2022

Investigating Spatiotemporal Kinetics, Dynamics, and Mechanism of Exosome Release

Anarkali Mahmood

University of Denver

Follow this and additional works at: https://digitalcommons.du.edu/etd

Part of the Biochemistry Commons, and the Chemistry Commons

Recommended Citation
https://digitalcommons.du.edu/etd/2066

This Dissertation is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu,dig-commons@du.edu.
Investigating Spatiotemporal Kinetics, Dynamics, and Mechanism of Exosome Release

Abstract
Exosomes are small lipid-based vesicles that can carry biomolecules from one cell to another. While exosomes are crucial to maintain homeostasis in healthy cells, they are exploited by unhealthy cells to aid disease progression. Exosomes likely facilitate disease progression via the transfer of disease-causing biomolecules from unhealthy to healthy cells. Exosomes are generated in Multivesicular endosomes (MVEs) and are then secreted into the extracellular space to travel to other cells. Despite being a crucial step, very little is known about exosomes release mechanism and dynamics. To further our understanding of exosomes, specifically their secretion, my work has focused on investigating spatiotemporal dynamics and kinetics of both MVEs and MVE interacting proteins. We visualized and characterized single MVE fusion events in lung cancer cells (A549) using CD63 fluorescent probes and total internal reflection fluorescence (TIRF) microscopy. The kinetics of release, or loss of fluorescence post-fusion, can relay information about the fate of exosomes. Using kinetic analysis we determined that a portion of exosomes are free to diffuse away from the fusion site, but some exosomes remain attached to the surface of the secreting cell. One challenge of measuring constitutive fusion events in a single vesicle fusion assay is the tedious process of manually finding and analyzing fusion events that occur at random points in time and in a relatively slow fashion (~1-3 events per minute). To overcome this, we developed a fast and fully automated algorithm to detect and analyze fusion events. To uncover some of the protein regulators we investigated the involvement of different SNAREs, and SNARE interacting proteins such as Syntaxins, SNAPs, and VAMP7 using in single vesicle fusion assays. Studying MVE fusion, exosome release, and potential fusion regulatory proteins provides insights into the exosome release mechanism, expands our understanding of the fusion process, and gives direction for future studies to identify potential therapeutic targets that can be used to modulate exosome secretion.

Document Type
Dissertation

Degree Name
Ph.D.

Department
Chemistry and Biochemistry

First Advisor
Michelle K. Knowles

Second Advisor
Daniel Linseman

Third Advisor
Scott Horowitz

Keywords
Biochemistry, Exosomes

Subject Categories
Biochemistry | Biochemistry, Biophysics, and Structural Biology | Chemistry
Investigating spatiotemporal kinetics, dynamics, and mechanism of exosome release

A Dissertation
Presented to
the Faculty of the College of Natural Sciences and Mathematics

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Anarkali Mahmood

June 2022

Advisor: Michelle K. Knowles
ABSTRACT

Exosomes are small lipid-based vesicles that can carry biomolecules from one cell to another. While exosomes are crucial to maintain homeostasis in healthy cells, they are exploited by unhealthy cells to aid disease progression. Exosomes likely facilitate disease progression via the transfer of disease-causing biomolecules from unhealthy to healthy cells. Exosomes are generated in Multivesicular endosomes (MVEs) and are then secreted into the extracellular space to travel to other cells. Despite being a crucial step, very little is known about exosomes release mechanism and dynamics. To further our understanding of exosomes, specifically their secretion, my work has focused on investigating spatiotemporal dynamics and kinetics of both MVEs and MVE interacting proteins. We visualized and characterized single MVE fusion events in lung cancer cells (A549) using CD63 fluorescent probes and total internal reflection fluorescence (TIRF) microscopy. The kinetics of release, or loss of fluorescence post-fusion, can relay information about the fate of exosomes. Using kinetic analysis we determined that a portion of exosomes are free to diffuse away from the fusion site, but some exosomes remain attached to the surface of the secreting cell. One challenge of measuring constitutive fusion events in a single vesicle fusion assay is the tedious process of manually finding and analyzing fusion events that occur at random points in time and in a relatively slow fashion (~1-3 events per minute). To overcome this, we developed a fast
and fully automated algorithm to detect and analyze fusion events. To uncover some of the protein regulators we investigated the involvement of different SNAREs, and SNARE interacting proteins such as Syntaxins, SNAPs, and VAMP7 using in single vesicle fusion assays. Studying MVE fusion, exosome release, and potential fusion regulatory proteins provides insights into the exosome release mechanism, expands our understanding of the fusion process, and gives direction for future studies to identify potential therapeutic targets that can be used to modulate exosome secretion.
ACKNOWLEDGEMENTS

First, I am very grateful to my advisor, Dr. Michelle Knowles for her patience, and guidance through the ups and downs of my graduate school. Without her support and guidance, none of this would have been possible. I am thankful to her for helping me with research, writing of this thesis, and for other opportunities she provided me with to further my career as a scientist. I would also like to thank the rest of my thesis committee members, Dr. Erich Chapman, and Dr. Scott Horowitz for their feedback during committee meetings.

I am also grateful to the Knowles Lab members (past and present) for their helpful comments and help with research. I am especially grateful to Alan Weisgerber and Broderick Bills. I am also thankful to my undergraduate mentees Rida Noor, Amira Reyad, and others for their contributions to my research projects. Lastly, I would like to acknowledge the support of my family and friends during my Ph.D.
# TABLE OF CONTENTS

## CHAPTER 1. EXTRACELLULAR VESICLES ................................................................. 1

1.1 Overview ........................................................................................................... 1
1.2 Extracellular Vesicles: Microvesicles and exosomes ...................................... 2
1.3 Diverse roles of exosomes ............................................................................... 4
1.4 Exosome and disease progression .................................................................... 5
1.5 Exosome Biogenesis and MVE fate ................................................................. 7  
\hspace{3ex} 1.5.1 MVE: Degradation or Secretion ......................................................... 8
1.6 MVE journey to the plasma membrane ............................................................ 9
1.7 MVE Fusion and exosome release ..................................................................... 11  
\hspace{3ex} 1.7.1 Exosome secretion ............................................................................... 13
1.8 Objectives of this work .................................................................................... 14

## CHAPTER 2 MATERIAL AND METHODS................................................................. 16

2.1 Cell Culture, Transfections, and labelling ......................................................... 16
2.2 Total Internal Reflection Fluorescence (TIRF) Microscopy .............................. 18
2.3 Image Analysis .................................................................................................. 19
2.4 Two-color TIRF imaging of docking and fusion: .............................................. 22
2.5 Fluorescence Recovery After Photobleaching (FRAP) .................................... 23
2.6 Modeling of MVE Fusion Decays: ................................................................. 23

## CHAPTER 3 CHARACTERIZATION OF EXOSOME SECRETION KINETICS ...... 26

3.1 Declaration of collaborative work ....................................................................... 26
3.2 Overview ............................................................................................................ 27
3.3 Introduction ....................................................................................................... 29
3.2 Results .............................................................................................................. 33  
\hspace{3ex} 3.2.1 Automated detection and quantification of MVE fusion events ............. 33
\hspace{3ex} 3.2.2 MVE fusion leads to exosome release and deposition of CD63 on plasma membrane .............................................................. 33
\hspace{3ex} 3.2.3 MVE fusion is temperature dependent. ................................................. 48
\hspace{3ex} 3.2.4 Simulation of exosome release, tethering, and CD63 membrane diffusion can account for the experimentally observed kinetics .................................................. 48
\hspace{3ex} 3.2.5 MVEs are docked prior to fusion .......................................................... 48
3.3 Discussion and Conclusion ............................................................................... 58
# LIST OF FIGURES

## CHAPTER 1

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extracellular vesicles (EVs) of different intracellular origins</td>
</tr>
<tr>
<td>2</td>
<td>Stages of Exosome Secretion. Exosome secretion begins with</td>
</tr>
<tr>
<td>3</td>
<td>Protein regulators of MVE docking, fusion and tethering</td>
</tr>
</tbody>
</table>

## CHAPTER 2

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>MVE fusion events were detected using CD63-pHluorin</td>
</tr>
<tr>
<td>5</td>
<td>Schematic overview of the automated data processing and analysis used</td>
</tr>
<tr>
<td>6</td>
<td>Differential movies are calculated to enhance the fusion events relative to stationary vesicles</td>
</tr>
<tr>
<td>7</td>
<td>Potential fusion events found in our analysis can be categorized as A) vesicle fusion B) vesicle motion or C) vesicle docking</td>
</tr>
<tr>
<td>8</td>
<td>Automated analysis of fusion events</td>
</tr>
<tr>
<td>9</td>
<td>MVE fusion events expand in time</td>
</tr>
<tr>
<td>10</td>
<td>Multiple kinetic modes are observed in MVE fusion events</td>
</tr>
<tr>
<td>11</td>
<td>Increase in temperature increases the percent of two component decay curves</td>
</tr>
<tr>
<td>12</td>
<td>Diffusion rate of CD63-pHluorin in plasma membrane are similar at 23C and 37C</td>
</tr>
<tr>
<td>13</td>
<td>Kinetics of single MVE fusion events depend on temperature</td>
</tr>
<tr>
<td>14</td>
<td>Rate of photobleaching does not account for the magnitude of fusion decay rate for fusion events</td>
</tr>
<tr>
<td>15</td>
<td>Fusion events were simulated by allowing CD63 particles to leave the center of the fusion site as tethered exosomes, free exosomes or membrane diffusion</td>
</tr>
<tr>
<td>16</td>
<td>Most MVEs are docked before fusion</td>
</tr>
</tbody>
</table>

## CHAPTER 3

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>Visualization of Syntaxin4 clusters and CD63-PHluorin using TIRF microscopy</td>
</tr>
<tr>
<td>18</td>
<td>Syntaxin4 is present at MVE fusion site</td>
</tr>
<tr>
<td>19</td>
<td>VAMP7 and SNAP23 are present at MVE fusion site</td>
</tr>
<tr>
<td>20</td>
<td>Knockdown of Syntaxin4 decreases MVB fusion events</td>
</tr>
</tbody>
</table>
CHAPTER 1. EXTRACELLULAR VESICLES

1.1 Overview

Extracellular vesicles (EV), such as exosomes and ectosomes, facilitate an important and ubiquitous mode of intercellular communication system in multicellular organisms[1-3]. EVs are composed of a lipid and protein membrane composition that renders them highly fusogenic, making them excellent tools for transfer of biomaterial. These vesicles help maintain homeostasis through intercellular transfer of biomolecules including nucleic acids, proteins, amino acids, and toxins. Interestingly, this mode of communication is also exploited to aid disease-progression via transfer of disease inducing material from unhealthy to healthy cells. EV secretion is upregulated in several illnesses including Alzheimers, Parkinsons, and various cancers[4]. Thus, secretion of these vesicles is a potential checkpoint that can be harnessed to modulate disease-progression. However, manipulation of EV secretion requires a complete understanding of the mechanistic details of this process. Much progress had been made in understanding EV biology, including their biogenesis, secretion, and post-secretion fate. However, many details remain unknown. Better understanding of extracellular vesicles not only holds enormous promises for medical and pharmaceutical applications, but is also of invaluable impact for fundamental research.
1.2 Extracellular Vesicles: Microvesicles and exosomes

The term EV is used to collectively address to a heterogenous group of lipid-protein based vesicles with two main subdivisions: exosomes and microvesicles (Figure 1)[5]. These EVs enable highly efficient transfer and exchange of biomaterial both between neighboring and distal cells. While exosomes size ranges from 50-100 nm, microvesicles can be much wider, ranging from 50 -1000 nm (in diameter) [4]. Both exosomes and microvesicles have been shown to support intercellular communication in healthy and diseased cells. From aiding homeostasis to transfer of oncogenes, these vesicles can act as instrumental players in physiological and pathological processes.
Figure 1: Release of Extracellular vesicles including microvesicles and exosomes. Schematic representation exosomes and microvesicles being released from a cell showing that microvesicles bud from plasma membrane and exosomes are formed inside larger vesicles called MVEs.

One of the biggest challenges in the EV field is characterization of different EV populations. Identification of different EVs is difficult as in addition to overlap in vesicle size, exosomes and microvesicles also have overlapping membrane components. For example, both exosome and microvesicle membranes have integrins, tetraspanins and high concentrations of cholesterol and sphingomyelin. However, differences in membrane composition between EVs are slowly being unraveled. Studies have shown that in contrast to exosomes, microvesicle membrane can includes matrix metalloproteinase MT1-MMP, glycoprotein receptors (GP1b and GPIIb/ GPIIa), adhesion protein P-selectin and integrin Mac-1[6-9]. Interestingly, not all microvesicles are composed of all (aforementioned) proteins. Therefore, though EVs are currently binned in two categories, heterogeneity within each category points towards the addition of further classifications as we learn more about these vesicles. Heterogeneity in EV composition is accompanied by differences in enclosed content and most likely plays an important role in determining post-fusion EV targets. The current EV classification is based on differences in biogenesis of microvesicles and exosomes.

Microvesicles are formed by accumulation of cargo at plasma membrane (PM) microdomains [6, 7]. Cargo accumulation at microdomains causes outward bending and is followed by vesicle fission and release of the cargo enclosed microvesicle into
extracellular space (Figure 1). Ectosome biogenesis likely involves rearrangement of phospholipids via Ca2+ dependent enzymes called flipases and flopases. Lipid rearrangement can lead to formation of microdomains and mark the site of ectosome biogenesis. While microvesicles are formed at the plasma membrane and are immediately released via vesicle fission, exosomes are created as intraluminal vesicles (ILVs) inside MVEs. These MVEs are located at perinuclear cytoplasm and remain there until signaled to move the plasma membrane (Figures 1 and 2) [6, 10].

MVE trafficking in the cytoplasm is followed by MVE-plasma membrane fusion and the concomitant release of exosomes into the extracellular space (Figure 2) [4-6]. MVE fusion and exosome release mechanism and its regulation are not complexly understood. In this work, we are primarily investigating MVE fusion kinetics and potential fusion regulatory machinery including SNARE and SNARE interacting proteins. Therefore, the remainder of this chapter is focused on more details relating to MVEs and exosomes. It is important to note that the majority of exosome related studies have been conducted on sEV (small extracellular vesicles). sEVs are a mixture of exosomes and microvesicles, smaller than 150 nm in diameter [11, 12]. When referring to such studies, the term sEVs will be used.

### 1.3 Diverse roles of exosomes

As mentioned earlier, the primary characteristic used to differentiate microvesicles from exosomes is the mode of biogenesis for each vesicle. Discovery of
small vesicles that form within multivesicular endosomes, and their designation as exosomes took place in the early 1980s [6, 13]. However, exosomes were concluded to only play the role of cellular waste-carriers that were secreted to be disposed of and interest in exosomes was cast aside for over a decade. In the 1990s, involvement of EVs in modulation of immune response was first observed. In this work, sEVs secreted from B-cells were shown to be enriched in major histocompatibility complexes (MHC) II class molecules[14]. These sEVs were able to activate T-cells. This was followed by reports of sEV mediated tumor suppression by dendritic cells [15]. In addition to regulation of immune response, EVs have been reported to play a role in many other biological processes including metabolism and cell motility [15-17]. The discovery of exosomes as immune response mediators and potential immunotherapeutic agents revived overdue interest in these small extracellular vesicles and a diverse array of not only physiological, but also pathological roles played by exosomes came to light.

1.4 Exosome and disease progression

The discovery of exosome involvement in disease progression was an important milestone for the field as it not only led to advances in further unraveling this cellular process, but also opened new avenues of research aimed at early disease detection and hindering exosome-mediated disease progression. Exosomes have been shown to participate in several illnesses including cancer, neurodegeneration, and viral spread [18, 19]. Additionally, EVs carrying tumor-associated, or misfolded protein aggregates can be detected in bodily fluids as biomarkers of disease [4, 20].
Exosomes can carry oncogene up-regulators, including mRNAs, miRNAs, lncRNAs, proteins, and lipids that can aid cancer progression by various mechanisms including development of pre-metastatic niches [21]. Exosome secretion is upregulated in cancer cells, where ILVs are loaded with disease-inducing genetic material and then directed towards the plasma membrane to release that genetic material for uptake by surrounding healthy cells. Many different cancer cell types have exhibited this behavior – including breast, lung, uterine, skin, ovarian and prostate cancers[22-24]. Several studies have demonstrated that exosomes secreted by tumor cells can upregulate tumorigenic properties in otherwise healthy cells and that blocking exosome release in tumor cells suppresses tumorigenic properties[5, 25, 26]. Hence, the secretion of exosomes plays a crucial role in the progression and eventual metastasis of several types of cancers.

EVs also play a role in several neurodegenerative diseases including Parkinsons and Alzheimers. EVs can carry and spread pathological amyloid proteins. Amyloid proteins can be delivered as intraluminal EV cargo, e.g. alpha-synuclein or EV membrane can be enriched with the disease inducing material such as amyloid-B peptide [7, 27, 28]. Furthermore, several viruses, including SARS-COV2, Vaccinia, and Hepatitis virus have been shown to hijack MVE pathway to aid their progression and proliferation [18].

Therefore, studying MVE fusion, exosome secretion, and proteins integral to this process will provide insights about the mechanism and identify potential therapeutic
targets that can be harnessed to hinder exosome-mediated disease progression in a wide range of illnesses.

### 1.5 Exosome Biogenesis and MVE fate

While exosome modulation holds enormous promises for medical applications, these efforts are limited by and are heavily dependent on a complete mechanistic understand of exosome biology. Efforts into understanding EV biology and mechanism has shed light on the involvement of a numerous biomolecules involved in EV biogenesis and regulation, however many aspects of the mechanism are still being unraveled.

The process of ILV biogenesis is accompanied by cargo loading and can be facilitated by ESCRT machinery (Figure 2). The ILV formation process takes place via three main steps. First, cargo is accumulated on microdomains on endosomal membrane, cargo accumulation is then followed by inward budding and the process finally ends with fission of exosomes into the MVE lumen [4, 6, 29]. However, retrofusion of ILVs has also been reported (4,6). The accumulated cargo is encapsulated within the exosome.
**Figure 2: Stages of Exosome Secretion.** Exosome secretion begins with 1) an endosome invaginating to form intraluminal vesicles (ILVs) and become a multivesicular endosome (MVE). As MVEs mature, they can fuse with lysosomes (green arrow) for cargo degradation by lysosomal enzymes (yellow stars) or they can 2) mature as MVEs, 3) traffic to the plasma membrane to 4) dock and 5) fuse with the plasma membrane (blue arrows). Fusion releases ILVs as exosomes outside the cell but exosomes can remain attached to the plasma membrane (purple), limiting them to local interactions. 6) Once exosomes diffuse away from the source cell into the environment, they can be found in body fluids or interact with neighboring or distant cells [30].

Following ILV formation, while some MVEs fuse with the plasma membrane and secrete exosomes, some MVEs are directed towards the lysosome for degradation (Figure 2).

MVE and ILV biogenesis can take place via different mechanisms including ESCRT-dependent and ESCRT-independent pathways [6]. The different pathways most likely determine the fate and content of the MVE. For instance, in ESCRT dependent pathway, ESCRT-0 recruits and sequesters ubiquitinated transmembrane proteins[6, 10]. In the ESCRT independent pathway, tetraspanins, and ceramides, can facilitate the entire ILV biogenesis process from cargo recruitment to ILV release into the MVE lumen [31-34]. Tetraspanins, including CD63, CD81, and CD9 are commonly found on exosomes, and are well-established marker for exosomes detection [35]. Thus, MVE that undergo tetraspanin pathway are likely to be directed towards PM for exosome secretion.

Different exosome biogenesis methods most likely lead to biogenesis of exosomes with different membrane and cargo compositions and could result in the observed exosome heterogeneity.

**1.5.1 MVE: Degradation or Secretion**
Interestingly, MVEs can either be directed towards lysosomes for degradation or towards plasma membrane for exosome secretion [4, 6]. The determinants of MVE fate are currently unknown. However, MVE maturation, a process that determines the compositional repertoire of MVEs, most likely determines whether MVEs are degraded or secreted. During maturation, endosomes transform into MVEs through a process involving changes in protein and lipid content of the endosomal membrane [6].

For instance, MVE maturation involves acidification of MVEs by the major proton pump of the cell, the H+-vacuolar ATPase (V-ATPase) [36]. The exact mechanism of V-ATPase on MVE maturation in mammalian cells is still unknown, however, V0 subunit of V-ATPase is proposed to be the mediator of MVE-PM fusion [6]. Knockdown of V-ATPase in HeLa cells has been shown to increase release of exosomes and the accumulation of cholesterol in ILVs [37].

1.6 MVE journey to the plasma membrane

While microvesicles are formed at the plasma membrane, exosome biogenesis occurs in the perinuclear region. Therefore, for secretion, these exosomes must first be transported to the plasma membrane (Figure 2, Steps 3 and 4,). Like other vesicles, MVEs undergo directed transport via interactions with the cytoskeleton (actin and microtubules), molecular motors (dynein, kinesins and myosin) and GTPases (such as Rabs) [38-40]. In polarized cells, directed trafficking of MVEs to different sides of the cell has been observed, but the exact mechanism remains to be identified. EVs collected
from the apical side of polarized MDCK cells are enriched in CD63+ sEVs, whereas EVs collected from the basolateral side are rich in CD9+ and CD81+ sEVs [41].

Transportation of MVE towards the plasma membrane is also facilitated Rab GTPases, such as Rab27b [4]. Silencing of Rab27b leads to accumulation of MVEs at the perinuclear region in Hela cells [42]. Furthermore, following transportation, additional Rab GTPases can aid docking of MVEs at the fusion site (Figure 2,3). Docking is a process that allows vesicles to sit at the site of exocytosis close to the plasma membrane.

Docking is facilitated by interactions between vesicle and plasma membrane proteins. Studies have shown that Rab27a, Rab11 and Rab35 are important for vesicle docking in several cell lines including K562, A549 cancer cells [43]. However, exact identity of the Rab facilitating MVE docking can differ by cell type and have only been studied in a limited number of cell lines.
Figure 3: Protein regulators of MVE docking, fusion and tethering. A) Rab11 (cyan), Rab27 (yellow), and Rab35 (green) assist with trafficking and docking MVEs at the plasma membrane. Granuphilin (pink) and cortactin (blue) associate with other proteins, not included here, to aid in docking MVEs. The inset zoom of an ILV shows the cargo (DNA, RNA, fibrils/proteins) and the lipid bilayer shell. Tetraspanins, CD9, 83 and 81 (black) decorate the ILV, MVE and plasma membrane. B) During trafficking in polarized endothelial cells, CD9+ (red) and CD81+ (green) exosomes are secreted to the apical membrane (top), whereas, CD63+ (blue) exosomes are secreted to the basolateral membrane (bottom). However, the fusion of both types of MVEs is likely facilitated by SNARE proteins and the Ca2+ sensor Syt7 (yellow). Long-lived exosome attachment has been observed for both CD81+ and CD63+ exosomes, but not CD9+ exosomes[35]. This likely occurs via tethering (orange) and/or adhesion proteins (pink and brown) and may depend on the cell type [30].

1.7 MVE Fusion and exosome release

Following docking, MVEs can fuse with the plasma membrane. MVE fusion with the plasma membrane is the final step need for exosome secretion. Though MVE
fusion is poorly understood, membrane fusion involved in release of neurotransmitters and insulin secretion has been extensively studied. Investigation of membrane fusion of different vesicles has shed light on several conserved mechanisms employed in membrane fusion processes in eukaryotic cells. It is well-known that vesicle fusion requires energy, and this energy barrier is generally overcome via aid of SNARE proteins [38, 44, 45]. Briefly, the SNARE complex operates via the interaction of R-SNARE present on vesicle membrane and two Q-SNARES present on plasma membrane. This interaction aids vesicle fusion and release of the enclosed vesicular cargo into the extracellular space. Recent studies point towards employment of SNARE proteins in the MVE fusion process as well, with vesicle associated membrane proteins (VAMPs) acting as R-SNARE and Syntaxins and SNAP proteins as Q SNAREs.

Knockdown of Syntaxin4 (Syx4) and SNAP23 leads to decreased MVE fusion in HeLa cells [35]. Additionally, sEV studies have shown that exosome secretion can be modulated by altering SNAP23 activity via phosphorylation of SNAP23 [35, 46]. Phosphorylation of SNAP23 increases exocytosis by enabling formation of stable SNARE complexes [47]. Interestingly, while blocking SNAP23 phosphorylation suppressed secretion of CD63+ exosomes, it did not suppress secretion of CD81 and CD9 bearing exosomes. This suggests that different MVBs might utilize different mechanisms for membrane fusion and exosome release. Lastly, studies have shown VAMP7 to play a role in MVB fusion and exosome secretion. Inhibition of VAMP7 reduces sEV secretion in MDCK and K562 cells [48, 49]. These studies show that SNAREs play a role in
exosome release. More studies are needed to confirm SNARE involvement and identities of specific SNARE proteins utilized by different cell types and different MVBs.

1.7.1 Exosome secretion

MVB fusion with the plasma membrane releases exosomes into the extracellular space. Released exosomes can carry their content to cells near and distal to the secreting cell. Interestingly, while some of the secreted exosomes diffuse away from the fusion site, some of the released exosomes remain attached to the cell surface. Surface-bound exosomes were first observed in EM studies, and restraining of exosomes is considered to be a regulation mechanism employed by the secreting cell [6, 14, 35, 50]. This anchoring can involve several cell and matrix adhesion molecules, such as CD9, CD63, and CD11, present on exosome surface. Investigations of surface bound vesicles have also shown that a protein called tetherin can aid exosomes anchoring [4, 51]. Tetherin is a transmembrane protein with a GPI anchor at its C-terminus and plays a role in inhibiting spread of enveloped viruses such as HIV. Tetherin homodimers can anchor vesicles at plasma membrane, and their tethering capacity depends on their GPI anchor. Removal of the GPI anchor diminishes tetherin’s ability to sequester both viral and exosomal vesicles. Tetherin KO has been shown to cause a 4-fold decrease in surface-bound exosomes [51]. This makes tetherin a potential candidate to modulate exosome mediated signaling. Furthermore, as tetherin KO only led to a 4-fold decrease, it is possible that other tethering molecules are also present, and remain to be discovered.
Study of exosomes bearing different tetraspanins (CD63, CD81, and CD9) has provided insights into dispersion rate of different exosomes post fusion. Upon secretion, CD9+ exosomes diffuse an order of magnitude faster than CD81+ and CD63+ exosomes [35]. The prolonged presence of CD63+ and CD81+ exosomes is likely due to exosomal anchoring at cell surface [35]. Interestingly however, tetherin KO does not impact the slow CD63 secretion kinetics in HeLa cells. This further points towards potential presence of tethering molecules, other than tetherin itself, that can limit exosome travel range of different types of exosomes.

1.8 Objectives of this work

The discovery of exosomes and their involvement in disease progression was an important milestone in science. Discovery of exosomes has shed light on an important mode of intercellular communication and has opened new avenues of research aimed at hindering exosome-mediated disease progression. However, it is clear, that to develop exosome related therapeutic targets we need to fully understand exosome secretion mechanism and regulation. This information remains lacking due to exosomes being a relatively new field. One area of MVE biology lacking a complete understanding is MVE fusion and exosome secretion. To further understand the MVE fusion and secretion process, my work has focused on characterizing secretion of CD63+ exosomes, and potential SNAREs that facilitate their secretion. We have additionally invested in developing a fully automated algorithm to enable fast processing of MVE fusion data.
Chapter 2 contains all methods used to obtain results discussed in Chapter 3. Chapters 3 is focused on my work related to investigating exosome release kinetics, our automated approach to characterization of MVE fusion, and Chapter 4 is focused on investigating involvement of SNARE proteins in CD63+ exosome release. Hence, this will work will help increase our understanding of exosome release and its regulation.
CHAPTER 2. MATERIAL AND METHODS

2.1 Cell Culture, Transfections, and labelling

A549 cells were cultured in Dulbecco-modified Eagle’s minimum essential medium (DMEM; Life Technologies). DMEM was supplemented with 10% fetal bovine serum (Sigma Aldrich). Cells were grown and maintained in a humidified 37°C, 5% CO₂ incubator. Cells used for microscopy were plated in LabTek 8 well dishes and were transiently transfected using Lipofectamine 3000 (ThermoFisher) using 2.5 mg/ml of CD63-pHluorin plasmid DNA (gift from D.M Pegtel[35]) according to the manufacturer’s protocol. Approximately, 0.3 µg of DNA was added to each well, and each well had a final volume of 250 ul. Cells were imaged between 24-48 hours post transfection. For Syx4, SNAP23, and VAMP7, and Syx4 KD experiments A549 cells were co-transfected with CD63-pHuji with Syx4-myc-myc-His tag, SNAP-eGFP, VAMP7-eGFP, siSyx4, siNT. SiRNAs were obtained from Dharmacon and transfections were carried out using DharmaFECT Transfection reagent following manufacturers protocol. In co-transfection experiments, 0.15 µg of each plasmid was added to keep the final DNA amount of 0.3 µg. Anti-myc 9E10 antibodies for myc-tag conjugated to Alexa Fluor 488 was obtained from Santa Cruz Biotech and was used to tag and visualize Syntaxin 4 clusters using manufacturer’s protocol. Briefly, 1:100 dilution of Anti-myc
Alexa Fluor 488 was used to visualize Syntaxin 4. 1:100 dilution of Anti-myc was made in blocking solution, and cells were incubated with the diluted antibody in blocking
solution for 20 minutes at 37 °C. After 20 minutes of incubation, cells were washed 3x using DPBS buffer. All cell were imaged in imaging buffer with 140mM NaCl, 3mM KCl, 1mM MgCl₂, 3mM CaCl₂, 10mM D-glucose, 10mM HEPES (pH 7.4). For Slot Blot, cells were plated in 6-well plates. Approximately, 0.5 µg of DNA was added to each well, and each well had a final volume of 2500 ul. Cells were collected after 48-hours post transfection. Briefly, 1000 uL of DPBS was added to the adhered cells, and cells were scraped using a cell scraper. 1% of total cell pellet was added to each slot of slot blot. 1% of cell pellet was mixed in 1XBE1 buffer before being applied to the Nitrocellulose filter paper slot. 1XBE1 buffer was composed of 300 mM NaCl, 5 mM DTT, 40 mM HEPES at pH 7.5. For imaging of slot blot, Nitrocellulose Paper was washed 5x with TBS (10 minutes per wash). Secondary Ab for Syntaxin4 (Alexa 488) was added and incubated for 1 hour. Nitrocellulose Paper 5x was rinsed with TBS (10 minutes per wash) and imaged. TBS buffer included 500 mM NaCl and 20 mM Tris at pH 7.5.

2.2 Total Internal Reflection Fluorescence (TIRF) Microscopy

The transiently transfected cells were imaged using an inverted Nikon microscope equipped with 491 nm and 561 nm lasers on an acousto-optic laser launch (Solamere) and an EMCCD camera (Andor iXon897). The laser power entering the back of the microscope was set to 30 mW for each laser and then reduced using ND filters in the laser path prior to excitation. To direct excitation light to the sample, a dual color, a TIRF-specific, dichroic beamsplitter was used (Chroma Technologies) and emission was split into red and green fluorescent channels using a Dual-view (Optical Insights). For
pHluorin detection, a green laser (491 nm) was used for excitation and a 525/50 filter (Chroma Technologies) was used to detect emission. For pHuji detection, a yellow laser (561 nm) was used for excitation and a 605/75 nm filter (605/75) was for detection. For magnification, a 60x 1.49 NA objective and an additional 2.5x lens were used such that one pixel is 109 nm. To improve our ability to focus on both colors simultaneously a very long focal length (1000 mm) plano-convex lens was placed in the red emission channel of the dual view. Image acquisition software (Micromanager http://www.micro-manager.org/) was used to obtain data with a 50ms exposure time continuously for 500-1100 frames. For two-color imaging experiments, the two TIRF channels were aligned with 200 nm yellow-green fluospheres (ThermoFisher). Yellow-green fluospheres show up in both channels and are used for registration of green and red images in conjunction with a custom Matlab code. The depth of field is also indirectly measured by the Fluospheres. Sample temperature was maintained for the duration of each imaging experiment using an on-stage incubator system (Bioscience Tools TC-1-100S).

2.3 Image Analysis

To identify fusion locations, image sequences with the pH sensitive MVE fusion marker were subject to the following analysis using scripts designed in MATLAB (Figure 5). MATLAB and Python versions of the analysis code is available at GitHub (https://github.com/MKKnowlesLab?tab=repositories).

Step 1 - Calculation of Differential Movies: Images were subtracted from one another to highlight the cellular locations where intensity has changed in time. An earlier
time frame (t) was subtracted from a later time frame (t + 25 frames, t + 1.25sec). The maximum projection of the difference movie was used for the subsequent spot finding step. Bandpass filtering of the movie can help with finding spots (Figure 6) but was not needed for the fusion events processed here.

**Step 2 – Fusion Event Localization:** The average intensity of the maximum projection is used as the threshold to locate spots and a cell mask is applied to find the average cell intensity. Spots above the calculated threshold and greater than size 40 pixels were located using MATLAB tracking algorithm freely available by Eric R. Dufresne, Yale University and originally designed for use in IDL[52]. A spot size of 40 pixels was chosen because the spot expands if secretion occurs and the max projection of a fusion site is larger than the vesicle and initial fusion event. Tracking was performed on a subset of the potential fusion events and the rate of motion was used to filter out fusing vesicles from moving vesicles.

**Step 3 – Cropping raw data at fusion locations:** A 25 x 25 pixel region with the potential fusion spot centered was cropped to aid individual event analysis. If two color data was obtained, the two channels were aligned using Fluospheres (Yellow-Green 505/515, 200 nm diameter carboxylate modified, ThermoFisher) and a home-build code in Matlab[53] prior to cropping the fusion event locations.

**Step 4 – Output the intensity in time:** The average intensity from a circle 7 pixels in diameter was measured. The average cell intensity within the cell boundary was defined as the background and subtracted from fusion event. Intensity traces were normalized by first subtracting the minimum before fusion then dividing by the maximum intensity.
**Step 5 – Alignment of data in time**: Since the fusion events are not synchronous, all events were aligned by setting the frame prior to fusion to be 0 s. A background average is calculated using first 35 data points of the fusion event intensity data and the onset of fusion is defined as the point where intensity spikes to 1.4 times the background average.

**Step 6 – Sort events**: Events that are identified in the above steps are either fusing, moving or docking vesicles. Initially, events were sorted by viewing the cropped movies prepared in Step 3 or by the intensity traces in Step 5. Fusion is noted by a sudden spike in intensity (1.4x the background), which is followed by gradual loss of fluorescence over time. With a subset of vesicles, an automated protocol was developed to separate fusion from other events. Here, docking events are removed based on the slope of the decay after the maximum intensity. Moving vesicles that visit the cell surface are removed based on the mean square displacement after tracking. A diffusion coefficient greater than 0 um^2/s denoted a motile vesicle. These non-fusion events are shown in Figure 6.

**Step 7a – Calculate kinetic parameters**: From the aligned intensity traces in time several pieces of information are noted in the results. The slope of the decay, fraction lost at 1 and 5 seconds was calculated. To determine the rate of decay, a double exponential decay shown in Eq. 1, was used to fit each event from the maximum to the end of the trace

\[ F(t) = Ae^{-k_1 t} + Be^{-k_2 t} + C \] (1)
Where A and B are the relative amounts of the fast and slow components that have corresponding rates $k_1$ and $k_2$. C is a plateau because some traces do not return to the background level over the time intensity was measured. The rates, k, were converted in $t_{1/2}$ where $t_{1/2} = \ln (2)/k$. All fitting was performed in GraphPad Prism.

**Step 7b – Calculate size changes in time:** Separately, the radial plot of the fusion event is determined as a function of time; fusion events expand in time[54]. To determine the size of the fusion events, the cropped 25x25 pixel image at the moment of fusion (t = 0) was averaged with the frame before and the frame after (t = -0.05 to 0.05s). This was done to improve the signal to noise. A radial plot was calculated from the image such that the center pixel is position 0 and the equidistant pixels are averaged, as described in our past work[55]. The FWHM was determined by interpolating between points.

**2.4 Two-color TIRF imaging of docking and fusion:**

To determine whether MVE vesicles dock prior to fusion, A549 cells were co-transfected with CD63-eGFP and CD63-pHuji. CD63-pEGFP C2 was a gift from Paul Luzio ([http://n2t.net/addgene:62964](http://n2t.net/addgene:62964); RRID:Addgene_62964) and CD63-pHuji was a gift from D.M. Pegtel[35]. Dual-channel TIRF microscopy was utilized for simultaneous detection of eGFP and pHuji. CD63-pHuji was utilized for automated fusion detection, and corresponding images and intensity traces of both channels were output, as described above. To count whether the MVE was docked prior to fusion, the intensity before fusion in the green channel was measured for events where green was observed during fusion. Green fluorescence during fusion ensured that CD63-EGFP was present in the MVE as some events could be noted as not docked prior to fusion due to the absence of CD63-
EGFP in that particular MVE. Presence of CD63-eGFP at the fusion site was confirmed both manually using CD63-eGFP images and via the small increase in CD63-eGFP brightness.

2.5 Fluorescence Recovery After Photobleaching (FRAP)

FRAP was performed on A549 cells expressing CD63-pHluorin using an Olympus Fluoview 3000. Cells expressing CD63 on the plasma membrane were selected and a 1.99 µm radius spot was bleached using the tornado raster setting. Three images were collected prior to bleaching and images were collected for a total of 100 frames by taking one frame every 1.085s. The rate of recovery was fit, as described previously, to determine the diffusion coefficient of CD63 on the plasma membrane and the fraction mobile[56]. FRAP was performed at 23°C and 37°C.

2.6 Modeling of MVE Fusion Decays

MVE fusion events were modeled as a point source where 100 CD63-pHluorin molecules are deposited. Post deposition, molecules can escape a circle of 0.76 µm diameter (7 pixels in the image analysis) around the fusion site in several ways: 1) As an untethered exosome moving at 6.5 µm²/s. This rate of diffusion is calculated using the Stokes-Einstein diffusion equation: \( D = k_B T / (6\pi\eta r) \), where \( k_B \) is the Boltzmann constant, \( T \) is 310 K, \( \eta \) is the viscosity of the aqueous buffer (0.69 cP), \( r \) is the radius of an exosome, which ranges from 15-60 nm and 50 nm was used. The rate an exosome would
diffuse from the site of fusion varies by less than 10% over the range of temperatures measured in the experimental data. 2) As a molecule diffusing in the membrane at 0.038 \( \mu m^2/s \). This rate of diffusion comes from the FRAP measurements of CD63-pHluorin diffusing on the plasma membrane of live A549 cells. 3) As tethered exosomes. The rate of motion of the tethered exosome is immobile unless the tether is broken, then the exosome diffuses at a rate of 6.5 \( \mu m^2/s \). In the simulation, the time constant at which half of the tethers are broken (\( t_{1/2tether} \)) was varied from 1-50 seconds to best fit the data and exosomes were attached with only one tether. The fraction of CD63-pHluorin in the membrane, in tethered exosomes, and in free exosomes was a second variable that was altered to best match the data. The simulation used 50 \( \mu s \) steps and allowed the molecules to move via diffusion with the rates above if untethered and checked at each 50 \( \mu s \) time point if the tether was still intact. The location of the molecule was recorded every 50 or 100 ms, depending on the experimental data that the simulation was matching to; 23°C data was taken at 100 ms/frame to obtain longer data sets due to the observed plateau in the events and the rest was taken at 50 ms/frame. The simulation lasted between 12.5 and 25 seconds and the number of molecules remaining in a 0.76 \( \mu m \) circle, the size over which the intensity was measured for the data analysis, was output.

Simulations were run and a best match to the average decays was obtained for all temperatures and individual decays for 37°C data (n = 98 fusion events). For the average data, the ratio of free exosomes to endosomal CD63 diffusing on the membrane was determined by fitting the first 10 frames of the experimental data. The tethered exosomes do not contribute to intensity loss at this stage. Next, the \( t_{1/2tether} \) was varied and the
percent of tethered exosomes was determined to fit the long-time tail (frames > 100) of the experimental data. At this point, the endosomal CD63 and the free exosomes do not contribute significantly to the experimental data. For fitting averages 5 simulations were performed. For individual traces, the initial 10 frames were fit as described for the average data. The $t_{1/2,\text{tether}}$ was kept constant after being determined from fitting the average, but the fraction tethered was varied in 2% increments from 0 to 100%. The remainder of the CD63 was split between endosomal and free exosomes at a ratio of 0.6, as determined from fitting the averages. The absolute value of the differences between the simulation and the data was measured for each time point and the lowest sum of these differences was considered the best match for both average and individual traces.
CHAPTER 3 CHARACTERIZATION OF EXOSOME SECRETION KINETICS

3.1 Declaration of collaborative work

The work mentioned in this chapter is the result of a collaborative effort in the Knowles lab. Alan Weisgerber performed the FRAP experiments and analyzed FRAP data. Zdeněk Otruba performed all modeling and analyzed modeling data. Max Palay calculated and analyzed MSD data.
3.2 Overview

When multivesicular endosomes (MVEs) fuse with the plasma membrane exosomes are released into the extracellular space where they can affect other cells. Whether exosomes regulate cells nearby or further away depends on whether they remain attached to the secreting cell membrane. The regulation and kinetics of exosome secretion are not well characterized, but probes for directly imaging single MVE fusion events have allowed for visualization of the fusion and release process. In particular, the design of an exosome marker with a pH sensitive dye in the middle of the tetraspanin protein CD63 has facilitated studies of individual MVE fusion events. Using TIRF microscopy, single MVE fusion events were measured in A549 cells held at 23-37°C and events were identified using an automated detection algorithm. Stable docking precedes fusion almost all of the time. Fusion events exhibit a proportional relationship with temperature; a decrease in temperature was accompanied by decrease in the rate of content loss and a decrease in the frequency of fusion events. The loss of
CD63-pHluorin fluorescence was measured at fusion sites and fit with a single or double exponential decay, with approximately 50% of the events requiring two components and a plateau. To interpret the kinetics, fusion events were simulated as a point source release of tethered/untethered exosomes coupled with the plasma membrane diffusion of CD63. The experimentally observed decay required three components in the simulation: free exosomes, tethered exosomes and CD63 membrane diffusion. CD63 moves from the endosomal membrane into the plasma membrane at a rate of 0.04 µm²/s, as measured by FRAP. Another component of the decay arises from exosomes being secreted but tethered to the surface with one tether that has a lifetime of 8 seconds at 37°C and longer at lower temperatures; simulating with fixed tethers or the absence of tethers fails to replicate the experimental data. Majority of fusion events were docked before fusion. Syntaxin4 clusters, VAMP7, and SNAP23 were present at the site of fusion. Lastly, knockdown of Syntaxin4 decreased MVB fusion by almost 50% and ionomycin increased fusion activity in control and Syntaxin4 KD cells. This kinetic analysis increases our understanding of exosome secretion, and how it is regulated.
3.3 Introduction

Exosomes are a subset of small extracellular vesicles (sEVs) secreted from cells. They range from 30-100 nm in diameter and are formed from the inward budding of vesicles into the intraluminal space of late endosome \([29, 57]\). Exosomes are secreted into the extracellular space when these multivesicular endosomes (MVEs) fuse with the plasma membrane\([5, 10, 19, 58]\). Exosome secretion is used by healthy cells to maintain essential processes such as homeostasis and cell motility, however the release of exosomes can be exploited by unhealthy cells to aid in disease progression\([7, 22]\). Specifically, exosomes and other sEVs have been shown to play an important role in various diseases such as Parkinson’s, Alzheimer’s and cancer \([59-62]\). Once exosomes are secreted into the extracellular space, they can affect cells nearby and further away \([61, 63]\). Exosomes likely facilitate disease progression via the transfer of disease-causing biomolecules from unhealthy to healthy cells\([5, 25, 26, 64-66]\). Many studies regarding exosomes isolate and study sEVs, which are rich in exosomes, but are not exclusively exosomes, therefore the term “sEV” is used. The discovery of exosomes and their involvement in disease states has motivated research into exosomes as markers for early detection and potential avenues for intervention\(^1,8,9\).

Though MVE membrane fusion is likely an integral check point that can be harnessed to modulate exosome release, the lack of a mechanistic understanding of the fusion process is lacking. MVE fusion with the plasma membrane requires energy. Several proteins, including several SNAREs, Rabs, and tethering factors have been shown to aid reduction of energy barriers and catalyze fusion. Knockdown or impaired
activity of SNAREs, including SNAP23, Syntaxin4, and VAMP7 has been shown to decrease exosome secretion in K562 and HeLa cancer cells[35, 48]. Similarly, knockdown of another SNARE protein YKT6 reduces exosome secretion in lung cancer cell (A549)[67]. The specific molecular machinery required for exosome secretion can differ between different cell types, and sEVsubtypes. It is also known that MVE fusion is a constitutive process but enhanced in the presence of Ca^{2+} [68, 69]. In a bulk sEV collection assay, more sEVs were collected from cells treated with Ca^{2+} ionophores^{22}. This was also observed in single fusion events, where ionophores increased the number of events in a variety of cell types [68, 69], however the magnitude of the effect of Ca^{2+} depended dramatically on the cell type [68]. Basic information, that can further our understanding of the fusion process, including whether MVEs are stably docked, factors that control the fusion kinetics and dependence of MVE fusion on temperature are unknown. This information can provide insights into similarities and differences between well-understood fusion mechanisms utilized by other vesicles, such as dense core vesicles, and potentially identify avenues for modulating exosome secretion.

Membrane fusion has been well-studied in various vesicles such as synaptic vesicles (SV), dense core vesicles (DCV) and insulin granules. The kinetics of fusion between the vesicle and the plasma membrane in synaptic vesicle fusion has led to the identification of different fusion mechanisms, such as kiss and run fusion and information about pore dilation [70]. In larger vesicles, such as DCVs and insulin granules, fusion events have been observed to be readily endocytosed as a regulation mechanism. Studies of different vesicles has shed light on conserved and distinct steps and mechanisms that accompany fusion and exocytosis of different types of vesicles. For example, exocytosis
of SVs and DCVs vesicles involves docking, priming and fusion. Additionally, kinetic analysis of different vesicular exocytosis has also identified differences in fusion pores of different vesicles. Shared fusion characteristics such as docking, and differences in release kinetics is indicative of involvement of both shared and distinct fusion machinery at the respective steps of the fusion process [70-72]. As kinetic studies of different vesicles have provided highly important insights into exocytosis, similar studies regarding MVEs fusion and exosomes release will be of major importance to increase our understanding of exosome release.

To elucidate the kinetics of exosome secretion, a marker of intraluminal vesicles (ILVs), CD63, has been used to visualize single fusion events[35, 73, 74]. CD63 is a tetraspanin protein present on the endosome and plasma membranes of cells and it is enriched on the membrane of ILVs that become exosomes upon MVE fusion [25, 75]. Several labs have tagged CD63 on the first extracellular loop with a pH-sensitive fluorescent protein (pHluorin or pHuji) such that the probe is quenched on the inside of the late endosome or when on the surface of an ILV[35, 73, 74]. CD63-pHluorin is a pH dependent probe that remains quenched inside the acidic environment of an MVE and signals the onset of fusion by a sudden spike of fluorescence that gradually disperses radially [76-78]. The rate at which fluorescence dissipates depends on how CD63 labeled exosomes leave the fusion site. Interestingly, long lasting fluorescence has been observed with CD63+ and CD81+ exosomes but not CD9+ exosomes [35]. Post fusion, exosomes have been proposed to either diffuse away from the site of secretion or remain attached to the cell surface [19, 51]. EM data has revealed that some exosomes remain attached to the cell surface and that filamentous molecules appear in MVEs, possibly connecting
ILVs [51]. Tethering molecules, such as tetherin, may be involved with limiting the widespread release of a portion of exosomes [51]. By observing and modeling the kinetics of single fusion events, the extent of exosome attachment to the cell surface, free exosome diffusion and the diffusion of CD63 from the endosomal to the plasma membrane can be measured and quantified.

In this work, single MVE fusion events were visualized using CD63 fluorescent probes and total internal reflection fluorescence (TIRF) microscopy. A non-small cell lung cancer cell line (A549) was used. A549 cells constitute a good model to investigate MVE fusion because they readily release exosomes, facilitating the imaging process, and exosomes have been shown to play an integral role in cancer progression [22, 65, 79, 80]. One challenge of measuring constitutive fusion events in a single vesicle fusion assays is the tedious process of manually finding and analyzing fusion events that occur at random points in time and in a relatively slow fashion (~1-3 events per minute). To overcome this, an automated approach to detection and analysis capable of capturing both large and small fusion events that vary over 2 orders of magnitude in intensity with a background of CD63-pHluorin on the membrane was developed. Our results reveal that, prior to fusion, almost all MVEs are docked for at least 1 second, typically longer. The frequency of fusion increases and the kinetics of release is faster at higher temperatures. The kinetics of release, or loss of fluorescence post-fusion, can relay information about the fate of exosomes. Through two types of analyses, 1) fitting the fluorescence loss and 2) simulations of the release event, we determine that a portion of exosomes are free to diffuse away from the fusion site, but some exosomes remain attached to the surface in many, but not all, fusion events. This attachment depends on temperature, with fewer
exosome remaining attached at higher temperatures. Lastly, we tested whether Syntaxin4 is important for MVE secretion in A549 cells. To our knowledge this is the first measurement, simulation and interpretation of MVE fusion kinetics.

3.2 Results

3.2.1 Automated detection and quantification of MVE fusion events

MVE fusion events were visualized using TIRF microscopy and quantified in A549 cells using CD63-pHluorin, a pH-sensitive fluorescent probe [35, 81], where the pHluorin moiety is located between two transmembrane domains of CD63 such that it is localized to the interior of the MVE and on the exterior of the ILV (Figure 4A). While enclosed in the acidic environment of an MVE vesicle, pHluorin is quenched. Once the vesicle fuses with the plasma membrane the pH inside the MVE changes to physiological pH of 7.4, the pH change unquenches pHluorin leading to an abrupt fluorescence spike, marking the onset of MVE fusion (Figure 4B-D). The fluorescence rapidly appears at fusion site and dissipates into the surrounding area (Figure 4D), as exosomes diffuse into the extracellular space and as CD63-pHluorin on the MVE membrane diffuses into the plasma membrane. The kinetics of MVE fusion can be measured using the intensity profile of single events in time (Figure 4C).
Figure 4: MVE fusion events were detected using CD63-pHluorin. a) Diagram depicting the assay used to detect MVE fusion events. Cells are fluorescently labeled with CD63-pHluorin, which is quenched when in an acidic vesicle. Once the MVE fuses, the pH increases and pHluorin emits green fluorescence. b) Membrane fusion is observable in live cells using TIRFM. A maximum projection of an A549 cell with potential fusion events encircled in green, scale bar = 5 µm. A total of 8 events were found, shown in green circles c) Example of a single exocytosis event, where 0 s is defined as the onset of fusion. Scale bar = 1 µm d) The intensity profile of a sample fusion event showing a rapid increase in fluorescence upon MVE fusion. The intensity shown is the average intensity of one fusion event within a 0.76 µm diameter circle and normalized to the maximum intensity. d) Single images of a difference movie and the raw data for a fusion event. Maximum projections of difference movies were used to identify fusion locations and difference movies were generated by subtracting earlier time (dt) frame from later time frame (dt+1.25 seconds) (details can be found in methods).

Although sites of fusion are visible by eye, manually locating and detecting each event is time consuming and subjective. To circumvent such issues, we developed an algorithm in MATLAB for the rapid detection, and quantification of MVE fusion events (summarized in Figure 5). Detection begins by enhancing the signal of the fusion events relative to the background via the generation of a difference movie, like others have done.
for constitutive fusion measurements [82]. The purpose of this step is to remove noise rendered by vesicles that are visible and stationary on the cell surface throughout the movie; fusing vesicles show a rapid florescence intensity spike due to pH change upon fusion and are generally not visible before the onset of fusion. The difference movie is calculated by subtracting an earlier frame from a later frame. Examples of the effects of different time intervals and filtering on the signal to noise of fusion events are shown in Figure 6. Signal enhancement obtained from the difference movie is apparent in Figure 1E, a single fusion event.
Figure 5: Schematic overview of the automated data processing and analysis used. A) Sample image of raw movie data. Raw data is processed and signal-to-noise ratio of fusion events is increased by calculating difference movies so fusion events appear as bright spots on the cell. B) Sample image of difference movie (of raw data shown in a), showing decrease in background noise and enhanced bright spots representing fusion events. C) Maximum projection of difference movie (shown in B). From the difference movies, a maximum projection is calculated, fusion events are detected (by locating the x and y coordinates of bright spots on the cell), and are isolated by cropping 2.7 µm x 2.7 µm individual fusion regions. D) Sample cropped fusion event, with event circled in green (left), a mask centered on fusion event (right), and quantified fluorescence intensity plot for the fusion event over the duration of the movie (right). The mask, centered on the fusion event, is first multiplied with the potential fusion event to output intensity signal for the fusion event, excluding surrounding background signal. And the calculated output fluorescence intensity is then averaged, to obtain the average time course intensity of
the fusion event. Real fusion event peaks are filtered from non-real events (see Methods and Figure S3 for more details), and both fusion event movies and traces are aligned by artificially setting the peak at $t = 0$ s. F) List of various analyses output by the algorithm. Following time alignment, fusion events are analyzed and a variety of results related to the kinetics, vesicle size and the number of events per minute are output.

Figure 6: Differential movies are calculated to enhance the fusion events relative to stationary vesicles (Steps 1-2 in Figure 5). A) A single image of a transiently transfected A549 cell expressing CD63-pHluorin with 50 ms exposure with six fusion event spots highlighted by pink and yellow boxes. B) A maximum projection of a difference movie is generated using a time interval of 25 frames (1.25 s) and depicts increase in signal of fusion spots and decrease in background noise. C) Ten potential fusion events were
detected using the difference image shown in B. The regions are overlaid on the maximum projection of the raw data. D) Approximately 140 potential fusion events were detected using a maximum projection of the raw data, overlaid on the maximum projection of the raw movie. Scale bar in A-D is 40 pixels. E) Maximum projection images of difference data showing changes in fusion event S/N (for events highlighted in A) after applying bandpass filter of 3, 5, or 7 pixels. Difference movie was calculated using time interval of 150 ms, after application of the bandpass filter for each of the three sizes. F) Maximum projection images showing changes in fusion event S/N (for events highlighted in A) upon application of a bandpass filter of 7 pixels and different time delay.
Upon the fusion of an acidic vesicle, the fluorescence intensity profile of a pH dependent probe exhibits certain criteria, such as a rapid increase in fluorescence intensity at a localized spot followed by a cloud-like spread of intensity into a region around the initial spike location (Figure 4B-D)[35, 54, 82, 83]. Therefore, fusing MVEs were identified by the rapid onset (1-2 frames) of fluorescence intensity, followed by a gradual loss of fluorescence over time. Our approach to locating the fusion events works well for avoiding stationary, fluorescent vesicles, however, fluorescent vesicles that move or dock are detected and binned respectively. Among the potential fusion events identified using our automated approach, approximately 70% were fusion events. The remaining 30% of non-fusion events comprised of a combination of moving vesicles (Figure 7B) and docking vesicles (Figure 7C).
Figure 7: Potential fusion events found in our analysis can be categorized as A) vesicle fusion B) vesicle motion or C) vesicle docking D) Slope of fusion event helps distinguish docked vesicles from moving, and fusing vesicles. Docking is noted by the lack of loss of fluorescence (over time) in the intensity trace. The slope of a line fit from the maximum to one second later is substantial. 

While all three events show a rapid spike in fluorescence, the subsequent kinetics are different. Fusion events decay exponentially (Figure 7A). Moving vesicles typically have to remain stationary for a few frames to be detected as a spot, then move away causing a slower onset of fluorescence followed by a plateau or slow decay (Figure 7B). Docking events do not decay, resulting in a plateau (Figure 7C). The slope from the maximum intensity to one second later was fit and used to bin the events into fusing, moving, and docking (Figure 7D). While slope is useful for binning docked vesicles from the others, significant overlap was observed for fusing and moving. Therefore, tracking
was performed to determine a diffusion coefficient (Figure 7E); moving vesicles can be identified by a diffusion coefficient greater than 0. Once the fusion events were isolated from other detected events, the intensity traces were further analyzed to characterize MVE size and MVE fusion kinetics.

Automated analysis of fusion events portrayed the diversity in fusion events (Figure 8). Fusion events curves from two different cells highlight different types of decay profiles that are present in MVE fusion events (Figure 8A). Different fusion event curves lost intensity at different rates; majority of fusion events lost 30% to 40% intensity within 1 second of fusion onset (max intensity) (Figure 8b, top). However, majority of fusion events showed 50% – 80% intensity loss 5 seconds post fusion onset (Figure 8b, bottom). Our automated algorithm results show that fusion event decay curves from two cells had an average slope of 0.15 seconds, however, the range of decay slope varies for each event (Figure 8C). Results from our automated algorithm can be used to quickly calculate average decay curve, slope and events per minute for the analyzed data set (Figure 8d).
Figure 8: Automated analysis of fusion events. a) Sample normalized fusion events from two cells imaged at 37 °C (12 out of 14 events are shown) b) Histogram showing distribution of percent lost after one second (top) and five seconds (bottom) post fusion. c) Histogram showing slope distributions of data in a. d) Average fusion event trace of the 12 events in a. e) Average Inversed slope (left) and average events per minute (right).
Following the fusion event onset, CD63-pHluorin gradually moves away from the fusion site and a radial spread of the fluorescence is typically observed. This presumably occurs as exosomes diffuse from the fusion site and the CD63-pHluorin from MVE surface laterally diffuses into the plasma membrane (Figure 9A). Using the images of a fusion event (Figure 9B), radial averages were calculated from the site of fusion (the center of the image), in time (Figure 9C) and compared to the size of docked, dense core vesicles in PC12 cells and 200 nm diameter yellow-green fluospheres (Figure 9D). Quantification of the width of the radial plots at the onset of fusion (t = 0 s) yielded an average vesicle diameter of 440 ± 11 nm (Figure 9E) and an increase in size, denoted by the black dashed lines compared to the size at the start (blue dashed line), was observed as the event progressed (Figure 9C). The vesicle diameter appeared to increase due to the diffusion of CD63-pHluorin into the plasma membrane.
Figure 9: MVE fusion events expand in time. A) Depiction of MVE fusion showing how CD63+ exosomes and CD63 in the membrane lead to spread of the fluorescence signal in time. B) Sample fusion event, unfiltered with contrast the same for all images. Scale bar is 1 µm. C) Average radial plots of the fusion event showing a change in MVE radius (HWHM) over time. The blue dashed line is the initial radius (t = 0s) and the black dashed line is the radius at that point in time. D) For comparison, the size of NPY-mCherry granules in PC12 cells and yellow-green fluorescent nanoparticles (d = 200nm). The blue dashed line is the HWHM shown in C. E) Histogram showing range of MVE diameters (FWHM) at the onset of fusion (t = 0 s).
3.2.1 MVE fusion leads to exosome release and deposition of CD63 on plasma membrane

By observing individual fusion events, the rate of the loss of intensity in time were analyzed to determine the kinetics of MVE fusion. Heterogeneity in the rate of fluorescence decay was noted with two types of decay profiles (Figure 10A-B); the decay profiles of some secretion events showed a gradual loss of intensity signal (Figure 10B), other events underwent a biphasic decay with an initial fast downward slope for approximately 3s, followed by a gradual intensity loss phase (Figure 10A).

Figure 10: Multiple kinetic modes are observed in MVE fusion events. A) A single fusion event (black dots) fit to a one component (pink) or a two component (blue) fit. Pink one component fit line shows that the fast decay is not fit well with a one component, single
exponential fit. B) Sample fusion event that is fit well with a one component decay. C) Average time course of 120 fusion events. D) Pie chart showing percent of one vs two component decay profiles. E) Average decay constants, calculated using fast and slow rate constants from decay fit data and FRAP data, calculated based on the diffusion coefficient and circle size. All data taken at 37°C. F) Percent of the fast component in decay curves for fusion events at 37°C. G) Plateau for decay curves at 37°C. All averages or medians (±SEM) represent at least three independent replicates.

A majority of the fusion events exhibited two-component decay profiles (Figure 10D), however this varied slightly with temperature where higher temperatures had more decays that required a two-component fit (Figure 11). In the two-component fit, a plateau (Figure 10G) is also included as many decays do not reach the background. It is worth noting that single fusion events are noisy and Figure 8A shows examples of fusion traces.

![Bar chart showing percent of one phase, two phase, and neither for different temperatures.](image)

*Figure 11: Increase in temperature increases the percent of two component decay curves. Percent of one-component, two-component, and decay curves that fit neither model for decay at 23°C, 27°C, 32°C, and 37°C.*

To determine the cause of the biphasic decays, we considered the likely location of the fluorescence probe. The fluorescence marker of ILVs, CD63-pHluorin, is present on the ILV surface and on the MVE surface. Recent EM data shows that, on average, 30% of
CD63 is retained on the endosomal limiting membrane (dashed line in Figure 10F), whereas the remainder is placed on ILVs. When fusion occurs, it is expected that part of the decay is due to exosomes diffusing from the fusion site and the part of the decay is due to the CD63-pHluorin diffusing from the MVE membrane into the plasma membrane. Analysis of the rate of decay was performed to identify the amount of fluorescence loss due to each of these components. Those traces that fit well to a two-component exponential function (n = 66 events), displayed fast and slow components with $t_{1/2}$ of $0.40 \pm 0.04$ s and $11.5 \pm 2.4$ s, respectively. To determine which component of the kinetics trace was due to CD63-pHluorin diffusion in the plasma membrane, fluorescence recovery after photobleaching (FRAP) experiments were performed with CD63-pHluorin on the surface of A549 cells (Figure 12). Recovery traces (Figure 12) show that CD63-pHluorin is mobile on the plasma membrane and fitting of the data, as described in methods, revealed that CD63-pHluorin on the plasma membrane recovered with a diffusion coefficient of at 0.038 $\mu m^2/s$ (Figure 12), which corresponds to a $t_{1/2}$ of $1.47 \pm 0.10$ s when accounting for the size of the region ($d = 0.76 \mu m$) over which intensity was measured for imaging a fusion event. The plateau, although small on average because many events fully decay, was slightly above 0 (Figure 13E). Fluorescence often remained post fusion (Figure 8 for example traces).
Figure 12: Diffusion rate of CD63-pHluorin in plasma membrane are similar at 23 °C and 37 °C. A) Montage showing photobleaching and recovery of CD63-pHluorin on plasma membrane of A549 cells. Confocal image of the cell before photobleaching (-3s) is followed by the photobleach-recovery process at intervals of 10s. B) FRAP recovery of CD63-pHluorin at 23 °C and 37 °C. N=10 cells. C) Very similar mobile fraction of 0.74 and 0.78 are seen at 23 °C and 37 °C, respectively. D) Diffusion coefficient for CD63 at both temperatures is almost identical. Standard error is shown in all figures.

3.2.3 MVE fusion is temperature dependent.

MVE fusion kinetics were analyzed at four different temperatures: 23°C, 27°C, 32°C, and 37°C. On average, the kinetics of MVE fusion are directly related to temperature; a decrease in temperature caused a decrease in the rate at which the CD63-pHluorin fluorescence signal was lost (Figure 13A-D). Although the lower temperatures reduced the rate of content loss, two phases were required when fitting the average decay. In Figure 13B, the decays are fit well with a bi-exponential but the single exponential fit in Figure 13C clearly misses the fast component for all temperatures. This suggests that the release event contains at least two molecular mechanisms of leaving, likely the loss of CD63 on the exosomal limiting membrane and loss of exosomes themselves, occur for all
temperatures studied. The fast portion of the decay lasted about 0.4 s for 37°C so we compared the amount of fluorescence remaining after 1s for all of the different temperatures to isolate the fast component(Figure 13A). The decays after 1 second trend as a function of temperature (Figure 13D), however the mobility of CD63 on the membrane shows no temperature dependence in FRAP measurements (Figure 12).

Another key difference in temperature is the long-time plateau observed for 23°C; the remaining fluorescence in the plateau was much more prominent at lower temperatures (Figure 13E) with many MVE fusion events at 23 °C retaining approximately 50% of their maximum intensity for over 10 s whereas most of the fusion events at 37 °C only retained 50% of maximum intensity for approximately 3 s post fusion onset (Figure 13A). The median plateau observed at 23°C was approximately 15 times larger than the median plateaus observed at 23°C and 3-5 times larger than 27°C or 32°C. The presence of a plateau suggests that content is not fully released from the fusion site or remains attached to the cell surface and higher temperatures allow more content to leave the fusion site. The rate of fusion events per minute (Figure 13F) also trended higher at higher temperature. Overall, higher temperatures make fusion more frequent, faster to release content and less fluorescence is retained at long times.
Figure 13: Kinetics of single MVE fusion events depend on temperature. a) Average intensity traces in time of single MVE fusion events at 23°C (n=86), 27°C (n=83), 32°C (n=77), and 37°C (n=110). For fitting, the time alignment was done with respect to the maximum intensity (0s) and individual traces were normalized by the maximum. Solid lines are fits with a biexponential function, b) zoom in on the biphasic fit and the c) single exponential fit at short times. d) The percent loss in intensity one second after the maximum. All are significantly different in t-tests (p > 0.05) from the nearest temperature (Average +/- SEM). e) The plateau from the biexponential fit relates the long time, remaining intensity (median +/- SEM, only 23°C is significantly different from others). f) Fusion events observed per minute of data acquisition at different temperatures.
3.2.4 Simulation of exosome release, tethering, and CD63 membrane diffusion can account for the experimentally observed kinetics.

We applied a simple model to determine what types of CD63 motion were sufficient for explaining the release kinetics at different temperatures. In the experimental data, the loss of signal at the fusion site is due to CD63 leaving. It is likely that CD63 can leave in several ways: 1) CD63 is sorted onto ILVs that become exosomes and these exosomes can diffuse freely from the fusion site. 2) CD63 exosomes can remain at the fusion site due to tethering. 3) CD63 on the endosomal limiting membrane can diffuse into the plasma membrane and leave the fusion site. To simulate the data, molecules were released at the center of the fusion site, which is defined as the center of a 0.76 µm circle (Figure 15A). The molecules move with one of the three motions above. Free exosomes (green in Figure 14A) diffuse away almost instantly at a rate of 6.5 µm²/s. CD63-pHfluorin located on the endosomal membrane (blue in Figure 15A) moves at a rate of 0.04µm²/s as measured by FRAP of CD63-pHfluorin on the plasma membrane (Figure 12). Finally, tethered exosomes (red) do not move. To account for the long-time (t > 5s) decay observed in the data (shown in Figure 14 and Figure 15B, grey open circles), we considered that the intensity loss could be due to photobleaching of CD63-pHfluorin. Photobleaching was measured (Figure 14 and Figure 15B, grey open squares) and the rate of photobleaching is extremely slow under the imaging conditions. Therefore, to account for the long-time decay observed, tethers were allowed to break, and without breakage, the long-time data could not be replicated with the loss of fluorescence from photobleaching alone (Figure 14). Therefore, tethers were
considered breakable with some time constant, $t_{1/2,tether}$. The newly released exosomes are shown in Figure 15A (red) and allowed to diffuse at the rate of a free exosome when tethers break. In all simulations, the amount of CD63 remaining in the 0.76 µm circle was measured and an average of 3 or more simulations, each with 100 molecules, of these three dynamic components separately is shown in Figure 15B alongside the experimental data at 37°C and the photobleaching loss.

Figure 14: Rate of photobleaching does not account for the magnitude of fusion decay rate for fusion events at 23°C and 37°C, whereas addition of tethers helps simulate the experimental data. a) Comparison of fusion event decay curve obtained by fusion data and by simulation of fusion data in presence of unbreakable tethers at 23°C. b) Comparison of fusion event decay curve obtained by fusion data and by simulation of fusion data presence of unbreakable tethers at 37°C. c) Plateau observed for fusion data decays faster than the photobleaching rate at 23°C and d) 37°C. Therefore, difference, faster decay for fusion events, is the result of factors other than photobleaching.
When comparing the simulation results to the average data, the variables in the simulation were: \( t_{1/2} \) tether and the fraction of free exosomes; the fraction of CD63 on the endosomal membrane. The best fit for the average data at each temperature is summarized in Table 1 and shown in Figure 14C. The \( t_{1/2} \) tether trended longer with lower temperatures and fraction of attached exosomes also increased with lower temperatures. The amount of CD63 on the endosomal membrane was relatively constant for all temperatures. Simulation of data shows that exosomes are most likely tethered at the site of secretion. Interestingly, the tethers break, and exosomes eventually diffuse away, exosomes are tethered for longer at lower temperatures (Table 1, Figure 15).

Table 1: Parameters for simulated fusion events best fits. Fusion events were simulated for the average trace at each temperature. \( T_{1/2} \), the fraction attached, and the fraction free were unconstrained. The fraction on the endosomal membrane was constrained to 0.3+/−0.1. See Methods for details.

<table>
<thead>
<tr>
<th>( T_{1/2} ) tether (s)</th>
<th>23°C</th>
<th>27°C</th>
<th>32°C</th>
<th>37°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fraction attached exosomes</td>
<td>40</td>
<td>10</td>
<td>8.5</td>
<td>8</td>
</tr>
<tr>
<td>Fraction free exosomes</td>
<td>0.6</td>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>Fraction endosomal</td>
<td>6.5</td>
<td>0.08</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>N events</td>
<td>0.04</td>
<td>0.32</td>
<td>0.35</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>--</td>
<td>86</td>
<td>83</td>
<td>77</td>
</tr>
</tbody>
</table>

The individual fusion event traces at 37°C were also modeled to determine the heterogeneity in MVE fusion that would be missed by looking at the average alone. All traces were modeled, both the biphasic and single-phase types. The \( t_{1/2} \) tether was set to 8s,
as determined from the average, however, the fraction of CD63 on the endosomal membrane was also varied as there is likely a distribution of CD63 location from MVE to MVE. The range of CD63 on the endosomal membrane ranged from 0 to 0.6 (Figure 15D, blue) and the fraction free and tethered exosomes (Figure 15D green and red, respectively) also varied widely to accommodate the distribution of kinetics observed from single traces (examples shown in Figure 8). Overall, the kinetics of individual MVE fusion events are highly variable, and similarly simulation shows that though exosomes are tethered at all temperatures, the duration and percent of free versus tethered exosomes is highly variable.
Figure 15: Fusion events were simulated by allowing CD63 particles to leave the center of the fusion site as tethered exosomes, free exosomes or membrane diffusion. a) depicts the simulation in time. 100 particles are deposited at the center of the 0.76 µm circle and allowed to diffuse unless they are tethered. A portion of the particles are freely diffusing as 100 nm diameter exosomes (green), diffusing on the plasma membrane (blue), or tethered, which are not mobile (red, in the center) until the tether breaks and then these move as free exosomes (red, black outline). The larger green and blue circles depict the motion of the bulk of particles and only a few individual particles are shown. b) The output as number of particles remaining within the circle (d = 0.76 µm) over which the intensity was measured for all data shown. The loss of particles in time under the scenario where all particles move as untethered exosomes at a rate of 6.5 µm²/s (green), all particles move as CD63 does in the plasma membrane at a rate of 0.04 µm²/s (blue), or all are tethered with a tether that has a half-life of 8.0 seconds (red). The data is shown as grey circles and the photobleaching rate as grey squares. c) The mean of each
the experimental data for temperature shown with the best fit from simulations (n = 5). Simulation parameters are shown in Table 1. d) The fraction of each component was varied to fit individual traces at 37°C. To recreate fusion data, individual events ranged from 0-92% tethered.

3.2.5 MVEs are docked prior to fusion

Whether fusion happens with newly arrived MVEs (Figure 16A, left) or MVEs that are docked before fusion (Figure 16A, right) was investigated as it can potentially point towards different mechanisms of fusion regulation. While CD63-pHuji is only visible upon the onset of fusion and serves as a fusion marker, EGFP-CD63 can be observed prior to fusion[35, 68]. Using MVEs marked with both CD63-pHuji and CD63-eGFP, MVEs were visible before and during fusion (Figure 16B). In the CD63-pHuji channel (Figure 16C), fluorescence is not visible before fusion but these events were used to locate fusion sites. In the EGFP-CD63 channel, green fluorescence is observed prior to fusion. Visual assessment of vesicle docking revealed that 87 of 94 events showed a visible vesicle docked at the site of fusion at least 1 second, generally longer, prior to fusion onset.
Figure 16: Most MVEs are docked before fusion. a) Diagram illustration of crash fusion, and docking and fusion, b) Depiction of assay used to detect MVE docking prior to fusion, c) Single images and quantified average intensity plots of the raw data for a fusion event that is docked prior to fusion (n=80). Events shorter than 5 s were excluded in average but were included calculation of percent fusion events that were visible 1s before fusion. d) Percent of fusion events that were visible at least 1s before onset of fusion (87 out of 94 MVE vesicles were visibly docked at least 1s prior to fusion). All vesicles were docking at the beginning of data collection, though fusion events took place at various time points, all events took place at least 1s after start of data collection.
3.3 Discussion and Conclusion

In this work we describe an automated analysis for locating MVE fusion events from TIRF microscopy time series data and use it to characterize single MVE fusion events in A549 cells. To temporally characterize the fusion events, images were taken at a high frame rate (10-20 Hz) as events spontaneously occurred. Fusion events were located by a sharp, transient, increase in fluorescence that occurs upon a somewhat bright background since the CD63 probe is not entirely contained within MVEs, but also appears on the cell membrane, similar to VAMP2 in stimulated exocytosis experiments[83]. To reduce this background, difference movies were calculated before plotting a maximum projection to locate fusion events (Figures 4, 5 and 6). This identified fusion events as well as other transient events where non-acidified, fluorescent vesicles moved near the cell surface or docked (Figure 7). The traces of the event in time, measured from the raw data, allowed for differentiation between the types of events detected. Fusion was marked by a rapid increase in fluorescence and an exponential decay, however, diffusion and docking events were slower to rise, decayed differently or had a faster diffusion coefficient when tracked (Figure 7). Although this work focused on MVE fusion, the docked and moving events could be separately assessed in the future. This approach is capable of high throughput, automated detection and characterization of membrane fusion events and adds to currently available protocols for fusion analysis [83, 84].
One major strength of measuring single fusion events is that it can lead to determination of where specific activators, inhibitors or regulators of the exosome secretion process take effect. Using imaging methods, such as EM or a single fusion assay, a reduction in fusion can be separately deduced from a reduction in ILV formation, but both would result in loss of exosomes in bulk collection.

MVEs range in size depending on the cells and organism. In HeLa cells, EM data shows that MVE range from 400 to 600 nm in diameter [35]. Using super-resolution fluorescence method, MVEs were measured to be 1 um in diameter in MDA-MD-231 cancer cells [50], but recent studies have further narrowed that range to 400-600 nm for 75% of the population [19, 35]. In the work here, MVEs secreted from A549 cells were on average 440 nm and range between 200-800 nm (Figure 9). It should be noted, however, that the size is calculated at the first observed moment of fusion. One limitation of this approach is that expansion could occur during the 50 ms exposure time; this could lead to overestimation of size in events but does not appear to be an issue.

Prior to fusion, most MVEs dock at the plasma membrane (Figure 15). In past work docking was noted with CD63eGFP and fusion events were measured [68]. In this work, the CD63 carriers were visible prior to fusion. It is unclear what proteins are needed for this step, but MVE docking has been shown to be regulated by Rab27a, cortactin, Ral1 and our current understanding of the regulation of MVE docking is reviewed here [96]. After docking, MVEs can fuse with the plasma membrane to release content. Fusion of MVE vesicles can be mediated by several different SNARE proteins,
and our data shows that CD63+ MVE fusion events in A549 seem to use Syx4 mediated fusion. It would be interesting to compare the decay profiles of fusion events obtained from control, Syx4KD and experiments with addition ionomycin. Differences in kinetic profiles can be indicative of a heterogenous population of MVEs.

MVE fusion events are clearly visible in TIRF microscopy images (Figure 4) where the CD63-pHluorin probe is lost from the fusion site and expands laterally in the image (Figures 4,9). Our results show that most of the MVE fusion events in A549 cells proceed via a biphasic decay (Figure 4) where the loss of intensity from the fusion site had a rapid initial decay in fluorescence followed by a slow decay that typically persisted longer than 10 s (Figures 10 and 13a-c). During MVE fusion, CD63-pHluorin is likely released in at least two different ways: as exosomes and by diffusion from the endosomal limiting membrane into the plasma membrane. At 37°C, two thirds of the traces show biphasic decays (Figure 10D) with the faster rate on the same order of magnitude as the rate of CD63-pHluorin diffusion in the plasma membrane (Figure 10E). The fraction of the intensity that is lost at this fast rate matches well to the fraction of CD63 reported to be on the endosomal limiting membrane, where 30-34% of MVE localized CD63 was found on the limiting membrane by EM data [85], shown in Figure 10F (dashed line). The fraction fast and likely endosomal diffusion was constant (0.40-0.46) and not significantly different over all temperatures measured. Together, this suggests that one way CD63 leaves during MVE fusion is via membrane diffusion.
It is worth noting that we assume the rate of diffusion of CD63 into the plasma membrane from the endosomal membrane is similar to the rate of diffusion of CD63 on the plasma membrane. However, the rate of CD63 loss from the fast component of fusion is 2.4 times faster than CD63 plasma membrane diffusion (Figure 10e). Here, the time it takes to leave the circle where fusion occurs, $t_{1/2}$, is 0.40 s (red) for the fast component and the expected $t_{1/2}$ if CD63 moved at a rate of 0.039 μm²/s on the membrane would be 0.95 s (blue). Other transmembrane proteins have also been noted to diffuse faster during membrane fusion. In a study of constitutive fusion using TIRF microscopy, VSVG membrane diffusion from fusion sites was measured and compared to VSVG on the plasma membrane. The diffusion from a fusion site is 3x faster than diffusion of protein already existing within the PM. It is possible that our assessment of CD63-pHluorin motion on the plasma membrane using FRAP is not representative of how newly delivered protein moves, yet this aligns well with the interpretation of the fast component of the decay being membrane diffusion.

To gain a mechanistic understanding of the experimental kinetics, a simple model for CD63 loss from the fusion site was designed. In an independent analysis of the data using a diffusion model to simulate the decay curves, as opposed to fitting, approximately 35% of the CD63 needed to be on the endosomal membrane to recreate the experimental data (Table 1). Although the simulations model the decays well, simulations could not recreate the fast loss with only membrane diffusing CD63-pHluorin (Figure 15C, blue). Therefore, a small portion of CD63 needed to reside on free exosomes (8-24%, Table 1) to model the experimental data at all temperatures. This further supports that CD63
leaves fusion sites by diffusing into the plasma membrane and free exosomes contribute a small amount to the initial loss.

The third component of the intensity loss relates to the long time, slow loss of CD63-pHluorin. The slow rate is almost 30 times slower than the fast at 37°C, but it is not as slow as the photobleaching rate (Figure 14 and Figure 15c), suggesting that CD63 molecules are leaving the fusion site in a delayed fashion. In past work, others have also noted long-lived fluorescence in single fusion events and suggested this was due to tethering of exosomes[35], although cells lacking one of the known tethers, tetherin [37], did not remove the long-lived fluorescence [35]. Many molecules could potentially tether exosomes to the surface, such as tetraspanins, adhesion proteins, and attachment of exosomes to the cell surface has been noted in EM data for many years [5, 51, 86, 87]. However, in the analysis of imaging MVE fusion exosomes could not remain attached eternally, the rate of loss is faster than photobleaching. To model this rate of loss, exosomes were considered tethered and the tethers dissociate or break over the course of many seconds (8-40s). Without considering details of what molecules are at work (i.e tetherin, adhesion proteins, etc), this approach was able to model the time data well at all temperatures (Figure 15D). Several other mechanisms could be responsible for the slow loss, such as kiss and run fusion where the fusion pore remains confined and then closes again [70, 88] or the incomplete flattening of the fusion event followed by endocytosis and reacidification [53, 89, 90] In DCV fusion, small molecules are released first [72]. However, once we lose some exosome fluorescence, the pore is big enough and all exosomes should escape because even small exosomes range from 30-120 nm. Last of
all, exosomes could temporarily stick to the glass surface used for imaging, however, different types of exosomes, CD9+ instead of CD63+, display fast fusion kinetics [35].

Although A549 cells undergo MVE fusion at all temperatures studied, higher temperatures led to increase in the number of fusion events per minute (Figure 13F). Increase in frequency of fusion events with increase in temperature has been observed in several vesicle exocytosis studies before [91]. Higher rate of fusion events facilitates the study of single MVE fusion events as fusion events are slow and infrequent, even in cell lines, such as cancer cells, that are well-established at undergoing exosome secretion. Overall, the release kinetics at all temperatures likely depend on the cell type, growth conditions, and the construct design, as others have observed in stimulated secretion studies (R Chow BJ 2004). Gandisi et al demonstrated a wide variety of kinetics from the dense core vesicles of insulin cells by varying the fluorescent moiety. While all green fluorescent probes decayed similarly, red probes varied over a factor of 7 in their speed. This could be due to the photophysical properties of these red probes, as others have noted for pHuji [92, 93] and new red probes may avoid this issue. In light of this, all of the kinetics of release observed by Gandisi et al were fit well with a single exponential function, suggesting one dominant mechanism driving the decay kinetics for the secretion of vesicles that contain NPY, a vesicle content marker. In our work, kinetics were only evaluated from CD63-pHluorin events and CD63-pHuji events were solely used for docking studies.

In summary, the kinetics of MVE fusion can relate the mechanism by which fluorescently labeled exosomes leave the fusion site. In this work we hypothesize and test
a model where exosomes are tethered the tethers can dissociate in time and do so more readily under higher temperatures.
CHAPTER 4 ROLE OF SNARE PROTEINS IN MVE FUSION EVENTS

4.1 Overview

To test the involvement of SNARE proteins in MVE fusion events we at SNARE-protein activity at the site of MVE fusion event. Syntaxin4 clusters, VAMP7, and SNAP23 were present at the site of MVE fusion. Additionally, knockdown of Syntaxin4 decreased MVB fusion by almost 50% and ionomycin increased fusion activity in control and Syntaxin4 KD cells. Primary results reported in this chapter increases our understanding of exosome secretion, how it is regulated, and provide guidance for future experiments.
4.2 Introduction

Exosomes are a subset of small extracellular vesicles (sEVs) secreted from cells. They range from 30-100 nm in diameter and are formed from the inward budding of vesicles into the intraluminal space of late endosome [29, 57]. Exosomes are secreted into the extracellular space when these multivesicular endosomes (MVEs) fuse with the plasma membrane[5, 10, 19, 58]. Exosome secretion is used by healthy cells to maintain essential processes such as homeostasis and cell motility, however the release of exosomes can be exploited by unhealthy cells to aid in disease progression[7, 22]. Specifically, exosomes and other sEVs have been shown to play an important role in various diseases such as Parkinson’s, Alzheimer’s and cancer [59-62]. Once exosomes are secreted into the extracellular space, they can affect cells nearby and further away [61, 63]. Exosomes likely facilitate disease progression via the transfer of disease-causing biomolecules from unhealthy to healthy cells[5, 25, 26, 64-66]. Many studies regarding exosomes isolate and study sEVs, which are rich in exosomes, but are not exclusively exosomes, therefore the term “sEV” is used. The discovery of exosomes and their involvement in disease states has motivated research into exosomes as markers for early detection and potential avenues for intervention1,8,9.

Though MVE membrane fusion is likely an integral check point that can be harnessed to modulate exosome release, the lack of a mechanistic understanding of the fusion process is lacking. MVE fusion with the plasma membrane requires energy. Several proteins, including several SNAREs, Rabs, and tethering factors have been shown to aid reduction of energy barriers and catalyze fusion. Knockdown or impaired
activity of SNAREs, including SNAP23, Syntaxin4, and VAMP7 has been shown to decrease exosome secretion in K562 and HeLa cancer cells[35, 48]. Similarly, knockdown of another SNARE protein YKT6 reduces exosome secretion in lung cancer cell (A549)[67]. The specific molecular machinery required for exosome secretion can differ between different cell types, and sEVsubtypes. It is also known that MVE fusion is a constitutive process but enhanced in the presence of Ca^{2+} [68, 69]. In a bulk sEV collection assay, more sEVs were collected from cells treated with Ca^{2+} ionophores. This was also observed in single fusion events, where ionophores increased the number of events in a variety of cell types [68, 69], however the magnitude of the effect of Ca^{2+} depended dramatically on the cell type [68]. Basic information, that can further our understanding of the fusion process, including whether MVEs are stably docked, factors that control the fusion kinetics and dependence of MVE fusion on temperature are unknown. This information can provide insights into similarities and differences between well-understood fusion mechanisms utilized by other vesicles, such as dense core vesicles, and potentially identify avenues for modulating exosome secretion.

In this chapter, we continued to look at single MVE fusion events using CD63 fluorescent probes and total internal reflection fluorescence (TIRF) microscopy. We expanded our MVE fusion event assay by adding simultaneous detection of SNARE proteins at the site of MVE fusion events. All experiments were conducted in non-small cell lung cancer cell line (A549). As mentioned before, A549 cells constitute a good model to investigate MVE fusion because they readily release exosomes, facilitating the imaging process, and exosomes have been shown to play an integral role in cancer
progression [22, 65, 79, 80]. In this chapter we report results for preliminary studies regarding importance of SNARE and SNARE interacting proteins at the site of MVE fusion. We tested whether Syntaxin4 and Syntaxin4 interacting proteins are present at the site of MVE fusion events in A549 cells.

4.3 Results

4.3.1 Syntaxin4 clusters colocalize with CD63-pHuji fusion events

Knockdown of Syntaxin4 (Syx4) has been shown to reduce secretion of CD63+ exosomes in HeLa cells [35]. However, fusion machinery needed for MVB fusion can differ between cell types. As majority of MVB vesicles appeared to be docked before fusion (Figure 16), and syntaxin cluster formation is part of docking process, we tested whether syntaxin 4 is present at the site of MVB fusion. Alexa-488 Ab labelled Syx4 and CD63-pHuji were simultaneously imaged for >30 s using TIRF microscopy. Colocalization between average Syx4 images and maximum projection image of CD63 MVBs in A549 cells was assessed. Syx4 is recruited upon MVB docking, and majority of MVBs were pre-docked within our imaging time frame, therefore average Syx4 clusters should be present at MVB fusion site in average images. While some CD63-pHuji fusion spots colocalized visually with average Syx4 clusters tagged with Alexa 488 (Figure 17a and b), some fusion spots were not colocalized with the Syx4. Lack of colocalization can be due to two reasons. It is possible that not all Syx4 clusters were labeled, or because not all CD63+ MVBs utilize Syx4 for fusion.
4.3.2 Time course of VAMP7 and SNAP23 during over the course of fusion events

To test if syntaxin4 interacts with MVE before fusion onset, congregation of Syx4 clusters near fusion spot was assessed. Presence of Syx4 clusters in a circle (diameter, 7 pixel) centered on fusion spot fluorescence was visually assessed (Figure 18a) and quantified (Figure 18b). Syx4 clusters were in 7 out of 10 fusion events (n= 10 cells). Syntaxin4 clusters on A549 plasma membrane were highly similar to Syntaxin1 clusters reported in PC12 [94].

We quantified and assessed two additional exocytosis proteins: VAMP7 and SNAP 23. VAMP7 is a vesicle marker that has been shown to catalyze MVE fusion via interactions with Syntaxin4 on plasma membrane. eGFP-VAMP7 was co-transfected with CD63-pHuji and was visible prior to onset of fusion (Figure 18a). Interestingly,
eGFP-VAMP7 fluorescence spikes at fusion onset, showing that VAMP7 might be present on the exosomes as well. This could be due to a mechanistic role played by VAMP7 in medicating exosome fusion with designated tissues, could be a result of VAMP7 overexpression, or due to movement of eGFP-VAMP7 closer to the imaging area.

Figure 18: Syntaxin4 is present at MVE fusion site. a) Example image sequences of sample MVE marker CD63-pHuji (top) and Alexa-488 labelled Syx4 (bottom). Images are centered on the fusion spot. To emphasize changes in granule-associated
Though SNAP23 is known to participate in Syntaxin4 mediated fusion activity, SNAP23 accumulation at fusion site most likely occurs at a very slow rate similar to SNAP25 [95] and was not captured within our imaging time frame. SNAP23 also dissipates gradually post-fusion and is also not visible (Figure 19d). However, quantification of SNAP23 at images shows that SNAP23-eGFP is present at the fusion site as the average intensity of SNAP23 is above zero before, during, and after fusion of MVE (Figure 19d).
Figure 19: VAMP7 and SNAP23 are present at MVE fusion site. a) Example image sequences of sample fusion spot images are centered on the fusion event. b) Example image sequences of MVE marker CD63-pHuji and EGFP-labelled VAMP7 and quantification images. c) Example image sequences of MVE marker CD63-pHuji and EGFP-labelled SNAP23 during fusion event. Images are centered on the fusion spot. d) Quantification of SNAP23 and CD63 fluorescence intensity for 7 different events. To emphasize changes in protein and MVE fusion-associated fluorescence, the image sequences were linearly scaled for best contrast. N= 10 cells.

4.3.3 Knockdown of Syntaxin4 decreases MVB fusion events and ionomycin increases exosome release

To further test involvement of Syx4 in MVB fusion activity, we knocked down Syx4 using siRNA. CD63-pHluorin was expressed in control and Syx-4 KD cells and the rate of MVB fusion events was obtained using TIRF microscopy. Cells transfected with CD63-pHluorin with or without non-targeting (siNT) RNA showed a similar rate of fusion events per minute (Figure 20a). Whereas fusion rate, upon Syx4 KD, was reduced by almost 50%. We next tested the effect of Ca^{2+} influx via addition of ionomycin. Increase in Ca^{2+} has been shown to increase exosome secretion, though a complete mechanistic understanding and all regulatory proteins involved in Ca^{2+} based secretion are not known. SNARE proteins play an important role in Ca^{2+} dependent synaptic transmission, and KD of some SNAREs can impair SNARE-Ca^{2+} dependent secretion[43, 96, 97]. Interestingly, addition of ionomycin increased fusion activity in control and Syx4 KD cells. These results suggest that fusion events observed in Syx4 KD cells are either mediated by remaining Syx4 due to incomplete depletion by KD, or utilize a Syx4-independent fusion pathway. MVB heterogeneity is well-established in
literature and at least two distinct MVB populations are present in our kinetic analysis. Different MVBs can fuse via different mechanisms.

Figure 20: Knockdown of Syntaxin4 decreases MVB fusion events a) Effect of Syntaxin4 knockdown on fusion activity in A549 cells. n ≥ 10 cells per condition ctrl, siNon-targeting, siSyntaxin-4. b) Addition of ionomycin increases fusion activity despite Syntaxin4 knockdown in A549 cells n>10 cells per condition. c) Confirmation of Syntaxin4 knockdown (KD) at the protein level in A549 cells.

Further characterization of the fusion events observed in Syx4 KD before and after ionomycin addition is needed. Their comparison with our kinetic analysis of fusion observed in control cells can shed light on the any differences in MVB populations that are impacted by Syx4 and ionomycin.

4.4 Discussion and Conclusion

In this work we conducted preliminary studies to check whether Syntaxin4 plays a role in MVE fusion events using TIRF microscopy time series data. We used TRIF to it
to characterize single MVE fusion events in A549 cells, and simultaneous activity of fusion machinery including SNAREs and SNARE interacting proteins. We specifically looked at Syntaxin4, and Syntaxin4 interacting proteins including SNAP23 and VAMP7. All events were detected and characterized using the automated method described in Chapters 2 and 3.

As discussed in Chapter 3, most MVEs dock at the plasma membrane prior to fusion (Figure 15). In past work docking was noted with CD63eGFP and fusion events were measured [68]. In this work, the CD63 carriers were visible prior to fusion. It is unclear what proteins are needed for this step, but MVE docking has been shown to be regulated by Rab27a, cortactin, Ral1 and our current understanding of the regulation of MVE docking is reviewed here (96). After docking, MVEs can fuse with the plasma membrane to release content. Fusion of MVE vesicles can be mediated by several different SNARE proteins, and our data shows that CD63+ MVE fusion events in A549 seem to use Syx4 mediated fusion. It would be interesting to compare the decay profiles of fusion events obtained from control, Syx4KD and experiments with addition ionomycin.
REFERENCES


[46] Y. Wei et al., "Pyruvate kinase type M2 promotes tumour cell exosome release via phosphorylating synaptosome-associated protein 23," (in En), Nature


[78] "<Fusion of Constitutive Membrane Traffic with the Cell Surface Observed by Evanescent Wave Microscopy.pdf>.


function pro_cess_fusion_data(imdata,imdata2) 
% pro_cess(imdata)
% imdata is the imaging data in tiff file format
% PURPOSE: 1) to process imaging data and detect fusion event. 2) Analyze
% detected fusion events to obtain various kinetic parameters.
% movie
% INPUT:
% imdata: movie stack, in tiff format.
% OUTPUT: an excel containing:
% 1) Expression level of cell (background corrected)
% 2) fusion event traces with fusion event onset aligned to t=0,
% 3) Normalized event traces
% 4) Slope of each fusion event
% 5) % intensity loss at different time points
% 6) Decay curve fit and rate of decay
% 7) Radial plots of each curve starting t=0
% 8) Full width half max at different time points starting with t=0
% 9) Area under curve
%
% PROCESSING OVERVIEW (please look at each program below for details):
% I. Read tiff movie
% II. Generate Cell Mask, Background Mask and Relative Expression Level
% III. Generate Max projection of difference movie and find fusion event
% IV. Separate each fusion event and obtain fluorescence intensity of the
% fusion event (circle) and of the background (annulus)
% V. Obtain decay rates, radial plots, HWHM, Area under RP.
% VI. Generate time aligned movies and montages for each fusion event
%
% I. Read tiff movie
info = imfinfo(imdata);
numframe = length(info);
for K = 1 : numframe
    rawframes(:,;:,K) = imread(imdata, K);
end
% convert data to 3D
rawframes = rawframes(:,;,:);
%% for dual channel input, nargin>1, process second input movie
if nargin>1
   info2 = imfinfo(imdata2);
   numframe2 = length(info2);
   for K = 1 : numframe2
      rawframes2(:,:,K) = imread(imdata2, K);
   end
   %convert data to 3D
   rawframes2 = rawframes2(:,:);
end

%% II. Generate Cell Mask, Background Mask and Relative Expression Level
%% Use first frame to obtain masks and write out image
   CFimage = rawframes(:,:,1);
   imwrite(CFimage,'CFimage.tif','compression','none');
   %Obtain cell and background masks
   [~,cellmask,~,backgroundmask]=obtaincellmasks('CFimage.tif');
   %calculate relative expression level(RExLevel) =cell intensity in the first
   %image minus background
   CFimage=double(CFimage);
   % multiply by cell mask so intensity value outside cell will become
   % zero
   CF2=CFimage.*cellmask;
   % Calculate Avg cell intensity in first image of the cell
   CFavg = mean(nonzeros(CF2));
   % Calculate Avg background in first image of the cell using background
   %mask
   CFb=CFimage.*backgroundmask;
   CFbAvg = mean(nonzeros(CFb));
   % Relative Expression level = Average Cell intensity minus average
   % background intensity
   %RExLevel=CFavg-CFbAvg;
   % Create excel to output data
   curr_directory =  pwd;current_folder_name = curr_directory(end-3:end);
   % current_folder_name = name of folder
   output_excel_file = sprintf('Analyzed Data_%s.xls',current_folder_name);
   % Data is output in an excel file, and the data is converted to excel
   % compatible format
   %RExLevel=array2table(RExLevel);
   %writetable(RExLevel, output_excel_file,'Sheet', 'RExLevel');

%% III. Generate Max projection of difference movie and find fusion event
%% Generate Max projection image
   [~,diff_max_proj] = CalcDifferenceMovie(rawframes,7,25);
% Multiply Max projection and cell mask to eliminate background
% intensity, this converts all background values to 0
diff_max_proj = diff_max_proj .* cellmask;
% Intensity values within cell perimeter are > 0
cell_Intensity_values = (diff_max_proj) > 0;
% Increase brightness of the bright spots for enhanced contrast and
% easier peak selection
max_proj_2 = diff_max_proj .* cell_Intensity_values;
% Enhanced intensity values located within cell perimeter
cell_Intensity_enhanced = (nonzeros(max_proj_2));
% Find threshold to detect fusion event locations using pkfnd
% Set threshold using average brightness of the cell
pk_thresh = (mean(cell_Intensity_enhanced));
% Find location of bright peaks
pk_loc = pkfnd (diff_max_proj, pk_thresh, 40);

% IV. Separate each fusion event and obtain fluorescence intensity of the
% fusion event (circle) and of the background (annulus)

% Crop out 25 x 25 pixel regions with fusion events centered for individual
% analysis of each event
% Create 7x7 pixel circular mask
Indiv_events = ministk_movie(rawframes, pk_loc, 25, 0);
cir_mask = create_mask(25, 7);
outcir_mask = ~ (cir_mask);
outcir_mask(1:5,:) = 0; outcir_mask(:,1:5) = 0;
outcir_mask(:,20:25) = 0; outcir_mask(20:25,:) = 0;

if nargin>1
% Obtain array with timecourse average intensity
% Array containing fusion event intensity and protein event intensity
Indiv_eventsP = ministk_movie(rawframes2, pk_loc, 25, 0);
Int_ArrayP=ministk_intensityavg(Indiv_eventsP, cir_mask);
Int_ArraPO=ministk_intensityavg(Indiv_eventsP, outcir_mask);
% Array containing fusion event intensity
Int_DiffP=Int_ArrayP - Int_ArraPO;
end

% Obtain array with timecourse average intensity
% Array containing fusion event intensity and protein event intensity
Int_Array=ministk_intensityavg(Indiv_events, cir_mask);
Int_ArraO=ministk_intensityavg(Indiv_events, outcir_mask);

% Array containing fusion event intensity
Int_Diff=Int_Array-Int_ArraO;
%Filter unreal peaks and
%Align Fusion events by setting onset of fusion as T=0
if nargin < 2
    [Aligned_Pks, Aligned_Pks_I] = pk_align(Int_Diff, 50, Int_Array, CFbAvg);
    [Aligned_PksC, Aligned_PksCI] = compile_data(Aligned_Pks, Aligned_Pks_I);
    [ND1] = normalizedata2(Aligned_PksC);
end
%Filter unreal peaks and
%Align Fusion events by setting onset of fusion as T=0
if nargin > 1
    %Compiled Aligned pkgs-filter out 0s
    %Normalize Pks
    %Output intensity of fusion events in excel
    % V. Obtain decay rates, radial plots, HWHM, Area under RP
    % Fit Decay Curves to a double exp function and
    % Obtain fit parameters
    % decayfit2(ND1);
    % MAKE Radial Plots, and calculate HWHM, and Area under radial plot
    radial_plot(Aligned_Pks, Int_Diff, Indiv_events, Aligned_PksC);
end
%Compiled Aligned pkgs-filter out 0s
%Normalize Pks
%Output intensity of fusion events in excel

% VI. Generate time aligned movies and montages for each fusion event
% Create a 25 x 25 pixel movies for each fusion event with fusion event of
% middle of the movie
if nargin == 1
    align_movies(Aligned_Pks, Indiv_events);
    create_montage(Aligned_Pks, Indiv_events);
elseif nargin > 1
    align_movies(Aligned_Pks, Indiv_events, Indiv_eventsP);
    create_montage(Aligned_Pks, Indiv_events, Indiv_eventsP);
end
% Create 1s interval montages
% create_montage(Aligned_Pks, Indiv_events, Indiv_eventsP);
% Save Matlab Workspace
curr_directory = pwd; current_folder_name = curr_directory(end-3:end);
save_workspace = sprintf('%s Analysis.mat', current_folder_name);
save(save_workspace);
end

%DIFFERENCE MOVIE
function [mov_out, maxproj, max_sub] = CalcDifferenceMovie( mov, filter, dt )
%PURPOSE: to calculate the difference in intensity from one frame to the next, output is a movie of the difference images.
% Call:mot=CalcDifferenceMovie(a, 9, 1);
% Variables: a is a movie. The format is usually (256,256, nframes)
% filter is the size of the bandpass filter, and dt is the time lag in frames over which the difference movie is calculated
% Output: this calculates the percent of pixels that changed from one frame to the next
% Steps - calculate difference images (frame t+1) - (frame t).
% MKK Dec 2 2019
%
max_sub=zeros(256,512);
nframes = max(size(mov(1,1,:))); x = max(size(mov(:,1,1))); y = max(size(mov(1,:,1)));

if nargin <2
    filter=7;
end
if nargin <3
    dt=25;
    %dt=floor(nframes*(0.025));
end

mov_out = zeros(x, y, nframes-dt);
for i = 1:nframes-dt
    % filter images at frame t and t+dt
    im = bpass(mov(:,:,i),1,filter);
    im2 = bpass(mov(:,:,i+dt),1,filter);

    % take the difference image from frame t+1 and frame t to determine new pixels with values. This is where signal moved TO.
    delta = im2 +1000 - im;
    %store the image in a movie file
    mov_out(:,:,i) = delta;
end
%should backgrd always be 35 frames long?
Try backgrd with fewer frames to check how to automate length of bck.

```matlab
max_bckgrd = max(mov_out(:,:,1:35), [], 3);
max_rem= max(mov_out(:,:,36:end), [], 3);
max_sub=max_rem-max_bckgrd;

maxproj = max(mov_out, [], 3);
%figure;imshow(maxproj, [1000,1500]);
end
```

%CIRCLE MASK

```matlab
function [cir_mask] = create_mask(sz,d)
    %[cir_mask] = create_mask(25,7)
    % PURPOSE: to make a mask of size sz and diameter d
    % INPUT:
    % sz: in units of pixels, size of mask
    % d: the diameter of the circular mask
    % OUTPUT: an array that is very long (number of spots found in each frame)
    % that has X, Y, time

    mask = zeros(sz,sz);
    cntr =ceil(sz/2);
    % total rows/columns -1
    trc=d-1;
    % rows/columns above/below center-halof of total rows/col=3
    h=trc/2;
    %center=13
    for n=cntr-3:cntr+3
        mask(n,cntr-1:cntr+1)=1;
    end
    for n=cntr-2:cntr+2
        mask(n,cntr-2:cntr+2)=1;
    end
    for n=cntr-1:cntr+1
        mask(n,cntr-3:cntr+3)=1;
    end
    [cir_mask] =mask;
end
```

%AVERAGE INTENSITY

```matlab
function out=miniEstk_intensityavg(mov,msk)
% out=miniEstk_intensity(images, mask)
%
% PURPOSE: to measure the intensity in the center of a 25 x 25 image
```
% INPUT:
% mov: movie stack to measure from
% msk: an image that is 1 in the center and 0 outside the center
%
% OUTPUT: an array that is N_frames long by the number of spots. Each column is a new location.
%
% CREATED: Michelle Knowles Jan 2018

nrgn = length(mov(1,1,1,:));
nframes = length(mov(1,1,:,1));
intensity = zeros(nframes,nrgn);
size(intensity)
%loop through all spots
for i =1:nrgn;
    %loop through all frames - there's got to be a faster way!
    for j =1:nframes;
        a=double(mov(:,:, j, i)).*msk;
        intensity(j,i)=sum(sum(a))/sum(msk(:));
    end
end
out=intensity;
end

%CROP 25 x 25 pixel regions
function out=ministk_movie(im,rgn,sz,sepdist)
% out=ministk(im,rgn,sz)
%
% PURPOSE: to cut out small regions of an image based on spots found in
% the image or a corresponding image of a different color. This is used to
% measure colocalization based on the work of Knowles and Barg in two PNAS
% 2011 papers. Spots are found using the work of tracking routines
% available on Eric Weeks' website (Emory University) and made into Matlab
% by Eric Dufrense.
% INPUT:
% im: movie stack to cut from
% rgn: spots (x,y) about which regions should be cut
% sz: size of cut out (a square of sz by sz pixels)
% sepdist: is the minimum separation distance between two spots. If two
% spots are within this distance of one another, neither are counted.
%
% OUTPUT: a sz x sz x N_frames array for each spot
%
% CREATED: Michelle Knowles May 2012
% EDITED: to crop movies rather than single images by Michelle Knowles and
% Aubrie Blevins December 2017, note that X and Y from the peak finder seem
% to be switched when cropping in this way.

%if sz/2 == floor(sz/2)
%warning('sz must be even so that the spots can be centered on a pixel: 1 pixel added');
%sz = sz+1
%end

% scott's code for a mask
pix=(sz+1)/2;
dimy = length(im(1,:,:,1));
dimx = length(im(:,1,:));
rgn = length(rgn(:,1,:));
nframes = length(im(1,:,:));
sepdist = sepdist;
% create a blank image array that you can fill
msk=zeros([sz,sz,nframes]);
% loop through all regions that locate spots in an image
for i = 1:nrgn;
  x = rgn(i,1);
  y = rgn(i,2);
  % don't include regions within pix distance from the edge of the image.
  if ((x>pix) && (x+pix)<dimy) && (y>pix) && ((y+pix)<dimx)
    % don't include regions within 'sepdist' pixels of another region
    % calculate an array that contains the difference between the
    % current particle's position and all others.
    if sepdist < 0
      diffy=rgn(:,2)-rgn(i,2);
      diffx=rgn(:,1)-rgn(i,1);
      mag=((diffx.*diffx)+(diffy.*diffy).^0.5);
      % find all the locations in the magnitude array that are non-zero.
      % This should remove the comparison between particle i and itself,
      % which will always be zero.
      w = find(mag);
      mag = mag(w);
      if (min(mag) > sepdist)
        % how can we crop a MOVIE, do we need to loop over all images,
        % concatenate cutout then concatenate cutout to msk?
        cutout=im(y-pix+1:y+pix-1, x-pix+1:x+pix-1,:);
        msk = cat(4, msk, cutout);
      end
    elseif sepdist==0

89
%Align FusionEvent Traces

function [out_Aligned_PksC,out_Aligned_PksC_P,outFoverS]=
  pk_align(Int_Array,exposure_n,Int_2Array,Int_DiffP,CFbAvg)

%PURPOSE: to determine the location of fusion event onset and artifically set the
determined
%fusion onset point to Time = 0s
%
% When calling function pk_align(Int_Array,exposure_n,Int_DiffP,CFbAvg), each
% INPUT
% file is as follows:
%   1) Int_Array: An array containing Average Intensity values, each
%      fusion event as one column
%   2) exposure_n = exposure value (in milliseconds) used when collecting data, this
%      represents the time interval used to collect each data point and this value is
%      used to calculate the time range and interval for the
%      fusion event
%   3) Int_DiffP:  An array containing Average Intensity values from the second
%      color channel, if any.
%   4) CFbAvg= Cell background value calculated before function is
%      called and used for FoverS calculation.
% OUTPUT includes:
%   1) out_Aligned_PksC = fusion event intensity data, with onset of fusion aligned to
%      T=0s
%   2) out_Aligned_PksC_P= for two channel data, this output aligns the
%      intensity from the second channel to the onset of fusion aligned to T=0s
%   3) outFoverS = F over S is divides Fluorescence intesnity values by the Cell
%      background
%      value (CFbAvg-if provided when calling the function)
%   
% Determine number of columns (ncol) and number of rows (nr) of the input
%file
ncol = length(Int_Array(1,:));
[nr,_] = size(Int_Array);
%
% Create empty arrays to fill with output data
out_Aligned_PksC = zeros(nr,ncol);
out_Aligned_PksC_P = zeros(nr,ncol);
out_Aligned_PksC_I = zeros(nr,ncol);
outFoverS=zeros(nr,ncol);
%
%Loop through all fusion events, first column has zeros, so loop begins at
%column (c) =2

for c = 2:ncol

%Isolate the fusion event being aligned &
%Find the max value (Max_value), and location of the max value (Loc_max) for the
%isolated
%event
Fusion_event = Int_Array (:,c);
[Max_value,Loc_max] = max (Fusion_event);

%keep peaks with at least a few points (0.99* nr (total number of rows))
%if not enough points before fusion event, do not include event in analysis
%and continue analysis of the next column

if Loc_max >0.99*nr
    continue
end

% Find the range of data points to calculate background average for the
% fusion event
% Loc_max = location of max, st= starting point for background avg. calculation
%en = end point for background avg. calculation
if Loc_max<20
    start_time =1;
en=5;
else
    start_time=1;
en=20;
end

% Calculate background average for fusion event (Bck_avg_FE)
%and only keep fusion events with max value > 1.4x Bck_avg_FE

Bck_avg_FE = mean(Fusion_event(start_time:en,1));
if Bck_avg_FE>1
Bck_avg_FE=Bck_avg_FE;
else
Bck_avg_FE=1;
end

Max_Bckgrd_ratio= Max_value/Bck_avg_FE;

%Additonaly, only keep peaks that maintain more that 60% of the Intensity at Max+1 % data point. This is to exclude pixel noise which generally only lasts one data %point

if Max_Bckgrd_ratio > 1.4 && Fusion_event(Loc_max+1,1)/Max_value > 0.6

%For peaks that meet conditions, begin alinging of peak onset at T=0 %Divide all Intensity values for fusion event (Fusion_event(1:end)) by the background average %Bck_avg_FE)
%Search for point where
Fusion_Int_Bckgrd_ratio= Fusion_event(1:Loc_max)./Bck_avg_FE;
Max_Int_ratio=Fusion_event(1:Loc_max)./Max_value;

%Find te point were fusion begins(Fusion_onset_Point)
Fusion_onset_Point = find(Fusion_Int_Bckgrd_ratio >= (Max_Bckgrd_ratio - 0.5) & Max_Int_ratio >= 0.5 ,1,'first');

%Align the Intensity data with fusion onset point at T=0, T=0 is located at %row 200 %If fusion onset point is located in a row < 200, data will be pushed %forward so onset point is at T=0 and empty cells will be filled with NaNs

if Fusion_onset_Point < 200
  %Figure out how many data points data needs to be shifted by so onset of %fusion is at T=0 % Solve for x, were x = the number of data points that need the data needs % to be pushed forward so Fusion onset is at row 200 (T=0s).
  %Find x
  syms x
  x_shift = solve(x + Fusion_onset_Point == 200, x);
  %x_shift = x, the number of points data needs to be pushed forward
x_shift = double(x_shift);
%convert x_shift to double
%create an array of dimensions [S 1], filled with NaNs
array_nans = NaN([x_shift 1],'double');
%Create an array containing fusion event intensity data
%Concatenate the array containing NaNs to the beginning of the array containing
%fusion event intensity data
array_Fusionevent = Fusion_event;
Concatenate_arrays = [array_nans;array_Fusionevent];
%add concatenated array into output file
out_Aligned_PksC(:,c) = Concatenate_arrays(1:nr);

%Convert row numbers to Time points, using exposure value provided by user
%convert exposure value from milliseconds to seconds, exposure value =
%time interval between each frame recorded
expo = exposure_n/1000;
%total duration of data collection
total_time = nr*expo;
start_time = -9.95;
end_time = total_time+(start_time-expo);
%create array with time interval from expo
time_array = (start_time:expo:end_time); time_array_h = time_array';
out_Aligned_PksC(:,1) = time_array_h(:,1);

% For data with two channels i.e simultaneous analysis of two diff. fluo.
% molecules, the second fluorescent molecule is referred to as protein in this data
% analysis
%And we need to shift the protein data from the second channel so it
%matches with our aligned data
if nargin > 4

Protein_event = Int_DiffP(:,c);
%FoverSP= Protein event intensity data divided by Average Background
%Value(CFbAvg) for the data
FoverSP = Protein_event/CFbAvg;
%Create an array containing protein event intensity data
array_proteinevent = Protein_event;
%Concatenate the array containing NaNs to the beginning of the array containing
%protein event intensity data
Concatenate_p_array = [array_nans;array_proteinevent];
out_Aligned_PksC_P(:,c) = Concatenate_p_array(1:nr);
Concatenate_FoverS_array = [array_nans;FoverSP];
outFoverS(:,c) = Concatenate_FoverS_array(1:nr);

%Convert row numbers to Time points, using exposure value provided by user
% convert exposure value from milliseconds to seconds, exposure value =
% time interval between each frame recorded
expo = exposure_n/1000;
%total duration of data collection
total_time = nr*expo;
start_time=-9.95;
end_time=total_time+(start_time-expo);
%create array with time interval from expo
time_array = (start_time:expo:end_time); time_array_h = time_array';

out_Aligned_PksC_P(:,1) = time_array_h(:,1);
outFoverS(:,1) = time_array_h(:,1);
end

if nargin > 2

Int_array_event = Int_2Array(:,c);
Concatenate_I_array= [array_nans;Int_array_event];
out_Aligned_PksC_I(:,c) = Concatenate_I_array(1:nr);

elseif Fusion_onset_Point > 200

% Figure out how many data points data needs to be shifted by so onset of
% fusion is at T=0
% Solve for x_shift, were x = the number of data points that need the data needs
% to be pushed backward so Fusion onset is at row 200 (T=0s).
% Find x
x_shift = Fusion_onset_Point - 200 ;
%x_shift = x, the number of points data needs to be pulled backward
% create an array of dimensions [x_shift 1], filled with NaNs
array_nans = NaN([x_shift 1]);
shifted_fusionevent = Fusion_event(x_shift+1:end);
Concatenate_array = [shifted_fusionevent;array_nans];
out_Aligned_PksC(:,c) = Concatenate_array(1:nr);

% Convert row numbers to Time points, using exposure value provided by user
% convert exposure value from milliseconds to seconds, exposure value =
% time interval between each frame recorded
expo = exposure_n/1000;
% total duration of data collection
total_time = nr*expo;
start_time = -9.95;
end_time = total_time + (start_time-expo);
% Create array with time interval from expo
time_array = (start_time:expo:end_time);
time_array_h = time_array';
out_Aligned_PksC(:,1) = time_array_h(:,1);

% For data with two channels i.e simultaneous analysis of two diff. fluo.
% molecules, the second fluorescent molecule is referred to as protein in this data
% analysis
% And we need to shift the protein data from the second channel so it
% matches with our aligned data
if nargin > 4
    Protein_event = Int_DiffP(:,c);
sifted_proteinevent = Protein_event(x_shift+1:end);
Concatenate_p_array = [sifted_proteinevent;array_nans];
out_Aligned_PksC_P(:,c) = Concatenate_p_array(1:nr);
FoverSP = Protein_event/CFbAvg;
shifted_FoverSP = FoverSP(x_shift+1:end);
Concatenate_FoverSP = [shifted_FoverSP;array_nans];
outFoverS(:,c) = Concatenate_FoverSP(1:nr);

% Convert row numbers to Time points, using exposure value provided by user
% convert exposure value from milliseconds to seconds, exposure value =
% time interval between each frame recorded
expo = exposure_n/1000;
% total duration of data collection
total_time = nr*expo;
start_time = -9.95;
end_time = total_time + (start_time-expo);
% Create array with time interval from expo
time_array = (start_time:expo:end_time);
time_array_h = time_array';
out_Aligned_PksC_P(:,1) = time_array_h(:,1);
outFoverS(:,1) = time_array_h(:,1);
%Output of ALigned Int_Array to check if any peaks were misses
if nargin > 2

Int_array_event = Int_2Array(:,c);
sifted_intensityevent = Int_array_event(x_shift+1:end);
Concatenate_I_array= [sifted_intensityevent;array_nans];
out_Aligned_PksC_I(:,c) = Concatenate_I_array(1:nr);

%Convert row numbers to Time points, using exposure value provided by user
% convert exposure value from milliseconds to seconds, exposure value =
% time interval between each frame recorded
expo = exposure_n/1000;
%total duration of data collection
total_time = nr*expo;
start_time = -9.95;
end_time = total_time+(start_time-expo);
%create array with time interval from expo
time_array = (start_time:expo:end_time);
time_array_h = time_array';
out_Aligned_PksC_I(:,1) = time_array_h(:,1);
end
end
end
end
end

%CURVE FITTING
function [fitparam] = decayfit2(file)

% PURPOSE: to obtain decay fit parameters for fusion event peak onwards
% INPUT:
% file: fusion event intensity data peak onwards
% OUTPUT: fit parameters

ncol = length(file(1,:));
fitparam = zeros(ncol,5);
time_x = file(:,1);

ft = fittype( 'p+a*exp(-b*x)+c*exp(-d*x)', 'independent', 'x', 'dependent', 'y' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.StartPoint = [1 1 1 0 1];
opts.Lower=[0 0 0 0 0]; opts.Upper=[100 100 100 100 100];
% loop through all decay curves
    for c = 2 : ncol
        decay_curve = file(:,c);
    end
% isolate fusion event decay curve to onwards
% find location of peak (I)
    [~,I] = max(decay_curve(:,1));
% Time-aligned fusion event traces have Nans either before or after the % onset of fusion event. Nans help adjust the onset of fusion event to t=0
% Check if Nan's were added before or after fusion event in order to exclude % Nans from Nan's were added before or after fusion event in order to exclude
    y1 = isnan(decay_curve(1,1));
% if the first value of fusion event is a Nan we can fit the entire decay curve from peak to end
    if y1 == 1
        [~,I] = max(decay_curve(:,1));
        decaycurve = decay_curve(I:end,:);
        length_y = length(decaycurve(:,1));
        time_x2 = time_x(200:length_y+199,1);
        xandy=horzcat(time_x2,decaycurve);
        x2=xandy(:,1);y2=xandy(:,2);
        fp = fit( x2, y2, ft, opts );
        fitparam(c,:)=coeffvalues(fp);
    else
        y1==0
        % find first Nan value post peak intensity
        fnd_nan_val=find(isnan(decay_curve),1,'first');
        % last point
        last_point_dcurve=fnd_nan_val-1;
        decaycurve = decay_curve(I:last_point_dcurve,:);
        % length of x (time) and decay curve need to be equal
        length_y = length(decaycurve(:,1));
        time_x2 = time_x(1:length_y,1);
        xandy=horzcat(time_x2,decaycurve);
        x2=xandy(:,1);y2=xandy(:,2);
        % f = fit(x2,y2,'Exp2');
        fp = fit(x2,y2,ft,opts);
        fitparam(c,:)=coeffvalues(fp);
    end
end
% name each column and write out fit parameters for each decay curve
fitparam=fitparam(2:end,:);
curr_directory = pwd; current_folder_name = curr_directory(end-3:end);
C = cell(ncol,1);
for n2 = 2: ncol
filnam = sprintf('%s_%d',current_folder_name,n2);
C(n2,:) = {filnam};
end
rowNames = C(2:end,:);
fitparam = array2table(fitparam,'RowNames',rowNames);
curr_directory = pwd; current_folder_name = curr_directory(end-3:end);
output_excel_file = sprintf('Analyzed Data_%s.xls',current_folder_name);
writetable(fitparam, output_excel_file,'Sheet','Decayfit Data','WriteRowNames',true);
end

%COMPILE DATA
function [out_Aligned_PksC2,
out2Aligned_PksC_I2,out2_Aligned_PksC_P2,out_slope,out_percentloss]=
compile_data(Aligned_Pks,Aligned_PksC_I,Aligned_Pks_P,FoverS)
%
% PURPOSE:
% 1) To refine data by excluding non-real curves. Non real curves
% have all rows ==0
% 2) Calculate Slope, and % Loss in intensity at Time points = 1s and 5s
% for each decay curve
% When calling function pk_align(Int_Array,exposure_n,Int_DiffP,CFbAvg), each
INPUT
% file is as follows:
% 1) Aligned_Pks: An array containing Average Intensity values with onset of fusion
% event
% aligned to T=0, each fusion event as one column
% 2) Aligned_Pks_P = An array containing Average Intensity values for second color
channel (if any) with onset of fusion event
% aligned to T=0, each fusion event as one column
% 3) FoverS: An array containing BAckground corrected Average Intensity values
for second color channel (if any) with onset of fusion event
% aligned to T=0, each fusion event as one column
% OUTPUT includes:
% 1) out_Aligned_PksC = fusion event intensity data, with non real
% curves (=0) excluded.
% 2) out2_Aligned_PksC_P = for two channel data, this output as intensity data from
the second cannel, with non real
% curves (=0) excluded.
% 3) outFoverS = F over S is divides Fluorescence intensity values by the Cell
background

98
% file: Aligned peaks intensity data
% OUTPUT:  Aligned peaks intensity data excluding non real peaks

ncol = length(Aligned_Pks(1,:));
nr = length(Aligned_Pks(:,1));

% Create empty arrays to fill with output data
out_Aligned_PksC = Aligned_Pks(:,1);
out_slope = Aligned_Pks(1,1);
out_percentloss = Aligned_Pks(1:2,1);
out_Aligned_PksC_I = Aligned_PksC_I(:,1);
%for two cannel data, nargin >1
%if nargin>1 create empty arrays to fill second channel data
if nargin > 2
out2_Aligned_PksC_P = Aligned_Pks_P(:,1);
outFoverS = FoverS(:,1);
end

% Loop through all fusion events, first column has time, so loop begins at
% column (c) =2
for c = 2 : ncol
% Sum of column with fusion event data >0 if column contains data
Sum_Fusionevent(:,c) = sum(Aligned_Pks(:,c),'omitnan');
%if Sum_Fusionevent >0, fusion event is output and further analyzed via calculation of
%slope and % loss
if Sum_Fusionevent(:,c) > 0
out_Aligned_PksC = [out_Aligned_PksC Aligned_Pks(:,c)];
out_Aligned_PksC_I = [out_Aligned_PksC_I Aligned_PksC_I(:,c)];
% Find Max and location of max to calculate slope and % loss
[Max_value,Max_location] = max(Aligned_Pks(:,c));
% Calculate slope

if Max_location+20 <nr
avg_time= mean(Aligned_Pks(Max_location:Max_location+20,1));
avg_decaycurve = mean(Aligned_Pks(Max_location:Max_location+20,c));
for i = Max_location:Max_location+20

end

end

end
slope_decaycurve(i,1) = ((Aligned_Pks(i,1) - avg_time)*(Aligned_Pks(i,c) - avg_decaycurve));
slope_decaycurve(i,2) = ((Aligned_Pks(i,1) - avg_time)^2);
end
slope_decaycurve = slope_decaycurve(Max_location:Max_location+20,:);
slope_decaycurve_y = sum(slope_decaycurve(:,1));
slope_decaycurve_x=sum(slope_decaycurve(:,2));slope_rise_overrun = slope_decaycurve_y/slope_decaycurve_x;
slope_final = (slope_rise_overrun/Max_value);

%Calculate % intensity lost after 1s data points, baseline (first 150 points)

intensity_initial= Max_value;
intensity_final= (Aligned_Pks(Max_location+20,c));
baseline= mean(Aligned_Pks(1:150,c),'omitnan');
percent_loss_1s= ((intensity_initial-intensity_final)/(intensity_initial-baseline))*100;
else
  %if the the maximum value is not followed by 20 data points, we cannot
  %calculate the slope and an output value of 0 is generated for the slope
  %and for % loss
  slope_final=0;
  percent_loss_1s=0;
end

%if the the maximum value is followed by 100 data points, we can
%calculate the % intensity loss after 5s of peak value
if Max_location+100 < nr
  %Calculate % intensity lost after 5s data points, baseline substracted
  intensity_initial= Max_value;
  intensity_final= (Aligned_Pks(Max_location+100,c));
  baseline= mean(Aligned_Pks(1:150,c),'omitnan');
  percent_loss_5s= ((intensity_initial-intensity_final)/(intensity_initial-baseline))*100;
else
  percent_loss_5s=0;
end
compiled_percent_loss=[percent_loss_1s;percent_loss_5s];
out_percentloss = [out_percentloss compiled_percent_loss];
out_slope=[out_slope slope_final];

%for two channel data, compile second channel data
if nargin > 2
  out2_Aligned_PksC_P = [out2_Aligned_PksC_P Aligned_Pks_P(:,c)];
  outFoverS = [outFoverS FoverS(:,c)];
end
end
end

% Folder name is the same as data name (#), so file names are assigned using the folder name
curr_directory = pwd; current_folder_name = curr_directory(end-3:end);
C = cell(1, ncol);
for n = 1: ncol
    filnam = sprintf('%s_%d', current_folder_name, n);
    C(:,n) = {filnam};
end

% Data is output in an excel file, and the data is converted to excel compatible format
out_Aligned_PksC2 = out_Aligned_PksC;
out_Aligned_PksC = mat2dataset(out_Aligned_PksC);
out_Aligned_PksC = dataset2table(out_Aligned_PksC);
out_slope = mat2dataset(out_slope);
out_slope = dataset2table(out_slope);
out_percentloss = mat2dataset(out_percentloss);
out_percentloss = dataset2table(out_percentloss);
out2Aligned_PksC_I2 = out2Aligned_PksC_I;
out2Aligned_PksC_I = mat2dataset(out2Aligned_PksC_I);
out2Aligned_PksC_I = dataset2table(out2Aligned_PksC_I);

% Data is output in an excel file, and the data is converted to excel compatible format for second channel
if nargin > 2
    out2_Aligned_PksC_P2 = out2_Aligned_PksC_P;
    out2_Aligned_PksC_P = mat2dataset(out2_Aligned_PksC_P);
    out2_Aligned_PksC_P = dataset2table(out2_Aligned_PksC_P);
    outFoverS = mat2dataset(outFoverS);
    outFoverS = dataset2table(outFoverS);
end

% Folder name is the same as data name (#), so column names are assigned to both
% te fusion event and the corresponding slope/percentloss data using the folder name
ll = size(out_Aligned_PksC(1,:));
ll = ll(:,2);
for l = 1 : ll
    out_Aligned_PksC.Properties.VariableNames{l} = C{l};
    out2_Aligned_PksC_P.Properties.VariableNames{l} = C{l};
    outFoverS.Properties.VariableNames{l} = C{l};
end
% All data is written out in excel that is named according to the folder name, Fusion, Slope, Protein loss etc are all output to different tabs in excel with labels.

output_excel_file = sprintf('Analyzed Data_%%s.xls',current_folder_name);
writetable(out_Aligned_PksC, output_excel_file,'Sheet','Fusion');
writetable(out_slope, output_excel_file,'Sheet','Slope');
writetable(out_percentloss, output_excel_file,'Sheet','% Loss');
writetable(out_Aligned_PksC_I, output_excel_file,'Sheet','Fusion_RawData');
if nargin > 2
    writetable(out2_Aligned_PksC_P, output_excel_file,'Sheet','Protein');
    writetable(outFoverS, output_excel_file,'Sheet','Protein_FoverS');
end
end

% NORMALIZE DATA
function [out_Norm_Data_Fusion, out_Norm_Data_Protein] = normalizedata2(Aligned_PksC, Aligned_PksC_P)

% PURPOSE: To normalize fusion and protein data using minimum value from data points before the onset of fusion event. This is to avoid changing the shape and kinetics of the decay curve.

% When calling function normalizedata2(Aligned_PksC, Aligned_PksC_P), each input file is as follows:
% 1) Aligned_Pks: An array containing Average Intensity values with onset of fusion event aligned to T=0, each fusion event as one column
% 2) Aligned_Pks_P = An array containing Average Intensity values for second color channel (if any) with onset of fusion event aligned to T=0, each fusion event as one column
% OUTPUT includes:
% 1) Norm_Data_Fusion = Normalized fusion event intensity data, with onset of fusion aligned to T=0s
% 2) Norm_Data_Protein= for two channel data, this output normalizes the intensity from the second channel

% Determine number of columns (ncol) and number of rows (nr) of the input file
nframes = length(Aligned_PksC(:,1));
ncols = length(Aligned_PksC(1,:));
% Determine number of columns (ncol) and number of rows (nr) of the input
% in multi channel data
if nargin>1
ncols2 = length(Aligned_PksC_P(1,:));
end
% Create empty arrays to fill with output data
Norm_Data_Fusion = zeros(nframes,ncols);
Norm_Data_Protein = zeros(nframes,ncols);

% Loop through all fusion events, first column has time, so loop begins at
% column (c) = 2
for c = 2:ncols

% Find the Max value of the fusion event i.e. peak
% Find a minimum value from frames before fusion event
max_data_fusion = max(Aligned_PksC(:,c));
min_data_fusion = min(Aligned_PksC(1:199,c));

% Loop through each data point and normalize using the min and max values
% obtained above nframes=number of rows
    for i = 1:nframes
        i_intensity_value = Aligned_PksC(i,c);
        Norm_Data_Fusion(i,c) = ((i_intensity_value - min_data_fusion)/(max_data_fusion - min_data_fusion));
    end

% Copy the Time column from input file
    Norm_Data_Fusion(:,1) = Aligned_PksC(:,1);

% For two channel data, nargin > 1
% if nargin>1 cnormalize data from second channel
if nargin > 1

% Loop through all columns of second data set
for k = 1:ncols2

% Find the min and max values for each column of second data set
max_data_protein = max(Aligned_PksC_P(:,k));
min_data_protein = min(Aligned_PksC_P(:,k));

k_intensity_value = Aligned_PksC_P(i,k);
Norm_Data_Protein(i,k) = ((k_intensity_value - min_data_protein)/(max_data_protein - min_data_protein));

end
end
end
end

end

% Data is output in an excel file, and the data is converted to excel
% compatible format
out_Norm_Data_Fusion = Norm_Data_Fusion;
Norm_Data_Fusion = mat2dataset(Norm_Data_Fusion);
Norm_Data_Fusion = dataset2table(Norm_Data_Fusion);
% Data is output in an excel file, and the data is converted to excel
% compatible format for second channel
if nargin > 2
out_Norm_Data_Protein = Norm_Data_Protein;
Norm_Data_Protein = mat2dataset(Norm_Data_Protein);
Norm_Data_Protein = dataset2table(Norm_Data_Protein);
end
% Folder name is the same as data name (#), so file (column) names are assigned using
% the folder name
curr_directory = pwd; current_folder_name = curr_directory(end-3:end);
C = cell(1, ncols);
for n2 = 1: ncols
filnam = sprintf('%s_%d', current_folder_name, n2);
C(:,n2) = {filnam};
end
if nargin > 2
% Column names for second data set
ll = size(Norm_Data_Fusion(1,:));
ll = ll(:,2);
for l = 1 : ll
Norm_Data_Fusion.Properties.VariableNames{l} = C{l};
Norm_Data_Protein.Properties.VariableNames{l} = C{l};
end
end
% All data is written out in excel that is named according to the folder
% name, and data sets are put in tabs with their corresponding names
output_excel_file = sprintf('Analyzed Data_%s.xls', current_folder_name);
writetable(Norm_Data_Fusion, output_excel_file, 'Sheet', 'Normalized Fusion');

if nargin > 1
writetable(Norm_Data_Protein, output_excel_file, 'Sheet', 'Normalized Protein');
end
end
function [sh_movie1,sh_movie2] = align_movies(Aligned_Pks,movie1,movie2)
% PURPOSE:
% 1) To create mini movies with fusion events in the middle of the mini
% movie e.g for a 200 frame length mini move the onset of fusion event is at
% approximately frame 100
% 2) Create montages of input movies
% When calling function align_movies(Aligned_Pks,movie1,movie2), each INPUT
% file is as follows:
% 1) Aligned_Pks: An array containing Average Intensity values with onset of fusion
% event
%   aligned to T=0, each fusion event as one column. This file is
%   used to create minis with the onset of fusion event centered.
% 2) movie1 = A 25x25 pixels 4D array containing all fusion events.
% 3) movie2 = A 25x25 pixels 4D array containing all second channel frames (if any)
% accompanying the fusion event.
% OUTPUT includes:
% 1) sh_movie1 = A 25x25 pixels 4D array containing all fusion event with fusion
% event aligned to the center of each mini for the respective fusion
% event.
% 2) sh_movie2 = A 25x25 pixels 4D array containing all second channel frames (if
% any) accompanying the fusion event.
% Each event is aligned to the center of each mini for the respective fusion
% event.
% Determine number of columns (ncol) and number of rows (nr) of the input
% file
ncol = length(Aligned_Pks(1,:));
[nr,~] = size(Aligned_Pks);

% figure out how big the output arrays will be
n_movies = ncol-(length(find(Aligned_Pks(:,1) == 0))+1);

% find all the time shifts, note that the first column
% contains time info (just 0s).
idx = 1;
% find the peak onset time (T=0s) that the peaks are aligned to in the array
% if data includes no fusion event Aligned_Pks(:,1)==0
if Aligned_Pks(:,1)==0
    sh_movie1=[];
    disp('no events to make minis');
return;
end

% Loop through all fusion events, first column has time, so loop begins at
% column (c) = 2
for c = 2 : ncol
% isolate each fusion event and check if the column contains a real event
% e. sum of column > 0
    Fusion_Event_x(:,c) = Aligned_Pks(:,c);
    sum_Fusion_Event_x(:,c) = sum(Aigned_Pks(:,c),'omitnan');
    if sum_Fusion_Event_x(:,c) > 0
        % Determine length of output mini for the fusion event
        % Determine if fusion event has 100 frames before the onset of the event,
        % for events with less than 200 frames before onset, the first row is nan
        real_fusion_event = Fusion_Event_x(1,c);
        if isnan(real_fusion_event)
            % Determine the last nan location to determine how many frames of data are
            % present before the fusion onset
            last_nan = find(isnan(Fusion_Event_x(:,c)),1,'last');
            if last_nan < 100
                % Determine location of the first frame for output mini for the fusion event
                start_frame = 200 - last_nan;
                % Create empty arrays to store the mini, and to create the mini, these files
                % are output
                sh_movie1 = uint16(zeros(25,25,200,idx));
                shmovie1 = uint16(zeros(25,25,200));
                % Write mini into the output file
                sh_movie1(:,:,start_frame-100:start_frame+99,c) = movie1(:,:,start_frame-100:start_frame+99,c);
                shmovie1(:,:,start_frame-100:start_frame+99,c) = sh_movie1(:,:,start_frame-100:start_frame+99,c);
            end
        end
    end
end
% Loop through each frame to write out mini as a tif file
% Each mini is named according to the column number of the fusion event
% For 2 channel data the second mini is also created

for frame_num = 1:200
    shmovie1(:,:,frame_num) = shmovie1(:,:,frame_num);
    mini_fusion = sprintf("Spot%dmini.tif",c);
end
imwrite(shmovie1(:,:,frame_num),mini_fusion,'WriteMode','append','compression','none');

if nargin>2
    shmovie2(:,:,frame_num) = shmovie2(:,:,frame_num);
    mini_protein = sprintf('Spot%dminip.tif',c);
end

imwrite(shmovie2(:,:,frame_num),mini_protein,'WriteMode','append','compression','none');
end

end

%Create montages for the fusion event and the accompanying second channel data, if any
%Write montages out as tif

shmovie1_montage=(shmovie1(:,:,1:20:end));
if nargin>2
    shmovie2_montage=(shmovie2(:,:,1:20:end));
end
mont_size=size(shmovie1_montage,3);
for num_frame = 1:mont_size
    shmovie1_montage(:,:,num_frame) = shmovie1_montage(:,:,num_frame);
    mininmontage = sprintf('Spot%dminimontage.tif',c);
    imwrite(shmovie1_montage(:,:,num_frame),mininmontage,'WriteMode','append','compression','none');
end

if nargin>2
    shmovie2_montage(:,:,num_frame) = shmovie2_montage(:,:,num_frame);
    mininpmontage = sprintf('Spot%dminipmontage.tif',c);
    imwrite(shmovie2_montage(:,:,num_frame),mininpmontage,'WriteMode','append','compression','none');
end

end

end

end

%Determine length of minis when fusion event has less than 100 frame length
%data before onset of fusion (T=0s,row=200)
if last_nan > 100 && last_nan < 200
    frames_before_fusion = 200-last_nan;
    strt = frames_before_fusion -1;
    pkstrt = frames_before_fusion-strt;
    mini_length = frames_before_fusion*2;
end

%Create empty arrays to store the mini, and to create the mini, these files are output
sh_movie1 = uint16(zeros(25,25,mini_length,idx));
shmovie1 = uint16(zeros(25,25,mini_length));
sh_movie1(:,:,idx) = movie1(:,:,pkstrt:mini_length,c);
shmovie1(:,:,:) = sh_movie1(:,:,idx);
if nargin >2
sh_movie2 = uint16(zeros(25,25,mini_length,idx));
sh_movie2(:,:,idx) = movie2(:,:,pkstrt:mini_length,c);
shmovie2(:,:,:) = sh_movie2(:,:,idx);
end

%loop through each frame to write out mini as a tif file
%each mini is named according to the column number of the fusion event
%for 2 channel data the second mini is also created

    for frame_num = 1:mini_length
        shmovie1(:,:,frame_num) = shmovie1(:,:,frame_num);
        mini_fusion= sprintf('Spot%dmini.tif',c);
        imwrite(shmovie1(:,:,frame_num),mini_fusion,'WriteMode','append','compression','none');
        if nargin>2
            shmovie2(:,:,frame_num) = shmovie2(:,:,frame_num);
            mini_protein = sprintf('Spot%dminip.tif',c);
            imwrite(shmovie2(:,:,frame_num),mini_protein,'WriteMode','append','compression','none');
        end
    end

%Create montages for the fusion event and the accompanying second channel
%data, if any
%Write montages out as tif

    shmovie1_montage=(shmovie1(:,:,1:20:end));
    if nargin>2
        shmovie2_montage=(shmovie2(:,:,1:20:end));
    end
    mont_size=size(shmovie1_montage,3);
    for num_frame = 1:mont_size
        shmovie1_montage(:,:,num_frame) = shmovie1_montage(:,:,num_frame);
        mininmontage = sprintf('Spot%dminimontage.tif',c);
        imwrite(shmovie1_montage(:,:,num_frame),mininmontage,'WriteMode','append','compression','none');
        if nargin>2
...
shmovie2_montage(:,:,num_frame) = shmovie2_montage(:,:,num_frame);
mininpmontage = sprintf('Spot%dminipmontage.tif',c);
imwrite(shmovie2_montage(:,:,num_frame),mininpmontage,'WriteMode','append','compression','none');
end
end

% Determine length of output mini for the fusion event
% Determine how many frames of data are present post onset of the event,
% for events with greater than 200 frames before onset, the first row is not nan
elseif ~isnan(real_fusion_event)
  first_nan = find(isnan(Fusion_Event_x(:,c)),1,'first');
total_nan = nr-first_nan+201 ;
if total_nan < nr-99
  lengthvid = 200;
% Create empty arrays to store the mini, and to create the mini, these files
% are output
  sh_movie1 = uint16(zeros(25,25,200,idx));
  shmovie1 = uint16(zeros(25,25,200));
  sh_movie1(:,:,total_nan-100:total_nan+99,c) = sh_movie1(:,:,total_nan-100:total_nan+99,c);
  if nargin>2
    sh_movie2 = uint16(zeros(25,25,200,idx));
    shmovie2 = uint16(zeros(25,25,200));
    sh_movie2(:,:,total_nan-100:total_nan+99,c) = sh_movie2(:,:,total_nan-100:total_nan+99,c);
  end
% Loop through each frame to write out mini as a tif file
% each mini is named according to the column number of the fusion event
% for 2 channel data the second mini is also created
  for frame_num = 1:lengthvid
    shmovie1(:,:,frame_num) = shmovie1(:,:,frame_num);
    mini_fusion = sprintf('Spot%dmini.tif',c);
imwrite(shmovie1(:,:,frame_num),mini_fusion,'WriteMode','append','compression','none');
  end
  if nargin>2
    shmovie2(:,:,frame_num) = shmovie2(:,:,frame_num);
    mini_protein = sprintf('Spot%dminip.tif',c);
imwrite(shmovie2(:,:,frame_num),mini_protein,'WriteMode','append','compression','none');
  end
end
%Create montages for the fusion event and the accompanying second channel
%data, if any
%Write montages out as tif
if nargin>2
    shmovie1_montage=(shmovie1(:,:,1:20:end));
    shmovie2_montage=(shmovie2(:,:,1:20:end));
    mont_size=size(shmovie1_montage,3);
    for num_frame = 1:mont_size
        shmovie1_montage(:,:,num_frame) = shmovie1_montage(:,:,num_frame);
        shmovie2_montage(:,:,num_frame) = shmovie2_montage(:,:,num_frame);
        mininmontage = sprintf('Spot%dminimontage.tif',c);
        mininpmontage = sprintf('Spot%dminipmontage.tif',c);
        imwrite(shmovie1_montage(:,:,num_frame),mininmontage,'WriteMode','append','compression','none');
        imwrite(shmovie2_montage(:,:,num_frame),mininpmontage,'WriteMode','append','compression','none');
    end
end
%Determine length of minis for fusion events with less than 100 frames post
%fusion
if total_nan> nr-99
    mini_length = nr-(total_nan);
    start_frame = total_nan-(mini_length);
    lengthvid= nr-(start_frame);
%Create empty arrays to store the mini, and to create the mini, these files
%are output
    sh_movie1 = uint16(zeros(25,25,lengthvid,idx));
    shmovie1 = uint16(zeros(25,25,lengthvid));
    sh_movie1(:,:,1:lengthvid) = movie1(:,:,start_frame:total_nan+mini_length-1,c);
    shmovie1(:,:,1:lengthvid) = sh_movie1(:,:,1:lengthvid);
    if nargin>2
        sh_movie2 = uint16(zeros(25,25,lengthvid,idx));
        shmovie2 = uint16(zeros(25,25,lengthvid));
        sh_movie2(:,:,1:lengthvid) = movie2(:,:,start_frame:total_nan+mini_length-1,c);
        shmovie2(:,:,1:lengthvid) = sh_movie2(:,:,1:lengthvid);
    end
%loop through each frame to write out mini as a tif file
%each mini is named according to the column number of the fusion event
%for 2 channel data the second mini is also created
for frame_num = 1:lengthvid
    shmovie1(:,:,frame_num) = shmovie1(:,:,frame_num);
mini_fusion = sprintf('Spot%d mini.tif', c);

imwrite(shmovie1(:, :, frame_num), mini_fusion, 'WriteMode', 'append', 'compression', 'none');
    if nargin > 2
        shmovie2(:, :, frame_num) = shmovie2(:, :, frame_num);
        mini_protein = sprintf('Spot%d mini protein.tif', c);
        imwrite(shmovie2(:, :, frame_num), mini_protein, 'WriteMode', 'append', 'compression', 'none');
    end
end

% Create montages for the fusion event and the accompanying second channel data, if any
% Write montages out as tif
    shmovie1_montage = (shmovie1(:, 1:20:end));
    if nargin > 2
        shmovie2_montage = (shmovie2(:, 1:20:end));
        end
        mont_size = size(shmovie1_montage, 3);
        for num_frame = 1:mont_size
            shmovie1_montage(:, num_frame) = shmovie1_montage(:, num_frame);
            mininmontage = sprintf('Spot%d mini montage.tif', c);
            imwrite(shmovie1_montage(:, :, num_frame), mininmontage, 'WriteMode', 'append', 'compression', 'none');
            if nargin > 2
                shmovie2_montage(:, num_frame) = shmovie2_montage(:, num_frame);
                mininpmontage = sprintf('Spot%d mini pm montage.tif', c);
                imwrite(shmovie2_montage(:, :, num_frame), mininpmontage, 'WriteMode', 'append', 'compression', 'none');
            end
        end
end
end
end

% CREATE_MONTAGE
function [sh_movie1, sh_movie2] = create_montage(Aligned_Pks, movie1, movie2)
% PURPOSE:
% 1) To create mini movies with fusion events in the middle of the mini
% movie e.g for a 200 frame length mini move the onset of fusion event is at
% approximately frame 100
% 2) Create montages of input movies
% When calling function align_movies(Aligned_Pks,movie1,movie2), each INPUT
% file is as follows:
% 1) Aligned_Pks: An array containing Average Intensity values with onset of fusion
%      event aligned to T=0, each fusion event as one column. This file is
%      used to create minis with the onset of fusion event centered.
% 2) movie1 = A 25x25 pixels 4D array containing all fusion events.
% 3) movie2 = A 25x25 pixels 4D array containing all second channel frames (if any)
%      accompanying the fusion event.
% OUTPUT includes:
% 1) sh_movie1 = A 25x25 pixels 4D array containing all fusion event with fusion
%      event aligned to the center of each mini for the respective fusion
%      event.
% 2) sh_movie2 = A 25x25 pixels 4D array containing all second channel frames (if
%      any) accompanying the fusion event.
% Each event is aligned to the center of each mini for the respective fusion
% event.
% % % % % % % % %
% Determine number of columns (ncol) and number of rows (nr) of the input
% file
ncol = length(Aligned_Pks(1,:));
[nr,~] = size(Aligned_Pks);

% figure out how big the output arrays will be
n_movies = ncol-(length(find(Aligned_Pks(1,:)==0))+1);

find all the time shifts, note that the first column
% contains time info (just 0s).
idx = 1;
find the peak onset time (T=0s) that the peaks are aligned to in the array
if data includes no fusion event Aligned_Pks(:,1)==0
    sh_movie1=[];
    disp('no events to make minis');
    return;
end

Loop through all fusion events, first column has time, so loop begins at
% column (c) =2
for c = 2 : ncol
%isolate each fusion event and check if the column contains a real event
%i.e. sum of column > 0
Fusion_Event_x(:,c) = Aligned_Pks(:,c);
sum_Fusion_Event_x(:,c) = sum(Aligned_Pks(:,c),'omitnan');
if sum_Fusion_Event_x(:,c) > 0

%Determine length of output mini for the fusion event at 20 frame interval

    length_montage = floor(nr/20);
%Convert the movie at c to 3D
mont_at_c = uint16(zeros(25,25,nr,idx));
mont_at_c(:,:,idx) = movie1(:,:,c);
mont_at_c = mont_at_c(:,:,,:);
if nargin > 2
    mont_at_c_p = uint16(zeros(25,25,nr,idx));
mont_at_c_p(:,:,idx) = movie2(:,:,c);
mont_at_c_p = mont_at_c_p(:,:,,:);
end
%Create empty arrays to store the mini, and to create the mini, these files
%are output
    sh_movie1 = uint16(zeros(25,25,length_montage));
    shmovie1 = uint16(zeros(25,25,length_montage));
if nargin > 2
    sh_movie2 = uint16(zeros(25,25,length_montage));
    shmovie2 = uint16(zeros(25,25,length_montage));
end
%write mini into the output file
    montage_interval = 1:20:nr;
    for i = 1:length_montage
        first_frame = montage_interval(1,i);
        montage_frame_mean = mean(mont_at_c(:,:,first_frame:first_frame+19),3);
        montage_frame_mean = uint16(montage_frame_mean);
        shmovie1(:,:,i) = montage_frame_mean;
    end
    if nargin > 2
        montage_frame_meanp = mean(mont_at_c_p(:,:,first_frame:first_frame+19),3);
        montage_frame_meanp = uint16(montage_frame_meanp);
        shmovie2(:,:,i) = montage_frame_meanp;
    end
%loop through each frame to write out mini as a tif file
%each mini is named according to the column number of the fusion event
%for 2 channel data the second mini is also created
for frame_num = 1:length_montage
    shmovie1(:,:,frame_num) = shmovie1(:,:,frame_num);
    mini_fusion= sprintf('Spot%dmini_fullmontage.tif',c);
    imwrite(shmovie1(:,:,frame_num),mini_fusion,'WriteMode','append','compression','none');
end
if nargin>2
    for frame_num = 1:length_montage
        shmovie2(:,:,frame_num) = shmovie2(:,:,frame_num);
        mini_protein = sprintf('Spot%dminip_fullmontage.tif',c);
        imwrite(shmovie2(:,:,frame_num),mini_protein,'WriteMode','append','compression','none');
    end
end
%RADIAL PLOTS
function [r_avg3,area_rp2,HWHMO]=radial_plot(Aligned_Pks,Int_Diff,Indiv_events,totalpks)
% PURPOSE: loop though entire input movie to make calculate Half width full max from
% onset of fusion event to the end of movie using better_colo_v2(a) program included
% below
% INPUT:
% mov: movie stack, in the format of (512,512,300). It can be smaller in X
% and Y dimensions.
% time:
% OUTPUT: Half width half max in nanometers)and area under radial plot
% over time from fusion onset onwards.
% ncol = length(Aligned_Pks(1,:));
% nr = length(Aligned_Pks(:,1));
% totalpks1=length(totalpks(1,:));
% tp2=(totalpks1-1);
% totalpks=(totalpks1-1)*7;
% area_rp2=zeros(ncol,1000);
% radial_avg=zeros(83,totalpks);
% HWHMO=zeros(1,1001);
AREAO=zeros(1,1000);

% convert from pixels to nm
conversion =
[0,1,1.414213562,2,2.236067977,2.828427125,3,3.16227766,3.605551275,4,4.12310562
6.4.242640687,4.72135955,5,5.099019514,5.385164807,5.6568542490,5.83095189500
000,6,6.08276253000000,6.32455532000000,6.403124237000000,6.708203932000000,7,7.
071067812000000,7.211102551000000,7.280109889000000,7.615773106000000,7.81024967
6000000,8,8.062257748000000,8.246211251000000,8.485281374000000,8.5440037450000000
,8.602325267000000,8.944271910000000,9,9.055385138000000,9.219544457000000,9.43398
113200000,9.486832981000000,9.848857802000000,9.899494937000000,9.10,10.04987562000000000
,10.198039030000000,10.295630140000000,10.440306510000000,10.630145810000000,10.7
703296100000,10.816653830000000,11,11.045361020000000,11.180339890000000,11.313708
5000000,11.401754250000000,11.661903790000000,11.704699910000000,12,12.0415945800000
00,12.083045970000000,12.165525060000000,12.206555620000000,12.369316880000000,12.529
96049000000,12.649110640000000,12.727922060000000,12.806248470000000,13,13.03840481
142135620000000,14.212670400000000,14.422205100000000,14.866068750000000,15,15.55634
919000000,15.620409350000000,16.278820600000000,16.97056275];
  x_rad=conversion';
  x_rad=x_rad*102.7;

r_avg2=x_rad;
r_avg3=x_rad;
% loop through all frames of fusion event mini movie if the movie
% corrensponding to c represents a real fusion event
for c =2:ncol
  HWHM=zeros(1,1001);
  radial_avg=zeros(83,totalpks);
  AS = sum(Aligned_Pks(:,c),'omitnan');
  AAS2 = Int_Diff(:,c);
  [~,I] = max(AAS2);
  if AS < 0
    continue
  end
  if AS>0

    r_avg2=x_rad;
    AAS2 = Int_Diff(:,c);
    [~,I] = max(AAS2);
    mini = Indiv_events(:,:,I-1:end);
    lenmini=size(mini,3);
%writeout mini for radial plot
    for i=1:lenmini
        radial_out(:,i) = better_colo_V2(mini(:,:,i));
    end
if lenmini<20
    continue
end
    for n=1:20:lenmini-20
        radial_avg(:,n)=mean(radial_out(:,n:n+2),2);
        r_avg(:,n)=radial_avg(:,n);
        maxn=max(r_avg(:,n));
        bckn=r_avg(83,1);
        bck=(maxn-bckn);
        hmaxya1=bck/2;
        hmaxya=maxn-hmaxya1;
        uAvgn=unique(r_avg(:,n), 'stable');
        lenuAvg=length(uAvgn);
        xn=x_rad(1:lenuAvg,1);
        HWHM(1,n)= interp1(uAvgn,xn,hmaxya, 'makima');
        r_avgx= [r_avg r_avg(:,n)];
        r_avgx=r_avgx(:,1:end);
    end
        r_avgx=r_avgx(:,1:end-1);
        HWHMO=[HWHMO;HWHM];
    r_avgx2=r_avgx(:,1:20:end);
    r_avg3=[r_avg3 r_avgx2];
    clear mini;

    area_rp=zeros(1,1000);
    for j = 1:20:size(r_avgx,2)
        s=r_avgx(:,j);
        ydatatemp=s(7:38,:);
        ypeak=s(1:7,:);
        base=mean(ydatatemp);
        rect=base*7;
        ypeak
    end
116
%are under the peak

\[ x = \text{sum}(y_{\text{peak}}); \]
\[ \text{area}_{\text{rp2}}(c,j) = x - \text{rect}; \]
\[ \% \text{area}_{\text{rp}} = \text{area}_{\text{rp2}}; \]
\[ \% \text{this gives a number for the area of the peak from the colocalization} \]
\[ \% \text{as the answer} \]
\[ \% \text{area}_{\text{rp2}} = [\text{area}_{\text{rp2}} \ \text{area}_{\text{rp}}(:,j)] \]
\[ \text{area}_{\text{rp}}(::) = \text{area}_{\text{rp2}}(c,:); \]

end

\[ \text{AREA}_0 = [\text{AREA}_0; \text{area}_{\text{rp}}]; \]

clear \text{radial\_avg}
clear \text{radial\_out}
clear \text{r\_avg}
clear \text{HWHM}
clear \text{r\_avgx}
clear \text{area\_rp}
end
end

\% Compile calculated HWHM and Area data and output both as matfile and in excel
\[ \text{HWHM}_0 = \text{HWHM}_0(:,1:20:end); \]
\[ \text{AREA}_0 = \text{AREA}_0(:,1:20:end); \]
\[ \text{sv} = \text{size}(\text{HWHM}_0,1); \]
\[ \text{curr\_directory} = \text{pwd}; \text{current\_folder\_name} = \text{curr\_directory}(\text{end}-3: \text{end}); \]
\[ \text{C3} = \text{cell}(\text{sv},1); \]
\[ \text{for } n2 = 1: \text{sv} \]
\[ \text{filnam} = \text{sprintf}(''\%s \_\%d'', \text{current\_folder\_name}, n2); \]
\[ \text{C3}(n2,:) = \{\text{filnam}\}; \]
\[ \text{end} \]
\[ \text{rowNames3} = \text{C3}(1:\text{end},:); \]
\[ \% \text{rowNames4} = \text{C2}(2:\text{end},:); \]
\[ \text{HWHM}_0 = \text{array2table}(\text{HWHM}_0,''\text{RowNames}'', \text{rowNames3}); \]
\[ \text{sva} = \text{size}(\text{AREA}_0,1); \]
\[ \text{curr\_directory} = \text{pwd}; \text{current\_folder\_name} = \text{curr\_directory}(\text{end}-3: \text{end}); \]
\[ \text{C3a} = \text{cell}(\text{sva},1); \]
\[ \text{for } n2 = 1: \text{sva} \]
\[ \text{filnam} = \text{sprintf}(''\%s \_\%d'', \text{current\_folder\_name}, n2); \]
\[ \text{C3a}(n2,:) = \{\text{filnam}\}; \]
\[ \text{end} \]
\[ \text{rowNames3a} = \text{C3a}(1:\text{end},:); \]
AREAO = array2table(AREAO,'RowNames',rowNames3a);
output_excel_file = sprintf('Analysed Data_%s.xls',current_folder_name);
writetable(AREAO, output_excel_file,'Sheet','area_rp','WriteRowNames',true);
writetable(HWHMO, output_excel_file,'Sheet','HWHM','WriteRowNames',true);
end

function [ radial_out ] = better_colo_V2(a)
%better_colo plots intensities of a location guided average image as a
%function of distance from the center of the image.
% The second half of the
%program finds the area under the peak of the radial intensity plot. It
%does this by subtracting the area of the rectangle under the peak (the
%peak ends 3 pixels away from the center) using the mean of the data after
%the third pixel as the height of the rectangle.
%Written by Mitch Alton on 7/1/15
%adapted for no user input and mass throughput of images by MKK 9/27/2016
%
%m,n=size(a);
%
%creates matrix of repeating columns of 1 through how ever many columns
%there are in the image
x=[1:n];
l=repmat(x,n,1);

%creates matrix of repeating rows
y=[1:m];
y';
p=repmat(y,m,1);
w=p';
%makes matrixes -# to +# . Not sure how this will work with even numbers
%though
x=l-(((n-1)*0.5)+1);
y=w-(((n-1)*0.5)+1);
%creates a matrix of intensities from image
z=a(1:m,1:n);

%converts three matrixes to a three dimentional polar coordinate matrix
[THETA,RHO,Z]=cart2pol(x,y,z);

%turns matrix into a list of values
P=reshape(Z,1,[]);
H=reshape(RHO,1,[]);
his averages the intensities (Z) for single distance (RHO) values

```matlab
[plt,I,J] = unique(H);
s = zeros(size(plt));
frequencies = zeros(size(plt));
for i = 1:max(J)
    I = find(J==i);
s(i) = mean(P(I));
frequencies(i) = length(I);
end
radial_out = s.;
end

CELL MASK

function [cellperimeter,cellmask,bckcormsk,backgroundmask]=obtaincellmasks(I);

% PURPOSE: to create cell mask
% INPUT:
% OUTPUT: Cellperimeter,cellmask,bckcormsk, and backgroundmask.
% OUTPUT: an array that is very long (number of spots found in each frame)
% that has X, Y, time

I = imread(I);
fudgeFactor = 0.4;
[~, threshold] = edge(I, 'sobel');
BWs = edge(I,'sobel',threshold * fudgeFactor);
se90 = strel('line',3,90);
se0 = strel('line',3,0);
BWsdil = imdilate(BWs,[se90 se0]);
BWdfill = imfill(BWsdil,8,'holes');
BWnobord = imclearborder(BWdfill);
bckcormsk=~BWnobord(:,:);
cellmask = BWdfill-BWnobord;
backgroundmask(:,:, :) = ~cellmask(:,:, :);
seD = strel('diamond',1);
BWhfinal1 = imerode(cellmask,seD);
cellperimeter = bwperim(BWhfinal1);
Segout = I;
Segout(cellperimeter) = 255;
end
```