

THESIS

COMPARTMENTALIZATION OF MEMBRANE PROTEINS BY THE ACTIN CYTOSKELETON

Submitted by

Jenny Higgins

School of Biomedical Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2013

Master's Committee:

Advisor: Diego Krapf

Michael Tamkun

James Bamburg

Mahmood Azimi-Sadjadi

Copyright by Jenny Higgins 2013

All Rights Reserved

## ABSTRACT

### COMPARTMENTALIZATION OF MEMBRANE PROTEINS BY THE ACTIN CYTOSKELETON

Acting as the point of contact for the outside world, the plasma membrane is crucial for cellular signaling events. Proper organization of membrane components is necessary to accomplish this task. Although a number of experiments have demonstrated the compartmentalization of lipids and proteins on the plasma membrane, direct observation of the mechanisms by which the organization occurs has been challenging, in part due to the imaging restrictions of a diffraction-limited system and the dynamic nature of the membrane compartmentalization.

Using photoactivated localization microscopy (PALM), a superresolution technique, we have captured the dynamics of compartments formed by the cortical actin cytoskeleton. Live human embryonic kidney (HEK293) cells were imaged with a temporal resolution of 2 s and a spatial resolution of 40 nm. The actin cytoskeleton forms compartments with a mean area of  $2.3 \pm 0.3 \mu\text{m}^2$  that are partially outlined by actin bundles. When the PALM images of actin were combined with single particle tracking of membrane proteins, we directly observed the cytoskeleton acting as a barrier to the diffusion of Kv2.1 and Kv1.4, two voltage-gated potassium channels. In addition, we used a novel compartment detection and tracking algorithm to show that Kv2.1 and Kv1.4 channels avoid actin when changing compartments. This work represents the first direct observations of individual membrane protein interactions with barriers formed by the actin cytoskeleton.

## ACKNOWLEDGEMENTS

A huge thank you goes to my advisor, Dr. Diego Krapf for sharing his wisdom, insight, and passion for science with me. Thank you also to Dr. Aubrey Weigel for all of the guidance and help she gave me along the way, without her help I could not have made it this far! Thank you Sanaz Sadegh, Bryce Schroder, and Kari Ecklund for their support and friendship along the way. Thank you to Dr. Michael Tamkun for his guidance and enthusiasm. Thanks also to Phil Fox and Liz Akin for their input and support. Thanks also goes to Dr. Olve Peersen and Grace Campagnola for their inspiration and enthusiasm on the RNA polymerase project. And of course, thanks also to my wonderful family and friends for their support.

The Kv2.1-loopBAD and Kv1.4-loopBAD plasmids were provided by Dr. Michael Tamkun. Preparation of the Dendra2-actin transfected ND723 cells was completed by Liz Akin. HEK cells were provided by Phil Fox. The ABP-tdEosFP plasmid was provided by Dr. Maxime Dahan. The PALM localization algorithm was provided by Dr. Keith Lidke.

## TABLE OF CONTENTS

ABSTRACT .....	ii
ACKNOWLEDGEMENTS.....	iii
Chapter 1. Introduction .....	1
1.1. Organization of the Plasma Membrane .....	1
1.2. Microscopy and the Diffraction Limit .....	4
1.3. Direct Observation of Membrane Compartmentalization.....	5
BIBLIOGRAPHY .....	8
Chapter 2. Superresolved Cortical Actin in Live Cells .....	13
2.1. Introduction .....	13
2.1.1. Actin is an Essential Cellular Protein.....	13
2.1.2. Superresolution Microscopy through PALM.....	14
2.2. Instrumentation and Characterization .....	16
2.3. Methods .....	16
2.3.1. Microscope.....	16
2.3.2. Spatial and Temporal Resolution .....	17
2.3.3. Cell Transfection .....	18
2.3.4. Image Acquisition .....	20
2.3.5. PALM Reconstruction .....	20
2.4. Results .....	21
2.4.1. Cortical Actin in Fixed Cells .....	21
2.4.2. Cortical Actin in Live Cells.....	21

2.5. Discussion .....	22
BIBLIOGRAPHY .....	25
Chapter 3. Interactions Between Membrane Proteins and the Actin Cytoskeleton .....	28
3.1. Introduction .....	28
3.1.1. The Landscape of the Plasma Membrane .....	28
3.1.2. Kv2.1 and Kv1.4 Characteristics .....	29
3.1.3. Single Particle Tracking .....	30
3.2. Methods .....	30
3.2.1. Visualizing Kv2.1 and Kv1.4 .....	30
3.2.2. Dual Color Imaging .....	31
3.2.3. Combining Channels .....	31
3.3. Results .....	32
3.3.1. Kv2.1 Clusters and Cortical Actin .....	32
3.3.2. Kv1.4 and Kv2.1 Interactions with Actin .....	33
3.4. Discussion .....	36
BIBLIOGRAPHY .....	39
Chapter 4. Actin Compartmentalization of Membrane Proteins .....	41
4.1. Introduction .....	41
4.2. Methods .....	42
4.2.1. Watershed .....	42
4.2.2. Compartment Details and Tracking .....	43
4.2.3. Euclidean Distance Mapping .....	43

4.3.	Results .....	44
4.3.1.	Actin is a Permeable Barrier to Membrane Protein Diffusion.....	44
4.3.2.	LatA and SwinA Effects on Kv2.1 Spatial Distribution Relative to Actin.....	46
4.3.3.	SwinA and LatA Effects on Actin Compartment Size .....	48
4.3.4.	SwinA Increases Compartment Size Over Time .....	48
4.4.	Discussion .....	50
BIBLIOGRAPHY .....		55
Chapter 5. Techniques for the Study of Viral RNA Dependent RNA Polymerases.....		57
5.1.	Introduction .....	57
5.1.1.	Role of RNA dependent RNA polymerases in viruses .....	57
5.1.2.	Studying DNA and RNA on the Single-Molecule Scale.....	58
5.2.	Methods .....	59
5.2.1.	Flow Cell Design and Preparation .....	59
5.2.2.	Imaging .....	59
5.3.	Results .....	60
5.3.1.	Flow Cell Proof-of-Concept.....	60
5.3.2.	Tethered RNA .....	61
5.4.	Discussion .....	62
BIBLIOGRAPHY .....		64
Chapter 6. Conclusion .....		66
Appendix A. An Additional Application of Euclidean Distance Mapping to Studying		
	Cellular Events .....	68

Appendix B. Code.....	71
Dependencies .....	71
Determining Nyquist Resolution .....	71
Channel Offset Calculation .....	73
Compartment Tracking .....	75
Compartment Change Analysis .....	79
Distance Distributions .....	86
ImageJ Macros .....	88
Correct Zoom Macro.....	88
Appendix C. Troubleshooting.....	89
PALM Imaging .....	89
Image background is too high.....	89
Initial PALM probe activation is too high.....	89
Too many/few probes activate during image acquisition.....	89
Appendix D. Protocols .....	90
Cell Transfection for PALM Imaging .....	90
Imaging Preparation for Combined SPT and PALM Imaging.....	91
Polystyrene Bead Dish for Colocalization.....	92
Image Acquisition for Combined SPT and PALM Imaging .....	93
Combined SPT and PALM Reconstruction.....	95
Generating EDM Based on Detected Image Edges.....	98
Flow Cell Buffer Preparation.....	99
Typical Flow Cell Protocol.....	100

## CHAPTER 1

# INTRODUCTION

### 1.1. ORGANIZATION OF THE PLASMA MEMBRANE

The plasma membrane is the “skin” of the cell, providing a layer that separates the components of the cell from the outside world. This complex system serves important functions beyond containment of intracellular pieces. For example, endocytosis and exocytosis, processes which allow the cell to pass materials between its interior and exterior, occur due to the dynamic capabilities of the plasma membrane. Signaling structures arranged on the membrane also allow the cell to receive and transmit information. Therefore, the organization of the plasma membrane is essential for the proper operation of these functions, though the mechanism behind the organization is not well understood.

First introduced by Singer & Nicolson in 1972, the fluid mosaic model was one of the earliest attempts to explain the distribution of lipids and proteins on the plasma membrane [1]. In this model, membrane proteins are dispersed at low concentration within a two-dimensional viscous soup made up of phospholipids. The physical implications for this model were further elucidated by Saffman & Delbrück through the two-dimensional fluid continuum model, which predicted diffusion rates for membrane proteins adrift in the lipid solvent of the membrane would be largely independent of size[2]. Over the past 40 years, plasma membrane organization has been studied using a variety of biophysical and biochemical techniques which have raised questions these models are unable to answer.

If membrane proteins are dispersed in a viscous lipid solvent, the diffusion constants on blebs and on the cellular surface should be the same. Some of the earliest experiments used fluorescent recovery after photobleaching (FRAP), a technique which can be used to extract

diffusion rates, to test this hypothesis. The diffusion rates of membrane proteins on blebs matched those predicted by the two-dimensional fluid continuum model [3], but multiple studies found that the diffusion rates on cells were lower than expected by as much as two orders of magnitude [4, 5]. Although the lack of an actin cytoskeleton in blebs was suspected to be a major contributing factor, this discrepancy between experimental measurements and theoretical predictions remained a mystery for many years.

There is also increasing evidence that lipids may form small, transient domains enriched with cholesterol and sphingolipids known as lipid rafts. The existence of these domains is controversial, in part due to the limitations of the techniques available to study them. Despite this, lipid rafts are thought to play important roles in signaling and membrane trafficking [9–11] and may be an important area of study for a variety of diseases, including Alzheimer’s and cancer [9]. FRAP measurements on live cells have detected lipid domains distinct from the typical lipid sea of the plasma membrane [6, 7]. The case for the existence of lipid rafts has been further strengthened by direct observation of spingomyelin- and cholesterol- enriched domains using superresolution imaging [15]. The presence of these patched domains and the signaling molecules found therein suggests a high degree of organization of the plasma membrane.

Other studies have demonstrated that a variety of membrane proteins are compartmentalized on the plasma membrane. The transmembrane protein MHC-1 forms temporary clusters which appear upon exocytosis and gradually diffuse into the surrounding membrane, the stability of which is governed by the actin cytoskeleton [12, 13]. Cadherin [14] and transferrin [28] receptors undergo hop diffusion, where they are thought to be temporarily confined within one compartment before moving to a neighboring one. Restricted within

micron-sized domains thought to be bordered by large actin bundles in the cytoskeleton, IgE also exhibits confined behavior [26]. Kv2.1 channels form stable, micron-sized clusters, the size of which is controlled by fluctuations in trafficking to and from the plasma membrane [27]. Thus, compartmentalization effects are felt both by lipids and by membrane proteins.

How are the components of the plasma membrane organized? The presence of barriers to membrane protein diffusion has been directly demonstrated through laser optical trap experiments which dragged individual membrane proteins across the plasma membrane [28]. In addition, the depth at which a membrane protein extends into the cytoplasm was found to have an effect on the distance they could travel before encountering a barrier (the barrier free path) [29]. A recent study used superresolution microscopy to reveal that hemagglutinin clusters colocalize with actin [30]. Since the actin cytoskeleton comes in close proximity to the plasma membrane, it may play a key role in the organization of membrane proteins by acting as a steric hindrance to diffusion [13, 26]. The compartments formed by these barriers exhibit dynamic behavior which likely contributes to the complex diffusion behavior of membrane proteins.

The cytoskeleton may influence the organization of membrane proteins by more than direct steric interactions with actin bundles. The picket fence model [18] describes a scenario where membrane proteins anchored to the actin cytoskeleton act as pickets, leading to the temporary confinement of lipids and other membrane proteins [8, 19]. By studying the diffusion path of individual lipid molecules, the sizes of the compartments are estimated to range from 32 nm to 230 nm, depending on cell type [8]. The effect that these barriers have on lipids and membrane proteins has been extensively studied [20–23, 31] and is commonly known as obstructed diffusion.

In light of all of these observations, researchers have modified the simplistic fluid mosaic model in favor of a compartmentalized plasma membrane. However, many of the measurements and observations to support compartmentalization have been made indirectly by studying diffusion rates and the diffusion paths of individual particles. Few direct measurements have been made where a membrane protein and the barrier are both visualized so that their interactions can be directly observed. This has been in large part due to limitations in imaging techniques which are now being overcome with the advent of superresolution imaging.

## 1.2. MICROSCOPY AND THE DIFFRACTION LIMIT

Traditional microscopes suffer from a fundamental limitation known as the diffraction limit. Initially noted in 1873, Ernst Abbe discovered the minimum spot diameter ( $d$ ) of a point light source followed the relationship:

$$d = \frac{\lambda}{2NA}$$

where  $\lambda$  is the emitted light wavelength and NA is the numerical aperture, which in modern systems can reach approximately 1.5. Thus, with the visible light spectrum ranging from 390 nm to 700 nm, a point source occupies a region with a diameter of approximately 130 nm to 233 nm, depending on the wavelength of light being viewed. As a result, if two point sources of light are within  $\sim 200$  nm of each other, they are impossible to distinguish. Actin filaments which make up the cortical cytoskeleton are smaller than the diffraction limit, requiring a system that is not diffraction limited for accurate imaging.

However, when multiple emitters are sufficiently separated so that their light patterns do not overlap, their location can be determined with an accuracy often below 10 nm. This technique, known as single-particle tracking (SPT)[17, 20, 24], is particularly suited for studying the dynamics of individual molecules and has been used to investigate the diffusion of a variety of lipids and membrane proteins. Since ergodicity breaking occurs in the plasma membrane of live cells [33], single molecule studies are important to understanding the mechanisms which govern cellular systems.

PhotoActivated Localization Microscopy (PALM) is a superresolution technique which has been revolutionizing the study of single molecule dynamics in cells[34–36]. PALM breaks the diffraction barrier by spreading out signals in time. Traditionally performed on fixed cells where the locations of hundreds of thousands of molecules could be gathered over the period of hours, recent studies have successfully applied PALM to the study of live cells [30, 37, 38]. Thus, PALM offers a way to image the cortical actin cytoskeleton in live cells with a spatial resolution superior to the diffraction limit. When combined with SPT, the interactions between cytoskeletal structures and membrane proteins can be directly observed with sub-diffraction limit accuracy.

### 1.3. DIRECT OBSERVATION OF MEMBRANE COMPARTMENTALIZATION

Here, we focus primarily on the role of cortical actin in close proximity to the plasma membrane as a steric barrier to membrane protein diffusion and thus as a potential mechanism of membrane compartmentalization. Specifically, we studied the interactions of actin with Kv2.1 and Kv1.4, two voltage-gated potassium channels. Kv2.1 (N-terminus: 186 aa, C-terminus: 430 aa) has a larger number of amino acids which extend into the cytoplasm compared with Kv1.4 (N-terminus: 308 aa, C-terminus: 85 aa), which could possibly result

in the proteins having different cytoplasmic depths and influence the barrier free path of the proteins [29]. The actin cytoskeleton is imaged in live cells with a temporal resolution of 2 s and a spatial resolution of 40 nm using PALM. Through single particle tracking, Kv2.1 and Kv1.4 are directly observed to interact with actin bundles in two distinct ways: (1) by reflecting away from actin boundaries and (2) by diffusing within actin bundles. In addition, the channels occasionally ignore actin boundaries, which may be due to the actin bundles not being close enough to the plasma membrane to act as a barrier.

Domains formed by the actin cytoskeleton were compartmentalized through use of the watershed algorithm, which fills in the gaps of the cytoskeleton with a best guess of the compartment boundary. The actin compartments observed have mean area of  $2.3 \pm 0.3 \mu\text{m}^2$  and increase significantly in size on the time scale of minutes upon application of 75 nM Swinholide A (SwinA), an actin-disrupting drug. However, the application of 100 nM Latrunculin A (LatA) did not significantly alter compartment size even after extended periods of time. The distribution of Kv2.1 channels in relation to the actin cytoskeleton was also determined using Euclidean distance mapping (EDM), demonstrating that Kv2.1 channels are more likely to be found further than 275 nm away from actin than the random case. The distribution of distances of Kv2.1 from the actin cytoskeleton broke down upon application of 100 nM LatA. However, the distinction of Kv2.1 distances from actin compared to the random case of a lipid on the plasma membrane was enhanced by application of 75 nM SwinA.

Compartment tracking combined with single particle tracking of membrane channels revealed that Kv2.1 channels show a preference of not crossing actin upon changing compartments. This preference is also shown by Kv1.4, though to a lesser degree, a difference that may be related to the cytoplasmic domain sizes of the proteins.

## BIBLIOGRAPHY

- [1] Singer, S. J., & Nicolson, G. L. (1972). The fluid mosaic model of the structure of cell membranes. *Science*, 175(23), 720-731.
- [2] Saffman, P. G., & Delbrück, M. (1975). Brownian motion in biological membranes. *Proceedings of the National Academy of Sciences*, 72(8), 3111-3113.
- [3] Barak, L. S., & Webb, W. W. (1982). Diffusion of low density lipoprotein-receptor complex on human fibroblasts. *The Journal of cell biology*, 95(3), 846-852.
- [4] Webb, W. W., Barak, L. S., Tank, D. W., & Wu, E. S. (1980). Molecular mobility on the cell surface. In *Biochemical Society symposium* (No. 46, pp. 191-205).
- [5] Murase, K., Fujiwara, T., Umemura, Y., Suzuki, K., Iino, R., Yamashita, H., Saito, M., Murakoshi, H., Ritchie, K., & Kusumi, A. (2004). Ultrafine membrane compartments for molecular diffusion as revealed by single molecule techniques. *Biophysical journal*, 86(6), 4075-4093.
- [6] Wolf, D. E., Kinsey, W., Lennarz, W., & Edidin, M. (1981). Changes in the organization of the sea urchin egg plasma membrane upon fertilization: indications from the lateral diffusion rates of lipid-soluble fluorescent dyes. *Developmental biology*, 81(1), 133-138.
- [7] Yechiel, E., & Edidin, M. (1987). Micrometer-scale domains in fibroblast plasma membranes. *The Journal of cell biology*, 105(2), 755-760.
- [8] Kusumi, A., Nakada, C., Ritchie, K., Murase, K., Suzuki, K., Murakoshi, H., ... & Fujiwara, T. (2005). Paradigm shift of the plasma membrane concept from the two-dimensional continuum fluid to the partitioned fluid: high-speed single-molecule tracking of membrane molecules. *Annu. Rev. Biophys. Biomol. Struct.*, 34, 351-378.
- [9] Brown, D. A., & London, E. (1998). Functions of lipid rafts in biological membranes. *Annual review of cell and developmental biology*, 14(1), 111-136.

- [10] Simons, K., & Gerl, M. J. (2010). Revitalizing membrane rafts: new tools and insights. *Nature Reviews Molecular Cell Biology*, 11(10), 688-699.
- [11] Lingwood, D., & Simons, K. (2010). Lipid rafts as a membrane-organizing principle. *science*, 327(5961), 46-50.
- [12] Lavi, Y., Edidin, M. A., & Gheber, L. A. (2007). Dynamic patches of membrane proteins. *Biophysical journal*, 93(6), L35-L37.
- [13] Lavi, Y., Gov, N., Edidin, M., & Gheber, L. A. (2012). Lifetime of major histocompatibility complex class-I membrane clusters is controlled by the actin cytoskeleton. *Biophysical journal*, 102(7), 1543-1550.
- [14] Kusumi, A., Suzuki, K., & Koyasako, K. (1999). Mobility and cytoskeletal interactions of cell adhesion receptors. *Current opinion in cell biology*, 11(5), 582-590.
- [15] Mizuno, H., Abe, M., Dedecker, P., Makino, A., Rocha, S., Ohno-Iwashita, Y., ... & Miyawaki, A. (2011). Fluorescent probes for superresolution imaging of lipid domains on the plasma membrane. *Chemical Science*, 2(8), 1548-1553.
- [16] Kusumi, A., Sako, Y., & Yamamoto, M. (1993). Confined lateral diffusion of membrane receptors as studied by single particle tracking (nanovid microscopy). Effects of calcium-induced differentiation in cultured epithelial cells. *Biophysical journal*, 65(5), 2021-2040.
- [17] Ghosh, R. N., & Webb, W. W. (1994). Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules. *Biophysical journal*, 66(5), 1301-1318.
- [18] Sheetz, M. P. (1983). Membrane skeletal dynamics: role in modulation of red cell deformability, mobility of transmembrane proteins, and shape. In *Seminars in hematology* (Vol. 20, No. 3, pp. 175-188).

- [19] Ritchie, K., Iino, R., Fujiwara, T., Murase, K., & Kusumi, A. (2003). The fence and picket structure of the plasma membrane of live cells as revealed by single molecule techniques (Review). *Molecular membrane biology*, 20(1), 13-18.
- [20] Saxton, M. J., & Jacobson, K. (1997). Single-particle tracking: applications to membrane dynamics. *Annual review of biophysics and biomolecular structure*, 26(1), 373-399.
- [21] Sung, B. J., & Yethiraj, A. (2006). Lateral diffusion and percolation in membranes. *Physical review letters*, 96(22), 228103.
- [22] Horton, M. R., Höfling, F., Rädler, J. O., & Franosch, T. (2010). Development of anomalous diffusion among crowding proteins. *Soft Matter*, 6(12), 2648-2656.
- [23] Weigel, A. V., Ragi, S., Reid, M. L., Chong, E. K., Tamkun, M. M., & Krapf, D. (2012). Obstructed diffusion propagator analysis for single-particle tracking. *Physical Review E*, 85(4), 041924.
- [24] Bannai, H., Lévi, S., Schweizer, C., Dahan, M., & Triller, A. (2007). Imaging the lateral diffusion of membrane molecules with quantum dots. *Nature protocols*, 1(6), 2628-2634.
- [25] Fujiwara, T., Ritchie, K., Murakoshi, H., Jacobson, K., & Kusumi, A. (2002). Phospholipids undergo hop diffusion in compartmentalized cell membrane. *The Journal of cell biology*, 157(6), 1071-1082.
- [26] Andrews, N. L., Lidke, K. A., Pfeiffer, J. R., Burns, A. R., Wilson, B. S., Oliver, J. M., & Lidke, D. S. (2008). Actin restricts Fc $\epsilon$ RI diffusion and facilitates antigen-induced receptor immobilization. *Nature cell biology*, 10(8), 955-963.
- [27] Weigel, A. V., Fox, P. D., Akin, E. J., Ecklund, K. H., Tamkun, M. M., & Krapf, D. (2012). Size of Cell-Surface Kv2.1 Domains is Governed by Growth Fluctuations. *Biophysical journal*, 103(8), 1727-1734.

- [28] Sako, Y., & Kusumi, A. (1995). Barriers for lateral diffusion of transferrin receptor in the plasma membrane as characterized by receptor dragging by laser tweezers: fence versus tether. *The Journal of cell biology*, 129(6), 1559-1574.
- [29] Edidin, M., Zuniga, M. C., & Sheetz, M. P. (1994). Truncation mutants define and locate cytoplasmic barriers to lateral mobility of membrane glycoproteins. *Proceedings of the National Academy of Sciences*, 91(8), 3378-3382.
- [30] Gudheti, M. V., Curthoys, N. M., Gould, T. J., Kim, D., Gunewardene, M. S., Gabor, K. A., ... Hess, S. T. (2013). Actin Mediates the Nanoscale Membrane Organization of the Clustered Membrane Protein Influenza Hemagglutinin. *Biophysical journal*, 104(10), 2182-2192.
- [31] Deverall, M. A., Gindl, E., Sinner, E. K., Besir, H., Ruehe, J., Saxton, M. J., & Naumann, C. A. (2005). Membrane lateral mobility obstructed by polymer-tethered lipids studied at the single molecule level. *Biophysical journal*, 88(3), 1875-1886.
- [32] Saxton, M. J. (1994). Anomalous diffusion due to obstacles: a Monte Carlo study. *Biophysical journal*, 66(2), 394-401.
- [33] Weigel, A. V., Simon, B., Tamkun, M. M., & Krapf, D. (2011). Ergodic and nonergodic processes coexist in the plasma membrane as observed by single-molecule tracking. *Proceedings of the National Academy of Sciences*, 108(16), 6438-6443.
- [34] Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., ... & Hess, H. F. (2006). Imaging intracellular fluorescent proteins at nanometer resolution. *Science*, 313(5793), 1642-1645.

- [35] Hess, S. T., Girirajan, T. P., & Mason, M. D. (2006). Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophysical journal*, 91(11), 4258-4272.
- [36] Rust, M. J., Bates, M., & Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature methods*, 3(10), 793-796.
- [37] Shroff, H., Galbraith, C. G., Galbraith, J. A., & Betzig, E. (2008). Live-cell photoactivated localization microscopy of nanoscale adhesion dynamics. *Nature Methods*, 5(5), 417-423.
- [38] Izeddin, I., Specht, C. G., Lelek, M., Darzacq, X., Triller, A., Zimmer, C., & Dahan, M. (2011). Super-resolution dynamic imaging of dendritic spines using a low-affinity photoconvertible actin probe. *PLoS One*, 6(1), e15611.
- [39] Kusumi, A., Suzuki, K. G., Kasai, R. S., Ritchie, K., & Fujiwara, T. K. (2011). Hierarchical mesoscale domain organization of the plasma membrane. *Trends in biochemical sciences*, 36(11), 604-615.

## CHAPTER 2

# SUPERRESOLVED CORTICAL ACTIN IN LIVE CELLS

### 2.1. INTRODUCTION

2.1.1. ACTIN IS AN ESSENTIAL CELLULAR PROTEIN. The cell is a complex machine capable of conforming to a variety of shapes, propelling itself through liquid, crawling along a surface, and even splitting in two. These essential processes are made possible by the cytoskeleton (Figure 2.1), a structural network capable of dynamic response to changes in the cell. Actin is an important component of the cytoskeleton which is partially responsible for the ability of the cell to rapidly react to signals. Initially monomeric, actin self-assembles into polymer filaments. A variety of linking proteins allow these filaments to form a range of structures, from thick, rope-like bundles which form stress fibers to web-like networks which make up the cellular cortex. Although actin is present throughout the cell, it is most dense in the cell cortex where it supports the plasma membrane.

Although the role of actin in major cellular events such as endocytosis [17] has been extensively studied, its effect on certain other cellular processes is unclear. Through experiments which indirectly probe the mechanical properties of the plasma membrane, actin has been suspected to play a role in organizing proteins on the plasma membrane [6, 7]. Recent discoveries have revealed that for some plasma membrane proteins, organization is linked to function [5]. Revealing the mechanism by which the cytoskeleton arranges the plasma membrane could open the door to many key insights about how the cell operates.

Traditional fluorescence microscopy techniques have revealed the macro-structure of actin, allowing visualization of large stress fiber bundles or a haze where actin meshworks are suspected to reside. However, due to the diffraction limit, the details of these structures cannot

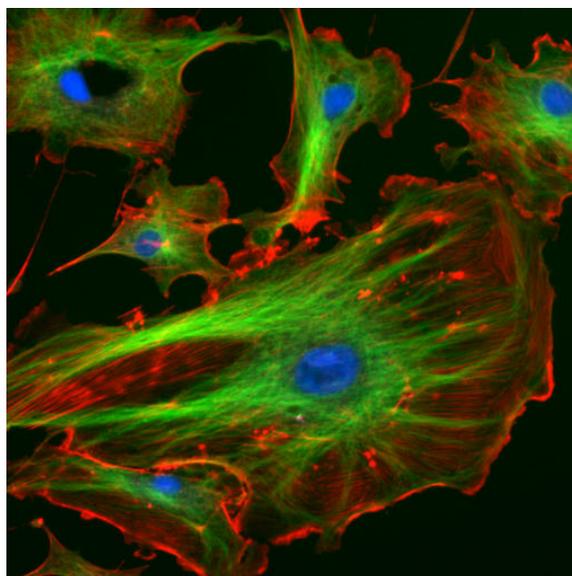


FIGURE 2.1. The cytoskeleton imaged using traditional, diffraction-limited fluorescence microscopy. Actin is labeled in red, microtubules in green, and nuclei in blue. Source: Wikipedia

be resolved in live cells using traditional techniques. Electron microscopy has been used to study the cortical actin cytoskeleton with nanometer resolution, but is incompatible with live cells. Studying the dynamics of the cytoskeleton in live cells is essential for understanding how the cell reacts to changes in its environment.

2.1.2. SUPERRESOLUTION MICROSCOPY THROUGH PALM. The first step in unlocking the mystery of how the actin cytoskeleton interacts with membrane proteins is to image the structures with sufficient temporal and spatial resolution for an accurate depiction of actin structures. Actin filaments are smaller than the diffraction limit which is the major contributing factor as to why traditional fluorescence microscopy techniques have been insufficient to reveal the mechanisms of the cellular cortex in arranging the plasma membrane. PhotoActivated Localization Microscopy (PALM) is a superresolution imaging technique that has revolutionized cellular imaging by revealing sub-diffraction limited details [8–10]. Dendritic spines in live neurons have been studied using this method with impressive results

[4], where a spatial resolution of as low as 44 nm with a temporal resolution of 50 s was achieved in live cells. By improving the temporal resolution so that changes that occur over the span of seconds can be observed, this same principle can be used to study the dynamics of the actin cytoskeleton.

PALM bypasses the diffraction limit by spreading out signals in time. To accomplish this, fluorophores with a particular property which allows them to be altered from an inactive state to an active state label the molecule of interest. Initially, all of the probes are in the inactive state. When an activation laser is directed at the pool of inactive fluorophores, a stochastic subset of the probes undergo a molecular change that results in their activation. The activated probes now emit light when exposed to the excitation laser, while the inactive probes are unaffected. After sufficient exposure to the excitation source, a probe will eventually enter the bleached state where it no longer emits light. Several strategies exist for controlling the density of probes activated, from tuning the laser power to changing laser exposure time.

The diffraction limit causes two objects that are closer than 250 nm to be indistinguishable. However, since only a fraction of the probes are activated at one time, they are less likely to occupy the same diffraction-limited region, and thus can be localized using Gaussian fitting techniques with nanometer accuracy. Once the activated pool is bleached, another subset of the inactive probes is activated, providing information about the location of additional molecules. This cycle is repeated many times, resulting in a stack of thousands of images which consist of the diffraction-limited spots created by individual probes. Once the images are processed such that each molecule is localized, a single image is reconstructed

using the molecular position information from hundreds or even thousands of frames. The result is an image that often bypasses the diffraction limit by an order of magnitude.

An obvious consequence of this process is that extended periods of time are required to generate a single superresolution image. In fixed cells, this is not an issue since the sample is static. In live cells, the structures being imaged are typically mobile. Requiring minutes or longer to acquire a single image is typically not sufficient to capture structure without significantly blurring the image. Here, we have imaged the cortical actin cytoskeleton in live human embryonic kidney cells (HEK293), achieving a spatial resolution as low as 40 nm with a temporal resolution of 2 seconds.

## 2.2. INSTRUMENTATION AND CHARACTERIZATION

### 2.3. METHODS

2.3.1. MICROSCOPE. A homebuilt microscope built around an IX71 Olympus base (Olympus, Melville, NY) was used in all imaging experiments. A 405 nm laser was used to activate the photoswitchable tdEosFP and Dendra2 probes, while 473 nm (blue) and 532 nm (green) lasers were used to excite the probes in their inactive and active states, respectively. Prior to combining the beams, the three lasers passed through separate telescopes which modified the beams to have the same diameter. In addition, the violet laser was directed through a 10  $\mu\text{m}$  pinhole for a cleaner beam profile. The beams were combined with dichroics before passing through a telescope for beam expansion and recollimation. An antireflection-coated achromatic 400 mm lens focused the beams on the back aperture of the objective (100x PlanApo N.A. 1.45; Olympus) and was adjusted so that the beam was totally internally reflected on the cell dish. Laser power after the objective was 20 mW for the blue laser, 1.0

mW for the violet laser, and 35 mW for the green laser. Samples were imaged using Total Internal Reflection Fluorescence Microscopy (TIRFM), which greatly reduced the power used to activate and excite the probes and also limited the imaging depth to approximately 200 nm. To optimize the number of PALM localizations per frame, power was further controlled for each individual laser through ND filters which could be changed during imaging. Light emitted by the probes was captured by an Andor iXon DU-888 EMCCD camera (Andor Technology, Belfast, Ireland).

2.3.2. SPATIAL AND TEMPORAL RESOLUTION. One of the largest challenges faced in PALM imaging is the tradeoff between spatial and temporal resolution. In fixed cells, thousands of frames collected over long time periods can be used to generate a single superresolution image with very high spatial resolution. However, this strategy becomes problematic in live cells because the features being imaged are not static. The more frames used to generate a PALM image, the more likely significant blurring will be introduced which will reduce the spatial resolution. In contrast, too few frames will result in an incomplete image with insufficient particles to determine structure accurately. A major goal of this project was to accurately image actin, which is a highly dynamic structure, by maximizing the spatial and temporal resolution of the PALM images.

The Nyquist Shannon theorem[2] states that a signal must be sampled at two times the maximum frequency present to preserve the signal. In terms of a 2D superresolution image, the maximum spatial resolution is twice the mean distance between neighboring molecules[3]. To determine the Nyquist limited spatial resolution of our system, the average distance to the nearest neighbor of each molecule in a superresolution image with a given temporal resolution was found. Figure 2.3 illustrates the tradeoff between spatial and temporal resolution. As

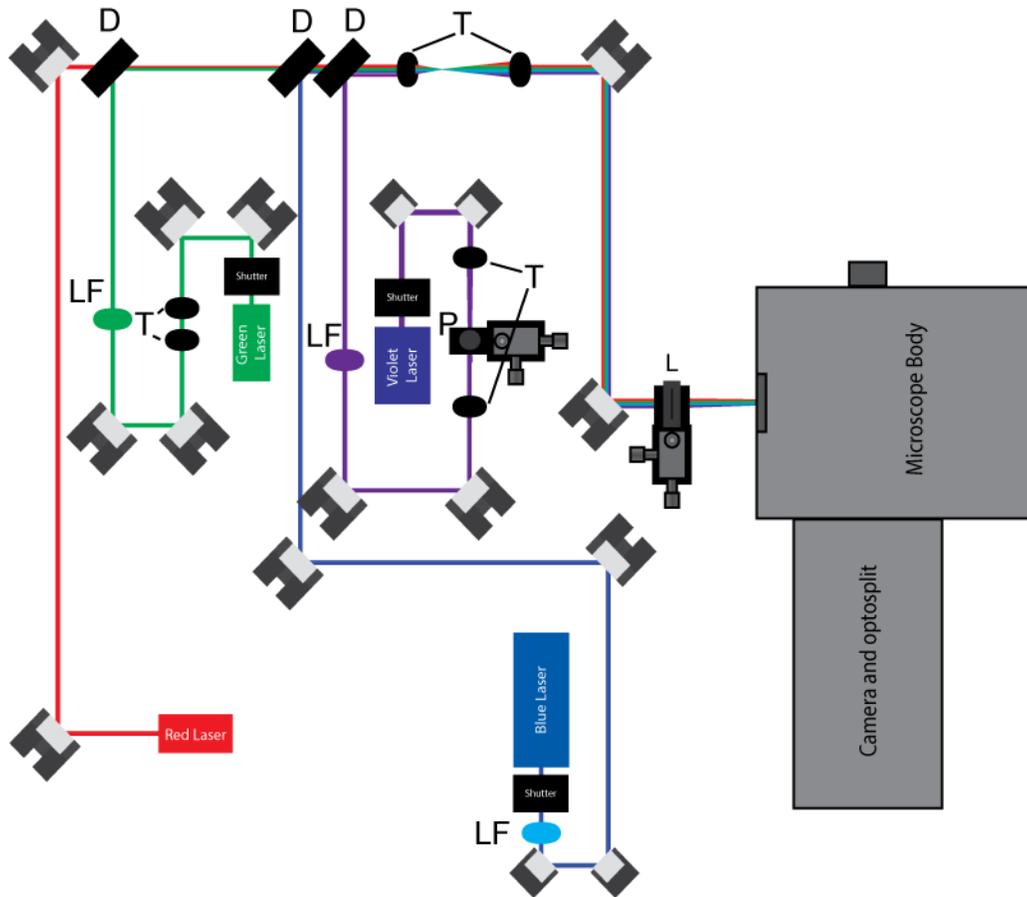


FIGURE 2.2. Microscope schematic. Three lasers were used in the PALM experiments: 405 nm (violet), 473 nm (blue), and 532 nm (green). A 638 nm (red) laser is also shown but was not used. LF indicates a line filter used to limit the wavelengths of the lasers to a more precise range. T: a pair of lenses used as a telescope to expand and columnate the beam. P: a pinhole used to clean up the beam profile of the violet laser. D: dichroics which combined multiple lasers into the same path. L: lens used to focus the beams on the back aperture of the objective.

more frames are included in the reconstruction, the spatial resolution improves. The majority of the PALM analysis completed used a temporal resolution of 2 s, which resulted in a spatial resolution as low as 40 nm.

2.3.3. CELL TRANSFECTION. For fixed cell PALM measurements, ND723 cells were transfected with 3  $\mu\text{g}$  of a Dendra2-actin plasmid and fixed. For live cell measurements,

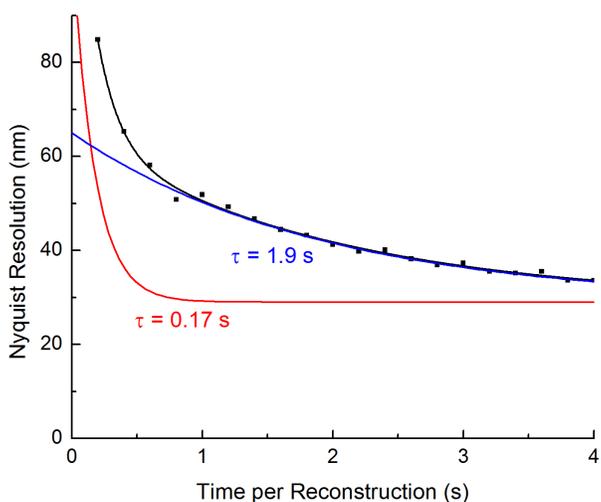


FIGURE 2.3. Nyquist limited spatial resolution versus time resolution. As the number of frames used to construct a single superresolution image increases (time resolution), the spatial resolution decreases, following a double decreasing exponential with a short time constant of 0.17 s and a longer time constant of 1.9 s. Typically, 2 seconds (100 frames) are used to construct a single superresolution image.

Human embryonic kidney (HEK) 293 cells (passage 42-49; American Type Culture Collection, Manassas, VA) were transfected in a 0.2 cm gap cuvette via electroporation using a 110 V 25 ms pulse (Genepulser Xcell; BioRad Laboratories, Hercules, CA) with the following plasmids: 3  $\mu$ g of BirA, 3  $\mu$ g of Kv2.1-loopBAD and 3 $\mu$ g of ABP-tdEosFP. ABP is the actin-binding sequence of ABP140 from *S. cerevisiae* consisting of 17 amino acids [4]. Cells were incubated overnight in phenol red free Dulbecco's Modified Eagle's Medium (DMEM; Gibco, Life Technologies, Carlsbad, CA) with 10% fetal bovine serum (FBS; Gibco) at 37  $^{\circ}$ C and imaged 12-24 hours after transfection. Transfected cells were cultured and imaged on round, glass-bottom dishes coated with Matrigel (BD Biosciences, San Jose, CA). Prior to imaging, the medium was replaced with HEK Imaging Saline (HIS; 146 mM NaCl, 4.7 mM KCl, 2.5 mM CaCl<sub>2</sub>, 0.6 mM MgSO<sub>4</sub>, 1.6 mM NaHCO<sub>3</sub>, 0.15 mM NaH<sub>2</sub>PO<sub>4</sub>, 0.1 mM ascorbic acid, 8 mM glucose, and 20 mM HEPES, pH 7.4).

2.3.4. IMAGE ACQUISITION. Images were acquired using IQ2.3 software (Andor Technology). To maximize the speed of acquisition, the vertical height of images was restricted to 130 px, which allowed the frame rate to reach 50 Hz when the frame transfer/overlap option was enabled. Camera gain was set to 250 with a pre-amp gain of 4.9. During imaging, the amount of activated fluorophores was controlled by modifying laser power using ND filters ranging ND 0.5 to ND 4 placed in the path of the individual beams and altered during imaging through use of a filter wheel. Typically, the violet laser power was gradually increased during imaging by initiating imaging with ND 4 and decreasing the ND by 0.5 as the non-bleached fluorophore density decreased.

2.3.5. PALM RECONSTRUCTION. PALM localization was completed using MATLAB code provided by Dr. Keith Lidke's lab [14]. A CCD Gain of 36.8 was used and the mean particle localization accuracy was 35 nm on average. Our goal was to maximize the number of molecules localized by the algorithm to improve the Nyquist spatial resolution. However, different regions in a cell had varying activated probes due to factors such as uneven excitation due to the Gaussian beam profile and differing probe densities in various regions of the cell. Regions of high density of activated probes were most efficiently and accurately localized after being processed by a Gaussian filter. However, this processing caused molecules in regions of lower density to not be detected. Thus, we utilized a double pass system with the reconstruction algorithm. During the first pass, the raw images were processed by the PALM localization algorithm which localized molecules in lower density regions. During the second pass, the images were first processed with a 0.8 px Gaussian filter before being analyzed by the PALM algorithm, which allowed particles within high density regions to be localized. A side effect of this method of analysis was that the number

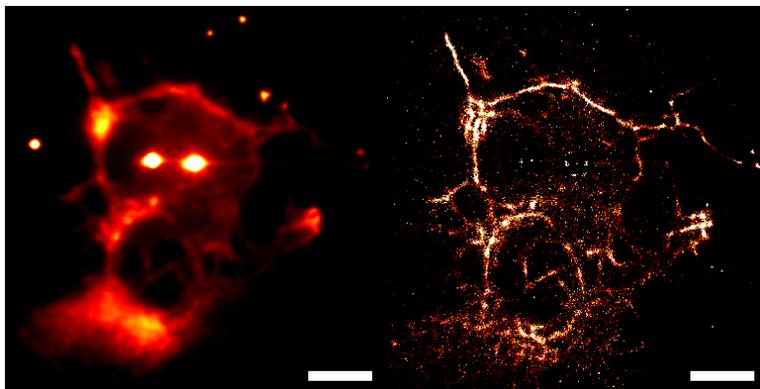


FIGURE 2.4. Summed TIRF (left) and PALM (right) image constructed from 1,200 frames with a frame rate of 10 Hz in fixed ND723 cell labeled with Dendra2-actin. The PALM image displays a higher level of detail than that of the TIRF image. Scalebar: 5  $\mu\text{m}$

of PALM localizations did not accurately reflect the number of molecules present in each frame, since some molecules could have been localized twice. However, since the analysis completed was focused on understanding the spatial relationship, structure, and dynamics of cortical actin, this was not a concern.

## 2.4. RESULTS

2.4.1. CORTICAL ACTIN IN FIXED CELLS. To verify the viability of our experimental setup to capture data for PALM images, fixed ND723 cells labeled with Dendra2-actin were imaged. Since the cells were fixed, over a thousand frames could be used to construct an individual image without introducing significant blurring effects. The PALM image (Figure 2.4) shows significant improvement of detail compared to the TIRF image, which was generated by summing all of the raw frames. Features with a full-width half maximum of as low as 60 nm were observed (Figure 2.5).

2.4.2. CORTICAL ACTIN IN LIVE CELLS. Imaging actin in fixed cells cannot reveal the dynamics of the actin cytoskeleton. To understand how cortical actin changes over time in

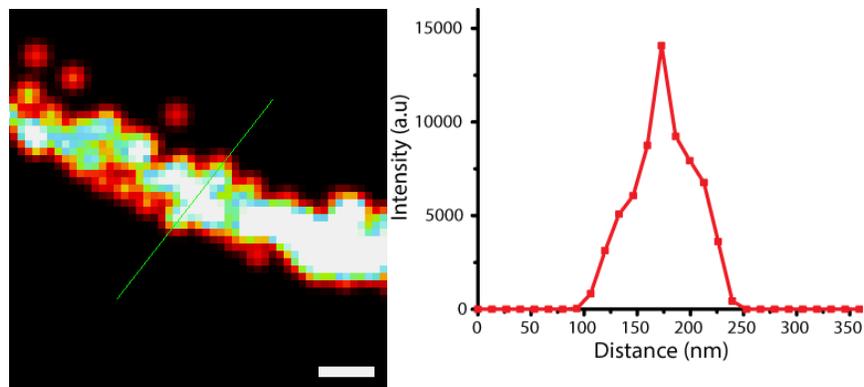


FIGURE 2.5. An actin bundle in a PALM image (left) and the associated intensity line profile (right). Intensity values for the graph were taken from the pixels indicated by the green line. Scalebar: 100 nm

a live cell, HEK293 cells were labeled with ABP-tdEosFP [4]. ABP is a protein which binds to actin filaments [15], while tdEosFP is a photoactivatable probe which switches from green to red upon activation [16]. PALM images were generated from 100 frames for a Nyquist limited spatial resolution of 53 nm, a temporal resolution of 2 s, and a sliding time window of 0.2 s. The resulting movie (Figure 2.6) reveals the actin cytoskeleton as a meshwork that undergoes changes over time that are not detectable with the summed TIRF image. Multiple compartments can be seen, some of which persist over the period of minutes, whereas others divide or expand. This illustrates the actin cytoskeleton as a highly dynamic structure.

## 2.5. DISCUSSION

The cortical actin cytoskeleton is a highly dynamic structure which requires an imaging technique with sufficient temporal and spatial resolution to detect changes in its organization. Although the achieved spatial resolution of 40 nm is not sufficient to see individual actin filaments, bundles of actin filaments which make up the cell cortex are clearly visible. The actin bundles form compartments, some of which are stable over timescales of minutes whereas others split into new compartments or merge with neighboring compartments. An

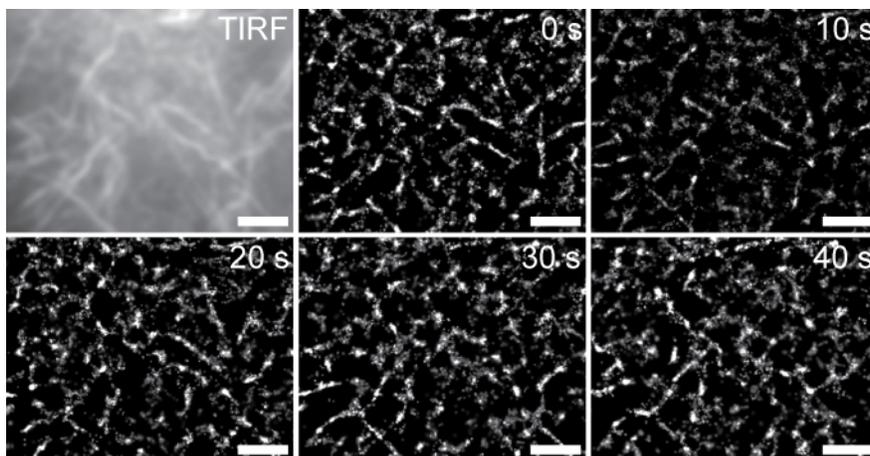


FIGURE 2.6. TIRF and PALM images of a live HEK293 cell labeled with ABP-tdEosFP. The TIRF image is the sum of 2500 frames. Each PALM image is constructed from 100 frames collected at 50 Hz of localized particles for a temporal resolution of 2 s and a Nyquist resolution of 53 nm. The PALM images reveal changes in the actin meshwork over time that are not detectable with the summed TIRF image. Scalebar: 2  $\mu\text{m}$

interesting future direction would be to determine whether compartment changes cause any organizational changes of proteins on the plasma membrane. Depending on how close the actin bundles are to the plasma membrane, it is also possible that actin acts to sterically hinder the movement of membrane proteins with a cytoplasmic domain.

This work has also succeeded in reducing the temporal resolution of the superresolution images to 2 s. Although works using another, similar technique known as STORM have achieved temporal resolutions of 0.5 s [11], 2 s is the fastest temporal resolution reported for PALM so far. Previous experiments have estimated that the time it takes for an actin filament to undergo complete turnover is approximately 5 min [12]. Thus, a temporal resolution of 2 s is sufficient to study the dynamics of actin.

Future directions could benefit by focusing on further improving the spatial and temporal resolution of the PALM images. Currently, these are limited by the stability and brightness of probes available. Development of probes with improved stability and brightness would

provide superior PALM images. Also, the development of faster, more efficient localization algorithms capable of multi-point fitting to increase the number of accurate localizations per frame would improve the spatial and temporal resolution of PALM reconstructions significantly.

## BIBLIOGRAPHY

- [1] Shroff, H., Galbraith, C. G., Galbraith, J. A., & Betzig, E. (2008). Live-cell photoactivated localization microscopy of nanoscale adhesion dynamics. *Nature Methods*, 5(5), 417-423.
- [2] Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1), 10-21.
- [3] Shroff, H., Galbraith, C. G., Galbraith, J. A., White, H., Gillette, J., Olenych, S., ... & Betzig, E. (2007). Dual-color superresolution imaging of genetically expressed probes within individual adhesion complexes. *Proceedings of the National Academy of Sciences*, 104(51), 20308-20313.
- [4] Izeddin, I., Specht, C. G., Lelek, M., Darzacq, X., Triller, A., Zimmer, C., & Dahan, M. (2011). Super-resolution dynamic imaging of dendritic spines using a low-affinity photoconvertible actin probe. *PLoS One*, 6(1), e15611.
- [5] O'Connell, K. M., Loftus, R., & Tamkun, M. M. (2010). Localization-dependent activity of the Kv2. 1 delayed-rectifier K<sup>+</sup> channel. *Proceedings of the National Academy of Sciences*, 107(27), 12351-12356.
- [6] Sako, Y., & Kusumi, A. (1995). Barriers for lateral diffusion of transferrin receptor in the plasma membrane as characterized by receptor dragging by laser tweezers: fence versus tether. *The Journal of cell biology*, 129(6), 1559-1574.
- [7] Fujiwara, T., Ritchie, K., Murakoshi, H., Jacobson, K., et al. 2002. Phospholipids undergo hop diffusion in compartmentalized cell membrane. *Journal of Cell Biology* 157(6): 1071-81
- [8] Betzig, E., Patterson, G. H., Sougrat, R., Lindwasser, O. W., Olenych, S., Bonifacino, J. S., ... & Hess, H. F. (2006). Imaging intracellular fluorescent proteins at nanometer

- resolution. *Science*, 313(5793), 1642-1645.
- [9] Hess, S. T., Girirajan, T. P., & Mason, M. D. (2006). Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophysical journal*, 91(11), 4258-4272.
- [10] Rust, M. J., Bates, M., & Zhuang, X. (2006). Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nature methods*, 3(10), 793-796.
- [11] Jones, S. A., Shim, S. H., He, J., & Zhuang, X. (2011). Fast, three-dimensional super-resolution imaging of live cells. *Nature methods*, 8(6), 499-505.
- [12] McGrath, J. L., Tardy, Y., Dewey, C. F., Meister, J. J., & Hartwig, J. H. (1998). Simultaneous measurements of actin filament turnover, filament fraction, and monomer diffusion in endothelial cells. *Biophysical journal*, 75(4), 2070-2078.
- [13] Thompson, R. E., Larson, D. R., & Webb, W. W. (2002). Precise nanometer localization analysis for individual fluorescent probes. *Biophysical journal*, 82(5), 2775-2783.
- [14] Smith, C. S., Joseph, N., Rieger, B., & Lidke, K. A. (2010). Fast, single-molecule localization that achieves theoretically minimum uncertainty. *Nature methods*, 7(5), 373-375.
- [15] Riedl, J., Crevenna, A. H., Kessenbrock, K., Yu, J. H., Neukirchen, D., Bista, M., ... & Wedlich-Soldner, R. (2008). Lifeact: a versatile marker to visualize F-actin. *Nature methods*, 5(7), 605-607.
- [16] Wiedenmann, J., Ivanchenko, S., Oswald, F., Schmitt, F., Röcker, C., Salih, A., ... & Nienhaus, G. U. (2004). EosFP, a fluorescent marker protein with UV-inducible green-to-red fluorescence conversion. *Proceedings of the National Academy of Sciences of the United States of America*, 101(45), 15905-15910.

- [17] Engqvist-Goldstein, Å. E., & Drubin, D. G. (2003). Actin assembly and endocytosis: from yeast to mammals. *Annual review of cell and developmental biology*, 19(1), 287-332.

## CHAPTER 3

# INTERACTIONS BETWEEN MEMBRANE PROTEINS AND THE ACTIN CYTOSKELETON

### 3.1. INTRODUCTION

3.1.1. THE LANDSCAPE OF THE PLASMA MEMBRANE. The plasma membrane is a busy place. This heterogenous system is packed with a variety of lipids and proteins, each of which have a specific role in keeping the cell functioning properly. Acting as the point of contact for the outside world, the plasma membrane is crucial for cellular signaling events, as well as endocytosis and exocytosis, which allows the cell to pass materials between its interior and exterior. In addition, the plasma membrane is home to a special class of membrane proteins which act as channels for ions. These ion channels are key components in allowing neurons to propagate signals.

Previous research has provided evidence that the components of the plasma membrane may not be randomly placed, but instead may be organized into specific domains. In yeast cells, lipid and protein domains form due to a variety of mechanisms, from protein and lipid interactions to cytoskeletal interactions [4]. In mammalian cells, cholesterol-enriched domains of sphingolipids have been observed to form clusters on the plasma membrane [1]. There is also evidence which suggests that membrane compartmentalization may be due to steric interactions with the cell cortex [5], however this has not yet been observed directly. A recent study used PALM to show that the organization of hemagglutinin, a membrane protein, is dependent on actin [3], although the underlying mechanism was not discovered. Proper

membrane organization is crucial to effective cellular signaling and thus understanding the mechanism behind this phenomena will reveal key insights into how cells communicate.

Several theories exist as to how the cellular cortex may facilitate the organization of the plasma membrane. The cortical cytoskeleton may interact sterically with membrane proteins, acting as a corral which prevents proteins from freely diffusing across the membrane [5, 10]. Another theory proposes that instead of a continuous barrier, a picket fence created by transmembrane proteins anchored to the actin cytoskeleton may limit the diffusion of phospholipids and membrane proteins [11].

Distinguishing amongst these possible schema has been hindered by the difficulties in observing the interactions directly. Actin filaments which make up the cellular cortex are orders of magnitude smaller than the diffraction limit, making traditional fluorescence microscopy insufficient to observe the dynamics of the relationship between membrane proteins and actin. However, single particle tracking techniques as well as recent advances in super-resolution imaging may hold the key to overcoming some of these challenges.

3.1.2. KV2.1 AND KV1.4 CHARACTERISTICS. Voltage-gated potassium channels are essential for neuronal communication, allowing electrical signals to propagate through the neuron. In particular, Kv2.1 is of interest because it forms micron-sized, stable clusters on the neuron soma [6, 7, 9]. Furthermore, it has been discovered that whether or not a Kv2.1 channel is clustered or free is linked to its function, because clustered Kv2.1 channels do not conduct ions [2]. Since clusters of Kv2.1 have been observed to occupy actin-deficient regions, the actin cytoskeleton may be involved in cluster maintenance [10].

In contrast, Kv1.4 is a voltage-gated potassium channel that is both mobile and homogeneous on the plasma membrane [12]. Since Kv1.4 does not form clusters, its interactions

with the cytoskeleton may be different from Kv2.1. If the actin cytoskeleton is responsible for cluster forming and maintenance of Kv2.1, Kv2.1 may be more strongly affected by the cellular cortex than Kv1.4. Thus, understanding how Kv2.1 and Kv1.4 interact with the cortical actin cytoskeleton could not only reveal information about the mechanism behind the clustering properties of Kv2.1, but also provide more general insights into how membrane proteins interact with actin.

3.1.3. SINGLE PARTICLE TRACKING. Single particle tracking (SPT) is a technique which allows the location of an individual molecule to be determined over extended periods of time. Since we want to characterize the interactions of individual proteins with the cytoskeleton, SPT of Kv2.1 and Kv1.4 allow those dynamics to be explored with nanometer resolution. Single particle tracking works by using the emitted light from a particle to find the most likely subpixel location of the molecule, often through fitting of a 2D Gaussian function. Then, particles localized in subsequent frames are linked together so that the movement of an individual particle can be traced over time. We have used this technique, combined with PALM images of actin, to reveal interactions between voltage gated potassium channels and the cortical actin cytoskeleton.

## 3.2. METHODS

3.2.1. VISUALIZING KV2.1 AND KV1.4. In order to visualize individual membrane channels such as Kv2.1 and Kv1.4, streptavidin conjugated Qdot 705 (Invitrogen, Carlsbad, CA) was used. QD705 has an emission spectrum distinct from both the inactive and active spectra of the actin probe, which makes it suitable for dual-color imaging. Both the expressed Kv2.1 and Kv1.4 constructs contained an engineered extracellular biotin acceptor domain. 1  $\mu$ g of the BirA plasmid, encoding the biotin ligase, was included in the transfection to

produce the biotinylated protein. Immediately prior to imaging, the medium in the cell dish was replaced with 1 nM QD705 in HEK imaging saline (HIS; 146 mM NaCl, 4.7 mM KCl, 2.5 mM CaCl<sub>2</sub>, 0.6 mM MgSO<sub>4</sub>, 1.6 mM NaHCO<sub>3</sub>, 0.15 mM NaH<sub>2</sub>PO<sub>4</sub>, 0.1 mM ascorbic acid, 8 mM glucose, and 20 mM HEPES, pH 7.4) containing 10 mg/mL BSA. Once this solution was incubated for 5-10 min, the cell dish was rinsed three times with HIS before being filled with 1 mL of HIS. For Kv2.1 cluster imaging, cells were stained with 1 µg/mL AlexaFluor 660 (Invitrogen, Carlsbad, CA) in HEK imaging saline containing 10 mg/mL BSA for 10 min instead of with the QD705 solution.

3.2.2. DUAL COLOR IMAGING. A Cairn Optosplit II (Cairn Research, Kent, England) was placed in the emission path of the of microscope, allowing the emissions of the ABP-tEosFP and the QD705 to be separated. In its active state, tdEosFP emits light at 590 nm, while QD705 emits light at 705 nm. A 655 nm dichroic was used to separate the two spectra by transmitting the QD705 light through to one side of the camera and reflecting the tdEosFP emission to the other side of the camera.

3.2.3. COMBINING CHANNELS. To determine the offset between the left channel (QD705) in the image and the right channel (tdEosFP) in the image, a white light image of polystyrene beads (1 µm; Polysciences, Inc., Warrington, PA) was taken prior to imaging cells. The bead dish was prepared by placing 100 µL of a 1:1000 dilution of the bead stock onto an imaging dish. After incubating at room temperature for 15 minutes, the dish was rinsed with isopropyl alcohol and dried with nitrogen. When the microscope focal point was placed slightly below the beads, a concentric pattern of diffraction rings appeared around the beads. This allowed the location of the beads to be determined with a localization accuracy of up to 8 nm through a custom autocorrelation algorithm written in LabView (National Instruments,

Austin, TX). Since each bead appeared in both sides of the image, once all of the beads were localized, the offset was determined by subtracting the location of one bead from one side of the image from the location of the same bead on the other side of the image. Once the offsets were determined, the two channels were overlaid precisely by subtracting the offsets from the location of each actin molecule, resulting in a super-resolution image with a region of interest that matched with the membrane protein image.

This process revealed an astigmatism induced by the dichroic used to split the channels. As the horizontal location of the bead increased, the offset between pairs of beads decreased linearly, which caused inaccuracies in the overlay of up to 140 nm when the average x offset was used. To address this problem, a linear fit was completed (Figure 3.1) and the astigmatism was corrected for by modifying the x location of each localized actin molecule based on the fit. The correction increased the accuracy of the the actin and membrane channel overlay by up to 140 nm.

PALM images were generated using the methods introduced in Chapter 2. Single particle tracking was completed using u-track [14] in MATLAB (MathWorks, Natick, MA).

### 3.3. RESULTS

3.3.1. KV2.1 CLUSTERS AND CORTICAL ACTIN. Previous experiments have revealed that upon application of Latrunculin A (LatA), an actin affecting drug which inhibits filament polymerization by sequestering free actin monomers [13], the size of Kv2.1 clusters increases [9]. This provides evidence that actin has a role in the clustering behavior of Kv2.1. We directly observed the effect of addition of 100 nM LatA on the actin meshwork in relation to Kv2.1 clusters. Kv2.1 channels were stained with streptavidin conjugated AlexaFluor 660 (AF660) through biotin-streptavidin linkages. Then, a snapshot of the Kv2.1 clusters was

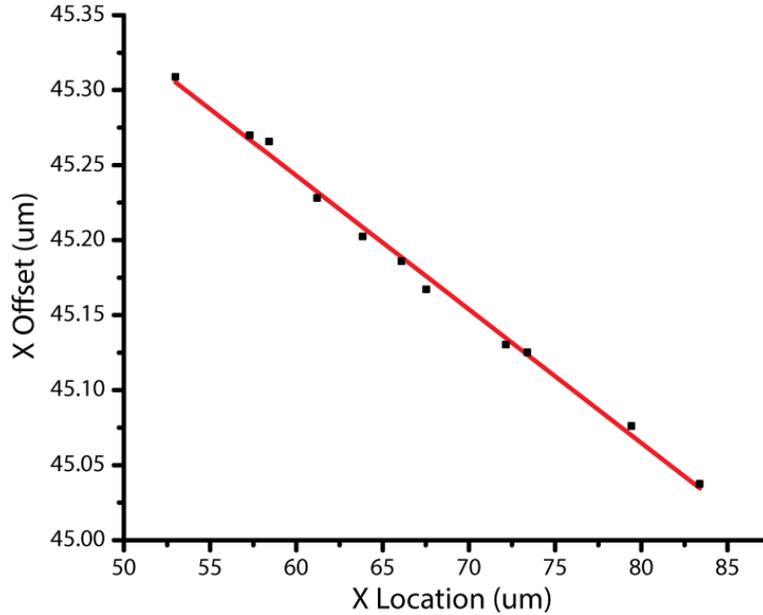


FIGURE 3.1. Astigmatism of the dichroic used to split the probe spectrums demonstrated by the decreasing x offsets with increasing x position. The offsets determined from the white light bead image are the black scatter points. The linear fit, represented as a red line, is used to determine the corrected x location of each actin molecule when the channels are overlaid.

taken, followed by PALM imaging of the ABP-tdEosFP. Prior to application of Lat A, the location of the clusters and actin appear anticorrelated, with actin bundles surrounding the clusters, suggesting that actin may act as a barrier which prevents clustered Kv2.1 diffusion (Top row; Figure 3.2). This result is consistent with previous diffraction-limited imaging results which indicated Kv2.1 clusters formed in areas of reduced actin concentration [10]. However, upon application of 100 nM LatA, the actin meshwork becomes markedly less dense and the Kv2.1 clusters do not appear bounded by the actin bundles, even though the clusters become larger and more diffuse (Bottom row; Figure 3.2).

3.3.2. KV1.4 AND KV2.1 INTERACTIONS WITH ACTIN. The previous experiment demonstrated that Kv2.1 is generally localized to actin deficient regions. However, the experiment only captured the behavior of ensemble groups of Kv2.1 channels. In order to understand

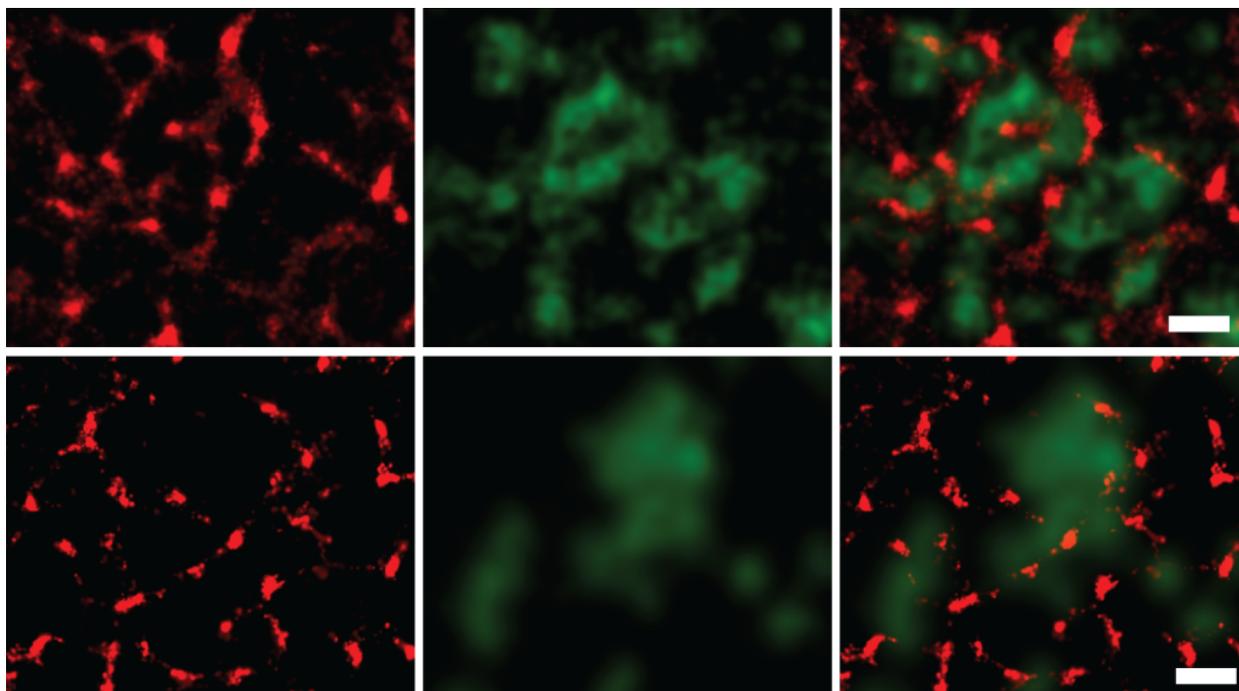


FIGURE 3.2. Kv2.1 clusters stained with streptavidin conjugated AlexaFluor 660 (green) and PALM of ABP-tdEosFP which binds actin filaments (red). PALM images are constructed from 5000 frames taken over a period of 100 s. The top row of images was taken prior to application of LatA and shows Kv2.1 clusters localized to areas of low actin density. The bottom row of images were taken 40 min after application of 100 nM LatA. Scalebar: 1  $\mu$ m

the mechanism behind actin interaction with Kv2.1, it is important to understand the behavior of individual channels. Kv1.4 is also a voltage-gated potassium channel which does not form clusters on the cell surface. Since Kv2.1 channels form clusters, we suspected that the interactions of Kv1.4 and Kv2.1 with the actin cytoskeleton may be different.

To test this hypothesis, we imaged cells transfected with BirA, ABP-tdEosFP, and either Kv2.1-loopBAD or Kv1.4-loopBAD. The Kv channels were labeled with a low density of QD705, allowing individual channels to be tracked. The tracks were combined with PALM reconstructions to study the interactions of individual channels with the cytoskeleton. Actin was commonly observed to act as a barrier to the diffusion of both Kv2.1 and Kv1.4 (Figure 3.3A and D), resulting in channels changing direction on contact. However, the cytoskeleton

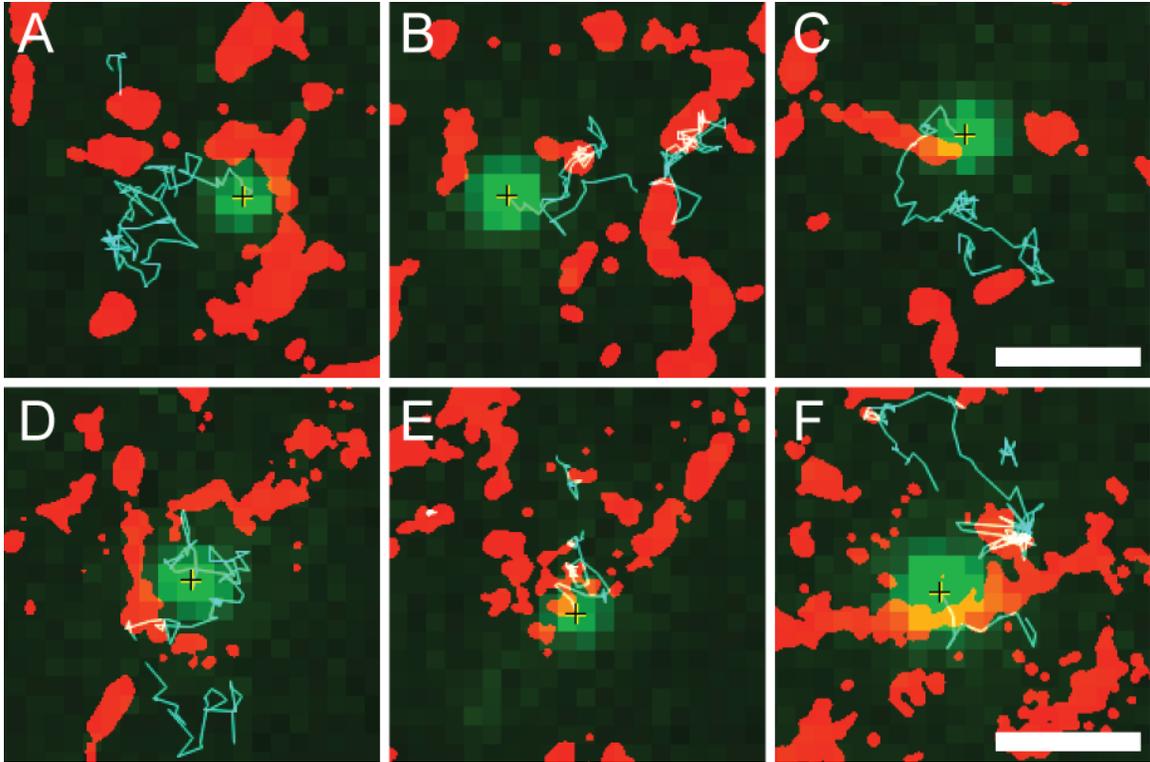


FIGURE 3.3. Trajectories of individual Kv1.4 (A-C) and Kv2.1 (D-F) channels combined with PALM images of actin (2 s frames, 42 nm resolution). Kv channel images are shown in green, thresholded actin images in red, and Kv tracks in cyan. Three different behaviors are demonstrated: channels reflecting off of actin (A, D), channels diffusing within an actin bundle (B, E), and channels diffusing through an actin bundle (C, F). Scalebar: 1  $\mu\text{m}$

was also observed to behave as a container (Figure 3.3B and E), causing channels to diffuse within actin bundles for periods of time before escaping. Channels also occasionally appear to ignore actin (Figure 3.3C and F), crossing large bundles, though the same bundles they cross frequently disappear shortly after (Figure 3.4), suggesting the actin may have allowed passage by pulling away from the plasma membrane. Thus, both Kv channels frequently treated actin as a barrier to diffusion. In addition, no significant difference between the interactions of Kv2.1 and Kv1.4 with the actin cytoskeleton were observed.

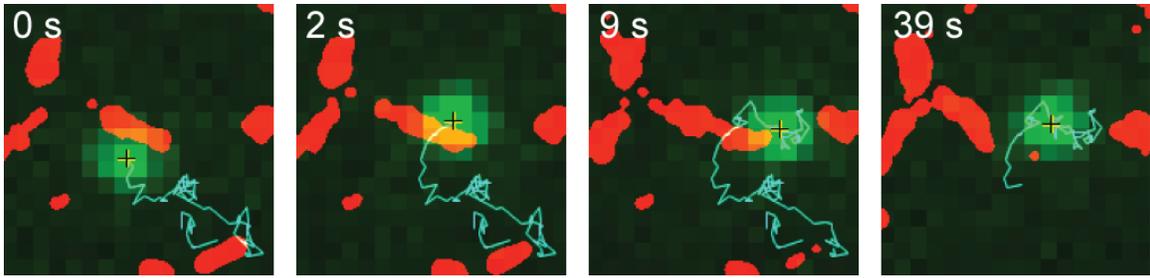


FIGURE 3.4. A Kv1.4 channel that crosses an actin barrier. Approximately 7 seconds after crossing actin, the actin bundle noticeably thins. At 37 seconds after crossing actin, the actin bundle has disappeared from the field of view completely.

### 3.4. DISCUSSION

Imaging Kv2.1 clusters combined with superresolved actin gave insight into the ensemble distribution of the channels. Previous experiments have demonstrated that the application of LatA leads to the formation of large clusters [9]. We found that the clusters tended to occupy actin-deficient regions. However, actin does not completely bound the Kv2.1 clusters on all sides, suggesting that actin may form an incomplete corral. Since the Kv2.1 clusters are excluded from regions of actin enrichment, it is possible that the cytoskeleton acts as a steric hindrance to either the Kv2.1 channels themselves or to whatever cluster maintenance mechanism is involved. With the application of the actin affecting drug LatA, the anti-correlated relationship of actin and Kv2.1 appears to break down. This suggests that actin polymers play a crucial role in organizing the Kv2.1 clusters and with the addition of LatA becomes a less effective barrier to channel diffusion.

We have used single particle tracking combined with PALM imaging to observe the interactions of membrane proteins with actin below the diffraction limit. Both Kv2.1 and Kv1.4 interact with actin in ways that fit into several different categories of behavior:

(a) Actin acts as a barrier. The actin cytoskeleton often acts as a steric hindrance for both types of channel, causing the channels to change direction after collision. Since both

Kv2.1 and Kv1.4 have a cytoplasmic domain, it is likely this domain is colliding with portions of the cell cortex that are in close proximity with the plasma membrane.

(b) Actin acts as a container. In the container case, the channel likely enters the actin meshwork via a pathway where the actin is further away from the membrane and then encounters obstacles which prevent its diffusion into actin-free space due to filaments which block its escape. This causes the channel to be trapped within the actin meshwork until it finds a region where the actin is not as close to the membrane and can diffuse out.

(c) Actin is ignored. The case where the channel appears to completely ignore actin is likely a situation where actin is not in close proximity to the plasma membrane. This is further supported by the instances where after a channel diffuses across an actin bundle, it often disappears from the field of view shortly after, likely due to pulling away from the plasma membrane.

These observations are most consistent with the model where the actin cytoskeleton behaves as a barrier to membrane protein diffusion. Even though Kv1.4 does not form clusters, it interacts with the actin cytoskeleton in similar ways as Kv2.1. Kv2.1 channels are present in two populations in the cell: clustered and free. The single particle tracking experiments presented here did not provide a clear method of distinguishing between free and clustered Kv2.1 channels. However, it is likely that the majority of the tracked Kv2.1 channels observed to interact with actin were free channels since they were very mobile. Thus, although we have elucidated the interactions of the cell cortex with Kv2.1 and Kv1.4, we have not discovered what mechanism is responsible for the clustering of Kv2.1.

A future direction would be to apply 3D imaging techniques to image acquisition. Currently, it is conjecture whether the reason the membrane channels cross or get captured

within actin bundles is due to the distance of the cytoskeleton from the plasma membrane in those regions. Imaging in 3D would allow for more insight into whether the membrane channels are crossing actin due to its distance from the plasma membrane or because of some other mechanism.

## BIBLIOGRAPHY

- [1] Mizuno, H., Abe, M., Dedecker, P., Makino, A., Rocha, S., Ohno-Iwashita, Y., ... & Miyawaki, A. (2011). Fluorescent probes for superresolution imaging of lipid domains on the plasma membrane. *Chemical Science*, 2(8), 1548-1553.
- [2] O'Connell, K. M., Loftus, R., & Tamkun, M. M. (2010). Localization-dependent activity of the Kv2. 1 delayed-rectifier K<sup>+</sup> channel. *Proceedings of the National Academy of Sciences*, 107(27), 12351-12356.
- [3] Gudheti, M. V., Curthoys, N. M., Gould, T. J., Kim, D., ... & Hess, S. T. (2013). Actin Mediates the Nanoscale Membrane Organization of the Clustered Membrane Protein Influenza Hemagglutinin. *Biophysical journal*, 104(10), 2182-2192.
- [4] Ziółkowska, N. E., Christiano, R., & Walther, T. C. (2012). Organized living: formation mechanisms and functions of plasma membrane domains in yeast. *Trends in cell biology*, 22(3), 151-158.
- [5] Sako, Y., & Kusumi, A. (1995). Barriers for lateral diffusion of transferrin receptor in the plasma membrane as characterized by receptor dragging by laser tweezers: fence versus tether. *The Journal of cell biology*, 129(6), 1559-1574.
- [6] Lim, S. T., Antonucci, D. E., Scannevin, R. H., & Trimmer, J. S. (2000). A Novel Targeting Signal for Proximal Clustering of the Kv2. 1 K<sup>+</sup> Channel in Hippocampal Neurons. *Neuron*, 25(2), 385-397.
- [7] Misonou, H., Mohapatra, D. P., Park, E. W., Leung, V., Zhen, D., Misonou, K., ... & Trimmer, J. S. (2004). Regulation of ion channel localization and phosphorylation by neuronal activity. *Nature neuroscience*, 7(7), 711-718.
- [8] Misonou, H., Mohapatra, D. P., Menegola, M., & Trimmer, J. S. (2005). Calcium- and metabolic state-dependent modulation of the voltage-dependent Kv2. 1 channel

- regulates neuronal excitability in response to ischemia. *The Journal of neuroscience*, 25(48), 11184-11193.
- [9] O'Connell, K. M., Rolig, A. S., Whitesell, J. D., & Tamkun, M. M. (2006). Kv2. 1 potassium channels are retained within dynamic cell surface microdomains that are defined by a perimeter fence. *The Journal of neuroscience*, 26(38), 9609-9618.
- [10] Tamkun, M. M., O'Connell, K. M., & Rolig, A. S. (2007). A cytoskeletal-based perimeter fence selectively corrals a sub-population of cell surface Kv2. 1 channels. *Journal of Cell Science*, 120(14), 2413-2423.
- [11] Fujiwara, T., Ritchie, K., Murakoshi, H., Jacobson, K., & Kusumi, A. (2002). Phospholipids undergo hop diffusion in compartmentalized cell membrane. *The Journal of cell biology*, 157(6), 1071-1082.
- [12] O'Connell, K. M., & Tamkun, M. M. (2005). Targeting of voltage-gated potassium channel isoforms to distinct cell surface microdomains. *Journal of cell science*, 118(10), 2155-2166.
- [13] Coué, M., Brenner, S. L., Spector, I., & Korn, E. D. (1987). Inhibition of actin polymerization by latrunculin A. *FEBS letters*, 213(2), 316-318.
- [14] Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S. L., & Danuser, G. (2008). Robust single-particle tracking in live-cell time-lapse sequences. *Nature methods*, 5(8), 695-702.

## CHAPTER 4

# ACTIN COMPARTMENTALIZATION OF MEMBRANE PROTEINS

### 4.1. INTRODUCTION

One of the greatest challenges in microscopy is extracting meaningful data from acquired images. As discussed in previous chapters, this challenge is partially due to the restrictions imposed by the diffraction limit. By exceeding the diffraction limit, features that are closer together than the typical 250 nm restriction can be resolved, allowing information about processes that occur on the order of tens of nanometers to be recorded. However, viewing these images provides mostly qualitative data. Observations become more meaningful when supported by statistical analysis and thus it becomes essential to quantitatively assess the gathered information.

Images contain an immense amount of information, but accurate and efficient extraction of this information is often difficult. The field of image processing has provided multiple ways to elucidate and highlight characteristics of images. Algorithms have been developed which can effectively eliminate noise, detect objects, determine distances, and compartmentalize images [6].

In particular, the watershed algorithm[8] has already found uses in many areas of biology [9–11]. The basic watershed principle can be explained by considering an image to be a topographic region of valleys and hills. First flood the region, filling the valleys. As the valleys overflow, the water will follow a path along the nearest minima. Watershed effectively follows the same procedure, creating lines that follow the path of the water. The result is an algorithm that is extremely effective at segmenting an image based on the available intensity

information. Here, we use the watershed algorithm to generate complete boundaries of cytoskeletal compartments.

Another useful algorithm is Euclidean distance mapping (EDM) [6]. EDMs are generated from a binary image, where the features of interest are set to “off” and the background pixels are set to “on”. The algorithm creates a new image based on the binary image where each pixel in the new image has a pixel value that corresponds to the Euclidean distance of that pixel from the nearest feature. Thus, a pixel that neighbors a feature of interest is assigned a pixel value of 1, a pixel neighboring pixels with value 1 are assigned a value of 2, and so on. EDMs are helpful in analysis of interactions between multiple structures in cells because they provide a way to determine numerical information on the distance of separation of molecules. Analysis of this information can lead to useful insights into cellular systems [7].

In the data presented in Chapter 2 and Chapter 3, the observations made about membrane channels in relation to the actin cytoskeleton were largely qualitative. Here, we take another look at the same data, using watershed and distance mapping to extract meaningful information encoded within the 2D matrices of images.

## 4.2. METHODS

4.2.1. WATERSHED. Watershed images were generated from PALM images of the actin cytoskeleton which were described previously in Chapter 2. First, images were segmented via thresholding and converted to binary in ImageJ [2]. Watershed images were generated from the binary images using functions from the image processing toolbox in MATLAB (MathWorks, Natick, MA). This resulted in images containing a best approximation of complete compartment outlines based on the available actin cytoskeleton information.

4.2.2. COMPARTMENT DETAILS AND TRACKING. Once watershed images were generated, areas were found using functions in MATLAB which detect compartments (`bwconncomp`) and provide information about individual sections of a compartmentalized image (`regionprops`). To track compartments, a custom algorithm written in MATLAB (MathWorks) first located compartment centroids before finding the nearest neighboring centroid in the next frame to connect compartments across multiple frames. To detect what percentage of channels crossed actin when changing compartments, a custom algorithm was written in MATLAB which detected the compartment occupied by a channel for each time point. On a compartment change, the location at which the particle crossed the compartment boundary was checked to see if actin was present. As a result, the percentage of compartment changes that occurred across actin were found. To ensure changes observed were complete compartment changes and not changes due to Brownian motion on the compartment boundary or noise in the watershed, only compartment changes that persisted for a certain number of frames were considered.

4.2.3. EUCLIDEAN DISTANCE MAPPING. Euclidean distance maps (EDMs) were generated in ImageJ [2] or MATLAB (MathWorks) from thresholded PALM images of the actin cytoskeleton. Images were inverted such that actin features became “off” pixels and background pixels became “on”. Then, the distance mapping function was completed, resulting in an image with pixel values that represented the distance of that pixel from the nearest actin structure. Once the EDMs were generated, all pixels within the region of interest (ROI) selected for analysis were used to form the control distribution. Specific protein distances from actin were found by collecting the EDM value at the sub-pixel localization as found in u-track [3].

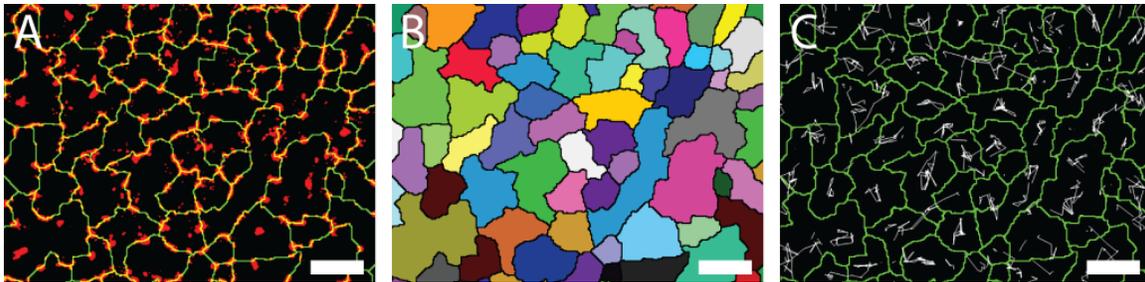


FIGURE 4.1. Actin cytoskeleton compartments as determined by watershed segmentation. PALM image of the actin cytoskeleton (red) with compartments completed by watershed segmentation (green) (A). The pixels belonging to each distinct compartment could then be found (B) which allowed the center of mass to be determined. Trajectories (gray) of the centroid location of each compartment across multiple frames were then found (C). Scalebar: 1  $\mu\text{m}$

### 4.3. RESULTS

4.3.1. ACTIN IS A PERMEABLE BARRIER TO MEMBRANE PROTEIN DIFFUSION. If actin acts as a barrier for membrane proteins such as Kv2.1 and Kv1.4, they should move across gaps in the actin cytoskeleton when changing compartments. Testing this hypothesis required the compartments formed by the actin cytoskeleton to be detected. To accomplish this, cells transfected with Kv2.1 and ABP-tdEosFP were imaged as described in Chapter 3. However, frequently in the superresolution images generated of the actin cytoskeleton compartments, boundaries were incomplete due to gaps in the actin cytoskeleton. Watershed segmentation[8], an algorithm that segments an image based on existing features, offered a way to complete the outline of the compartments in an image (Figure 4.1A). The enclosed areas represent the most likely shape and size of the compartments based on the actin data available in the image. Each distinct compartment was then identified (Figure 4.1B), its centroid determined in each frame, and the compartments were tracked over time (Figure 4.1C) by their centroid locations.

Once the space occupied by an individual compartment at each time point was found, each Kv channel could be assigned a compartment based on the channel location. When a channel changed compartments, our algorithm checked whether or not the location at which the channel crossed the compartment boundary was occupied by actin. Compartment boundaries derived directly from the actin cytoskeleton are fairly stable. However, boundaries derived from watershed are sensitive to noise, which can cause compartments to flicker or new compartments to appear and immediately disappear in the next frame. This caused the registration of erroneous compartment change events that were due to watershed variations and not cytoskeletal or membrane protein changes. To correct for this effect, the algorithm was designed to only count a compartment change event when the membrane channel remained within the new compartment for a certain minimum residence time.

The normalized crossed actin fraction was generated by dividing the percentage of compartment changes in which the particle crossed an actin barrier by the total fraction of the compartment boundary that was made up of the actin cytoskeleton (in other words, where the red and green overlap in Figure 4.1A). Figure 4.2A demonstrates the effect of various minimum residence time thresholds on the normalized crossed actin fraction for individual cells and also indicates that Kv2.1 and Kv1.4 cross actin less frequently on a compartment change than the random walk simulation. Normal 2D random walk trajectories were generated (Compartment Change N=17008) using an off-lattice Monte Carlo simulation of a Gaussian 2D random walk with a standard deviation similar to the Kv1.4 step size standard deviation, which is larger than the step size deviation of Kv2.1. With no minimum residence time requirement, both Kv1.4 and Kv2.1 crossed actin fractions are fairly close to the 2D random walk situation. As the minimum residence time increases, the Kv2.1 (Cell

N=5, Compartment Change N=4771) crossed actin fraction becomes significantly smaller than the random walk simulation, indicating compartmentalization is influenced by the actin cytoskeleton. Kv1.4 (Cell N=5, Compartment Change N=4145) also crosses actin less often on a compartment change than the control random walk situation, but more frequently than Kv2.1. Thus, actin appears to be a less effective barrier for Kv1.4 than for Kv2.1.

The residence times (Figure 4.2B) are highly dependent on the minimum residence time threshold that was set. As the minimum residence time is increased, the shorter residence times are excluded from the analysis, thus increasing the residence time for the 2D Random Walk, Kv2.1, and Kv1.4 cases. Similar to what was seen in Figure 4.2A, with no minimum residence time requirement, Kv2.1 and Kv1.4 residence times are similar to the random walk. However, as the minimum residence time increases so that erroneous compartment changes are excluded, Kv2.1 and Kv1.4 both have longer residence times than the random walk situation.

4.3.2. LATA AND SWINA EFFECTS ON KV2.1 SPATIAL DISTRIBUTION RELATIVE TO ACTIN. If Kv2.1 channels were tethered to the actin cytoskeleton in some way, they would likely spend more of their time near actin. Thus, we were interested in discovering the distribution of channel distances from actin. Euclidean distance mapping (EDM) is an algorithm that generates an image where the value of each pixel represents the distance of that pixel from the nearest feature. In our case, an EDM was generated from a binary image of the actin cytoskeleton. The random case was generated by sampling all EDM values within the region of interest, which gave the distribution of a molecule randomly positioned on the cell surface, which acted as the control case. To find the distribution of the distances of Kv2.1 channels from actin, the EDM values at each Kv2.1 channel location were found

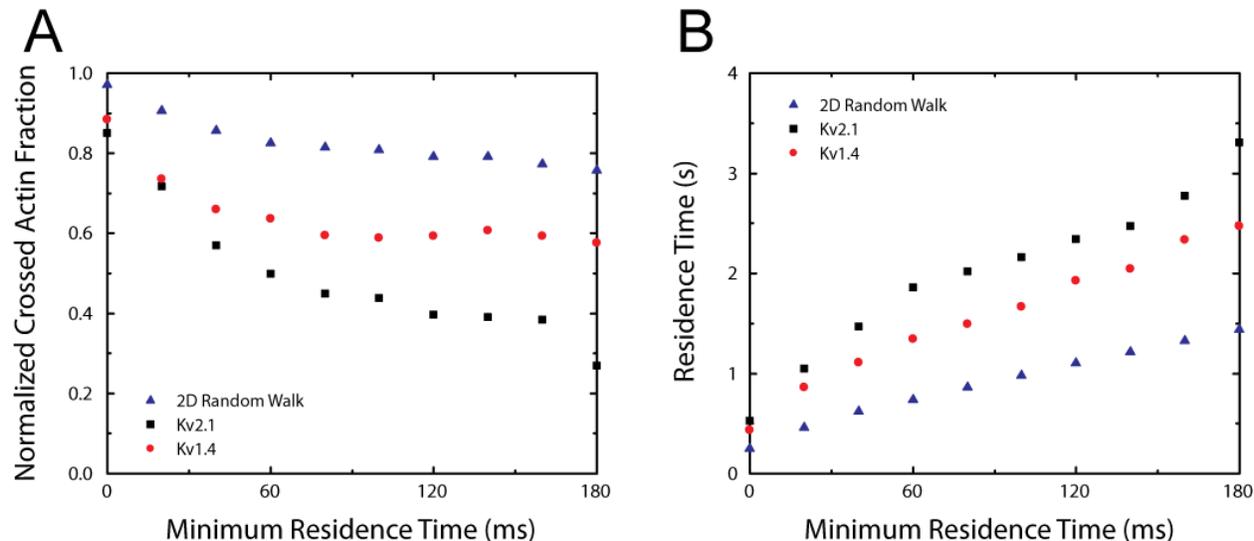


FIGURE 4.2. Compartment change analysis for various minimum residence times. The minimum residence time indicates how long the algorithm required a particle to stay in the new compartment before triggering a compartment change event. Data for the random walk (blue) was generated using a normal 2D random walk with a standard deviation similar to the Kv2.1 step size standard deviation. Data points were generated for Kv2.1 and Kv1.4 by taking the average on a cell-by-cell basis of the mean values for normalized crossed actin fractions and residence times, respectively. (A) The normalized fractions were determined by dividing the crossed actin on compartment change fraction by the mean actin fraction of the compartment boundary. Kv2.1 (black) is significantly less likely to cross an actin boundary on a compartment change than the random walk situation. Kv1.4 (red) displays the same tendency, but to a lesser extent. (B) The residence times were determined by dividing the frame rate of 20 ms by the fraction of all trajectory segments that changed compartments. The 2D Random Walk situation has the smallest residence time. Kv1.4 and Kv2.1 are distinct from the random walk and have longer residence times.

(cell  $N=3$ ). Kv2.1 channels are most likely to be found within 40 nm of an actin feature and show less preference of being within 40-244 nm of actin than the control case (Top row; Figure 4.3).

Latrunculin A (LatA) works by sequestering actin monomers, which inhibits actin filament polymerization [12] and has been shown to cause an increase in Kv2.1 cluster size [4]. Thus, understanding how the actin cytoskeleton may rearrange to allow for such size

increases was of interest. After the application of 100 nM LatA (10-30 minutes incubation), the Kv2.1 and control cases converge (Middle row; Figure 4.3), indicating a breakdown in the organization of Kv2.1 (cell N=3).

Swinholide A (SwinA) is also an actin disrupting drug, although it is more aggressive than LatA because it severs actin filaments as well as sequesters actin dimers [13]. However, at 75 nM concentration, SwinA has a similar effect on increasing Kv2.1 cluster size as 100 nM LatA [5]. In contrast to LatA, the application of SwinA (cell N=3; 10-30 minutes incubation) exaggerated the difference in the distance distribution of Kv2.1 channels to actin versus lipids (Figure 4.3).

4.3.3. SWINA AND LATA EFFECTS ON ACTIN COMPARTMENT SIZE. To further explore the effect of actin affecting drugs on actin structure, the area of each compartment as found by watershed was calculated for control cells, cells affected by 100 nM LatA, and cells affected by 75 nM SwinA. The mean compartment area for control cells (N=10) was  $2.4 \pm 0.3 \mu\text{m}^2$ . The mean compartment area of  $2.6 \pm 0.4 \mu\text{m}^2$  for LatA cells (N=10) was not significantly different from the control case ( $p = 0.11$ ). SwinA cells (N=10) had a mean compartment area of  $4.1 \pm 0.8 \mu\text{m}^2$  which was significantly different from the control case ( $p < 0.0005$ ). Thus, LatA does not have a strong effect on compartment areas of the actin cytoskeleton while SwinA has a dramatic effect, nearly doubling the mean compartment area.

4.3.4. SWINA INCREASES COMPARTMENT SIZE OVER TIME. SwinA causes a large increase in compartment area over the control case. To study the dynamics of this transition, a time lapse set of PALM data was gathered in Kv2.1 transfected cells, allowing the same cell to be observed every minute over a period of 10 minutes, beginning 1 minute after the application of 75 nM SwinA. During that time, the average compartment area (Figure 4.5)

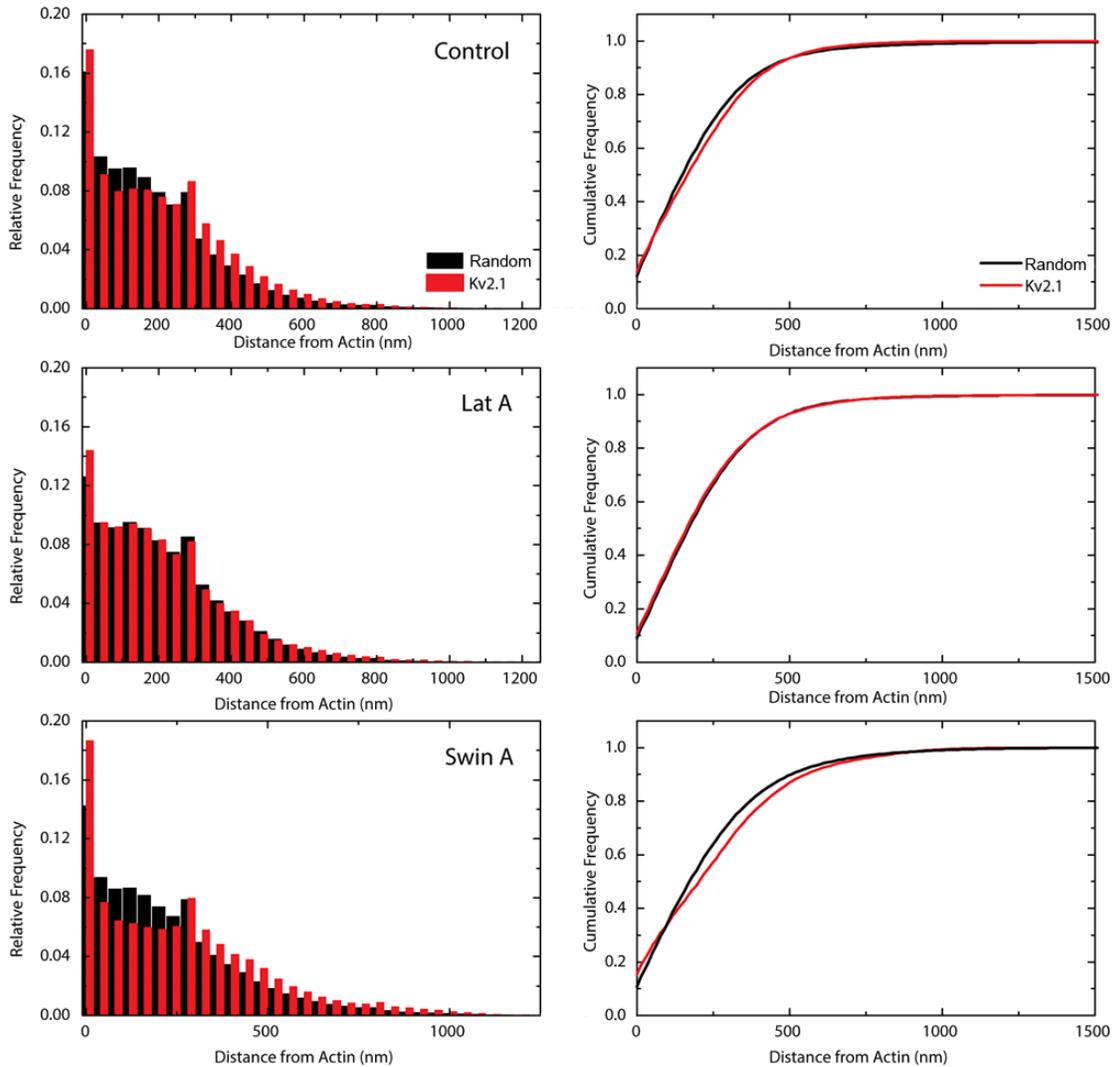


FIGURE 4.3. Distributions of distances of Kv2.1 from the actin cytoskeleton (red) plotted against the distribution of random locations (black) as the control case. The relative frequency (RF) and cumulative distribution function (CDF) are plotted for the control (pre-drug, cell N=3), 100 nM LatA (cell N=3), and 75 nM SwinA (cell N=3) cases. The intersection point for the pre-drug case (Top row) is at 244 nm, indicating that closer than that distance it is less likely to find a Kv2.1 channel compared to the control case. The RF and CDF for the post LatA case (Middle row) for the lipid and Kv2.1 channels are similar, indicating a breakdown in the organization of Kv2.1. However, the SwinA case (Bottom row) exaggerates the difference between the lipid and the Kv2.1 channel cases.

increased from  $2.2 \mu\text{m}^2$  at 1 minute after SwinA application to  $5 \mu\text{m}^2$  after 10 minutes. Although the application of SwinA and LatA have similar effects on Kv2.1 cluster size, their effect on the compartments of the actin cytoskeleton are different.

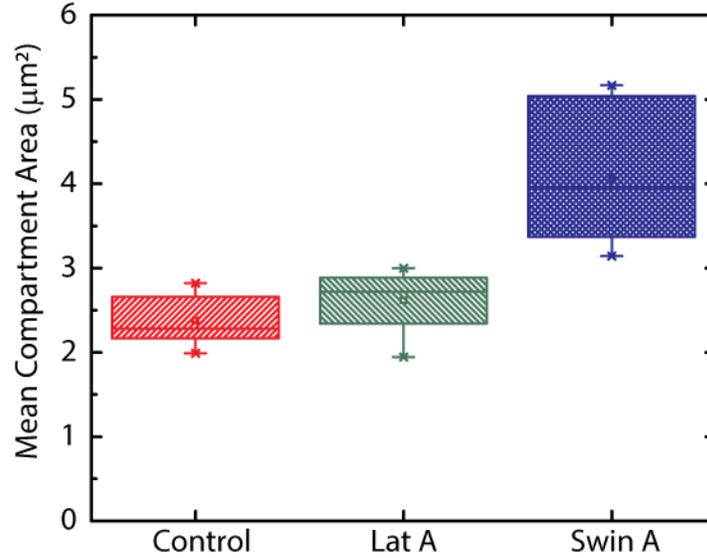


FIGURE 4.4. Actin cytoskeleton mean compartment areas, as determined by watershed compartmentalization. Control cells (N=10) had a mean compartment area of  $2.4 \pm 0.3 \mu\text{m}^2$ . The mean compartment area of  $2.6 \pm 0.4 \mu\text{m}^2$  for LatA cells (N=10) was not significantly different from the control case ( $p = 0.11$ ). SwinA cells (N=10) had a mean compartment area of  $4.1 \pm 0.8 \mu\text{m}^2$  which was significantly different from the control case ( $p < 0.0005$ ).

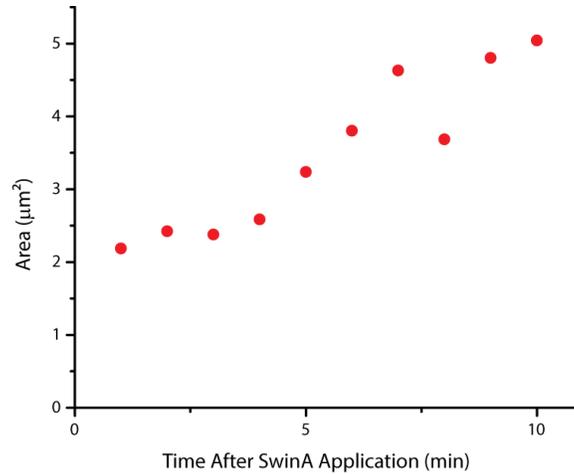


FIGURE 4.5. Mean area of actin cytoskeleton compartments in a single cell after application of Swinholide A over time. The mean area increased from  $2.2 \mu\text{m}^2$  at 1 minute after SwinA application to  $5 \mu\text{m}^2$  after 10 minutes.

#### 4.4. DISCUSSION

The cortical actin cytoskeleton is thought to play a role in organizing membrane proteins.

However, PALM images of cortical actin indicated that the actin bundles did not form

complete compartment boundaries, but instead the meshwork consisted of gaps between bundles. These gaps could be explained in several ways. (a) The actin meshwork may not form continuous boundaries, resulting in multiple gaps. (b) Since the images were taken in TIRF, the depth at which the excitation laser light extended into the cell was limited. Thus, actin bundles which were past the excitation laser depth would not have been imaged. The gaps in the actin cytoskeleton could represent regions where the actin did not come as close to the plasma membrane as would be required for them to be imaged. (c) The actin probe may not have completely labeled the actin cytoskeleton, possibly due to steric hindrances with other proteins which interact with actin. (d) The PALM images may not have sampled all of the labeled actin structures in the cell.

Through analysis of resident compartments for Kv2.1 and Kv1.4 channels, we observed that both channels treat actin as a barrier to compartment changes, preferring to cross gaps in the actin cytoskeleton more frequently than the random case. Kv1.4 does not show this tendency as strongly as Kv2.1, indicating that actin is not as effective of a barrier to its diffusion. This observation could be interpreted in several ways. (a) Kv1.4 channels do not form clusters which may be due to the reduced effect of the actin cytoskeleton on their movement when compared to Kv2.1. There may be modifications to Kv2.1 that cause actin to act as a more efficient barrier to their movement, thus encouraging clustering. (b) The actin cytoskeleton may surround another structure which constrains Kv2.1 channels. Kv2.1 clusters have been found to colocalize with junctions between the endoplasmic reticulum (ER) and the plasma membrane [7]. Thus, it is possible that the ER is responsible for the clustering mechanism of Kv2.1 and that actin is indirectly involved by structurally supporting the junction between the ER and the plasma membrane. (c) As previously noted by

Eddidin et al [14], the depth at which a membrane protein extends into the cytoplasm determines how frequently it encounters a barrier. Kv2.1 and Kv1.4 have a different number of amino acids which extend into the cytoplasm, and thus the mechanism responsible for the differences in how frequently they cross actin on compartment change may be a function of their cytoplasmic depth.

However, actin does not act as a perfect barrier, as there is a significant percentage of compartment changes that did cross actin. The images acquired are a two dimensional view of the actin cytoskeleton. The plasma membrane is not necessarily in close proximity with the actin filaments which make up the cellular cortex. For the cytoskeleton to act as a hindrance to channel movement, the structure would need to be close enough to the plasma membrane for the cytoplasmic domains of the membrane channels to sterically interact with the channel. The depth of view offered by TIRF microscopy is approximately 200 nm, which indicates that pieces of the actin cytoskeleton are being observed which are deeper into the cell than the cytoplasmic domains of the proteins extend. Since depth information was not captured in the actin cytoskeleton images used, it is likely that cases where a channel was allowed passage through an actin bundle were also cases where the actin bundle was not in close proximity to the plasma membrane. However, 3D imaging is necessary to verify this theory, and would be an interesting future direction for this project.

Kv2.1 and Kv1.4 both had residence times significantly longer than the control case of a 2D random walk, with Kv2.1 having the longest residence times of all. The fact that Kv2.1 and Kv1.4 treat actin as a barrier likely contributes to the difference between the random walk residence time and the Kv residence times. However, other factors due to the environment of the plasma membrane may also be at play. Molecular crowding could also

result in increased waiting times as well as obstructed diffusion as predicted by the picket fence model.

In Chapter 3, Kv2.1 channels were shown to occupy actin-deficient regions, but this relationship dispersed upon application of 100 nM LatA. Here, we showed that prior to the application of LatA, there was a distinct difference of the distribution of distances from actin for Kv2.1 compared to the control case. However, this distinction faded once LatA was applied, indicating that the dynamic population of actin filaments which undergo regular turnover are essential for the organization of Kv2.1 channels, as suggested previously [5]. Upon application of 75 nM SwinA, which has been shown to have a similar effect to 100 nM LatA on increasing Kv2.1 cluster size [5], the difference between the random lipid distance and the Kv2.1 distance distributions from actin was enhanced.

Application of both 100 nM LatA and 75 nM SwinA causes Kv2.1 clusters to become more diffuse and the overall cluster size to increase. SwinA nearly doubled the mean compartment area, likely due to reduction in F-actin. However, application of LatA did not change the size of the actin membrane compartments. This further emphasizes the difference in the effect of the two drugs on the actin compartments in the cell cortex. Each actin bundle visualized in the PALM images represents many filaments in close proximity that play a part in forming the barrier. As seen in Chapter 3, application of LatA reduces the density of actin, but the overall compartment structure is still present. The analysis done here shows that it is likely not compartment size changes that cause a change in the organization of Kv2.1 channels, at least not in the LatA case.

In this chapter, EDMs were used to study the distributions of distances of individual molecules from the cortical actin cytoskeleton. The use of EDMs in biology is extremely

useful in extracting meaningful data from images, especially when studying the spatial relationship between structures and proteins. In addition, the watershed transform was effective for modeling the actin compartments based on the cytoskeletal information available in the images. Although the cytoskeleton formed compartments, the boundaries formed by the actin bundles were often incomplete. Thus, to allow analysis of compartment sizes, the watershed transform allowed a best guess of the missing compartment boundaries to be made. Image processing provides ways to transform qualitative data into quantitative data fit for statistical analysis. Using techniques such as EDMs to determine proximity information and watershed for image segmentation can reveal meaningful information about biological processes.

## BIBLIOGRAPHY

- [1] Oberholzer, M., Östreicher, M., Christen, H., & Brühlmann, M. (1996). Methods in quantitative image analysis. *Histochemistry and cell biology*, 105(5), 333-355.
- [2] Rasband, W.S. ImageJ, U.S. National Institutes of Health, Bethesda, Maryland, USA, [imagej.nih.gov/ij/](http://imagej.nih.gov/ij/), 1997—2012.
- [3] Jaqaman, K., Loerke, D., Mettlen, M., Kuwata, H., Grinstein, S., Schmid, S. L., & Danuser, G. (2008). Robust single-particle tracking in live-cell time-lapse sequences. *Nature methods*, 5(8), 695-702.
- [4] O'Connell, K. M., Rolig, A. S., Whitesell, J. D., & Tamkun, M. M. (2006). Kv2. 1 potassium channels are retained within dynamic cell surface microdomains that are defined by a perimeter fence. *The Journal of neuroscience*, 26(38), 9609-9618.
- [5] Tamkun, M. M., O'Connell, K. M., & Rolig, A. S. (2007). A cytoskeletal-based perimeter fence selectively corrals a sub-population of cell surface Kv2. 1 channels. *Journal of Cell Science*, 120(14), 2413-2423.
- [6] Russ, J. C. (2006). *The image processing handbook*. CRC press.
- [7] Fox, P. D., Haberkorn, C. J., Weigel, A. V., Higgins, J. L., Akin, E. J., Kennedy, M. J., ... & Tamkun, M. M. (2013). Plasma membrane domains enriched in cortical endoplasmic reticulum function as membrane protein trafficking hubs. *Molecular biology of the cell*. (in press, doi: 10.1091/mbc.E12-12-0895)
- [8] Beucher, S. & Lantéjoul, C. (1979). Use of watersheds in contour determination. *Proceedings of the international workshop on image processing*. CCETT, Rennes (France)
- [9] Nandy, K., Gudla, P. R., Amundsen, R., Meaburn, K. J., Misteli, T., & Lockett, S. J. (2012). Automatic segmentation and supervised learning-based selection of nuclei in cancer tissue images. *Cytometry Part A*, 81(9), 743-754.

- [10] Ram, S., Rodríguez, J. J., & Bosco, G. (2012). Segmentation and detection of fluorescent 3D spots. *Cytometry Part A*, 81(3), 198-212.
- [11] Baker, L. A., & Rubinstein, J. L. (2011). Edged watershed segmentation: A semi-interactive algorithm for segmentation of low-resolution maps from electron cryomicroscopy. *Journal of structural biology*, 176(1), 127-132.
- [12] Coué, M., Brenner, S. L., Spector, I., & Korn, E. D. (1987). Inhibition of actin polymerization by latrunculin A. *FEBS letters*, 213(2), 316-318.
- [13] Bubb, M. R., Spector, I., Bershadsky, A. D., & Korn, E. D. (1995). Swinholide A is a microfilament disrupting marine toxin that stabilizes actin dimers and severs actin filaments. *Journal of Biological Chemistry*, 270(8), 3463-3466.
- [14] Edidin, M., Zuniga, M. C., & Sheetz, M. P. (1994). Truncation mutants define and locate cytoplasmic barriers to lateral mobility of membrane glycoproteins. *Proceedings of the National Academy of Sciences*, 91(8), 3378-3382.

## CHAPTER 5

# TECHNIQUES FOR THE STUDY OF VIRAL RNA DEPENDENT RNA POLYMERASES

### 5.1. INTRODUCTION

5.1.1. ROLE OF RNA DEPENDENT RNA POLYMERASES IN VIRUSES. Viruses are amazing machines that hijack cellular machinery by injecting their own genome into the cell. Since viruses largely use native cellular components to replicate, the development of treatments which specifically target and hinder the viral life cycle without harming the host can be challenging. However, these challenges can be overcome through increased knowledge of the specifics behind viral replication.

The mechanism by which viral replication occurs varies for each family of virus. Members of the *Picornaviridae* family are responsible for a variety of diseases ranging from the common cold to poliomyelitis. With a life cycle that is completely based on RNA, these viruses rely on the protein 3D<sup>pol</sup>, a RNA dependent RNA polymerase (RdRP) for replication. Since RdRP is essential to the viral life cycle, it provides an attractive target for antiviral treatments.

However, much about 3D<sup>pol</sup> is unknown. The crystal structures of RdRP elongation complexes have been studied, providing insight into possible mechanisms behind the kinetic cycle required for replication [3]. However, specifics of the kinetic cycle of the RdRP remain to be discovered. Further knowledge about the processes that assist viral effectiveness could allow for the development of new drugs which are able to specifically target the polymerase and prevent its proper function, thereby interrupting the viral life cycle.

5.1.2. STUDYING DNA AND RNA ON THE SINGLE-MOLECULE SCALE. Proteins which interact with DNA have been studied using various techniques and the majority of research has focused on polymerases native to the cell. Single molecule studies using optical traps have investigated the dynamics of DNA dependent RNA polymerases, revealing detailed information about dissociation rates, transcription rates, and pauses [5, 8]. Information about DNA polymerase kinetics have also been probed using Förster resonance energy transfer (FRET) [6, 7, 9], a phenomenon which allows the conformation of the molecule of interest to be probed.

However, a technique has been gaining increasing popularity which probes the dynamics of polymerases in multiplex while preserving the integrity of single-molecule measurements [1, 10]. In this method, DNA molecules are studied using a glass surface which has been coated with polyethylene glycol (PEG) molecules and a small fraction of biotin conjugated PEG (bioPEG) molecules. DNA molecules are bound to the bioPEG molecules, allowing the interactions of the DNA with fluorescently labeled polymerases to be observed while controlling the density of bound DNA. In addition, a flow cell is utilized, which causes the DNA strands to be stretched parallel to the flow. Thus, the position of the interacting polymerases can be tracked over time to determine detailed information about their behavior, such as transcription rates.

Although this technique has revealed detailed information concerning the behavior of various DNA associated proteins, it has yet to be extended to the study of RNA and RNA associated proteins. Here, we set the groundwork for multiplexed, single-molecule experiments which will allow the specifics of RdRP kinetics to be revealed.

## 5.2. METHODS

5.2.1. FLOW CELL DESIGN AND PREPARATION. A custom, re-usable flow cell holder was created from an aluminum base and a PDMS top that contained an inlet and outlet. The flow cell holder attached securely to the microscope stage through several screws. The flow cell itself was constructed by cutting 1-3 mm wide channel in a 4 x 2 mm piece of 0.12 mm thick double sided tape (SecureSeal Adhesive Sheet, Grace Bio-Labs, Bend, OR). A No. 1.5 glass coverslip (22x40 mm, Warner Instruments, Harmon, CT) was cleaned and pegylated following a previously established protocol [13] and was adhered to a No. 2 glass slide (Pearl) using the tape. The No. 2 glass slide formed the top of the flow cell and prior to use was drilled with a 1/16" diamond tip drill bit to form two ports. These ports lined up with the ports in the PDMS top and served as inlets and outlets for fluids.

5.2.2. IMAGING. The same microscope setup previously described in Chapter 2 was used to complete these experiments. To excite SYTOX green, a 473 nm laser was used. Experiments were completed using total internal reflection fluorescence (TIRF) so that the excitation was limited to the bottom portion of the flow cell.

Since the eventual goal of the system was to detect fluorescence from a single probe over an extended period of time, we used an imaging buffer designed to maximize the stability of the fluorophores. The imaging buffer used consisted of an oxygen scavenging system and a triplet state quencher. The presence of  $O_2$  in a buffer can cause premature bleaching of fluorophores through photo-oxidation [11]. However,  $O_2$  also acts as a triplet-state quencher, preventing fluorophores from entering their triplet state wherein they do not emit light. Thus, since the oxygen scavenging system removed excess  $O_2$ , Trolox was also added to the buffer to act as a triplet state quencher [12]. The imaging buffer consisted of 1 mM Trolox, 10 mM

Tris-HCL (pH 8.0), 75 mM NaCl, 0.8% w/v D-glucose, 0.04 mg/mL catalase, and 1 mg/mL glucose oxidase. The catalase and glucose oxidase were added immediately prior to imaging to ensure effectiveness. The final solution was adjusted to pH 8.0 using 0.1 M HCL.

Fluid was injected into the flow cell via a syringe pump (NE-500, New Era Pump Systems, Farmingdale, NY) controlled by a LabView (National Instruments, Austin, TX) application provided by the pump manufacturer. A high pressure dual mode sample injector valve (Model 7725; IDEX, Oak Harbor, WA) was used to facilitate accurate injection of various fluids into the chamber. Images were acquired using IQ2.3 software (Andor Technology, Belfast, Ireland). A 0.2 mg/mL neutravidin solution was incubated within the flow cell prior to injecting sample.

### 5.3. RESULTS

5.3.1. FLOW CELL PROOF-OF-CONCEPT. The first challenge of studying the RNA polymerases on the single molecule level was to verify the viability of our experimental setup. Pegylated flow cells have been used previously, successfully tethering DNA so that details about associated proteins could be probed [1, 10]. To verify that our system had the same capability, we assembled a flow cell using a pegylated glass slide. A syringe pump controlled through computer software was used to control flow rates and infusion volumes, while a HPLC valve allowed precise injection of various solutions. First, buffer was incubated in the chamber, followed by neutravidin which bound to the bio-PEG molecules on the pegylated slide surface. Next, 50 pM of a 13 kb DNA fragment which was biotinylated on one end was infused into the chamber and incubated for 10 minutes. After incubation, the unbound DNA was flushed out with buffer and 150 nM SYTOX Green was infused. The SYTOX dye intercalated between base pairs of bound DNA, revealing the DNA when excited with a 473

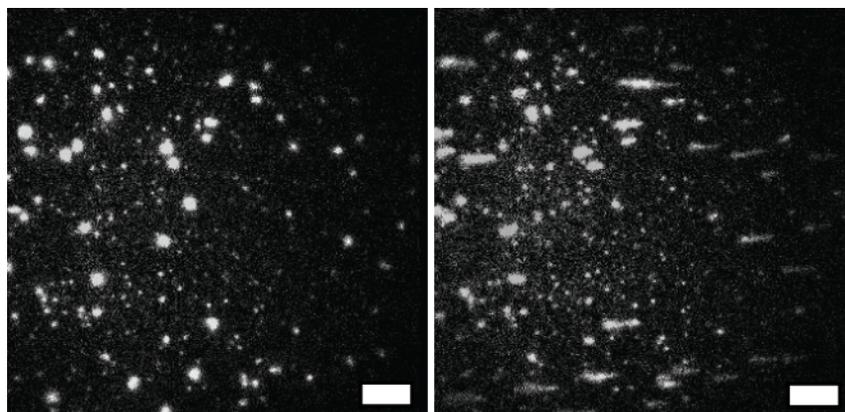


FIGURE 5.1. 13 kb biotinylated DNA fragments bound to a PEG slide and stained with 150 nM SYTOX Green. In the absence of flow, the DNA coils near its tether point (Left). When flow is applied, the DNA stretches out in the direction of flow (Right). Scalebar: 5  $\mu\text{m}$

nm laser. In the absence of flow, the DNA coiled near its tether (Left; Figure 5.1). However, when flow was applied, the DNA straightened out in the direction of flow (Right; Figure 5.1).

5.3.2. TETHERED RNA. The goal of these experiments was to allow single particle tracking of the movements of the RNA polymerase so that more information about the mechanism of its movement could be uncovered. To complete these measurements, it was necessary to stretch out the RNA so that the polymerase could be tracked with nanometer accuracy as it moved. Therefore, once our setup was confirmed to be capable of both binding DNA fragments at the desired density as well as stretching them, the next step was to achieve the same result with RNA.

The experiment mentioned previously was completed using a mixture including single-stranded RNA (ssRNA) hybridized to a biotinylated DNA fragment. SYTOX also binds to ssRNA, but the binding efficiency is much lower than with double stranded RNA or double stranded DNA. Individual ssRNA were observed which coiled near their tether location in

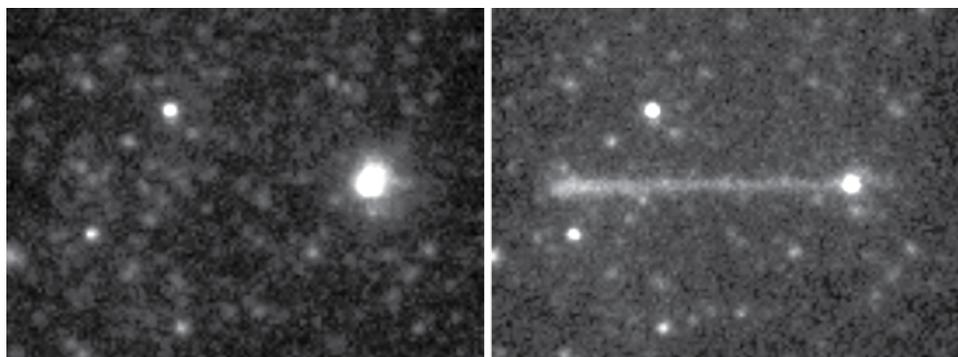


FIGURE 5.2. Single-stranded RNA tethered on one end and stained with SYTOX green. The ssRNA coils near its tether point in the absence of flow (Left) and stretches out when under the influence of flow (Right).

the absence of flow (Left; Figure 5.2) and stretched out under the influence of flow (Right; Figure 5.2).

#### 5.4. DISCUSSION

We showed that it was possible to tether a ssRNA molecule to a pegylated glass slide. However, several difficulties were encountered that need to be overcome in order for this project to move forward. First, we were unable to achieve the needed density of tethered RNA molecules within a single region of interest to run a multiplex experiment. One possible cause of this problem could have been a low binding efficiency of RNA due to secondary structures interfering with the biotin-neutravidin binding. In addition, although care was taken to avoid introducing RNAses into the samples, it is possible that the RNA could have been destroyed by RNAses.

We found that with dsDNA, even though laser power is greatly reduced by imaging in TIRF, using too high of power caused DNA to snap and break in multiple places. The effect is visible in Figure 5.1, where in the center of the image where the laser power was greatest, the majority of strands have broken. This placed a restriction on the imaging because powers

low enough to prevent the breaking from occurring made it difficult to see the dsDNA and ssRNA clearly.

An additional experiment was attempted using a mix including both a non-elongated, biotinylated RNA/DNA hybrid as well as a bound RdRP labeled with Atto 532 on the end of the RNA opposite the biotin. Once the hybrid was tethered to the flow cell, the flow was varied and the movement of the RdRP was observed. We expected the RdRP to locate near the tether point in the absence of flow, but to move to the end of its tether with the addition of flow. However, we were unable to produce the expected result. The RdRP observed did not move in response to the flow. Two possibilities that could have caused this behavior are either that (1) the pegylated surface may have had flaws which caused the nonspecific binding of the polymerase to the glass or (2) the RNA/DNA hybrid bound to the biotin-PEG correctly, but the flow force was not sufficient to overcome the secondary structure and lower persistence length of the RNA to stretch it out. Although this method of experimentation shows much promise, to probe the real-time kinetics of RdRP on RNA, more troubleshooting to overcome the challenges mentioned here is necessary.

## BIBLIOGRAPHY

- [1] Yardimci, H., Loveland, A. B., van Oijen, A. M., & Walter, J. C. (2012). Single-molecule analysis of DNA replication in *Xenopus* egg extracts. *Methods*, 57(2), 179-186.
- [2] Cohen, S. N., & Yielding, K. L. (1965). Inhibition of DNA and RNA polymerase reactions by chloroquine. *Proceedings of the National Academy of Sciences of the United States of America*, 54(2), 521.
- [3] Gong, P., & Peersen, O. B. (2010). Structural basis for active site closure by the poliovirus RNA-dependent RNA polymerase. *Proceedings of the National Academy of Sciences*, 107(52), 22505-22510.
- [4] Hobdey, S. E., Kempf, B. J., Steil, B. P., Barton, D. J., & Peersen, O. B. (2010). Poliovirus polymerase residue 5 plays a critical role in elongation complex stability. *Journal of virology*, 84(16), 8072-8084.
- [5] Davenport, R. J., Wuite, G. J., Landick, R., & Bustamante, C. (2000). Single-molecule study of transcriptional pausing and arrest by *E. coli* RNA polymerase. *Science*, 287(5462), 2497-2500.
- [6] Markiewicz, R. P., Vrtis, K. B., Rueda, D., & Romano, L. J. (2012). Single-molecule microscopy reveals new insights into nucleotide selection by DNA polymerase I. *Nucleic acids research*, 40(16), 7975-7984.
- [7] Luo, G., Wang, M., Konigsberg, W. H., & Xie, X. S. (2007). Single-molecule and ensemble fluorescence assays for a functionally important conformational change in T7 DNA polymerase. *Proceedings of the National Academy of Sciences*, 104(31), 12610-12615.
- [8] Harada, Y., Funatsu, T., Murakami, K., Nonoyama, Y., Ishihama, A., & Yanagida, T. (1999). Single-molecule imaging of RNA polymerase-DNA interactions in real time. *Biophysical journal*, 76(2), 709-715.

- [9] Rothwell, P. J., Mitaksov, V., & Waksman, G. (2005). Motions of the fingers subdomain of klentaq1 are fast and not rate limiting: implications for the molecular basis of fidelity in DNA polymerases. *Molecular cell*, 19(3), 345-355.
- [10] Kim, S., Blainey, P. C., Schroeder, C. M., & Xie, X. S. (2007). Multiplexed single-molecule assay for enzymatic activity on flow-stretched DNA. *Nature methods*, 4(5), 397-399.
- [11] Rasnik, I., McKinney, S. A., & Ha, T. (2006). Nonblinking and long-lasting single-molecule fluorescence imaging. *Nature Methods*, 3(11), 891-893.
- [12] Cordes, T., Vogelsang, J., & Tinnefeld, P. (2009). On the mechanism of Trolox as antiblinking and antibleaching reagent. *Journal of the American Chemical Society*, 131(14), 5018-5019.
- [13] Selvin, P. R., & Ha, T. (Eds.). (2008). *Single-molecule techniques: a laboratory manual* (pp. 3-36). Cold Spring Harbor, NY:: Cold Spring Harbor Laboratory Press.

## CHAPTER 6

### CONCLUSION

Here, we combined single particle tracking of membrane proteins with PALM of the actin cytoskeleton. Our observations allowed us to make educated assumptions about their interactions. The actin cytoskeleton was observed to act as a barrier to membrane protein diffusion, likely due to the steric hindrance of the actin bundles in close proximity to the plasma membrane with the cytoplasmic domains of the membrane proteins. Other behaviors were also observed, where actin appeared to act as a container to membrane protein diffusion and where actin was seemingly ignored. These events likely occurred in regions where the actin cytoskeleton was further away from the plasma membrane, though still within the field of view. In addition, we found that when Kv2.1 and Kv1.4 channels change compartments they show a preference for leaving compartments in a path unobstructed by actin. Kv1.4 showed this preference less strongly than Kv2.1, which could be due in part to the different depths at which the channels extend into the cytoplasm. Thus, Kv2.1 treats actin as a more effective barrier to diffusion than Kv1.4.

In addition, we studied the effects of the actin cytoskeleton on Kv2.1 channel organization by analyzing the distributions of channel distances from the actin cytoskeleton. Upon application of 100 nM Latrunculin A, the organization of the Kv2.1 channels broke down. However, upon application of 75 nM Swinholide A which has a similar effect on Kv2.1 cluster organization as 100 nM LatA, the organization effect of the actin cytoskeleton was enhanced. 75 nM SwinA and 100 nM LatA both increase the cluster size of Kv2.1, however the spatial relationship of the clusters to actin are different for the two drugs. This difference may

be related to the observation that application of 100 nM LatA does not significantly alter compartment area, whereas application of 75 nM SwinA doubles compartment area.

The groundwork for performing single-molecule studies on RNA dependent RNA polymerases was also performed. A flow cell was constructed that was capable of binding individual ssRNA molecules. With further advances to increase binding efficiency and to reliably stretch the RNA in the direction of the flow, the system has the potential to reveal the kinetics of a polymerase that is essential to the viral life cycle.

## APPENDIX A

# AN ADDITIONAL APPLICATION OF EUCLIDEAN DISTANCE MAPPING TO STUDYING CELLULAR EVENTS

Dr. Michael Tamkun and his colleagues have extensively studied the role of the endoplasmic reticulum (ER) in exocytosis, specifically examining the role of ER-plasma membrane junctions as trafficking sites [1]. One line of study was to determine whether the transferrin receptor (TfR) was exocytosed near the ER. Images of the ER and exocytosis locations of the ER were provided. The ER images were corrected for uneven illumination via subtraction of the same image blurred by a 20px Gaussian filter, then processed with a 5x5 Laplacian filter which detects edges to clarify ER features (Left; Figure A.1). Euclidean distance maps (EDMs) were then generated from the thresholded images (Right; Figure A.1).

The transferrin localizations were used in conjunction with the EDMs to determine the distance between the exocytosis site and the ER. First, the entire cell footprint was selected as an ROI and all localizations were considered. The same cell footprint was used to select values from the EDM which served as control values, since the sample of all EDM values within the cell is the distribution of all distances from the ER within the cell. The difference between the control and transferrin exocytosis site cumulative distribution functions (CDF) indicates that the exocytosis locations are generally closer to the ER than the control case. TfR delivery had a mean distance from the ER of  $0.25 \pm 0.38 \mu\text{m}$  (mean $\pm$ sd, n=131, from 5 cells) while the control case had a mean distance of  $1.5 \pm 2.6 \mu\text{m}$ ,  $p < 0.0001$ .

In addition, a second ROI was chosen within the ER-rich area. The control case now becomes closer to the TfR case because by limiting the ROI to ER-rich areas, the overall

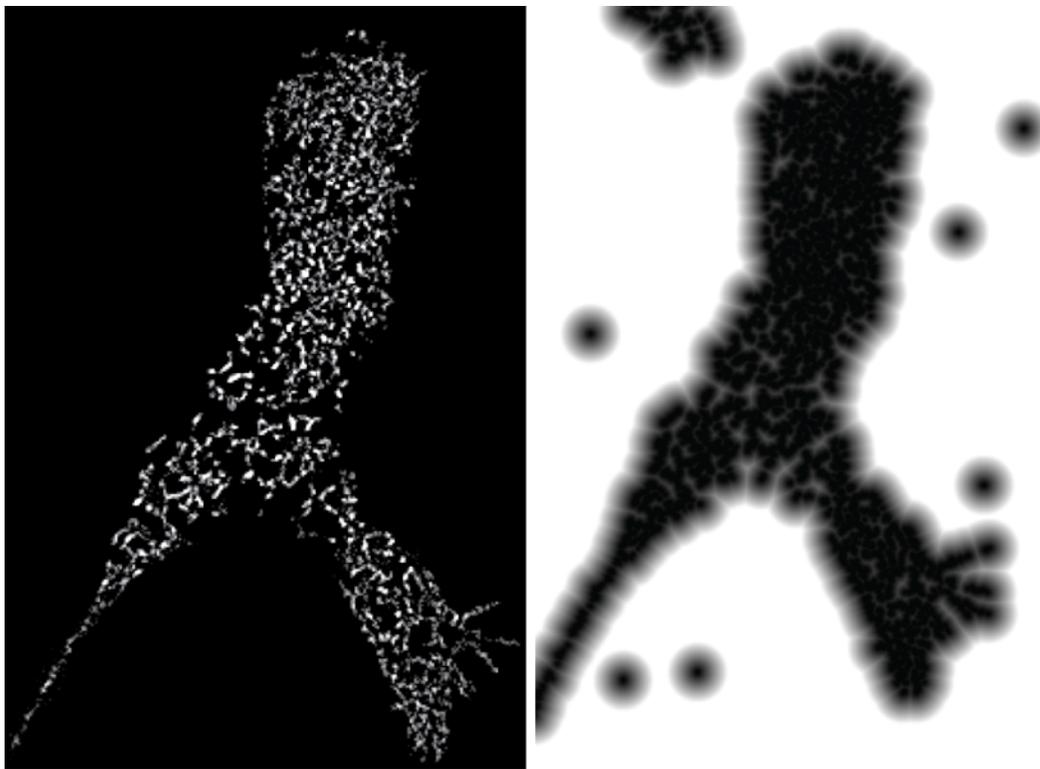


FIGURE A.1. Image of the ER that has undergone uneven illumination correction and Laplacian filtering (left) and the EDM of the same cell (right).

distance from ER features decreases. However, there is still a significant difference between the control case and the TfR case. The TfR delivery location had a mean distance from the ER of  $0.17 \pm 0.24 \mu\text{m}$  ( $n=114$ , from 5 cells) while the control case averaged  $0.25 \pm 0.33 \mu\text{m}$  ( $n=1.1 \times 10^7$  pixels),  $p < 0.05$  assuming equal variance.

In conclusion, analysis of EDM images of the ER allowed quantitative analysis of the proximity of transferrin exocytosis sites, showing that the exocytosis locations are not random, but are instead more likely to occur near the ER. This analysis contributed to research completed by Fox, et al. which explored ER-PM junctions as a trafficking hub for exocytosis and endocytosis<sup>1</sup>.

---

<sup>1</sup>Fox, P. D., Haberkorn, C. J., Weigel, A. V., Higgins, J. L., Akin, E. J., Kennedy, M. J., ... & Tamkun, M. M. (2013). Plasma membrane domains enriched in cortical endoplasmic reticulum function as membrane protein trafficking hubs. *Molecular biology of the cell*. (in press, doi: 10.1091/mbc.E12-12-0895)

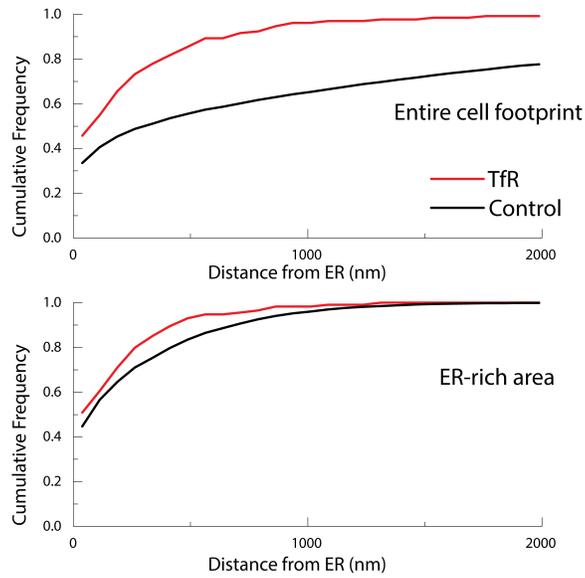


FIGURE A.2. Euclidean distance mapping analysis of the distance of TfR exocytosis sites from ER features. When the entire cell footprint was considered (top), TfR delivery had a mean distance from the ER of  $0.25 \pm 0.38 \mu\text{m}$  (mean $\pm$ sd,  $n=131$ , from 5 cells) while the control case had a mean distance of  $1.5 \pm 2.6 \mu\text{m}$ ,  $p < 0.0001$ . For the case where the region of interest was limited to ER-rich areas (bottom), TfR delivery location had a mean distance from the ER of  $0.17 \pm 0.24 \mu\text{m}$  ( $n=114$ , from 5 cells) while the control case averaged  $0.25 \pm 0.33 \mu\text{m}$  ( $n=1.1 \times 10^7$  pixels),  $p < 0.05$  assuming equal variance.

## APPENDIX B

### CODE

The following are MATLAB applications unless otherwise specified.

DEPENDENCIES. DIPimage (<http://www.diplib.org/>), which is free to use, is required for the majority of these algorithms.

DETERMINING NYQUIST RESOLUTION. The following code loads the output from the particle localization algorithm used for all .mat files within a specified directory. It calculates the

```
base_folder_name = 'K:\9-13-12\Processed\';
image_id = '71450';
% Number of frames to make one PALM reconstruction
reconstruction_frames = 100;
% ROI to use for analysis, based on original acquired images
roi = [604 36 40 40];
particle_files = dir([base_folder_name '*.mat']);
for k=1:length(particle_files)
    image_id = particle_files(k).name;

    load([base_folder_name image_id], 'SRtest');
    % Get particles within ROI
    particle_indexes = find(SRtest.Results.X >= roi(1) & SRtest.Results
        .X < roi(1) + roi(3) & ...
        SRtest.Results.Y >= roi(2) & SRtest.Results.Y < roi(2) + roi
            (4));
    particles = SRtest.getSubgroup(particle_indexes);
    max_frame = max(max(particles.frameenum));
    % Format: [molecules_in_roi mean_R min_R max_R std_R]
    reconstruction_stats = zeros(floor(max_frame/reconstruction_frames)
        +1, 5);
    stats_i = 1;
    for i=0:reconstruction_frames:max_frame
        % disp(['Frame ' num2str(i)]);
        % Get all particles in that frame
        frame_particles = find(particles.frameenum >= i & particles.
            frameenum <= i+reconstruction_frames-1);
        frame_all_particles = find(SRtest.Results.frameenum >= i &
            SRtest.Results.frameenum <= i+reconstruction_frames-1);
        num_particles = length(frame_particles);
        X = particles.X(frame_particles);
        Y = particles.Y(frame_particles);
        all_X = SRtest.Results.X(frame_all_particles);
        all_Y = SRtest.Results.Y(frame_all_particles);
        nearest_neighbors = zeros(num_particles, 1);
        for j=1:num_particles
```

```

        % For each particle, find its nearest neighbor
        current_x = X(j); current_y = Y(j);
        R = sqrt((all_X-current_x).^2+(all_Y-current_y).^2);
        R(R == 0) = []; % Remove any with radius 0, most likely same
            particle or double localization
        nearest_neighbors(j) = min(min(R));
    end
    nearest_neighbors = nearest_neighbors * 130;

    if(~isempty(nearest_neighbors))
        reconstruction_stats(stats_i, :) = [num_particles mean(
            nearest_neighbors) min(nearest_neighbors) max(
            nearest_neighbors) std(nearest_neighbors)];
        stats_i = stats_i + 1;
    else
        reconstruction_stats(end, :) = [];
    end
end

end
disp([image_id ' Resolution Mean/Min/Max (nm): ' num2str(mean(
reconstruction_stats(:, 2))*2) '/' ...
num2str(min(reconstruction_stats(:, 2))*2) '/' num2str(max(
reconstruction_stats(:, 2))*2)]);
disp([num2str(reconstruction_frames) ' ' ...
num2str(min(reconstruction_stats(:, 2))*2)]);
% disp('===== RESULTS =====');
% disp(['Molecules/Frame: ' num2str(mean(reconstruction_stats(:, 1)))]);
;
% disp(['Nearest Neighbor Mean/Min/Max (nm): ' num2str(mean(
reconstruction_stats(:, 2)) '/' ...
num2str(min(reconstruction_stats(:, 2)) '/' num2str(max(
reconstruction_stats(:, 2)))]);
% disp(['Nearest Neighbor STD (nm): ' num2str(mean(reconstruction_stats
(:, 5)))]);
% disp(['Nyquist Limited Resolution (nm): ' num2str(mean(
reconstruction_stats(:,2)*2)]);
% disp('=====');
end

```

CHANNEL OFFSET CALCULATION. This algorithm was used to determine the offsets between two channels in an image, particularly to allow for the overlay of Kv2.1 channel activity on PALM actin. Prior to using this algorithm, white light bead images were analyzed using a custom autocorrelation algorithm written in LabView. The LabView program outputs a tab delimited text file where each row bead information for a particular frame. The columns are in the format X0\_Left Y0\_Left I0\_Left X0\_Right Y0\_Right I0\_Right X1\_Left Y1\_Left I1\_Left X1\_Right etc...

The offset numbers are calculated as the location of the bead on the right side of the image minus the location of the same bead on the left side of the image. Thus, a negative y offset indicates that localizations on the right channel need to be shifted down, or in other words, to have the absolute value of the y offset added to their y coordinates. The x offset is always positive. However, due to the astigmatism induced by the dichroic, the x offset is dependent on the x location of the particle in the original image. To correct x locations, the x coordinate of the right channel particles should be subtracted by the linear formula output by this program.

```
% Takes an input tab delimited file with marker locations, assuming 0
is
% the left location of the marker, 1 is the right location of the
marker, 2
% is the left location of the next marker, and so on
roi_size = 30; % Size of ROI used in labview to generate localizations
filename = 'C:\Users\j\lh\Desktop\7-3-12 Analysis\Colocalization\dish1.
txt';
location_correct = (roi_size / 2) * 130;
trajectories = importdata(filename, '\t', 1);
trajectories = trajectories.data;
% To ignore certain data points
% trajectories(:, 67:72) = [];
traj_count = floor(size(trajectories, 2) / 6);
y_offsets = zeros(traj_count, 1);
x_offsets = zeros(traj_count, 1);
x_coord = zeros(traj_count, 1);
for i=1:traj_count
    x_column_i = (i-1)*6 + 1;
    % Correct for LabView giving localizations at top left corner
    instead
    % of center of bead
    left_x = mean(trajectories(:, x_column_i)) + location_correct;
    right_x = mean(trajectories(:, x_column_i + 3)) + location_correct;
    left_y = mean(trajectories(:, x_column_i + 1)) + location_correct;
    right_y = mean(trajectories(:, x_column_i + 4)) + location_correct;

    x_offsets(i) = right_x - left_x;
    y_offsets(i) = right_y - left_y;
    x_coord(i) = right_x;

    disp([num2str(i) ':' num2str(x_offsets(i))]);
end
x_offset = mean(x_offsets);
y_offset = mean(y_offsets);
x_std = std(x_offsets);
```

```

y_std = std(y_offsets);
% Find fit to camera x astigmatism
figure
line_fit = polyfit(x_coord, x_offsets, 1);
scatter((x_coord*10^-3), (x_offsets*10^-3))
hold on
plot((x_coord*10^-3), (x_coord*line_fit(1) + line_fit(2))*10^-3, '--r')
title({'Line Fit in nm - Dichroic Astigmatism;[num2str(line_fit(1)) 'x
+ ' num2str(line_fit(2))]});
xlabel('Right Channel X (um)');
ylabel('X Offsets (um)');
hold off
% Find fit to camera x astigmatism
% figure
% scatter((x_coord/130), (x_offsets/130))
% hold on
% plot((x_coord/130), (x_coord*line_fit(1) + line_fit(2))/130, '--r')
% title({'Line Fit in nm - Dichroic Astigmatism;[num2str(line_fit(1)) '
x + ' num2str(line_fit(2)/130)]});
% xlabel('Right Channel X (px)');
% ylabel('X Offsets (px)');
% hold off
disp(' ');
disp(' ');
disp(['FILE: ' filename]);
disp('--- Offset Info ---');
disp(['X Offset: ' num2str(x_offset) ' nm / ' ...
num2str(x_offset / 130) ' px']);
disp(['Y Offset: ' num2str(y_offset) ' nm / ' ...
num2str(y_offset / 130) ' px']);
disp(' ');
disp('--- StdDev Info ---');
disp(['X StdDev: ' num2str(x_std) ' nm / ' ...
num2str(x_std / 130) ' px']);
disp(['Y StdDev: ' num2str(y_std) ' nm / ' ...
num2str(y_std / 130) ' px']);
disp(' ');
disp('--- LineFit Info ---');
disp(['Formula (nm): ' num2str(line_fit(1)) 'x + ' num2str(line_fit(2))
]);
disp('-----');

```

COMPARTMENT TRACKING. Compartment tracking consists of two stages:

- (1) To identify the coordinates of the centroid of each compartment
- (2) To connect centroids across multiple frames

The input to this program was a multi-image tiff of watershed images. The output is an XYI file, where the value of the I column indicates whether that particular compartment was present or not in that frame.

```
folder_path = 'K:\9-13-12\Processed\compartment analysis\85617\';
base_name = 'centroids';
ROI = 100; % Size of box in pixels
max_blink = .1; % Maximum dark time for a particle in sec
frame_rate = .02; % frame rate in sec
frame_start = 0;
frame_end = 240;
max_dark_frames = round(max_blink / frame_rate);

watershed_file = [folder_path 'matlab_watershed.tif'];

disp('Loading watershed...');
watershed_i = readtimeseries(watershed_file, '[1 1]');
watershed_i = dip_array(watershed_i);
digits = 10;

stack_size = size(watershed_i, 3);

particle_x = [];
particle_y = [];
particle_frames = [];
for i=1:stack_size
    current_frame = watershed_i(:, :, i);
    watershedBW = im2uint8(current_frame);
    CC = bwconncomp(watershedBW, 4);
    s = regionprops(CC, 'centroid');
    centroids = round(cat(1, s.Centroid));

    new_particle_frames = zeros(size(centroids, 1), 1);
    new_particle_frames(:) = i;
    particle_frames = [particle_frames; new_particle_frames];
    particle_x = [particle_x; centroids(:, 1)];
    particle_y = [particle_y; centroids(:, 2)];

    disp(['Done With Frame ' num2str(i)]);
end
qd_particles = {};
qd_particles.X = particle_x;
qd_particles.Y = particle_y;
qd_particles.filename = particle_frames;

% For these datasets (psf localization) filename is actually the frame
% number
trajectories = zeros(size(qd_particles.X, 1), frame_end-frame_start+1);
trajectory_i = 0; % Holds the row # of the most recent trajectory
```

```

% Stores information about which trajectories ended, when, and why
% column 1: 1 if went dark, 2 if collided
% column 2: particle indexes, column 3: last frame seen in
trajectory_ends = zeros(size(qd_particles.X, 1), 3);
trajectory_ends_i = 1; % row # of the next empty row
for i=frame_start:frame_end
    frame_particles = find(qd_particles.filename == i);
    frame_particles_x = qd_particles.X(frame_particles);
    frame_particles_y = qd_particles.Y(frame_particles);
    frame_particles_found = size(frame_particles, 1);
    used_particles = zeros(frame_particles_found, 1);
    used_particles_i = 1;

    if(trajectory_i > 0)
        % Cycle through all trajectories, but do not consider
        % any trajectories where dark time has exceeded max_dark_frames
        % or where trajectory has been aborted
        for j=1:trajectory_i
            % Do not consider if trajectory is listed in trajectory_ends
            % (inefficient way of doing this, find a better solution
            % later)
            if(any(trajectory_ends(:, 2) == trajectories(j, 1)))
                continue;
            end

            % some logic here to connect the dots
            current_trajectory = trajectories(j, :);

            % Get index of most recent location for the trajectory
            [r, c] = find(current_trajectory == max(current_trajectory
            (:)));

            qd = current_trajectory(c);

            last_x = qd_particles.X(qd);
            last_y = qd_particles.Y(qd);
            last_frame = qd_particles.filename(qd);

            % If dark too long, stop tracking
            if(i - last_frame > max_dark_frames)
                trajectory_ends(trajectory_ends_i, :) = [1 trajectories(
                j, 1) last_frame];
                trajectory_ends_i = trajectory_ends_i + 1;
                continue;
            end

            % Find particles in current frame around a box of size ROI
            % centered at last_x, last_y
            linked = find(frame_particles_x >= (last_x - (ROI/2)) &...
                frame_particles_x <= (last_x + (ROI/2)) &...
                frame_particles_y >= (last_y - (ROI/2)) &...
                frame_particles_y <= (last_y + (ROI/2)));

            % Link particles if there's only one candidate

```

```

        if(size(linked, 1) == 1)
            % Make sure particle hasn't been claimed already
            particle = frame_particles(linked(1));

            if(~any(used_particles == particle))
                trajectories(j, c+1) = particle;

                % Mark particle as used so that it can't be used
                % again

                used_particles(used_particles_i) = particle;
                used_particles_i = used_particles_i + 1;
            end
            %% Add some logic here to look at distances between
            %% the particle and the previously linked particle
            %% and the particle and the currently examined particle
            %% to see which distance is least, then update links
            %% appropriately
        elseif(size(linked, 1) == 0)
            % Blink event
        else
            % Stop tracking -- too many particles in ROI
            trajectory_ends(trajectory_ends_i, :) = [2 trajectories(
                j, 1) i];
            trajectory_ends_i = trajectory_ends_i + 1;
            continue;
        end
    end
end

new_particles_i = find(frame_particles ~= used_particles);

for k=1:frame_particles_found
    if(~any(frame_particles(k) == used_particles))
        trajectory_i = trajectory_i + 1;
        trajectories(trajectory_i, 1) = frame_particles(k);
    end
end

disp(['Done with ' num2str(i)]);

end
% This next section puts the trajectory information in an XYI file
% format
% that we typically use. I is set to NaN during a blink, 1 otherwise
% Delete unused rows in trajectories
trajectories(find(trajectories(:, 1) == 0), :) = [];
trajectory_xyi = zeros(size(trajectories, 2), size(trajectories, 1)*3,
    'double');
trajectory_xyi(:) = NaN;
% Cycle through each trajectory, populating trajectory_xy with the
% proper
% values
for i=1:size(trajectories, 1)

```

```

x_column = (i-1)*3 + 1;
y_column = x_column + 1;
i_column = y_column + 1;

% Cycle through each link in the trajectory
for j=1:size(trajectories, 2)
    particle = trajectories(i, j);

    % If zero, break out of the for loop because no more links
    remain.
    if(particle == 0)
        break;
    else
        % Store the x/y/i values for the particular particle
        particle_frame = qd_particles.filename(particle);
        trajectory_xyi(particle_frame, x_column:i_column) = [
            qd_particles.X(particle) qd_particles.Y(particle) 1];
    end
end
end
format = '';
title = '';
fid = fopen([folder_path base_name '_Compartment_Track.txt'], 'w'); %%
    Filename to write to
traj_counter = 0;
for j=1:size(trajectories, 1)
    format = [format '%12.10f\t%12.10f\t%12.10f\t'];
    title = [title 'X' num2str(traj_counter) '\t' 'Y' num2str(
        traj_counter) '\t' 'I' num2str(traj_counter) '\t'];
    traj_counter = traj_counter + 1;
end
fprintf(fid, [title '\r\n']);
fprintf(fid, [format '\r\n'], trajectory_xyi');
fclose(fid);
disp('Done!');

```

COMPARTMENT CHANGE ANALYSIS. The following algorithm requires the following inputs:

- A file containing trajectories in a format where the columns are X0 Y0 I0 X1 Y1 I1 ... and each row represents a particular frame
- A file containing compartment trajectories (see compartment tracking algorithm)
- A multi-image tiff containing watershed images generated from PALM actin images
- A multi-image tiff containing the thresholded PALM actin images
- A multi-image tiff containing the EDM of the PALM actin images

The algorithm detects compartment changes based on the trajectories of individual particles combined with the compartment tracking trajectories. To avoid issues with a compartment change being due to noise in the watershed, a compartment change is only recorded if the change persists for at least `min_stick` frames, which is typically set to 5. To check whether or not a particle crossed actin on compartment change, a line is drawn between the particle's location immediately prior and after the compartment change using Bresenham's line drawing algorithm (<http://www.mathworks.com/matlabcentral/fileexchange/28190-bresenham-optimized-for-matlab>). Then, the point at which this line intersects the compartment boundary is checked based on the thresholded actin file provided to determine whether the crossing occurred on actin or not.

```

folder_path = 'K:\9-13-12\Processed\compartment analysis\90215\';
watershed_file = [folder_path 'matlab_watershed.tif'];
edm_file = [folder_path 'edm.tif'];
actin_file = [folder_path 'actin_thresholded.tif'];
tracks_file = [folder_path 'tracks.txt'];
compartment_trajectory_file = [folder_path 'centroids_Compartment_Track
.txt'];
reconstruction_frames = 100; % number of original frames used to
    reconstruct 1 PALM image
sliding_time_window = 10; % sliding time window used for original PALM
    image
track_frames_averaged = 1; % frames averaged for QD tracks
track_start_frame = 1; % frame (numbers based on PALM frames) to start
    looking at tracks
track_end_frame = 240;
min_stick = 5; % number of frames a particle must be in the same
    compartment before it can be considered for switches
track_offset_x = 0; % offset in pixels (after scaling) to apply to
    tracks
track_offset_y = 0; % offset in pixels (after scaling) to apply to
    tracks
make_movie = 0; % Set to 1 to make a movie of the skeleton with
    trajectories overlaid
% 9/13/12 Kv1.4
roi = [1251 486 1086 654]; % 90215 - Corrected
%% END SETTINGS =====
if(make_movie == 1)
    movie_save_path = [folder_path 'RNDM_Crossed_Actin_Trajectories.tif
'];
    movie_save_path_all = [folder_path 'RNDM_All_Trajectories.tif'];

    % Size of the image

```

```

total_width = roi(3);
total_height = roi(4);
trajectory_images = zeros(total_height, total_width,
    track_end_frame - track_start_frame + 1);
trajectory_all_images = zeros(total_height, total_width,
    track_end_frame - track_start_frame + 1);
lineDrawer = vision.ShapeInserter;
lineDrawer.Shape = 'Lines';
lineDrawer.BorderColor = 'White';
lineDrawer.Antialiasing = true;
end
disp('Loading watershed...');
watershedi = readtimeseries(watershed_file, '[1 1]');
watershedi = dip_array(watershedi);
disp('Loading skeleton...');
skeleton = readtimeseries(actin_file, '[1 1]');
skeleton = dip_array(skeleton) & ~watershedi;
disp('Loading EDM...');
edm = readtimeseries(edm_file, '[1 1]');
edm = dip_array(edm);
disp('Loading tracks...');
tracks = importdata(tracks_file, '\t', 1);
tracks = tracks.data;
track_num = floor(size(tracks, 2) / 3);
track_length = size(tracks, 1);
% Adjust track coordinates. Every 3 (starting at 1) is x
% and every 3 (starting at 2) is y
tracks(:, 1:3:size(tracks, 2)) = round((tracks(:, 1:3:size(tracks, 2))
    -1)*10) + track_offset_x - roi(1) + 1;
tracks(:, 2:3:size(tracks, 2)) = round(((tracks(:, 2:3:size(tracks, 2))
    )-1)*10) + track_offset_y - roi(2) + 1;
% Can store particle compartment information in every 3rd column (ie,
% where
% the intensities are normally kept) since I don't use the intensities
% at
% all
tracks(:, 3:3:size(tracks, 2)) = 0;
disp('Loading compartment trajectories...');
trajectories = importdata(compartment_trajectory_file, '\t', 1);
trajectories = trajectories.data;
trajectories(:, 3:3:size(trajectories, 2)) = 0;
% Go through tracks frame by frame
frame_offset = 1;
% Format: [frame #, QD #, result, crossed skeleton; start_x; start_y;
% end_x; end_y] where:
% result: -1 if out of bounds or on compartment line; 2 if same
% crossed skeleton: -1 if didn't cross, 1 if crossed
% moving: 1 if moving away from actin, 2 if moving toward actin, 3 if
% same
% distance
% compartment, 1 if different compartment
compartment_results = zeros(track_num*(track_end_frame -
    track_start_frame), 8);
compartment_result_i = 1;

```

```

for j=track_start_frame:track_end_frame
    disp(['Frame #' num2str(frame_offset)]);

    current_frame = watershedi(:, :, frame_offset);

    watershedBW = im2uint16(current_frame);
    CC = bwconncomp(watershedBW, 4);
    s = regionprops(CC, 'centroid');
    centroids = round(cat(1, s.Centroid));

    % Go through the compartment trajectories for this frame and add an
    % entry to indicate which CC.PixelIDxList index they are contained
    in
    for p=1:3:size(trajectories, 2)
        x = trajectories(frame_offset, p);
        y = trajectories(frame_offset, p+1);

        if(~isnan(x) && ~isnan(y))
            cc_link = find(centroids(:, 1) == x & centroids(:, 2) == y)
                ;
            trajectories(frame_offset, p+2) = cc_link;
        end
    end

    track_segment_start = (((j-1)*sliding_time_window) + (
        reconstruction_frames / 2)) / track_frames_averaged;
    track_segment_end = track_segment_start;

    disp(['Initial start: ' num2str(track_segment_start)]);

    track_segment_start = track_segment_start - ceil(
        sliding_time_window / (2 * track_frames_averaged)) + 1;
    track_segment_end = track_segment_end + floor(sliding_time_window /
        (2 * track_frames_averaged));
    disp(['Final start-end: ' num2str(track_segment_start) '-' num2str(
        track_segment_end)]);
    frame_trajectories = trajectories(frame_offset, :);
    frame_trajectories = frame_trajectories(:, 3:3:end);

    % Go through each track in the segment
    for n=track_segment_start:track_segment_end
        track_frame = n;

        % Go through tracks for this frame
        for k=1:track_num
            x_column_i = ((k - 1) * 3) + 1;
            y_column_i = x_column_i + 1;
            i_column_i = x_column_i + 2;

            x = tracks(track_frame, x_column_i);
            y = tracks(track_frame, y_column_i);
            if(~isnan(x) && ~isnan(y))
                % If track is out of bounds
                if(x < 1 || y < 1 || x > roi(3) || y > roi(4))

```

```

        cvalue = -1;
    else
        coord_index = (CC.ImageSize(1)*(x-1)) + y;
        qd_cc_link = find(cellfun(@(cellval) any(cellval ==
            coord_index), CC.PixelIdxList));
        if isempty(qd_cc_link)
            qd_compartment = 0;
        else
            % The actual compartment is based on the
            % tracked
            % compartment identities
            qd_compartment = find(frame_trajectories ==
                qd_cc_link);
        end

        tracks(track_frame, i_column_i) = qd_compartment;
        % Once it switches compartments, it must stay in
        % the
        % new compartment for at least min_stick time
        % points
        % before it is considered a true switch
        %

        % look at the
        % compartment data in
        % tracks(track_frame-min_stick:track_frame,
        % column_i)
        % If they are all the same, compare them with
        % compartment data in:
        % tracks(track_frame-min_stick-1). If this
        % compartment
        % is different than the compartments in future
        % frames,
        % log a compartment_changed event between
        % coordinates
        % of
        % (*) track_frame-min_stick-1 and ($) track_frame-
        % min_stick
        % If (*) compartment == 0, no change.
        % If in the frame ($) does not have (*) compartment
        % (x/y are NaN), no change.

        %% Reasoning for algorithm design:
        % But the problem with this is... what if (*) is 0?
        % This means the compartment didn't necessarily
        % change.
        % So let's say that if (*) is 0, no change is
        % triggered.

        % Another problem is that a compartment change
        % could
        % easily be triggered by small watershed variations
        % from frame to frame. These seem to last 3-5
        % frames
    
```

```

% (so typically 30-50 track points). To address
% this
% problem: when a compartment change is triggered,
% check ($) frame for the (*) compartment and see
% if
% the value was NaN. If it was, don't trigger a
% compartment change.
%%%%%%%%%

% Check if the particle has been in the current
% compartment for at least min_stick frames
start_frame_range = track_frame - min_stick;

% If start_frame_range <= 0, not enough compartment
% data has been gathered to determine if a
% compartment
% change has occurred
if(start_frame_range > 0)
    compartment_check = all(tracks(
        start_frame_range:track_frame, i_column_i) ==
        tracks(track_frame, i_column_i));

    previous_compartment = tracks(start_frame_range
        - 1, i_column_i);

    start_compartment_frame = round((
        start_frame_range*track_frames_averaged -
        (.5*reconstruction_frames))/
        sliding_time_window + .9) - track_start_frame
        + 1;

    % Check if this compartment varies from the one
    % prior to start_frame_range and make sure
    % compartment prior to start_frame_range is not
    % zero.
    % Finally, make sure prior compartment didn't
    % blink/disappear
    % Multiply by 3 to get to the compartment info
    % column, subtract by 1 to get the Y info
    if(compartment_check == 1 && ...
        tracks(start_frame_range, i_column_i) ~=
        previous_compartment && ...
        previous_compartment ~= 0 && ...
        ~isnan(trajectories(start_compartment_frame
            , previous_compartment*3-1)))

        compartment_changed = 1;
    else
        compartment_changed = 0;
    end

    % This effectively gives a min_stick lag for
    % adding an

```

```

% entry to compartment_results
x = tracks(start_frame_range - 1, x_column_i);
y = tracks(start_frame_range - 1, y_column_i);
x_next = tracks(start_frame_range, x_column_i);
y_next = tracks(start_frame_range, y_column_i);
% Make sure the coordinates are not nans
if(~isnan(x) && ~isnan(y) && ~isnan(x_next) &&
~isnan(y_next) ...
    && x > 0 && y > 0 && x_next > 0 &&
    y_next > 0 ...
    && start_compartment_frame > 0)
if(compartment_changed == 1)
    % Find closest point to actin
    [line_x line_y] = bresenham8(x, y,
    x_next, y_next);
    line_size = size(line_x, 1);
    smallest_distance = 0;
    first_distance = 0;
    last_distance = 0;
    crossed_skeleton = -1;
    for m=1:line_size
        distance = edm(line_y(m), line_x(m)
        , start_compartment_frame);
        if(m == 1 || distance <
        smallest_distance)
            smallest_distance = distance;
        end
        if(m == 1)
            first_distance = distance;
        elseif(m == line_size)
            last_distance = distance;
        end
        if(skeleton(line_y(m), line_x(m),
        start_compartment_frame) > 0)
            crossed_skeleton = 1;
            if(make_movie == 1)
                trajectory_images(:, :,
                start_compartment_frame) =
                step(lineDrawer,
                trajectory_images(:, :,
                start_compartment_frame), [x
                y x_next y_next]);
            end
        end
    end
end
else
    crossed_skeleton = 0;
end
compartment_results(compartment_result_i,
:) = [j k compartment_changed
crossed_skeleton x y x_next y_next];
compartment_result_i = compartment_result_i
+ 1;

```

```

        if(make_movie == 1)
            trajectory_all_images(:, :,
                start_compartment_frame) = step(
                lineDrawer, trajectory_all_images(:, :,
                start_compartment_frame), [x y x_next
                y_next]);
        end
    end
end
end
end
end
end
end

    frame_offset = frame_offset + 1;
end
if(make_movie == 1)
    disp('Saving movie...');
    save_stack(movie_save_path, trajectory_images);
    save_stack(movie_save_path_all, trajectory_all_images);
    disp('Done.');
```

*% remove empty compartment results*

```

compartment_results((compartment_results(:, 1) == 0), :) = [];
disp('=====');
disp('          SUMMARY          ');
compartment_all = size(compartment_results, 1);
compartment_same = size(find(compartment_results(:, 3) == 0), 1);
percent_same = (compartment_same / compartment_all) * 100;
disp(['Same Compartment: ' num2str(compartment_same) ' (' num2str(
    percent_same) '%)']);
compartment_different = size(find(compartment_results(:, 3) == 1), 1);
percent_different = (compartment_different / compartment_all) * 100;
disp(['Different Compartment: ' num2str(compartment_different) ' ('
    num2str(percent_different) '%)']);
crossed_skeleton_different = size(find(compartment_results(:, 4) == 1 &
    compartment_results(:, 3) == 1), 1);
percent_crossed_different = (crossed_skeleton_different /
    compartment_different) * 100;
disp(['Crossed skeleton (Different Compartment): ' num2str(
    crossed_skeleton_different) ' (' num2str(percent_crossed_different) '
    %)']);
%% Replace with the code that finds the average skeleton % of watershed
skele_percentages = zeros(size(skeleton, 3), 1);
time_points = (1:size(skeleton, 3)) * .2;
for i=1:size(skeleton, 3)
    skele_percentages(i) = (100*size(find(skeleton(:, :, i) == 1), 1)) /
        size(find(~watershedi(:, :, i) == 1), 1);
end
disp(['Mean Skeleton % of Watershed: ' num2str(mean(skele_percentages))
    ]);
disp(['Total: ' num2str(compartment_all)]);
disp('=====');
```

DISTANCE DISTRIBUTIONS. This algorithm takes a multi-image tiff of EDMs and the output from the detection stage of the u-track algorithm (<http://lccb.hms.harvard.edu/software.html>) and generates a file with the distances of the localized particles as determined by the EDM. A control case is also generated using all values in the EDM from a sample of frames in the stack.

```

folder_path = 'C:\Users\jllh\Desktop\7-3-12 Analysis\Distance
Distributions\';
roi = [357 165 1041 709]; % [x_start y_start width height] 82111
track_end_frame = 490;
result_file = [folder_path 'Tracks\Detection_' image_id];
edm_file = [folder_path 'EDMs\' image_id '.tif'];
reconstruction_frames = 100; % number of frames used to construct the
PALM image
sliding_time_window = 10; % sliding time window used for original PALM
image
track_frames_averaged = 1; % frames averaged for QD tracks
track_start_frame = 1; % frame (numbers based on PALM frames) to start
looking at tracks
make_movie = 0;
%% END SETTINGS =====
disp('Loading edm...');
edm = readtimeseries(edm_file, '[1 1]');
edm = dip_array(edm);
if(make_movie == 1)
    track_locs = zeros(size(edm));
end
disp('Loading localizations...');
load(result_file, 'movieInfo');
% 3 columns: (x, y, frame)
chunk_size = 100000;
tracks = zeros(chunk_size, 3);
track_i = 1;
for i=1:size(movieInfo, 1)
    for j=1:size(movieInfo(i).xCoord, 1)
        tracks(track_i, :) = [movieInfo(i).xCoord(j, 1) movieInfo(i).
            yCoord(j, 1) i];

        track_i = track_i + 1;
        if(track_i > size(tracks, 1))
            tracks = [tracks; zeros(chunk_size, 3)];
        end
    end
end
if(track_i <= size(tracks, 1))
    tracks(track_i:end, :) = [];
end
clear movieInfo;
track_num = size(tracks, 1);
tracks(:, 1) = round((tracks(:, 1)-1)*10) - roi(1) + 1;
tracks(:, 2) = round((tracks(:, 2)-1)*10) - roi(2) + 1;
% Go through tracks frame by frame
frame_offset = 1;

```

```

edm_values = [];
for j=track_start_frame:track_end_frame
    disp(['Frame #' num2str(frame_offset)]);

    track_segment_start = (((j-1)*sliding_time_window) + (
        reconstruction_frames / 2))) / track_frames_averaged;
    track_segment_end = track_segment_start;

    disp(['Initial start: ' num2str(track_segment_start)]);

    track_segment_start = track_segment_start - ceil(
        sliding_time_window / (2 * track_frames_averaged)) + 1;
    track_segment_end = track_segment_end + floor(sliding_time_window /
        (2 * track_frames_averaged));
    disp(['Final start-end: ' num2str(track_segment_start) '-' num2str(
        track_segment_end)]);

    segment_particles = find(tracks(:, 3) >= track_segment_start &
        tracks(:, 3) <= track_segment_end);

    for n=1:size(segment_particles, 1)
        x = tracks(segment_particles(n), 1);
        y = tracks(segment_particles(n), 2);

        if ~(x < 1 || y < 1 || x > roi(3) || y > roi(4))
            edm_values = [edm_values; edm(y, x, frame_offset)];

            if (make_movie == 1)
                track_locs(y, x, frame_offset) = 255;
            end
        end
    end

    frame_offset = frame_offset + 1;
end
lipid_values = edm(:, :, round(linspace(1, size(edm, 3), 5)));
lipid_values = lipid_values(:);
format = '';
fid = fopen([folder_path 'Localized\' image_id '.txt'], 'w'); %%
    Filename to write to
format = [format '%12.10f'];
fprintf(fid,[format '\r\n'] ,edm_values');
fclose(fid);
fid = fopen([folder_path 'Control\' image_id '.txt'], 'w'); %% Filename
    to write to
format = [format '%d'];
fprintf(fid,[format '\r\n'] ,lipid_values');
fclose(fid);
if (make_movie == 1)
    save_stack([folder_path 'movie_ustrack_' image_id '.tif'], track_locs
        );
end

```

## IMAGEJ MACROS

CORRECT ZOOM MACRO. Often it was necessary to zoom an image to overlay super-resolution data, such as PALM images or single particle tracking trajectories, on a raw image. This algorithm accurately resizes an image so that the sub-pixel trajectories can be accurately overlaid.

```
scale = 10;
interp = "None";
h=getHeight()*scale;
w=getWidth()*scale;
tweak_h = h - scale/2;
tweak_w = w - scale/2;
run("Scale...", "x="+scale+" y="+scale+" z=1.0 width="+w+" height="+h+"
    depth="+nSlices+" interpolation="+interp+" average process create
    title=test_small_spot_stack-1.tif");
run("Canvas Size...", "width="+tweak_w+" height="+tweak_h+" position=
    Bottom-Right zero");
run("Canvas Size...", "width="+w+" height="+h+" position=Top-Left zero
    ");
run("Canvas Size...", "width=4220 height=1300 position=Top-Left zero");
```

## APPENDIX C

### TROUBLESHOOTING

#### PALM IMAGING.

*Image background is too high.* High background is often due to improper laser alignment. Verify that lasers are properly aligned. Imaging in TIRF can reduce background. Cell autofluorescence can contribute to background noise in some cases, requiring the use of fluorophores that do not emit light at similar wavelengths to the autofluorescence. Check the imaging buffer, as some imaging buffers may contain particles that fluoresce.

*Initial PALM probe activation is too high.* Take care to keep the cell dishes protected from light after transfection, as that can cause premature activation of PALM probes. If initial activation is still too high, expose the cell to the excitation laser until the activated probes bleach, then begin imaging.

*Too many/few probes activate during image acquisition.* Multiple factors can be tuned to adjust the amount of activated probes to the desired density:

- Reduce/increase the activation laser power to alter the pool of activated probes.
- Reduce/increase the excitation laser power to bleach the activated pool less/more quickly.
- Reduce/increase the amount of plasmid used in transfection to alter the total probe pool.
- If imaging in TIRF, reduce/increase the TIRF angle to tweak the imaging depth.

## APPENDIX D

# PROTOCOLS

### CELL TRANSFECTION FOR PALM IMAGING.

- (1) Check cells for health. Cells should be confluent but not over-crowded. When using HEK cells, use between passage 42 and 49.
- (2) Prepare bioptechs cell dishes by coating dishes with Matrigel and allowing them to incubate at 37 deg C for at least 15 minutes.
- (3) Remove plasmids from freezer and allow them to thaw at room temperature.
- (4) Place media (Phenol red free DMEM + 10% FBS), trypsin, and optimum in a water bath and warm to 37 deg C.
- (5) Prepare plasmid solution by combining 200  $\mu$ L of optimum with plasmids. Place in 37 deg C water bath.
  - (a) Kv2.1-loopBAD: 2.63  $\mu$ L for a total of 3  $\mu$ g of plasmid
  - (b) Kv1.4-loopBAD: 2.44  $\mu$ L for a total of 3  $\mu$ g of plasmid
  - (c) birA: 1.33  $\mu$ L for a total of 1  $\mu$ g of plasmid
  - (d) ABP-tdEosFP: 6  $\mu$ L for a total of 3  $\mu$ g of plasmid
- (6) Aspirate media from the 10 mL cell dish and immediately add 4 mL of trypsin to the dish.
- (7) Place dish in incubator for 1 min.
- (8) Gently tap the sides of the dish to dislodge the cells from the bottom of the dish.
- (9) Add 10 mL of media to the dish.
- (10) Place the solution into a 15 mL conical.
- (11) Centrifuge at 1.5 rpm for 3 min to form a pellet of cells.
- (12) Carefully aspirate the media out of the conical, leaving the cell pellet intact.
- (13) Quickly add the plasmid solution into the conical, pipetting up and down a few times to suspend the cells in the plasmid solution.
- (14) Place the solution containing the plasmids and the cells in an electroporation cuvette.
- (15) Electroporate using the electroporator's built-in settings for HEK293 cells.
- (16) Replace the matrigel in the bioptechs cell dishes with 1 mL of media.
- (17) Place a small drop of the electroporated cells in each cell dish and gently swish the cell dishes to spread the cells.
- (18) Incubate at 37 deg C for at least 12 hours prior to imaging.

#### IMAGING PREPARATION FOR COMBINED SPT AND PALM IMAGING.

- (1) Place HEK Imaging Saline (HIS) in water bath and warm to 37 deg C.
- (2) Prepare the quantum dot solution for labeling Kv channels by combining 1 mL HIS with 10  $\mu\text{g}$  of BSA and 1  $\mu\text{g}$  QD705. Dilute QD further if needed.
- (3) Turn lights off to avoid activation of fluorescent probes.
- (4) Select a bioptechs cell dish for imaging by examining the dishes under a light microscope with a yellow light filter. Avoid dishes with cells that are too crowded (growing over each other) or too sparse (difficult to find a cell within a viewing region). Avoid dishes infected with bacteria.
- (5) Once a suitable dish has been chosen, rinse the dish gently by aspirating off the original solution while adding 1 mL of HIS, being careful never to dry the dish. Repeat this process once.
- (6) Aspirate off the HIS and immediately replace with 500  $\mu\text{L}$  of the QD705 solution.
- (7) Incubate at 37 deg C. Vary incubation time between 5 min and 20 min based on the desired density of QD.
- (8) Aspirate off the QD705 solution and rinse the dish several times with HIS to remove any remaining unbound QD.
- (9) Place 1 mL of HIS into the dish.

POLYSTYRENE BEAD DISH FOR COLOCALIZATION.

- (1) Clean a Bioptechs dish with 10% bleach, 1% alconox, and DI water. Rinse with IPA and dry with nitrogen to avoid leaving a residue.
- (2) Prepare a solution of 1:100,000 2  $\mu\text{m}$  polystyrene beads (Polyscience) in DI water. Vortex solution briefly then place 50  $\mu\text{L}$  of the solution onto the middle of a glass slide.
- (3) Allow to rest for at least 20 minutes then rinse with IPA and dry with nitrogen.

## IMAGE ACQUISITION FOR COMBINED SPT AND PALM IMAGING.

- (1) Select emission path filters suitable for the probes being imaged. For the experiments described in this thesis using ABP-tdEosFP and QD705:
  - (a) Dichroic: 6601pxr
  - (b) ABP-tdEosFP Filters: 600/60M and 473RS
  - (c) QD705 Filter: 710/40
- (2) Set up initial ND filters in place to avoid damage to cells or excessive activation while searching for a cell to image. Typically:
  - (a) Blue and Green Laser: ND 1
  - (b) Violet Laser: ND 4
- (3) Verify that the shutters are in place and are closed for each laser. Typically:
  - (a) Blue: Shutter 1
  - (b) Violet: Shutter 2
  - (c) Green: Shutter 3
- (4) Open Andor iXon2. This program controls the shutters and camera and is used to acquire and save images.
- (5) Place the lasers in TIRF.
- (6) Using a polystyrene bead dish and white light illumination, find a region where the beads are well separated and regularly distributed, particularly along the horizontal axis (to provide as much information as possible about the dichroic astigmatism). Move the objective slightly below focus so that multiple concentric rings are clearly visible around each bead. Acquire images of both channels simultaneously. Typically a stack of 100 images with an exposure time of 100 ms is sufficient. These images will be used for image registration which will allow the QD705 and ABP-tdEosFP signals to be accurately overlaid.
- (7) Replace the polystyrene bead dish with a cell dish.
- (8) Once the system is in focus, activate the green laser by opening the associated shutter. Search for a transfected cell with the desired density of QD. A typical transfected cell will respond with an initial burst of emission in the tdEosFP channel before hitting a plateau.
- (9) Briefly, check the cell under filtered white to verify health. Avoid imaging cells that appear sickly (typically round and lifting off of the dish) or that are overly crowded by other cells. Once cell health is verified, turn off the white light.
- (10) Verify that the ND filters are in place. Typically, start with ND 4 on violet and ND 1 on green.
- (11) Set the ROI (region of interest) to have a width that spans both channels, but a maximum height of 130 px. Restricting the height allows for high imaging speeds. Create a loop within the protocol which sets the channel such that the camera exposure time is set to 20 ms, camera gain to 250, and the green and violet lasers are open. After the channel is set, add a "SNAP" command and verify that the SNAP has the frame overlap/transfer enabled. Set the number of times the loop repeats to the desired number of frames to acquire (2500 typically works well).
- (12) Run the protocol. Monitor the acquired images closely and decrement the ND by 0.5 when the density of activated PALM probes is too low.
- (13) Save the image stack.

(14) Repeat steps 8-12 for each cell. Before switching to a new cell dish, repeat steps 6-7.

## COMBINED SPT AND PALM RECONSTRUCTION.

- (1) Determine the x and y offsets between the left and right channels
  - (a) In LabView, open the Autocorrelation Moving multiple ROI 2 VI. Specify settings as follows prior to running the vi:
    - (i) Pattern: \*.tif
    - (ii) ROI Size (pixels): 30
    - (iii) Start frame: 0
    - (iv) End Frame: determined by your data. For a 100 frame data set, 99
    - (v) Select Multi Image Folder: specify the directory which contains multiple tiffs of white light bead images of both channels
  - (b) Upon running the vi, you will be prompted to select the location of a bead. First, select the location of a bead in the left channel of the image and click “Check ROI” to finalize the selection. Next, select the same bead as it appears in the right channel of the image and click “Check ROI”. Repeat this process, selecting first the left version of the bead, then the right version of the same bead, until all distinct beads have been selected. Avoid beads which overlap with other beads or that are cut off by the image border.
  - (c) Click “Continue”. After the locations of each bead in each frame are calculated by the program, you will be prompted to save the resulting text file which contains that information.
  - (d) In MATLAB, open find\_offsets.m. Set filename to the text file saved from step 1c. Run the program.
  - (e) Record y\_offset as well as the linear fit for x\_offset. These two offsets will be used to modify the localization of the right channel data (typically ABP-tdEosFP emissions) so that they accurately overlay with the left channel data.
- (2) Generate the PALM reconstructions
  - (a) PALM reconstructions are generated using a MATLAB application provided by Dr. Keith Lidke. Refer to SR\_Demo/RC2/Docs/ExampleCodeDoc.pdf for information on system requirements as well as how to configure your system to run the application.
  - (b) Run `localize_particles_with_merge(image_id, use_filter, base_folder_name, read_folder_name, x_offset_1, x_offset_2, y_offset)`, where:
    - (i) image\_id: name of the file containing the tiff stack of raw data ONLY (no directory, no .tif extension)
    - (ii) use\_filter: 1 to process data with a 0.8 px Gaussian filter (function `gaussf`) prior to localization, 0 otherwise
    - (iii) base\_folder\_name: the folder where the results should be saved
    - (iv) read\_folder\_name: the folder where the raw data is located
    - (v) x\_offset\_1: as determined in step 1e – the “A” in the linear fit  $Ax + B$  (unitless)
    - (vi) x\_offset\_2: as determined in step 1e – the “B” in the linear fit  $Ax + B$  (px)

- (vii) `y_offset`: as determined in step 1e
  - (c) The function will localize all of the particles and store the results in `SRtest`, a variable which is saved under `base_folder_name/results/image_id.mat`. Then, the localizations in the right channel will be modified using the x and y offset information provided. Finally, reconstructions will be generated. By default, the reconstructions will consist of 100 frames with a 100 frame sliding time window. This setting is useful for screening for the best data sets – ie, the data sets with the highest density of localized particles. Once the best data sets have been determined, it is recommended to repeat the reconstruction process with a 10 frame sliding time window for more continuity between frames. To do so, scroll to the last line in the `localize_particles_with_merge` function and alter the third argument in `SRtest.SaveMovie` from 100 to 10.
- (3) Calculate trajectories for QD data
- (a) SPT was completed using u-track. To prepare a data set for use with u-track, open the tif stack in ImageJ. Crop the image such that only the left channel is visible. Be very careful to begin the crop at 0,0 (the top left corner). Beginning the crop at any other point will cause systematic errors in the PALM data overlay. Once the stack is cropped, go to File -> Save As -> Image Sequence, and save the cropped images in a separate folder.
  - (b) Open u-track/scriptDetectGeneral. Set:
    - (i) `movieParam.imageDir`: The directory containing the individual images, as determined in step 3a
    - (ii) `movieParam.filenameBase`: The prefix for each image, as determined in step 3a upon saving the image sequence
    - (iii) `movieParam.firstImageNum`: Typically 1, indicates which image to start at
    - (iv) `movieParam.lastImageNum`: Typically the total number of images in the data set
    - (v) `movieParam.digits4Enum`: Determined by how many images are in the data set. Typically, this will be 4 for data sets of 1000-9999 images
    - (vi) `saveResults.dir/.filename`: Where to save the detection output
    - (vii) `detectionParam.visual`: 0
    - (viii) `detectionParam.bitDepth`: 16
    - (ix) `detectionParam.psfSigma`: 1.278 for QD
    - (x) `detectionParam.integWindow`: 0
    - (xi) `detectionParam.alphaLocMax`: 0.005
    - (xii) `detectionParam.doMMF`: 0
  - (c) `detectionParam.alphaLocMax` must be carefully chosen for each individual data set, as the optimal value varies based on the images provided. To determine the optimal setting, change `movieParam.lastImageNum` to 1 and `detectionParam.visual` to 1. Start with `detectionParam.alphaLocMax` at 0.005. Run the script and examine the image that is displayed. If

noise is detected and treated as a particle, reduce the value of `detectionParam.alphaLocMax` by an order of magnitude. If the majority of the particles to be tracked are not detected, increase the value by an order of magnitude. Once the optimal value for `detectionParam.alphaLocMax` is found, restore the value of `movieParam.lastImageNum` and set `detectionParam.visual` to 0.

- (d) Run `scriptDetectGeneral`. This algorithm will localize all particles within the dataset.
- (e) Open `scriptTrackGeneral`. Typical settings are:
  - (i) `gapCloseParam.timeWindow`: 3
  - (ii) `gapCloseParam.mergeSplit`: 1
  - (iii) `gapCloseParam.minTrackLen`: 15
  - (iv) `gapCloseParam.diagnostics`: 0
  - (v) `parameters.linearMotion`: 2
  - (vi) `parameters.minSearchRadius`: 2
  - (vii) `parameters.maxSearchRadius`: 5
  - (viii) `parameters.brownStdMult`: 3
  - (ix) `parameters.useLocalDensity`: 1
  - (x) `parameters.brownScaling`: [0.5 0.01]
  - (xi) `parameters.timeReachConfB`: 4
  - (xii) `parameters.ampRatioLimit`: [0.7 4]
  - (xiii) `parameters.lenForClassify`: 15
  - (xiv) `parameters.linScaling`: [0.5 0.01]
  - (xv) `parameters.maxAngleVV`: 30
  - (xvi) `parameters.gapPenalty`: 1
  - (xvii) `parameters.resLimit`: 1.7
  - (xviii) `saveResults.dir/.filename`: Where to save the tracking output
- (f) Run `scriptTrackGeneral`. This will process the localizations from step 3d and connect them to generate trajectories.
- (g) To convert the output from `scriptTrackGeneral` into the typical XYI format used by many of our algorithms, open `convert_and_write_trajectories_utrack`. Modify `id_number` and `track_name` to match the desired text file save name.

## GENERATING EDM BASED ON DETECTED IMAGE EDGES.

- (1) Open image in ImageJ.
- (2) Go to Analyze -> Set Measurements. Check Min & max gray value and click OK. This will make sure that the min and max gray value will be calculated in step 3.
- (3) Select the Rectangle tool and draw a box in a background only area of the image. Next, go to Analyze -> Measure (CTRL+M) to find the max value within the box. This will provide a rough estimate as to the value of the low level noise in the image.
- (4) Go to Process -> Math -> Subtract and subtract the max background value found in Step 3 from the image. This will remove the low level noise from the image.
- (5) Click Edit -> Selection -> Select None (CTRL+SHIFT+A) to remove the rectangle.
- (6) A Gaussian blurred version of the original image can be subtracted from the original image to remove uneven background illumination. Go to Image -> Duplicate... (CTRL+SHIFT+D). If the image is a stack, check the box to duplicate the entire stack. Click OK.
- (7) Go to Process -> Filters -> Gaussian Blur. Set Sigma (Radius) to 20. Click OK.
- (8) Go to Process -> Image Calculator. For Image1, select the original image. For Operation, select Subtract. For Image2, select the Gaussian Blurred image. Check "Create new window" and click OK. This will remove uneven background illumination.
- (9) Select the resulting image.
- (10) To enhance the feature edges in an image to allow accurate thresholding, a Laplacian filter should be applied. Click Process -> Filters -> Convolve. Use the following 5x5 Laplacian kernel:
$$\begin{array}{ccccc} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{array}$$
- (11) To increase the precision of the distance map, which is particularly helpful when using the distance map with subpixel particle localization, the image can be zoomed. Run the Correct\_Zoom.txt macro with a zoom of 10. Set Interpolation to Bicubic.
- (12) Go to Image -> Adjust -> Threshold... (CTRL+SHIFT+T). Select Threshold values which exclude noise without compromising feature detail. Click Apply.
- (13) To generate the Euclidean distance map (EDM), go to Process -> Binary -> Make Binary. Next, click Edit -> Invert (CTRL+SHIFT+I). Finally, click Process -> Binary -> Distance Map

#### FLOW CELL BUFFER PREPARATION.

- (1) Buffer T150 + D-glucose (\*NOTE: Need final salt concentration of at least 75 mM or RNA polymerases likely to bind each other)
  - (a) Prepare 20 mM Tris-HCl (pH 8.0) in DEPC water (100  $\mu$ L 2M stock Tris in 10 mL DEPC water)
  - (b) Add 150 mM NaCl (8.77 g/L for 150 mM => 87.7 mg per 10 mL)
  - (c) Add D-glucose to 1.6% w/v (16 g per liter => 160 mg per 10 mL)
  - (d) Store at 4 deg C
- (2) 100X Gloxy (1 mL) (Provides the oxygen reduction system)
  - (a) Add 112  $\mu$ L of 34 mg/mL catalase to 882  $\mu$ L of buffer T50 for a concentration of 4 mg/mL
  - (b) Add 100 mg of glucose oxidase for a concentration of 100 mg/mL
  - (c) Tap to dissolve glucose oxidase, but do not vortex
  - (d) Centrifuge for 1 minute
  - (e) Collect the supernatant
  - (f) Store at 4 deg C for up to several months and limit exposure to air
- (3) Trolox (~2 mM; Typically prepared day of experiment) (Acts as a triplet state quencher)
  - (a) Add 30 mg Trolox to 10 mL DEPC water
  - (b) Vortex for 1 minute
  - (c) Filter through a 0.2  $\mu$ m syringe filter
  - (d) Store in the dark at 4 deg C for up to 15 days
- (4) Imaging Buffer (Final Concentrations ~1 mM Trolox, 10 mM Tris-HCl, 75 mM NaCl, 0.8% w/v D-glucose, 0.04 mg/mL catalase, 1 mg/mL glucose oxidase)
  - (a) Combine 5 mL 2 mM Trolox Solution with 5 mL Buffer T150 + D-glucose
  - (b) Adjust pH to 8.0 using HCl
  - (c) Place under vacuum for ~10 minutes to remove air
  - (d) Immediately before measuring, add 100X gloxy to 1X final concentration (100  $\mu$ L gloxy in 10 mL Buffer)

#### TYPICAL FLOW CELL PROTOCOL.

**TIB:** ~1 mM Trolox + T50 Buffer (pH 8.0) + D-Glucose + .1 mg/mL BSA

**TGIB:** TGIB: TIB + 1X Gloxy

- (1) Attaching a syringe filled with TIB to tubing which connects to the HPLC valve inlet.
- (2) Prep the tubing and the HPLC valve by injecting ~1 mL TIB.
- (3) Place the syringe in the syringe pump and connect tubing from the HPLC valve outlet to the flow cell inlet.
- (4) Using the LabView software, inject 500  $\mu$ L of TIB at 500  $\mu$ L/min and incubate for 5 min.
- (5) Record images for several minutes to establish baseline fluorescence prior to injecting sample.
- (6) To verify specificity of the flow cell, inject a 100  $\mu$ L of a 1 nM sample of a fluorescently labeled protein. Flush 500  $\mu$ L of TIB through the flow cell, then record images for several minutes to determine level of nonspecific binding. If the density of nonspecific binding is low, continue.
- (7) Inject 100  $\mu$ L 0.2 mg/mL neutravidin at 100  $\mu$ L/min and incubate for 5 minutes.
- (8) Inject 1 mL of TGIB at 500  $\mu$ L/min to replace TIB.
- (9) Inject 100  $\mu$ L of sample (typically 50 pM, but adjust based on desired density) and incubate for 30 minutes.
- (10) Inject 200  $\mu$ L of TGIB at 100  $\mu$ L/min to remove remaining unbound sample.