Genome Res*. 2003:2413-2422. doi:10.1101/gr.1195703.

7. Kumar A, Takada Y, Boriek AM, Aggarwal BB. Nuclear factor-kappaB: its role in health and disease. *J Mol Med*. 2004;82(7):434-448. doi:10.1007/s00109-004-0555-y.

8. Srinivasan M, Lahiri DK. Significance of NF-kappaB as a pivotal therapeutic target in the neurodegenerative pathologies of Alzheimer's disease and multiple sclerosis. *Expert Opin Ther Targets*. 2015;19(4):471-487. doi:10.1517/14728222.2014.989834.

9. Cheong R, Hoffmann A, Levchenko A. Understanding NF-kappaB signaling via mathematical modeling. *Mol Syst Biol*. 2008;4(192):192. doi:10.1038/msb.2008.30.

10. Alawdi SH, El-Denshary ES, Safar MM, Eidi H, David M-O, Abdel-Wahhab MA. Neuroprotective Effect of Nanodiamond in Alzheimer's Disease Rat Model: a Pivotal Role for Modulating NF-κB and STAT3 Signaling. *Mol Neurobiol*. 2016. doi:10.1007/s12035-016-9762-0.

11. Ihekwaba AEC, Broomhead DS, Grimley RL, Benson N, Kell DB. Sensitivity analysis of parameters controlling oscillatory signalling in the NF-kB pathway : the roles of IKK and IkBa. *Syst Biol (Stevenage)*. 2004;1(1):93-103. doi:10.1049/sb.

12. Schmidt H, Jacobsen EW. Linear systems approach to analysis of complex dynamic behaviours in biochemical networks. *Syst Biol (Stevenage)*.

2004;1(1):149-158. doi:10.1049/sb.

13.   Koh G, Lee D. Mathematical modeling and sensitivity analysis of the integrated
       TNF a -mediated apoptotic pathway for identifying key regulators. *Comput Biol
       Med*. 2011;41(7):512-528. doi:10.1016/j.compbiomed.2011.04.017.

14.   Nelson DE, Ihekwaba AEC, Elliott M, Johnson JR, Kell DB, White MRH.
       Oscillations in NF-kB Signaling Control the Dynamics of Gene Expression.
       Science (80-). 2004;306(October):704-709.

15.   Wang X, Yu W, Zheng L. The dynamics of NF-κB pathway regulated by circadian
       clock. *Math Biosci*. 2015;260:47-53. doi:10.1016/j.mbs.2014.07.012.

16.   Ihekwaba AE, Broomhead DS, Grimley RL, Benson N, Kell DB. Sensitivity
       analysis of parameters controlling oscillatory signalling in the NF-kappaB
       pathway: the roles of IKK and IkappaBalpha. *Syst Biol*. 2004;1(1):93-103.
       http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt
       =Citation&list_uids=17052119.

17.   Mothes J, Busse D, Kofahl B, Wolf J. Sources of dynamic variability in NF-κB
       signal transduction: a mechanistic model. *BioEssays*. 2015;37(4):452-462.
       doi:10.1002/bies.201400113.

18.   Zambrano S, Bianchi ME, Agresti A. High-throughput analysis of NF-kB
       dynamics in single cells reveals basal nuclear localization of NF-kB and
       spontaneous activation of oscillations. *PLoS One*. 2014;9(3).
       doi:10.1371/journal.pone.0090104.

19.   Rao BM, Lauffenburger D a, Wittrup KD. Integrating cell-level kinetic modeling

into the design of engineered protein therapeutics. *Nat Biotechnol*.
2005;23(2):191-194. doi:10.1038/nbt1064.

20.   Maguire O, Loughlin KO, Minderman H. Simultaneous assessment of NF- κ B /
p65 phosphorylation and nuclear localization using imaging fl ow cytometry ☆. *J Immunol Methods*. 2015;423:3-11.

21.   Megason SG, Fraser SE. Imaging in Systems Biology. *Cell*. 2007;130(5):784-795.
doi:10.1016/j.cell.2007.08.031.

22.   Herold J, Zhou L, Abouna S, et al. Computerized Medical Imaging and Graphics
Integrating semantic annotation and information visualization for the analysis of
multichannel fluorescence micrographs from pancreatic tissue. *Comput Med Imaging Graph*. 2010;34:446-452. doi:10.1016/j.compmedimag.2009.10.004.

23.   Carpenter AE, Jones TR, Lamprecht MR, et al. CellProfiler: image analysis
software for identifying and quantifying cell phenotypes. *Genome Biol*.
2006;7(10):R100. doi:10.1186/gb-2006-7-10-r100.

24.   Schneider C a, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of
image analysis. *Nat Methods*. 2012;9(7):671-675. doi:10.1038/nmeth.2089.

25.   Wittenberg T. Review of free software tools for image analysis of fluorescence
cell micrographs. *J Microsc*. 2015;257(1):39-53. doi:10.1111/jmi.12184.

26.   Keenan TM, Hooker A, Spilker ME, et al. Automated identification of axonal
growth cones in time-lapse image sequences. *J Neurosci Methods*. 2006;151:232-238. doi:10.1016/j.jneumeth.2005.07.010.

APPENDIX A: CEMSA MODULE 1

Module 1 is shown as adapted to Case A. Image input, time inputs, and sample inputs are adjusted for each experimental set.

```matlab
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%  Kasey Catalfomo
%  1/12/2016
%  Reading Cell Images
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%

%Endothelial cells treated with TNF-α
%First layer nuclei stain
%Second layer NFkB stain
%Each slide was evaluated at 63x

%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%                    Import Images with TNF-α Max Dose
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%

tifs=[];

%Images in each testing group
numtifs=2;

%Times
numtimes=3;

%10 min tif import
%Treated with TNF-α
tif10=dir('TNF10_10min*');

%15 min tif import
tif15=dir('TNF10_15min*');

%30 min tif import
tif30=dir('TNF10_30min*');

tifs=[tif10(1) tif10(2) tif15(1) tif15(2) tif30(1) tif30(2)];

%Initialize counts and structs
count=0;

timecount=0;

intensities={};
distances={};
outmean={};
```

```matlab
inmean={};
outstd={};
instd={};

for q=1:numtimes
    timecount=timecount+1;
    outactmean=[];
    tifcount=0;
    for w=1:numtifs

            count=count+1;
            tifcount=tifcount+1;
            fprintf('Reading Image %d \n', count)

            %Read TIF layers
            imstr=(tifs(count).name);
            img = imread(imstr);
            red = img(:,:,1); % Red channel = NFkB Stain
            green = img(:,:,2); % Green channel = cellular structure
stain
            blue = img(:,:,3); % Blue channel = DAPI nuclear stain

            intensities{q,w}=red;

            cell_mean_samp=[];


%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%                    Image Processing of Nucleus

%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
            [~, thresholdmin] = edge(blue, 'sobel');
            fudgeFactor = .5;
            BWs = edge(blue,'sobel', thresholdmin * fudgeFactor);
%             figure;
%             imshow(BWs)

            se95 = strel('line', 3, 90);
            se0 = strel('line', 3, 0);

            BWsdil = imdilate(BWs, [se95 se0]);

            BWdfill = imfill(BWsdil, 'holes');
%             figure;
%             imshow(BWdfill)

            %Clears cell boundaries that intersect border
            BWnobord = imclearborder(BWdfill, 4);
%             figure;
%             imshow(BWnobord)

            seD = strel('diamond',1);
            BWfinal = imerode(BWnobord,seD);
%             figure;
%             imshow(BWfinal)

            BWfinal = imerode(BWfinal,seD);
```

```matlab
            BWoutline = bwperim(BWfinal);



            nuc_im=imoverlay(red,BWoutline,[0 1 0]);
%              figure;
%              imshow(nuc_im)



            %Outline in Color:
            SegoutR = blue;
            SegoutG = blue;
            SegoutB = blue;
            %set color: "Cool Green"
            SegoutR(BWoutline) = 0;
            SegoutG(BWoutline) = 200;
            SegoutB(BWoutline) = 150;
            SegoutRGB = cat(3, SegoutR, SegoutG, SegoutB);

            bw=BWfinal;

            L = bwlabel(bw);

            %title('Label matrix-Jet')


            %figure;
            %nuc_im2=imoverlay(red,L,[0 1 0]);
            %imshow(nuc_im2)

            %'PixelList' — p-by-Q matrix specifying the locations of
pixels in the region
            s = regionprops(L, 'PixelIdxList', 'PixelList');
            %figure, imshow(blue)
            %hold on


%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
            %                         Remove Noise

%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.
            %Set threshold of number of pixels to make cell
            thresholdmin1=500;

            %Number of pixels in each "cell", regardless size
            cellpix1=[];
            %Cell labels
            cell_labels1=[];
            %Cell size: number of pixels in each cell
            cell_size1=[];
            %Average instensity of each cell
            cell_mean1=[];

            for k = 1:numel(s)
                idx = s(k).PixelIdxList;
```

33

```matlab
            pixels_in_region_k = blue(idx);
            number_pixels=size(pixels_in_region_k);
            numpix=number_pixels(1);
            cellpix1=[cellpix1 numpix];
            if numpix>thresholdmin1
                cell_labels1=[cell_labels1 k];
                cell_size1=[cell_size1 numpix];
                cell_mean1=[cell_mean1 (mean(pixels_in_region_k))];
            end
        end
        %hold off


%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
        %                     Remove Partial/Overlapping Cells

%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
        meancellsize=mean(cell_size1);
        deviation=std(cell_size1);

        %Narrow Cell size
        %Set threshold of number of pixels to make cell
        thresholdmintest=(meancellsize-1.5*deviation);
        if thresholdmintest<0
            fprintf('ERROR: Cell min size below zero. Image %d \n',
count)
            thresholdmin=thresholdmin1;
        else
            thresholdmin=thresholdmintest;
        end
        thresholdmax=(meancellsize+4*deviation);


        %'PixelList' — p-by-Q matrix specifying the locations of
pixels in the region
        s = regionprops(L, 'PixelIdxList', 'PixelList');
        ss= regionprops(red, 'PixelIdxList', 'PixelList');
        out= regionprops(BWoutline, 'PixelIdxList', 'PixelList');
        figure, imshow(blue)
        hold on

        %Number of pixels in each "cell", regardless size
        cellpix=[];
        %Cell labels
        cell_labels=[];
        %Cell size: number of pixels in each cell
        cell_size=[];
        %Average instensity of each cell
        cell_mean=[];

%
%           figure, imshow(blue)
        hold on
        for k = 1:numel(s)
            idx = s(k).PixelIdxList;
            pixels_activation=red(idx);
```

```matlab
            pixels_in_region_k = blue(idx);
            pixels_perim=BWoutline(idx);
            number_pixels=size(pixels_in_region_k);
            numpix=number_pixels(1);
            cellpix=[cellpix numpix];
            if numpix>thresholdmin & numpix<thresholdmax
                cell_labels=[cell_labels k];
                cell_size=[cell_size numpix];
                cell_mean=[cell_mean (mean(pixels_activation))];
                [max_value, max_loc] = max(pixels_activation);
                max_location = s(k).PixelList(max_loc, :);
                plot(max_location(1), max_location(2), '*')
            end
        end

        %hold off

        cells=length(cell_labels);
        cell_mean_normal=cell_mean/(cells);

        cell_mean_samp=[cell_mean_samp (cell_mean)];


    %.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
    %            Activation Outside and Inside Nuclei           %
    %.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%

        II2=uint8(L==0);
        II=uint8(L>1);
        outmeanb=mean(mean(red.*II2));
        outstdb=std(std(double(red.*II2)));
        inmeanb=mean(mean(red.*II));
        instdb=std(std(double(red.*II)));

        outmean{q,w}=outmeanb;
        outstd{q,w}=outstdb;
        inmean{q,w}=inmeanb;
        instd{q,w}=outstdb;

%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%          Function of Activation Outside Cells
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
        outact=[];
        sumval=0;
        countval=0;
        disp('starting loop')

        CS=1024;
        FS=100;   % Filter size, row+- FS
        get_all=0;% Do you need the distance for all cells or just
the ones < FS?
        output=1; % do some reporting as you run

        II2=ismember(L,cell_labels)-1; %0 at cells, -1 otherwise

        %contour(II2)
        pause(.1)
```

```matlab
            MM2=II2*0;  % Pre-set resulting matrix to 0s
            tic;
            c1=0;  % Cases where inside FS range
            c2=0;  % Cases where outside of FS range
            tt=0;  % Total time (s) doing main calc
            tot=cputime;
            for row=1:size(II2,1)
%               if (output)
%                   disp(['Minute estimate: '
num2str(CS*toc/60,'%0.5f') ' ' num2str(row,'%05d')])
%               end
%               tic;
%               if ((mod(row,10)==0)&output)
%                   % disp(row);
%                   contour(MM2+II2*10);
%                   pause(.1)
%                   tic;
%               end  % print something every 10 iters

                for col=1:size(II2,2)

                    clow=max([1 col-FS]);
                    chi=min([CS col+FS]);
                    rlow=max([1 row-FS]);
                    rhi=min([CS row+FS]);
                    t=cputime;
                    [rows,cols]=find(II2(rlow:rhi,clow:chi)==0);  %
need distances for these in filtered image
                    dist=sqrt( (row-rows-rlow+1).^2+(col-cols-
clow+1).^2);  % Vectorized distance calc
                    tt=tt+cputime-t;
                    if (~isempty(dist)) %% only add if cell within
range
                        MM2(row,col)=min(dist);
                        c1=c1+1;
                    elseif (get_all==1)
                        [rows,cols]=find(II2==0);  % need distances for
the overall image
                        dist=sqrt( (row-rows).^2+(col-cols).^2);
                        MM2(row,col)=min(dist);
                        c2=c2+1;
                    end
                end   % End for each column
            end  % End for each row

            disp(['Total time run is: ' num2str(cputime-tot)])
            disp(['Time spent in calc: ' num2str(tt)])

            distances{q,w}=MM2;


        end

end
hold on
```

```matlab
save('distintA2.mat','distances','intensities','outmean','inmean','outs
td','instd');




%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.


% Kasey Catalfomo
% 11/30/2015

%Load distance and intensity matrices from outsideNFkB_003
fprintf('Loading Image Data \n')
load('distintA2.mat');

%Images in each testing group
numtifs=2;

%Times
numtimes=3;

timecount=0;
count=0;

intmeanset={};
devset={};


for q=1:numtimes
    timecount=timecount+1;
    sampcount=0;
    tifcount=0;
        for w=1:numtifs

            count=count+1;
            tifcount=tifcount+1;

            fprintf('Reading Image %d \n', count)

            MM2=distances{q,w};
            II2=double(intensities{q,w});
            steps=10:10:150;

            maxvals=max(II2);
            minvals=min(II2);


            intmeanmat=[];
            dev=[];


            %Distance intervals of 10 pixels
            for step=10:10:150;

                fprintf('Reading Distances between 0 and %d \n', step)
```

```matlab
                %Set all pixels to 0 except for distances between step
and
                %step-10
                MSTEP=MM2.*(MM2<step).*(MM2>(step-10));
                [r,c]=find(MSTEP>0);

                intval=[];
                for ro=1:length(r)
                    %Find intensity values at coordinates in distance
                    %interval
                    RO=r(ro);
                    CO=c(ro);
                    intval=[intval II2(RO,CO)];
                end
                %Average intensity within interval
                intmean=sum(intval)/length(intval);
                intmeanmat=[intmeanmat intmean];
                dev=[dev (std(intmeanmat))];

            end
            intmeanset{q,w}=intmeanmat;
            devset{q,w}=dev;



        end
end

save('intmeandataA2.mat','intmeanset','devset')
```

APPENDIX B: CEMSA MODULE 2

Module 2 is shown as adapted to Case A. Image input, time inputs, and sample inputs are adjusted for each experimental set.

```matlab
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%          Plots Intensity as Function of Distance & fit
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%Load distance and intensity matrices from outsideNFkB_TNF-α01 & mean
data
%from outactplot_001
fprintf('Loading Image Data \n')
load('distintA2.mat');
load('intmeandataA2.mat')

%Images in each testing group
numtifs=2;

%Times
numtimes=3;

timecount=0;
count=0;

estimate={};
figure;

line={('.-r') ('o-b') ('--g') ('--k') ('m') ('y') ('r') ('b')};

for q=1:numtimes
    timecount=timecount+1;
        tifcount=0;

        X=[];
        Y=[];
        XX=[];
        YY=[];
        Xx=[];
        Yy=[];

        for w=1:numtifs

            count=count+1;
            tifcount=tifcount+1;
            steps=10:10:150;

            fprintf('Reading Image %d \n', count)
```

```matlab
            XX=double(distances{q,w});
            YY=double(intensities{q,w});
            [estimates, model] = fitcurveexp(steps,intmeansetB{q,w});

        end


        fprintf('Time %d \n', timecount)

%          subplot(4,2,q)
%          plot(steps,intmeanset{q,k},'*')
        hold on
        [sse, FittedCurve] = model(estimates);
        plot(steps, FittedCurve, line{q})
        xlabel('Distance intervals');
        ylabel('Intensity');
        axis([0 100 0 30]);
        hold on


        estimate{timecount}=estimates;
        save('estimatematA2.mat','estimate');



    end
    hold on
```

APPENDIX C: CEMSA MODULE 3

Module 3 is shown as adapted to Case A. Image input, time inputs, and sample inputs are adjusted for each experimental set.

```matlab
% Kasey Catalfomo
% 11/30/2015


%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%           Plots Intensity as Function of Distance & fit
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%

%Load distance and intensity matrices from outsideNFkB_TNF-α01 & mean
data
%from outactplot_001
fprintf('Loading Image Data \n')
load('distintA2.mat');
load('intmeandataA2.mat');
load('estimatematA2.mat');

%Images in each testing group
numtifs=2;

%Times
numtimes=3;

timecount=0;
count=0;

estimate={};
figure;

line={('-or') ('-sb') ('-dk')};
sets={'10 minutes' '15 minutes' '30 minutes'};

for q=1:numtimes
    timecount=timecount+1;
        tifcount=0;

        X=[];
        Y=[];
        XX=[];
        YY=[];
```

```matlab
        Xx=[];
        Yy=[];


        for k=1:numtifs

            count=count+1;
            tifcount=tifcount+1;
            steps=10:10:150;

            fprintf('Reading Image %d \n', count)

            XX=double(distances{count});
            YY=double(intensities{count});
            [estimates, model] = fitcurveexp(steps,intmeanset{q,k});

%               Xx=reshape(XX,1,[]);
%               Yy=reshape(YY,1,[]);

        end

%           X=[X Xx];
%           Y=[Y Yy];
%           XY=[X; Y];
%           [x1,x2]=sort(XY(1,:));
%           XY=XY(:,x2);
%           xxx=XY(1,:);
% %           yyy=XY(2,:);
%
%           [estimates, model] = fitcurveexp(steps,yyy);

        fprintf('Time %d \n', timecount)

%           subplot(4,2,q)
%           plot(steps,intmeanset{q,k},'*')
        hold on
        subplot(1,2,1)
        [sse, FittedCurve] = model(estimates);
        plot(steps, FittedCurve, line{q},'LineWidth',2)
        xlabel('Distance intervals (pixels)');
        ylabel('Intensity');
        legend(line,sets);
        axis([0 150 0 100]);

        hold on

%


    hold on

    end
    hold on

%Load distance and intensity matrices from outsideNFkB_TNF-α01 & mean
data
%from outactplot_001
```

```matlab
fprintf('Loading Estimate Data \n')
load('estimatemat.mat');

%Images in each testing group
numtifs=2;

%Times
numtimes=3;

timecount=0;
count=0;

A=[];
k=[];
B=[];

for q=1:numtimes
    timecount=timecount+1;
    tifcount=0;
    fprintf('Parameters for Time Interval %d \n', timecount)
    param=estimate{timecount}
    AA=param(1);
    kk=param(2);
    BB=param(3);

    format short e

    A=[A AA];
    k=[k kk];
    B=[B BB];

end

TimeInts=[10,15,30];

for q=1:numtimes

    subplot(1,2,2)
    plot(TimeInts(q),.1*A(q),'or','LineWidth',2);

    %xlabel('Distance intervals');
    %ylabel('Parameter A');
    axis([0 45 -10 55]);
    hold on

    subplot(1,2,2)
    plot(TimeInts(q),1000*k(q),'sb','LineWidth',2);

    xlabel('Time (minutes)');
    %ylabel('Parameter k');
    %axis([0 250 0 .05]);
    hold on

    subplot(1,2,2)
    plot(TimeInts(q),B(q),'dk','LineWidth',2);

    %xlabel('Distance intervals');
```

```matlab
    %ylabel('Parameter B');
    %axis([0 250 0 6]);

    legend('A x 0.1','k x 1000','B');
    hold on

end
hold on

%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%          Average Intensity Inside and Outside
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
lines={('-ok') ('-sb')};
set={'Outside Nucleus' 'Inside Nucleus'};
figure;
for q=1:numtimes
        outavgg=[];
        hold on

        for k=1:numtifs
            OUT=outmean{q,k};
            outavgg=[outavgg OUT];
        end
        inavgg=[];
        hold on

        for k=1:numtifs
            IN=inmean{q,k};
            inavgg=[inavgg IN];
        end


errorbar(TimeInts(q),mean(outavgg),std(outavgg),lines{1},'LineWidth',2)

errorbar(TimeInts(q),mean(inavgg),std(inavgg),lines{2},'LineWidth',2)
        xlabel('Time (minutes)');
        ylabel('Average NFkB Intensity');
        axis([0 40 0 70]);
        legend(lines,set);
        hold on

end




%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%
%             Raw Data Plots               %
%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%.%

% lines={('.r') ('.b')};
%
% for q=1:numtimes
%     figure;
%     for w=1:numtifs
%
%         plot(distances{q,w},intensities{q,w},lines{w})
```

```matlab
%          xlabel('Distance from Nucleus');
%          ylabel('NFkB Intensity');
%
%          axis([0 100 0 300]);
%
%          hold on
%      end
%      hold on
%
% end
%      hold on
```