

8-1-2021

## Analysis of Microscopic Objects Using Computer Vision Methods

Yuan Dao

Follow this and additional works at: <https://digitalscholarship.unlv.edu/thesesdissertations>



Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

---

### Repository Citation

Dao, Yuan, "Analysis of Microscopic Objects Using Computer Vision Methods" (2021). *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 4238.

<http://dx.doi.org/10.34917/26341170>

This Thesis is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact [digitalscholarship@unlv.edu](mailto:digitalscholarship@unlv.edu).

ANALYSIS OF MICROSCOPIC OBJECTS USING COMPUTER VISION

METHODS

By

Yuan Dao

Bachelor of Science - Electronic and Information Engineering  
Beijing Jiaotong University  
2020

A thesis submitted in partial fulfillment  
of the requirements for the

Master of Science in Engineering - Electrical Engineering

Department of Electrical and Computer Engineering  
Howard R. Hughes College of Engineering  
The Graduate College

University of Nevada, Las Vegas  
August 2021



## Thesis Approval

The Graduate College  
The University of Nevada, Las Vegas

July 15, 2021

This thesis prepared by

Yuan Dao

entitled

Analysis of Microscopic Objects Using Computer Vision Methods

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Engineering - Electrical Engineering  
Department of Electrical and Computer Engineering

Mei Yang, Ph.D.  
*Examination Committee Chair*

Kathryn Hausbeck Korgan, Ph.D.  
*Vice Provost for Graduate Education &  
Dean of the Graduate College*

Shahram Latifi, Ph.D.  
*Examination Committee Member*

Grzegorz Chmaj, Ph.D.  
*Examination Committee Member*

Mo Weng, Ph.D.  
*Graduate College Faculty Representative*

## Abstract

As an essential and powerful tool to observe living organisms, three-dimensional fluorescence microscopy is widely used in biological research and diagnosis. The 4D fluorescence microscopy data can be obtained using time-lapsed videos of 3D images. To analyze and extract useful information from the increasingly large and complex biological image dataset, efficient and effective computational tools are in need but still lagging behind. In analyzing biological data, two major challenges are faced. First, time-lapsed fluorescence microscopic images typically have a low SNR. Second, biological objects often change their morphology and internal structure frequently. As such, conventional image processing methods may not be suitable for analyzing fluorescence microscopic images.

The dataset we used is the 4D microscopic images of developing drosophila (fruit fly) embryos, in which three types of proteins (Myosin, Ajuba and E-cad) play important roles. Located in the center part of the cells, Myosin contraction initiates the physical tension. E-cad is the core component of intercellular adhesion complexes located on cell membranes. Ajuba is the tensor sensor which is recruited by Myosin to adhere junctions.

In our previous work, we proposed a multi-object tracking method with portion matching to track those protein clusters. The goal of this research is to quantitatively analyze the correlation between these proteins based on the tracking results. In this thesis, the following tasks have been conducted.

- ◆ Sort out effective objects according to their fusion and split actions and visualize their development process clearly.
- ◆ Quantify the amount and intensity of another type of object in adjacent to the first type of object.
- ◆ Analyze the correlation between two types of microscopic objects.

## Table of Contents

Abstract.....	iii
List of Tables.....	vii
List of Figures.....	viii
Chapter 1: Introduction.....	1
1.1 Challenges.....	1
1.2 Motivations.....	3
1.3 Objectives.....	6
1.4 Outlines.....	7
Chapter 2: Literature Review.....	8
2.1 Filtering.....	8
2.2 Thresholding.....	9
Chapter 3: Sorting and Visualization of Microscopic Objects.....	11
3.1 Cycle Time.....	12
3.2 Fusion & Split.....	12
3.3 Create “Fusion Button” & Multiple Inspections.....	14
3.4 Drawing Family Tree.....	17
Chapter 4: Quantification.....	20
4.1 Removing Photobleaching.....	20
4.2 Masking Method.....	25
4.3 Normalization.....	28

4.4 Thresholding.....	31
4.5 Opening.....	32
Chapter 5: Correlation Analysis.....	34
5.1 Dataset.....	34
5.2 Five Methods.....	36
5.3 Method Comparison and Analysis.....	37
5.4 Correlation Results.....	39
Chapter 6: Conclusions.....	46
References.....	47
Curriculum Vitae.....	51

## List of Tables

Table 1 Fluorescent Spectral Colors versus Wavelength Band .....	3
Table 2 “Fusion Button” Pseudo Code .....	15
Table 3 “Test Splitting Process” Pseudo Code .....	16
Table 4 Five Methods .....	37
Table 5 Correlation Results .....	43
Table 6 Curriculum Vitae .....	51

## List of Figures

Figure 1 (a) Ajuba & (b) Myosin .....	4
Figure 2 Heatmap of Ajuba Size .....	4
Figure 3 Heatmap of Ajuba Average Intensity .....	5
Figure 4 Relationships (a) Birth, (b) Death, (c) Split, (d) Merge .....	11
Figure 5 Objects Existing Less than One Cycle Time will be Deleted .....	12
Figure 6 Fusion & Split Relationship of Ajuba ID 31 .....	13
Figure 7 Protein Fusion and Split Example .....	17
Figure 8 Deleting Short-time IDs .....	18
Figure 9 Final Family Tree of ID 31 .....	18
Figure 10 Gaussian Filter Using Different Sigma Values .....	23
Figure 11 Filters (a) Median Filter (b) Mean Filter .....	24
Figure 12 Compariosns (a) Original Image (b) Removing Photobleaching .....	25
Figure 13 Mask Dilation Examples - Red: Original, Green: Dilation .....	27
Figure 14 SMA Example - Window Size 15 subtract from Size 3 .....	30
Figure 15 Comparisons (a) No thresholding (b) Mean .....	31
Figure 16 Comparisons (a) No opening (b) Opening .....	33
Figure 17 Heat Map Example - Part of Ajuba Intensity .....	35
Figure 18 Quantification (a)Ajuba (b)Myosin .....	35
Figure 19 ID 2782 Quantification (Red - Ajuba, Green - Myosin) .....	38
Figure 20 Normalized Intensity Value (Blue - Ajuba, Red - Myosin) .....	38

Figure 21 First Method (NT_NE) .....	41
Figure 22 Second Method (T_NE) .....	41
Figure 23 Third Method (NT_E) .....	42
Figure 24 Fourth Method (T_E) .....	42
Figure 25 Fifth Method (T_E_O) .....	43
Figure 26 “Ecad & Ajuba” Correlation (NT_NE) .....	44
Figure 27 “Ecad & Myosin” Correlation (NT_NE) .....	44
Figure 28 “Ecad & Myosin” Correlation (NT_E) .....	45

## Chapter 1: Introduction

Multidimensional fluorescence microscopes are widely used in biological research and become an efficient tool to observe living organisms. The most commonly used data sets are 2D and 3D fluorescence microscopy images. The 4D dataset can be obtained from time-lapse video of 3D images. To analyze and extract useful information from the increasingly large and complex biological image dataset, efficient and effective computational tools are in need. However, existing image processing tools are not suitable for analyzing microscopic objects and their relations, such as tracking objects, object relation analysis and correlation analysis.

Open-source software tools such as Image J [1], Fiji [2] and CellProfiler [3],[4] can perform basic operations, including projection, histogram, filtering and so on. Generally, these software can process and analyze the entire image very well, but they lack the capability in analyzing specific objects and customized analysis. For example, it is not possible to quantify specific objects and do object tracking using these tools. In our research, we aim to track, quantify, and analyze the correlation between microscopic objects which can help researchers analyze biological data in-depth.

### 1.1 Challenges

In analyzing biological data, three major challenges are faced. The first challenge is time-lapsed fluorescence microscopic images typically have a low signal-to-noise ratio (SNR) [5]. The equipment physical size, imaging speeds and the number of

pixels per image are decided by the temporal resolution. At the same time, the temporal resolution depends on the time scale of the observed biological activities. The SNR is a key factor in determining the quality of microscope images. The SNR of the original image affects the image quality, a low SNR ratio will obscure the original perfect modulation contrast and cause errors in the segmentation process. In biological samples, the SNR is often low due to bleaching, and it further drops in the process of acquiring time series [6],[7]. Photobleaching is a process that the fluorophore loses its ability to fluoresce. It occurs when fluorescent molecules accumulate chemically damaged electrons to excite fluorescence. Because fluorescent proteins used for real-time imaging are particularly sensitive to photobleaching, long-term imaging results in a severe reduction in fluorescence signal, resulting in a low SNR ratio.

The second challenge is that biological objects often change their morphology and internal structure frequently. Research on tracking living cells has been studied for years, however, it is still a challenge to track subcellular structures, specific targets such as proteins, or lipids. Unlike macro objects (e.g. cars), those protein complexes always have continuous morphological changes, including split and merge. Because cells may perform deformations such as expansion, shrinkage [8] and irregular. Highly dynamic morphology already became the biggest challenge to analyze those small size biological structures.

The third challenge is fluorescent molecules can only absorb light within a limited wavelength range determined by the nature and range of delocalized electrons,

ranging from about 340 to 700 nanometers. Each fluorescent pigment or fluorophore has its own unique absorption and emission spectra (Citation of Table 1), which depends on the internal structure of the fluorescent molecules, as well as the environment in which they are located [9]. In practice, the spectral cross-talk can be significant when using more than 3 fluorophores, which consequently limits the types of objects that can be observed and studied simultaneously.

<b>Wavelength Range (nm)</b>	<b>Perceived Color</b>
340-400	Ultraviolet
400-430	Violet
430-465	Blue
465-500	Cyan
500-550	Green
550-580	Yellow
580-620	Orange
620-700	Red
Over 700	Near-Infrared

Table 1 Fluorescent Spectral Colors versus Wavelength Band

## 1.2 Motivations

Unfortunately, conventional image processing methods may not be suitable for analyzing fluorescence microscopic images. The dataset we used is the 4D microscopic images of developing drosophila (fruit fly) embryos, in which three types of proteins (Myosin, Ajuba and E-cad) play important roles. Located in the center part of the cells, Myosin contraction initiates the physical tension. E-cad is the core component of intercellular adhesion complexes located on cell membranes. Ajuba is

the tensor sensor which is recruited by Myosin to adhere junctions. It is important to quantify the correlation among those three kinds of proteins.

Due to the third challenge explained in Section 1.1, we can only obtain three kinds of images: “Ajuba& Myosin”, “Ajuba & E-cad” and “E-cad & Myosin”.

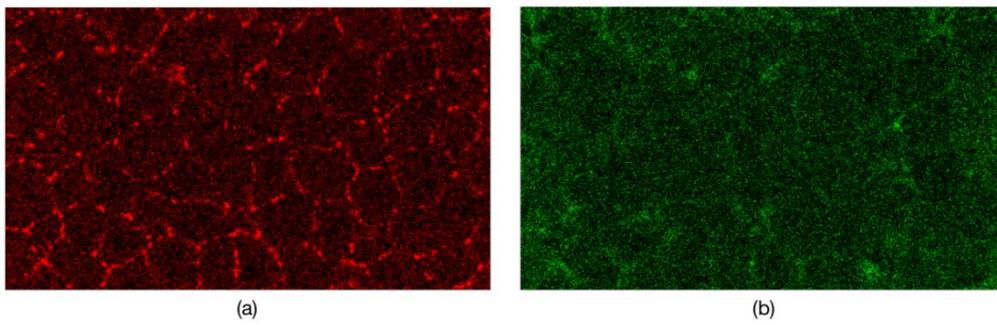


Figure 1 (a) Ajuba & (b) Myosin

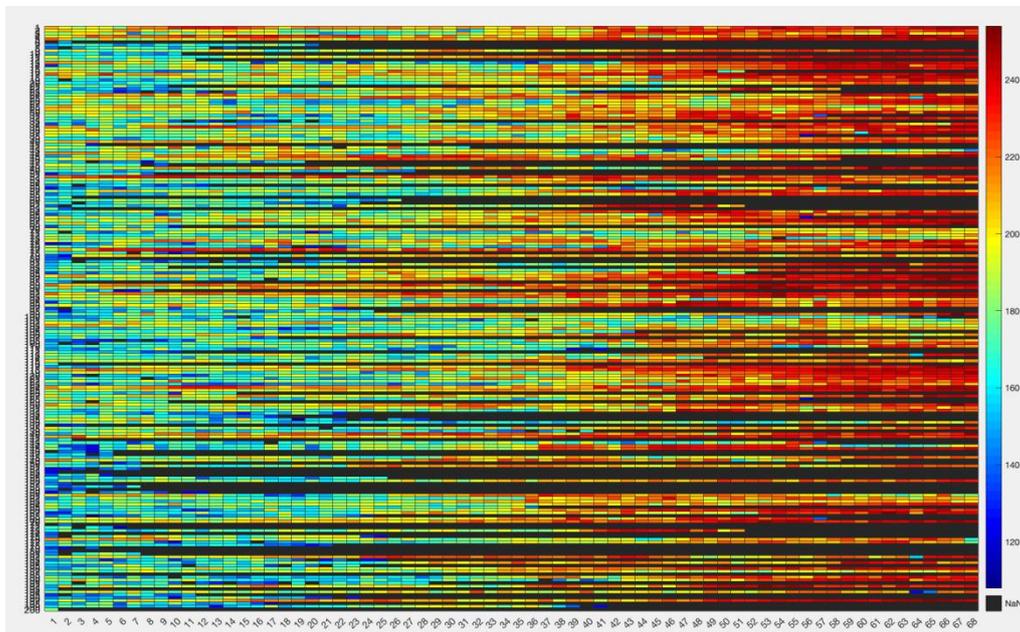


Figure 2 Heatmap of Ajuba Size

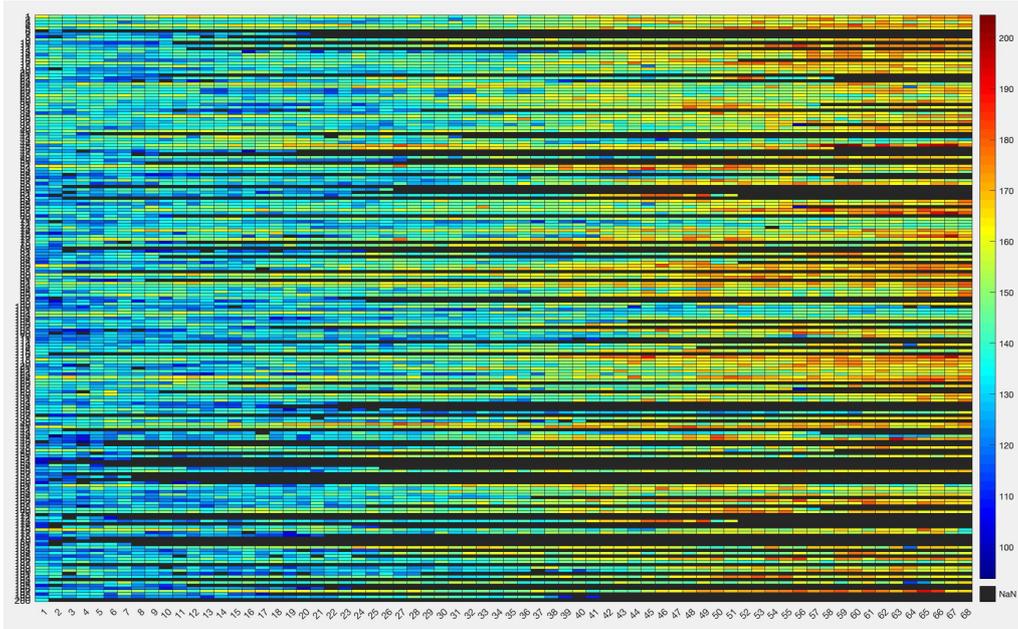


Figure 3 Heatmap of Ajuba Average Intensity

Ajuba and E-cad can be segmented or tracked, but not Myosin due to its dissolved morphology. In the previous work, we proposed a multi-object tracking method with portion matching to track Ajuba/E-cad clusters. The object information of one type of protein including size, intensity and location are available, as shown in Figure 2 & Figure 3.

In those two figures, two types of problems exist. 1) There are a large amount of objects with various shape, size and pixel intensity. 2) The relationship between objects is not shown. Before we analyze the correlation between two kinds of proteins, we need to sort out major objects of the first type of protein and derive their evolution processes. Then mask out the second type of protein, and quantify it.

### 1.3 Objectives

The goal of this thesis is to analyze the correlation between biological proteins quantitatively based on the tracking results. In specific, we need to sort the tracked Ajuba/E-cad clusters into different groups according to their development relation through fusion and split process, then build the family tree (a tree containing the following five generations of object information) of each specified object and visualize their features (including size, intensity, etc.) clearly. We also aim to design a common method to analyze correlation between biological objects, and overcome existing difficulties including highly dynamic and irregular morphology.

- Our first task is to visualize the microscopic objects that have been tracked. Out of the tracked objects (e.g. E-cad), some objects are either small or weak in intensity. We will sort out the effective objects according to their fusion and split actions and visualize their development process clearly.
- Our second task is to quantify the amount of another type of object in adjacent to the first type of object. In specific, for each E-cad cluster, we need quantify the amount and intensity of Myosin next to it in a 3D adjacency box. The same method will be applied to quantify Myosin signals next to Ajuba clusters and Ajuba signals next to E-cad clusters.
- Our third task is to analyze the correlation between two types of microscopic objects. Specifically we will analyze the correlation between Myosin and Ajuba, Myosin and E-cad, as well as Ajuba and E-cad in terms of their amount and intensity.

## 1.4 Outlines

The rest of thesis is organized as follows:

Chapter 2 presents the literature review. Our review is focused on existing filtering and thresholding methods.

Chapter 3 describes the methods for sorting and visualizing the microscopic objects.

Chapter 4 presents the method for quantifying another type of object adjacent to one type of object.

Chapter 5 discusses the results of correlation analysis of different proteins.

Chapter 6 presents the summary and conclusion of this thesis. In addition, we will list the future work.

## Chapter 2: Literature Review

In this thesis, we mainly study the methods of filtering and thresholding. These two processes play important roles in our data analysis. The main function of filtering is to remove the effects of photobleaching by normalizing the pixel value in the 4D image data, which can help reduce the error in calculations. Thresholding can help distinguish effective objects (foreground) from background in the picture. There are already many types of 2D thresholding methods, but we want to find the right one for processing our multi-dimensional image data. In this chapter, we will list the existing filtering and thresholding methods and briefly describe their functions and effects.

### 2.1 Filtering

There are many common filtering methods in image processing, such as average filter, Gaussian Blur, median filter, and so on. Those filtering methods are widely used in graphics software, usually used to reduce image noise and details. Both the average filter and Gaussian blur are all smoothing linear filters. The average filter, also known as the mean filter, takes the average of pixels in a neighborhood, which is defined by the size of the mask. The important thing is to normalize the output value by dividing by the sum of the values in the neighborhood. Gaussian blur [10], also known as Gaussian smoothing, uses convolution and Gaussian functions for smoothing [11]. Median filter is a smoothing non-linear filter, and it is perfect for reducing impulse noise (salt-and-pepper noise) [12]. The main idea of the median filter is to traverse the signal entries one by one, replacing each entry with the median

value of adjacent entries. For small to medium-level Gaussian noise, the median filter is significantly better than Gaussian blur in removing noise while retaining the edges of a given fixed window size; For high-level noise, the median filter performance is not much better than Gaussian blur [13]. Different filter methods have different advantages, which need to be analyzed according to the image data.

## 2.2 Thresholding

Thresholding is another critical procedure in analyzing fluorescence microscopy images. One of the simplest thresholding methods is to convert the gray-scale image to binary [14]. If the image intensity is less than a certain fixed constant, each pixel in the image is replaced with a black pixel, other pixels in the image are replaced with a white pixel. This constant value is actually the threshold value we need to find. The histogram shaped-based method is commonly used to obtain the threshold value, it makes certain assumptions about the intensity probability distribution of the image which has bimodal distribution. In [15], Prinyakupt *et al.* used morphological processing and histogram to process biological cells. The histogram filter is an approximate Bayes' filter [16] based on the discretization of the state space. The main concept of the algorithm shows in the article is to use the morphological properties of biological cells and the size of real cells calibrated relative to the image resolution [17].

Another efficient thresholding method is Otsu [18], this algorithm exhaustively finds the threshold that minimizes the variance. However, Otsu is not suitable for

thresholding image whose object area is smaller than the background area. If the variance of the target and background intensity is greater than the mean difference, the sharp valley of the gray scale histogram will be degraded, it will eventually lead to wrong threshold result [19]. In [20], Jian Zhuang *et al.* compared the intensity value of a given pixel with the average intensity of its neighboring areas to improve the segmentation results. Khan *et al.* developed an improved method instead of estimating object size and potential thresholds. Their method solves the problems of thresholding large objects and inconsistent solutions [21]. Ma *et al.* improved Otsu's method and drew a clear boundary between the categories of gray values [22].

According to the above literature reviews, the selection of the thresholding method is mainly based on the distribution of histogram, noise interference and the intensity of the target object. In this thesis, we will choose several different methods to compare.

### Chapter 3: Sorting and Visualization of Microscopic Objects

The focus of this chapter is how to sort different biological object clusters and create corresponding family tree. As we mentioned earlier, we proposed a multi-object tracking method with portion matching to track protein clusters in our previous work. We recorded biology object events of protein clusters, including 1) **Birth**: a newborn object (New ID) which is not matched with any former time objects, 2) **Death**: a dead object which ID will never be assigned to other objects, 3) **Split**: at least 2 objects split from one parent object if the intersection of unions of the matching objects set is not empty, and 4) **Merge**: an object is merged with multiple elements, shown in Figure 4.

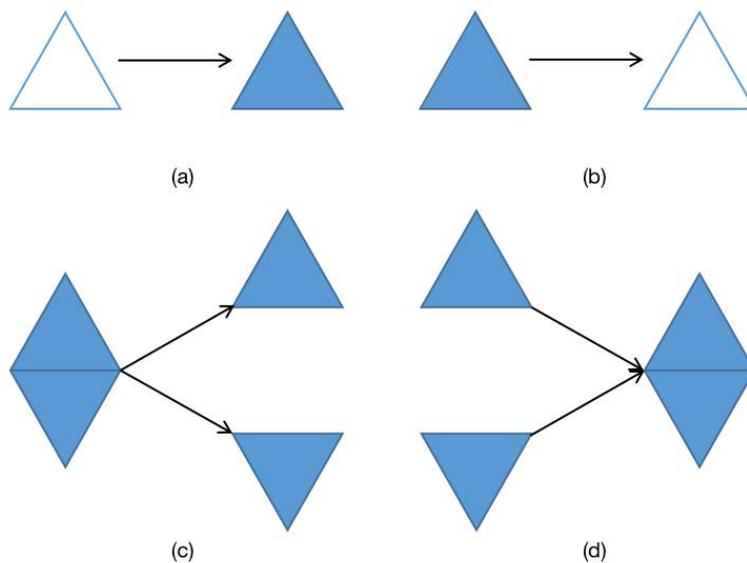


Figure 4 Relationships (a) Birth, (b) Death, (c) Split, (d) Merge

The establishment of a family tree is mainly divided into 4 small parts: 1)

confirm the cycle time, 2) find the relevant fusion and split process, 3) remove unnecessary or unreasonable processes, 4) draw the family tree.

### 3.1 Cycle Time

The first step of our sorting algorithm is setting an initial value of one cycle time. For example, there are about 5000 different IDs in one dataset, among those objects, some tracked protein objects only appear at one or two time points, we assume that those IDs are interfering signals with no practical meaning. After setting the cycle time of one entire biological action (e.g. 5 time points), every objects which appears less than the one cycle time will be ignored, shown in Figure 5.

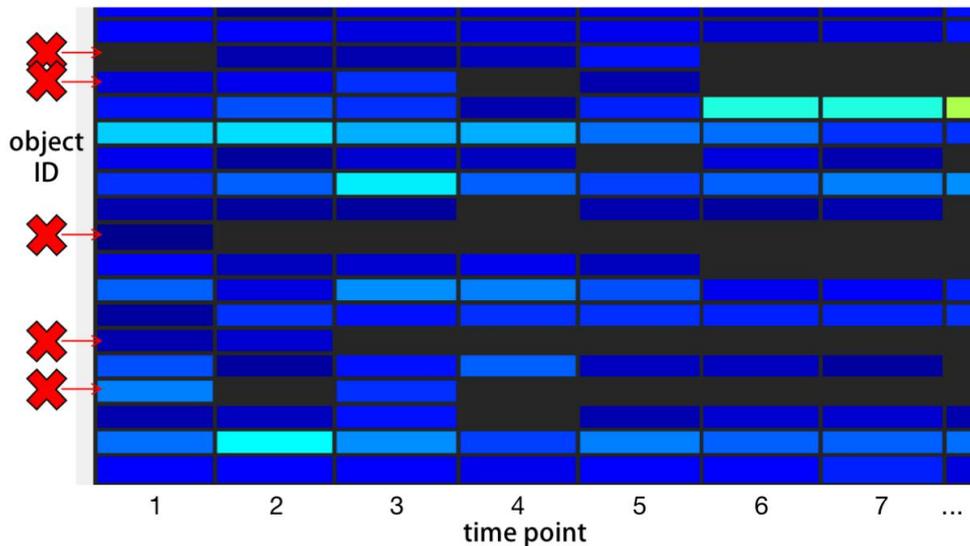


Figure 5 Objects Existing Less than One Cycle Time will be Deleted

### 3.2 Fusion & Split

The second step is based on biological fusion and split relationship. Objects

retained in the previous step will be reorganized. We will search for all proteins directly or indirectly related to the target object ID based on the fusion changes. After this operation, for each target object ID we will obtain the fusion information set  $V = \{V_1, V_2, \dots, V_n\}$ ,  $n$  is the number of fusion events related to the target object.  $V_i(T_v, A, B)$  is a triple element structure, where  $T_v$  is the time point of fusion,  $A$  is the object to be fused with other object and  $B$  is the merged and retained object (which absorbs another object).

Later, we will search for the split relationship of each target ID and obtain the split information set  $S = \{S_1, S_2, \dots, S_n\}$ ,  $n$  is the number of split events related to target ID.  $S_i(T_s, C, D)$  is a triple element structure, where  $T_s$  is the time point of split,  $C$  is the object to be split and  $D$  is the split object.

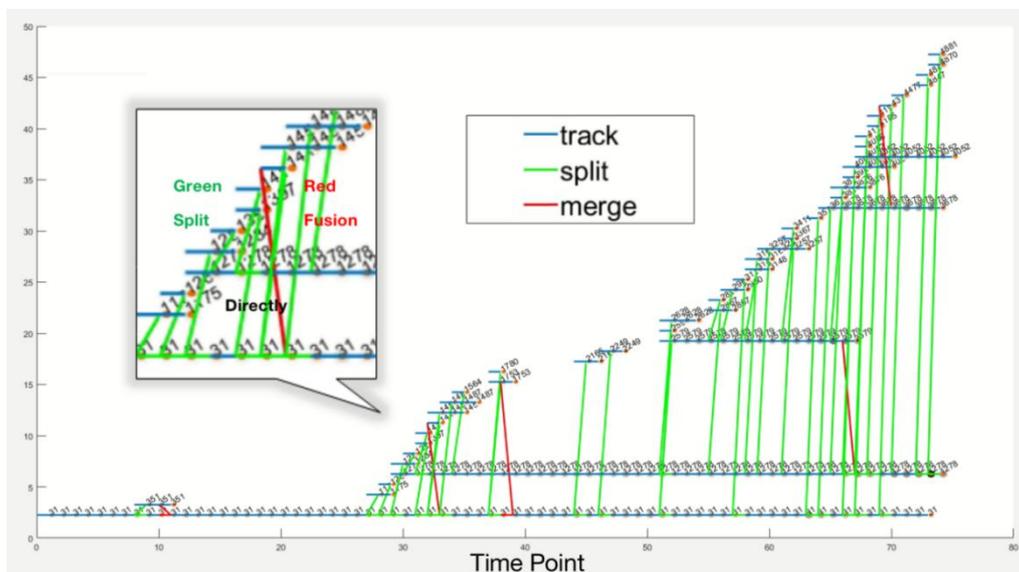


Figure 6 Fusion & Split Relationship of Ajuba ID 31

Basically after this operation, we got a rough statistical family tree, shown in

Figure 6. However, this generated family tree may have some logical and other unreasonable problems, such as 1) some fusion processes appeared earlier than the target ID; 2) the protein merged with the target ID has a split or fusion process with other proteins before the fusion relationship occurs. 3) complex multi-generational relationships may appear in the family tree, and it is necessary to carefully analyze whether these logical relationships are reasonable.

### **3.3 Create “Fusion Button” & Multiple Inspections**

The main reason for dividing proteins into family trees is to sort the effective objects accurately. That means we must confirm that this process of splitting or merging is truly related to our target ID. In this section, we propose the “Fusion Button” algorithm to solve the problem in 3.2. This fusion button can keep objects for two generations before and after the target ID, in specific, including 1) sons, 2) parents, 3) grandsons, and 4) grandparents. As a switch, the fusion button can be freely turned on or off. When we want to keep the subject part of the family tree, we can activate the fusion button, which can keep all the information of the five generations including the target ID (pseudo code shows in Table 2).

**Algorithm 1: Fusion Button Algorithm****Input:** Target object ID  $I$ ,  $P$ 's fusion information set  $V(T_v, A, B)$ **Output:** Obtain a final partition set  $Z$ 

```
1.   define  $i, j, C, P, n = 0$  ;
2.   for  $i = 1$  to  $|V|$  do
3.       find  $A$  and  $B$  from  $V_i$  ;
        /* trace fusions not directly related to target ID  $I$  */
4.       if  $(A \neq I) \ \&\& \ (B \neq I)$  then
5.            $C \leftarrow A$  ;
6.           for  $j = 1$  to  $i$  do
7.               find  $B$  from  $V_j$  ;
                /* trace fusions not related to target ID  $I$  */
8.               if  $(B == C)$  then
9.                   break ;
10.            end
11.           end
12.           if  $(j == i)$  then
13.                $Pn \leftarrow i$  ;
14.                $n++$  ;
15.           end
16.       end
17.   end
18.   for  $i = 1$  to  $n$  do
19.        $V_{P_i} \leftarrow \emptyset$  ;
20.   end
21.    $Z \leftarrow V$  ;
```

Table 2 “Fusion Button” Pseudo Code

In this loop, we have to make sure it starts from the first fusion case. We record fusion information in chronological order, so objects that meet the requirements before can be used as the basis for judging objects afterwards. The algorithm searches from the target ID as the starting point, eliminating the impact of multiple generations and multiple splits and fusions.

It should be noted that fusion button only excludes the impact of complex relationships, the problem mentioned earlier of how to test whether the split operation

is reasonable or not has not been solved. The solution we propose is to judge the time point of the relationship, and only keep the split occurred after the fusion (Pseudo code shows in Table 3).

<b>Algorithm 2:</b> Test Splitting Progress Algorithm	
<b>Input:</b>	Target object ID $I$ ; $I$ 's fusion information set $\mathcal{V}(T_V, A, B)$ ; ID $I$ 's split information set $\mathcal{S}(T_S, C, D)$
<b>Output:</b>	Obtain a final partition set $Z$
1.	define $i, j, k, F, P, n = 0$ ;
2.	<b>for</b> $i = 1$ to $ \mathcal{V} $ <b>do</b>
3.	find $B$ from $V_i$ ;
4.	<b>for</b> $j = 1$ to $ \mathcal{S} $ <b>do</b>
5.	find $C$ and $D$ from $S_j$ ;
	/* find merged object B is also split object C */
6.	<b>if</b> $(B == C)$ <b>then</b>
7.	find $T_V$ from $V_i$ and $T_S$ from $S_j$ ;
	/* find fusion case happened after split case */
8.	<b>if</b> $(T_V > T_S)$ <b>then</b>
9.	$F \leftarrow D$ ;
10.	<b>for</b> $k = i$ to $ \mathcal{V} $ <b>do</b>
11.	find $A$ from $V_k$ ;
	/* trace fusions happened not earlier than $T_V$ */
12.	<b>if</b> $(A == F)$ <b>then</b>
13.	<b>break</b> ;
14.	<b>end</b>
15.	<b>if</b> $(k ==  \mathcal{V} )$ <b>then</b>
16.	$Pn \leftarrow j$ ;
17.	$n++$ ;
18.	<b>end</b>
19.	<b>end</b>
20.	<b>end</b>
21.	<b>end</b>
22.	<b>end</b>
23.	<b>end</b>
22.	<b>for</b> $j = 1$ to $n$ <b>do</b>
23.	$S_{p_j} \leftarrow \emptyset$ ;
24.	<b>end</b>
24.	$Z \leftarrow S$ ;

Table 3 “Test Splitting Process” Pseudo Code

For example, in Figure 7, assume that  $ID_a$  is our target ID,  $ID_a$  and  $ID_b$  merge together at time point  $T_1$ .  $ID_b$  splits to  $ID_c$  at time point  $T_0$  and splits to  $ID_d$  at time point  $T_2$  ( $T_0 < T_1 < T_2$ ). In this case, we only keep the relationship among  $ID_a$ ,  $ID_b$  and  $ID_d$ .  $ID_c$  has definite relationship with  $ID_b$ , but it doesn't have any direct or indirect relationship with  $ID_a$ . The reason is at time point  $T_1$ ,  $ID_c$  is already split from  $ID_b$ , which means that it does not belong to the  $ID_a$ 's cluster at all. If  $ID_c$  is merged back to  $ID_b$  at time point  $T_3$  ( $T_1 < T_3$ ), in this case  $ID_c$  has indirect relationship with  $ID_a$ . We also need to be aware of this kind of case where objects split off and then merger back together.

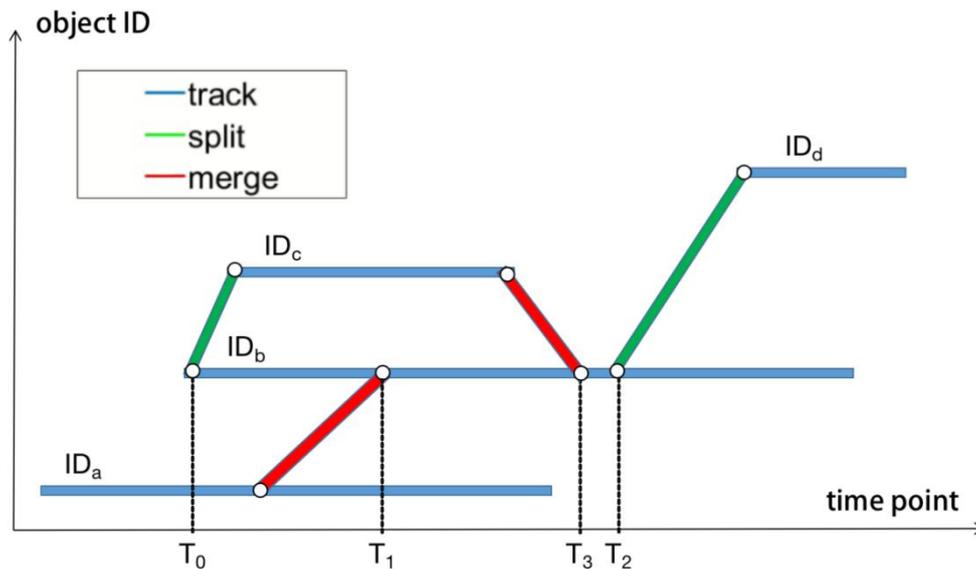


Figure 7 Protein Fusion and Split Example

### 3.4 Drawing Family Tree

Our final step is drawing the family tree. We filter the sorted family tree of Section 3.3 one last time, and retain IDs exist for more than one period (this operation

deletes the single ID that appears for less than 5 time points, which is same as Section 3.1 operation).

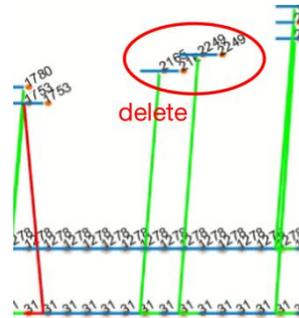


Figure 8 Deleting Short-time IDs

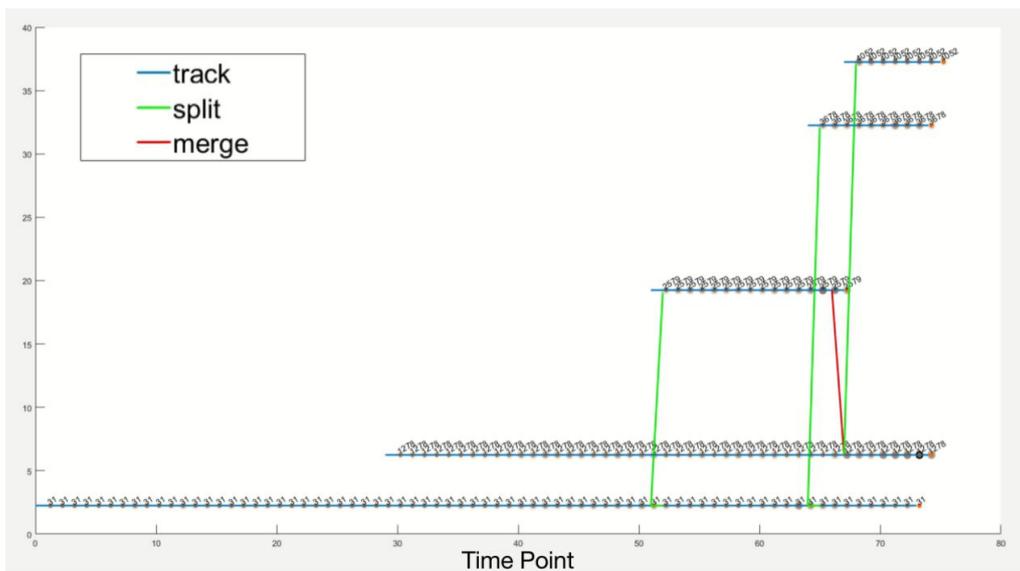


Figure 9 Final Family Tree of ID 31

Figure 8 shows some of the IDs that are removed. It is important to note that we are only deleting IDs that are no longer split or merged. We use blue lines to draw their parent-son tracking relationships, use red lines to draw their fusion relationships, and use green lines to draw their split relationships. We sort out the effective objects

according to their fusion and split actions and visualize their development process clearly in a family tree. Figure 9 is one example of our final results.

## Chapter 4: Quantification

Quantification is divided into five parts: removing photobleaching, masking method, data size normalization, thresholding and opening. Removing photobleaching can reduce the noise in the image and improve the SNR ratio. We will select suitable filters to process the image data. In the masking method part, we will talk about how to use thresholding method and quantify the amount of another type of object in adjacent to the first type of object. In this part, we mainly deal with objects and pictures that are not easy to track directly. The normalization of the data size is mainly for the convenience of comparing data in different processing conditions, so we can obtain relatively accurate information for correlation analysis. Thresholding method can theoretically reduce the error of noise. We will find the suitable one and compare it with the result without thresholding. Performing an opening operation on the picture after thresholding can remove more noise. In these five steps, we must do the step of data size normalization. The image filtering, thresholding and opening process may not be necessary.

### 4.1 Removing Photobleaching

Under ideal conditions, there is no photobleaching in fluorescence images, but in practice we need to consider its impact. This step is mainly to quantitatively analyze the impact of photobleaching on the final correlation result under the control of variables, and help us choose the most reasonable analysis method. The most straightforward way for removing photobleaching is to use filters. In this thesis, we

choose three different image filters, including Gaussian blur, mean filter and median filter. The major purpose of using filters is not to remove the noise in the picture, but to get the average intensity of the background of the picture by blurring the foreground (main object) of the picture. The Gaussian blur best fits our needs. Gaussian blur (also known as Gaussian smoothing) is a type of image blurring filter using Gaussian function [10]. Mathematically, using Gaussian blur to an image is the same as convolving an image with a Gaussian function. Since the Fourier transform of a Gaussian function is another Gaussian function, the use of Gaussian blur can reduce the high-frequency components of the image, which means Gaussian filter is a low pass filter [23].

The formula for the one-dimensional Gaussian function is: (“ $x$ ” is the distance from the origin on the  $x$ -axis, “ $\sigma$ ” is the standard deviation of the Gaussian distribution)

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Equation 1

The fluorescence images we use are 4-dimensional. We apply Gaussian blur at different time points, and we need to make a 3D Gaussian filter. In three dimensions, it is the product of three such Gaussian functions, one in each dimension.

The formula for the three-dimensional Gaussian function is: (“ $y$ ” is the distance

from the origin on the  $y$ -axis, “ $z$ ” is the distance from the origin on the  $z$ -axis)

$$G(x, y, z) = \frac{1}{\sqrt[3]{2\pi\sigma^2}} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}}$$

Equation 2

Distribution values are used to build a convolution matrix, which is applied to the original image. The new value of each pixel is set as the weighted average of the neighborhood of that pixel. The weight of the original pixel value is the largest, and the weight of the adjacent pixel decreases as the distance from the original pixel increases [10]. When Gaussian Blur is used for three-dimensional images, it is calculated as three independent one-dimensional calculations. In other words, applying a series of one-dimensional Gaussian matrices in the  $x$  direction and then repeating this process in the  $y$  and  $z$  directions can also achieve the effect of applying a three-dimensional matrix.

We need adjust the value of “ $\sigma$ ”- sigma. A higher sigma value will blur on a larger radius, which will use a larger kernel matrix to capture enough function energy. We want the kernel function to be large enough to cover most of the objects, but not so large that it starts to overlap multiple adjacent objects. As shown in Figure 10, if the sigma value is too small ( $\sigma = 0.5$  or  $1$ ), the foreground blur degree is not enough, and a large amount of foreground object information is still retained; if the sigma value is too large ( $\sigma = 10$ ), the foreground blur degree caused by overlapping objects

is too large, and the whole picture becomes almost the same color. Through the above comparison and analysis, we believe that sigma value equal to 5 is the best choice. Gaussian blur under this condition has the best effect on blurring the foreground of the Myosin image used in this thesis.

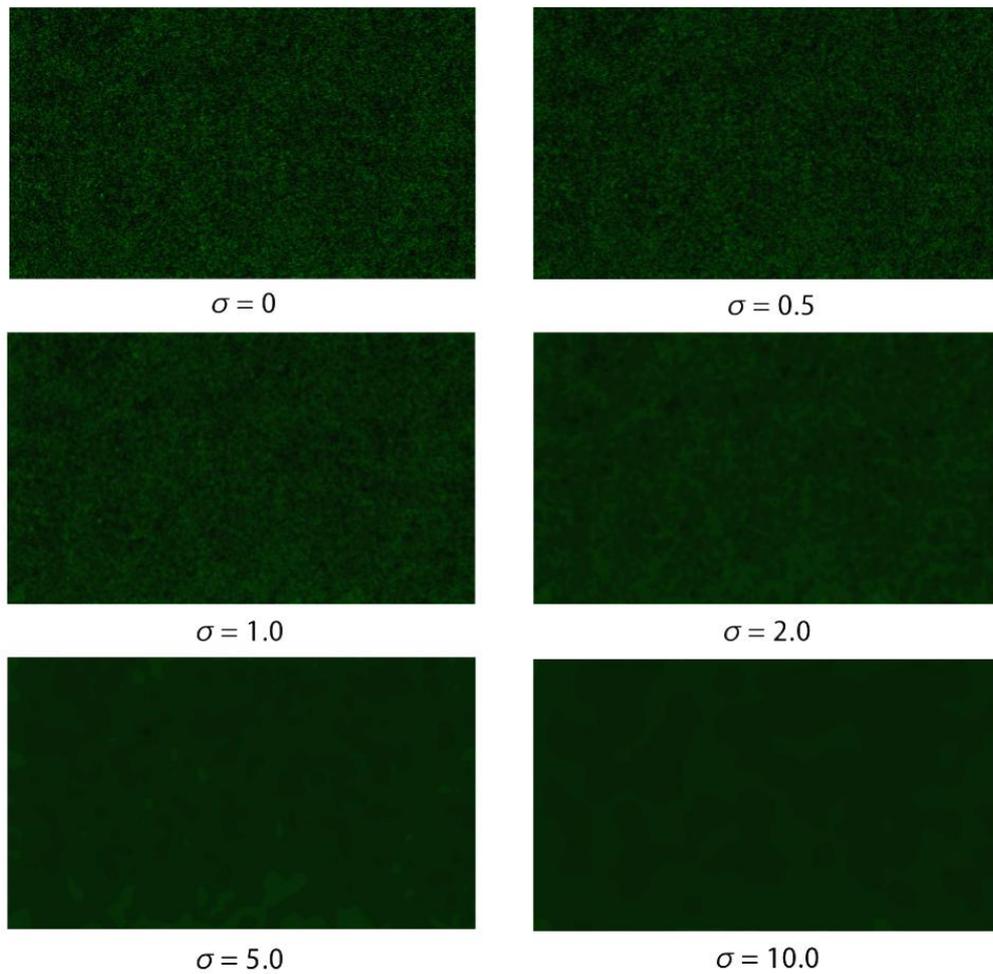


Figure 10 Gaussian Filter Using Different Sigma Values - Myosin,  $z=2$ ,  $t=20$

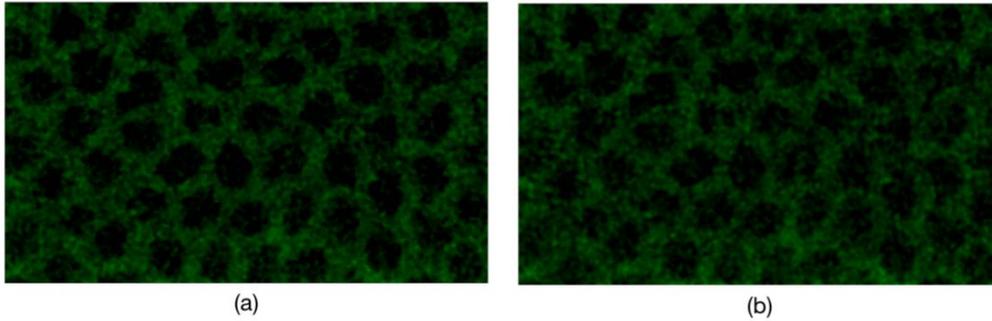


Figure 11 Filters (a) Median Filter (b) Mean Filter - Myosin,  $z=2$ ,  $t=20$

We also test both median filter and mean filter. The “window” size we selected is  $3 \times 3 \times 3$ . It can be seen from Figure 11 that both filters can filter noise by varying degrees, and the median filter has a better effect on impulse noise. But these two filters are not suitable for our purpose of removing photobleaching, which is mainly to remove the background color, rather than directly remove noises from the image, because some of “noises” are actually the parts of Myosin.

After the filtering method and the coefficient value are determined, we can calculate the average intensity through the filtered image. One thing to note is that we have to calculate the data of the 3D picture according to the time points. According to the previous results, photobleaching dynamics is fast at the beginning and slows down later. The Gaussian filter method chosen is simple and easy to do. Instead of looking for the constant area, the entire image is processed. The disadvantage of this method is that it will include non-constant areas, and the result will be unreasonable.

The biological changes at the earlier time points are less. At a later time point, the biological change begins, so the value of using Gaussian blurring to blur the foreground while preserving the background becomes unreasonable. In our thesis, we

use the value at the 20th time point to replace the value after the 20th time point, so as to avoid including non-constant areas. Figure 12 (b) & (d) are final results of removing photobleaching by subtracting background average value.

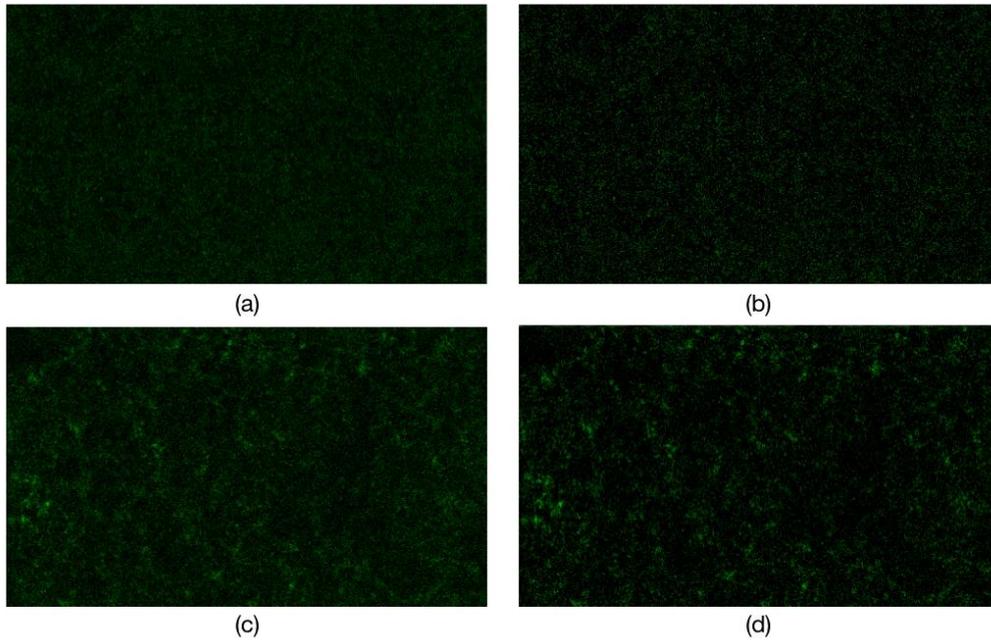


Figure 12 Comparisons (a) Original Image (b) Removing Photobleaching - Myosin,  $z=2, t=20$  (c)Original Image (d)Removing Photobleaching - Myosin,  $z=2, t=70$

## 4.2 Masking Method

In this section, we will show how to use Object  $A(O_A)$  to quantify Object  $B(O_B)$ . The data of  $O_A$  and  $O_B$  are stored in two different pictures. Among them,  $O_A$  can be tracked well because it has a strong intensity, which can be well distinguished from the background in the picture. In Chapter 3, the sorting method we proposed can help us calculate the intensity value of  $O_A$  easily. However, it is not easy to segment or track  $O_B$  directly because of its detection limit.

Here we have to divide into two different cases.

- Case 1: If the positions and sizes of the two objects are same then we can directly use  $O_A$ 's location and shape information to mask out  $O_B$ .
- Case 2: If the positions of two objects are relatively close but do not overlap, which means we cannot directly use the size and position of  $O_A$  to quantify  $O_B$ . We need to use a reasonably sized mask to do the quantification.

For Case 2, specifically, we want to expand the size of  $O_A$ , and use this expansion mask as  $O_B$ 's mask. Firstly, we need to extract the position and size information of  $O_A$  according to the time point. Then convert the original 16-bit image into binary image to obtain the original mask. Next we need to use binary dilation to derive the expansion mask. Binary dilation is a shift-invariant operator, which is equivalent to Minkowski addition [24]. If we explain dilation mathematically, let  $X$  and  $Y$  be subsets in 2-D space, where  $X$  is image undergoing analysis,  $Y$  is the structuring element.

The dilation of  $X$  by  $Y$  is defined by: ( $X_y$  is the translation of  $X$  by  $y$ .)

$$X \oplus Y = \bigcup_{y \in Y} X_y$$

Equation 3

If  $Y$  has a center at the origin, then  $Y$ 's expansion of  $X$  can be understood as the trajectory of the point covered by  $Y$  when the center of  $Y$  moves inside  $X$ . The dilation

can also be obtained by

$$X \oplus Y = \{ z \mid z = x + y \text{ for some } x \in X, y \in Y \}$$

Equation 4

For example, if  $X$  is a disk with a radius of 3 (centered at the origin), and  $Y$  is a square of size 10 (centered at the origin). The expansion result will be a square of side 16 (centered at the origin) with rounded corners (radius of round corners is 2).

The structuring element we use is a  $10 \times 10 \times 6$  cuboid in order to get a mask by extending 10 pixels (5 pixels - left/up and 5 pixels - right/down) on the  $x$  and  $y$  axis, respectively, and 6 pixels on the  $z$  axis (3 pixels - above and 3 pixels - below) , as shown in Figure 13.

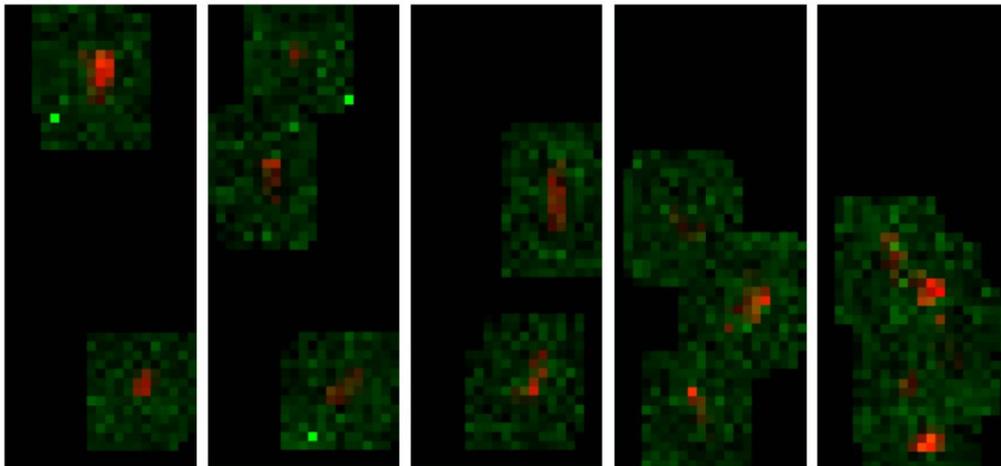


Figure 13 Mask Dilation Examples - Red: Original, Green: Dilation

This bounding box satisfies our quantify requirements. We expanded the mask

while still retaining the shape of the original mask. We also tried to use a cuboid that can frame the original object as the bounding box, but because the shape of the object is usually irregular, blindly choosing a constant shape will cause errors, such as fetching too many uncorrelated pixels at the corners, which leads to increased intensity and ultimately affect data analysis.

### **4.3 Normalization**

We have to consider that each fluorescence image may use different biological stains, which also means that the final calculated intensity will be different due to the difference in stains. This is something we can't avoid. So it's meaningless to compare the final intensity directly. Therefore, we need to find a way to normalize the final result, which can not only preserve the peak, valley and trend of the wave, but also compare the data in a relatively fair situation.

In statistics, a moving average (also called rolling average) is a calculation that analyzes data points by creating a series of averages of different subsets in a complete data set [25]. It is a finite impulse response filter. Moving averages are usually used for time series data to smooth short-term fluctuations and highlight long-term trends or cycles [26].

After consulting relevant information, there are two commonly used moving average methods in the financial industry: simple moving average (SMA) and exponential moving average (EMA). The main difference between EMA and SMA is their sensitivity to changes in the data used in the calculations. More specifically,

EMA assigns higher weight to recent prices, while SMA assigns equal weight to all values. Because the content of the biological data we use is different from financial data, this means we don't need to assign higher weight to recent value. We only need to put the biological (protein) data in different pictures under the same situation for comparative analysis. Here we choose SMA as our moving average method.

SMA adds the most recent data and divides the sum by the number of time periods to calculate the average. If we use mathematical language to explain, suppose there is a data set of  $n$  items, we want to find the mean value of last  $k$  items, denoted as  $SMA_k$ :

$$SMA_k = \frac{P_{n-k+1} + P_{n-k+2} + \dots + P_n}{k} = \frac{1}{k} \sum_{i=n-k+1}^n P_i$$

Equation 5

When calculating the new  $SMA_k$  value, denoted as  $SMA_{next}$  :

$$SMA_{next} = \frac{1}{k} \sum_{i=n-k+2}^{n+1} P_i = SMA_{prev} + \frac{1}{k} (P_{n+1} - P_{n-k+1})$$

Equation 6

The algorithm of SMA is clear and easy to use. The size of the selection window should be changed according to the data used. In this experiment, we used window sizes of 3 and 15. Specifically, we subtract the size 15 data curve from the size 3 data

curve, shown in Figure 14.

Using a big window size to process data will ignore some of data. Here we use data processed with a big window size as the new coordinate axis to observe the trend of data processed with a small window size, which can well preserve the peaks and valleys. The value of window size is selected according to the length of the data. The length of the data we observe in this example is 45 time points. 3 is a small value relative to 45, and 15 is a large value relative to 45.

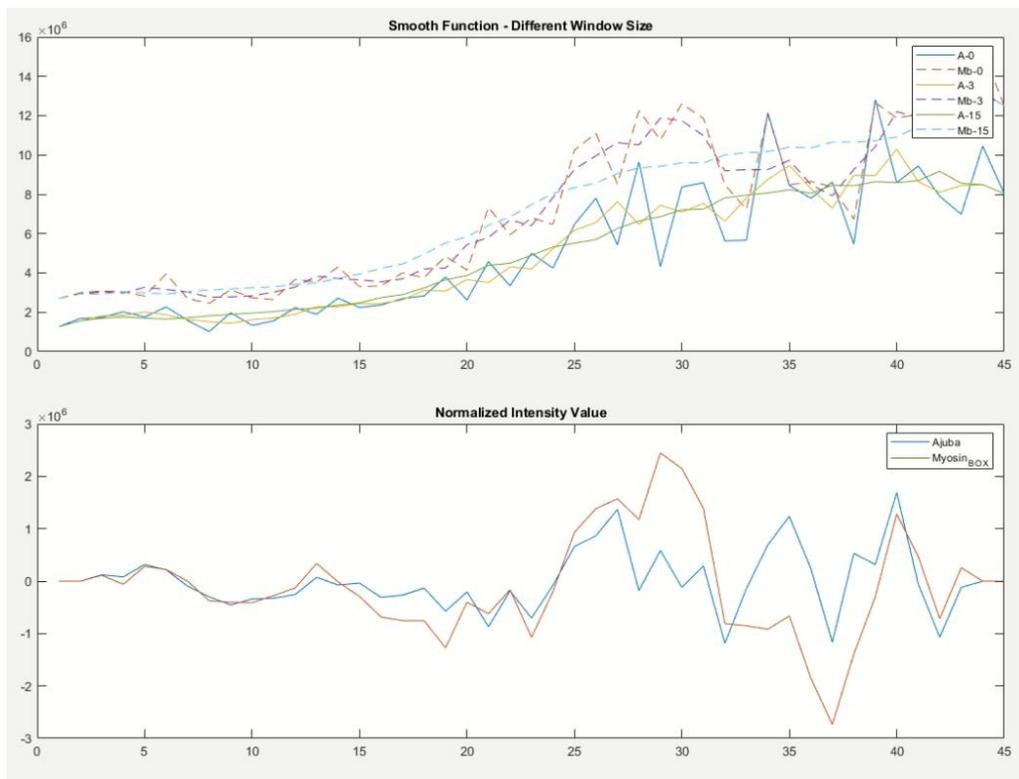


Figure 14 SMA Example - Window Size 15 subtract from Window Size 3

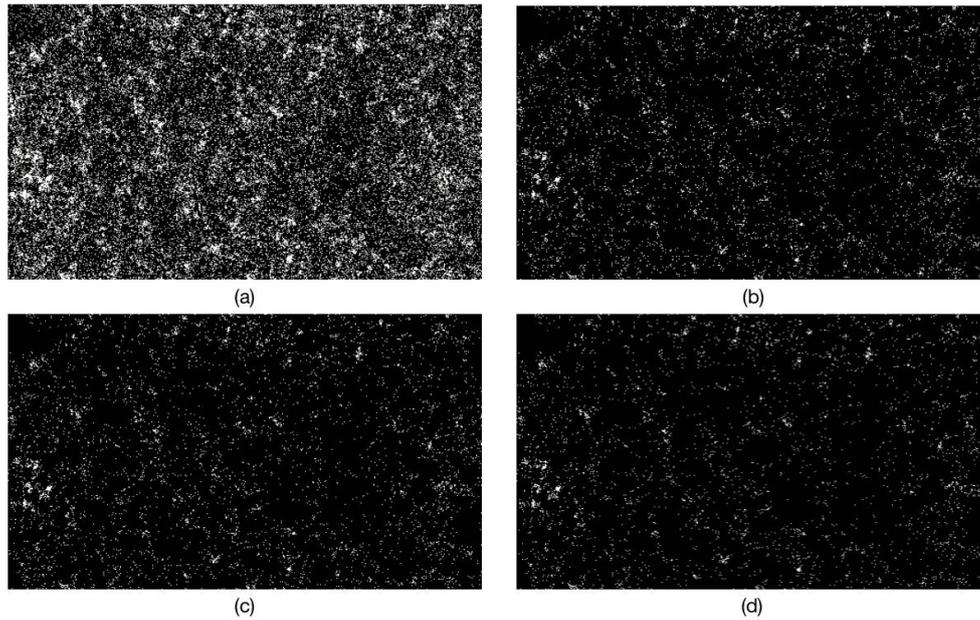


Figure 15 Comparisons (a) No thresholding (b) Mean  
(c) Mean + Standard Deviation (d) Otsu

#### 4.4 Thresholding

Before doing the thresholding step, we need to use the filtering in Section 4.1 to remove photobleaching, because noise may affect the setting of thresholding parameters. We compare three thresholding methods, including (b) mean, (c) mean + standard deviation, and (d) Otsu, shown in Figure 15.

The details of each result are explained below.

(a) No thresholding: Original Myosin image.

(b) Mean: We use the 3D image at the first time point and calculate the average value of all pixel intensity values. This average value is the thresholding value to be used.

(c) Mean + standard deviation: This method is adding the value of standard

deviations to the mean intensity value as the threshold.

(d) Otsu:

1. Calculate the histogram and probability of each intensity level
2. Set up initial probability  $\omega_i(0)$  and mean  $\mu_i(0)$
3. Step through all possible thresholds  $t = 1 \dots$  maximum intensity  
Update  $\omega_i(t)$  and  $\mu_i(t)$  and compute variance  $\sigma_i^2(t)$
4. The expected threshold corresponds to the maximum  $\sigma_i^2(t)$

It can be seen from the results that these three threshold methods are all very good. We want to use a constant threshold to segment all Myosin proteins in the 4D image, but Otsu will reset the threshold according to the 3D image at each time point, so this method does not meet our requirements.

Finally, “mean + standard deviation” is selected as the thresholding method because this algorithm can change the thresholding level easily. Its advantage is that when the image fluctuates too much, we can adjust the thresholding value by adding a number of standard deviations to the mean value.

## 4.5 Opening

Opening removes small objects from the foreground of an image, and placing them in the background [24]. In mathematical morphology, opening is the dilation of the erosion of a structural element B on the set A:

$$A \circ B = (A \ominus B) \oplus B$$

Equation 7

Result of opening is shown in the Figure 16.

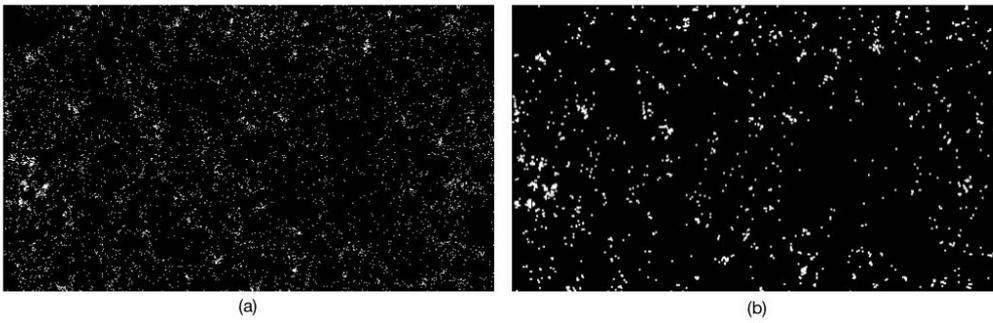


Figure 16 Comparisons (a) No opening (b) Opening

## Chapter 5: Correlation Analysis

In this chapter, based on quantification results obtained in Chapter 4, we will analyze the correlation results of between two types of proteins in several datasets processed in different methods. Specifically, we will choose one or more methods to process our new data set, including thresholding, masking method, and mathematical morphology. We select and combine the above methods by controlling variables, not a completely random combination. In the end, we will compare and analyze the correlation analysis results. For example, we will analyze why some results have strong correlation, but this strong correlation will become weaker after making morphological changes. We will also comprehensively compare all situations and select methods that are highly relevant and meet the actual biological significance.

### 5.1 Dataset

The dataset we selected is the 4D fluorescence microscopic images of three proteins (Ajuba, E-cad & Myosin) in drosophila embryos. In our previous work, we have already obtained the segmentation and tracking information of Ajuba. We use the sorting method proposed in Chapter 3 to divide Ajuba into different family trees. Then we can calculate the intensity of each Ajuba family tree in the image data.

As shown in the Figure 17, we have generated a heat map here to record time and intensity information, which can intuitively observe the changing trend of Ajuba. There will be many Ajuba clusters in total, and only a small part of them is shown in the picture. We can see that the intensity of Ajuba is basically gradually increasing.

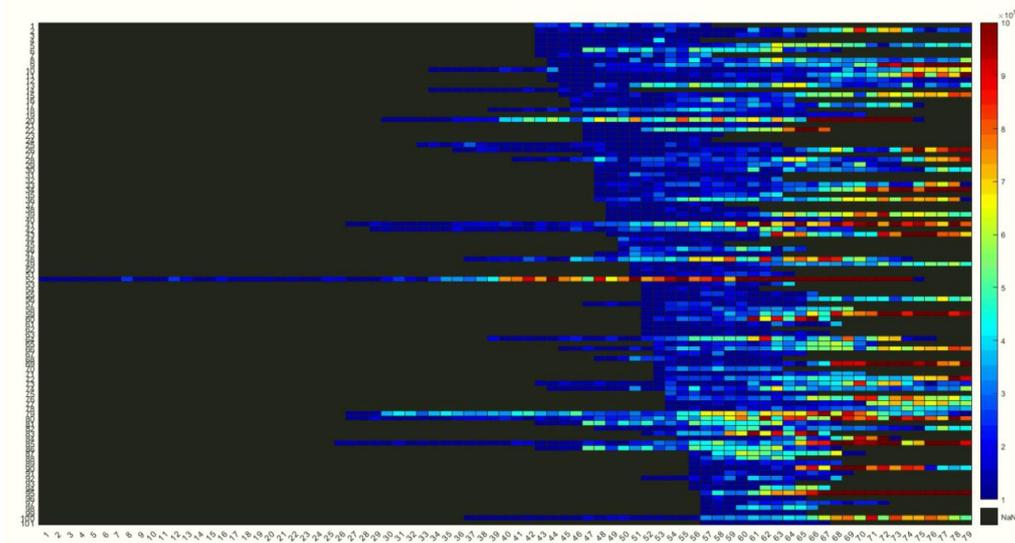


Figure 17 Heat Map Example - Part of Ajuba Intensity

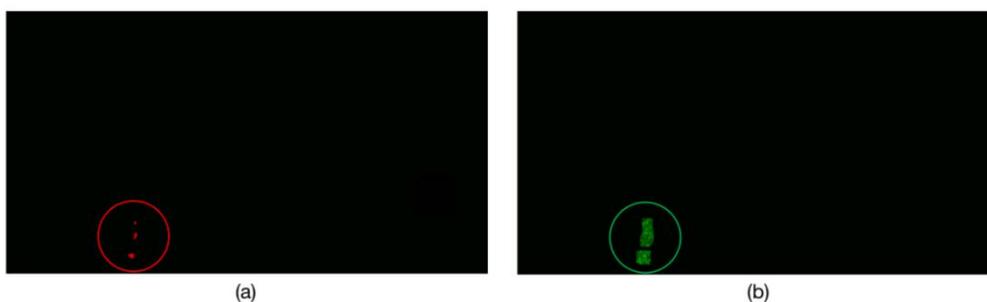


Figure 18 Quantification (a)Ajuba (b)Myosin

Next, we will use Ajuba's size and location information to quantify E-cad and Myosin. The size and position of E-cad and Ajuba are basically the same, so we can directly calculate the intensity of E-cad. Because of detection limit, the molecular proteins of Myosin cannot be fully detected, including the parts that exert force on the Ajuba protein cluster but do not overlap with their positions. When quantifying Myosin that does not overlap with Ajuba, we have to use the masking method

mentioned in Chapter 4. Figure 18 is an example of using Ajuba to quantify Myosin.

In the quantification of Myosin, we mainly choose Ajuba objects which last longer time, especially occurs in a later time point. We know that in biology process, Ajuba is the tensor sensor which is recruited by Myosin to adhere junctions. Myosin has not been formed at the beginning, and the intensity will be very low, so it is meaningless to calculate the correlation between Myosin and Ajuba at that time.

## 5.2 Five Methods

After we get the processed dataset of Ajuba, E-cad and Myosin, we can start to do correlation analysis. An E-cad object can be obtained using the Ajuba mask, so the intensity value of E-cad can be directly compared with Ajuba. This part is mainly for normalizing the intensity value of Myosin objects. We consider the following options:

- 1) Using the original Myosin image or applying the thresholding method directly on Myosin image.
- 2) Using the original (Ajuba) mask when quantifying Myosin or an expanded mask when quantifying Myosin.
- 3) Using Morphological operation (opening) on the image after thresholding.

Combining these options, we will have five different methods, shown in the Table 4.

#	Name	Thresholding	Expanded Mask	Opening
1	NT_NE	No	No	No
2	T_NE	Yes	No	No
3	NT_E	No	Yes	No
4	T_E	Yes	Yes	No
5	T_E_O	Yes	Yes	Yes

Table 4 Five Methods

### 5.3 Method Comparison and Analysis

First, we take out one of the correlation results (ID 2782) to analyze and verify the rationality of our methods, shown in Figure 19 and Figure 20.

First of all, the reason why we chose ID 2782 is because its family tree has existed for a long time, so that we can observe the changes in later time points. Figure 19 shows the changes of Ajuba and Myosin in the ID 2782 family tree from the 36th time point to the 79th time point, a total of 45 time points. Figure 20 shows the results obtained by using different methods to quantify Myosin. For the comparison, the sample chosen here are mostly of good correlations and the peak patterns from different myosin quantification approaches are similar. The big peaks produced by moving average/smooth appear to be true peaks rather than due to pixel numbers. The small peaks before smooth may be but the big peaks are not. It is consistent with Myosin average pixel value (the bottom curve in Figure 20 is average pixel value) showing a similar peak pattern. The average value does not change the patterns of the

large peaks produced without thresholding. The thresholding approach actually demonstrates worse correlation because it tends to remove the subtle increase in Myosin - basically the subtle increases are treated as background. This suppresses the peaks in early time point and reduces correlations. It also suppresses the ramping up portion of peaks in later time points and as a result, it appears that Myosin increases later than it actually does.



Figure 19 ID 2782 Quantification (Red - Ajuba, Green - Myosin)

Start from Time Point 36 to Time Point 79

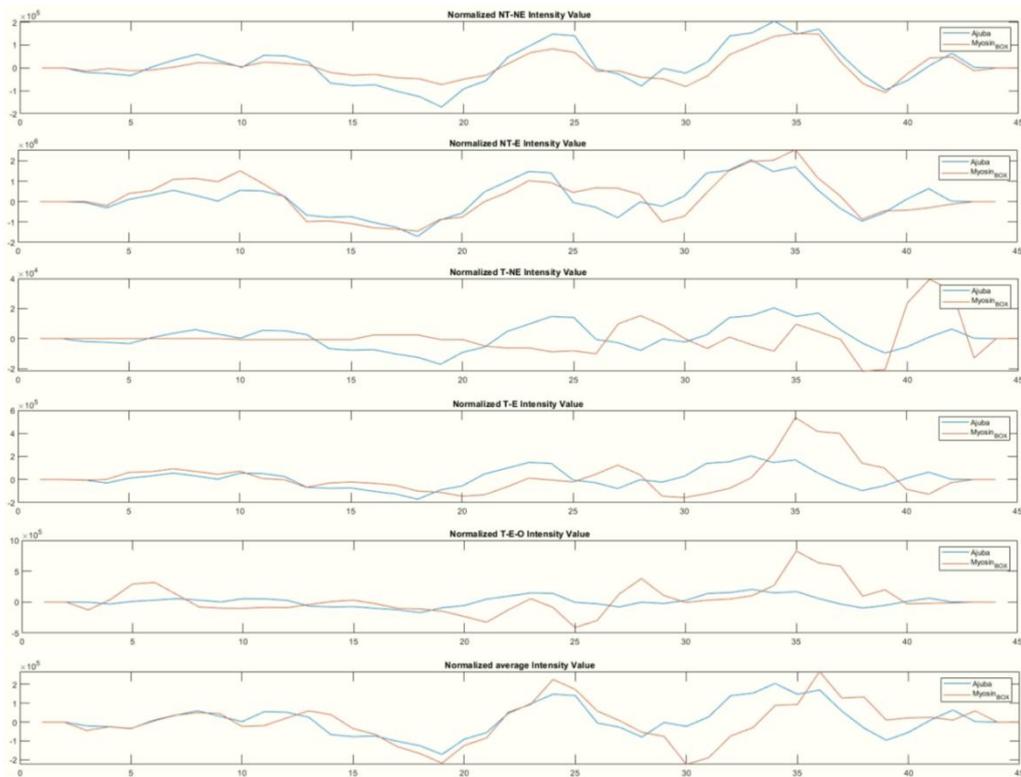


Figure 20 Normalized Intensity Value (Blue - Ajuba, Red - Myosin) (1) NT\_NE (2)

NT-E (3) T\_NE (4) T\_E (5) T\_E\_O (6) Average

Considering that without thresholding, artificial peaks may be generated due to fluctuations in the number of pixels, the following analysis is conducted.

1. Subtracting moving average with large window sizes is effectively a thresholding itself and allows for identification of signal peaks.

2. The periods of the resulted peaks are similar to what previously reported for Myosin pulses in this tissue: 90 sec.

3. The average pixel intensity gives the same peak patterns, indicating that the pixel number does not influence the final pattern significantly - most likely this is taken care of by the "smooth by 3" (this suppresses or smooths out the small fluctuations due to pixel number).

4. The two non-thresholding methods end up giving rise to artificial peaks because it leads to abrupt fluctuations in the total intensity value.

Overall the normalization by smooth function takes care of the thresholding already and is better at preserving the subtle changes.

#### **5.4 Correlation Results**

In the following, the process of correlation analysis is explained before the results are presented and analyzed. Considering the Ajuba/Myosin pair, we select Ajuba object IDs that exist at least 9 time points from ID 2000 -3000 to quantify Myosin and conduct correlation analysis. There are about 150 valid IDs. For each Ajuba object and corresponding Myosin intensity, we used the `xcorr` function to calculate the correlation coefficient:

$$[r,lags] = \text{xcorr}(x,y,20,'coeff')$$

Equation 8

In this function,  $x$  and  $y$  are input arrays which are the total intensity of Ajuba and Myosin respectively. And 20 is the maximum lag, specified as an integer scalar. The returned cross-correlation sequence ranges from -20 to 20. ' $coeff$ ' is normalizing the sequence so that the autocorrelation at zero lag is equal to 1. Its mathematical formula is:

$$R_{xy,coeff}(m) = \frac{1}{\sqrt{R_{xx}(0)R_{yy}(0)}} R_{xy}(m)$$

Equation 9

In output arguments,  $r$  is the autocorrelation result,  $lags$  is the lag indices. We use the five methods described before to quantify Myosin, calculate the correlation coefficient of Ajuba and Myosin, and derive the average result eventually.

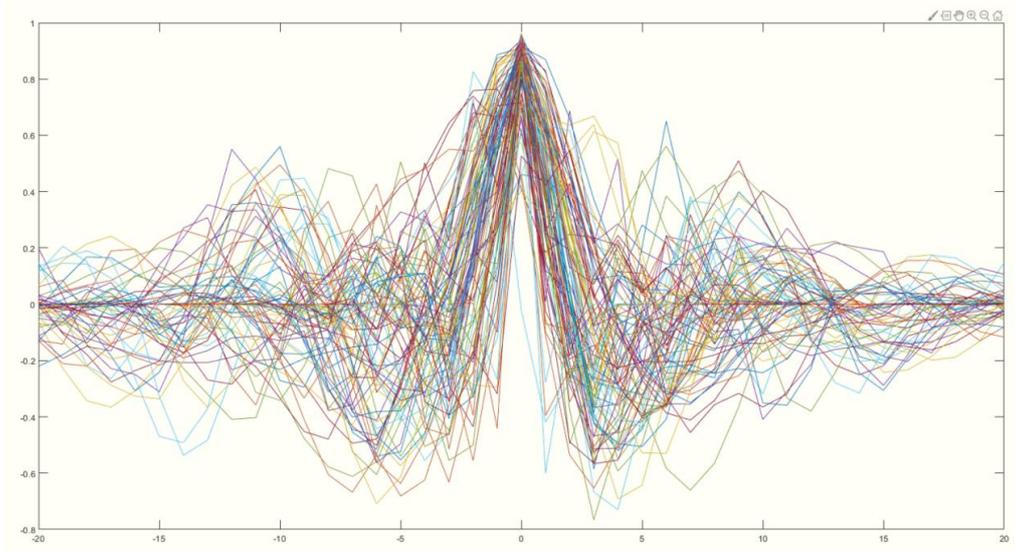


Figure 21 First Method (NT\_NE)

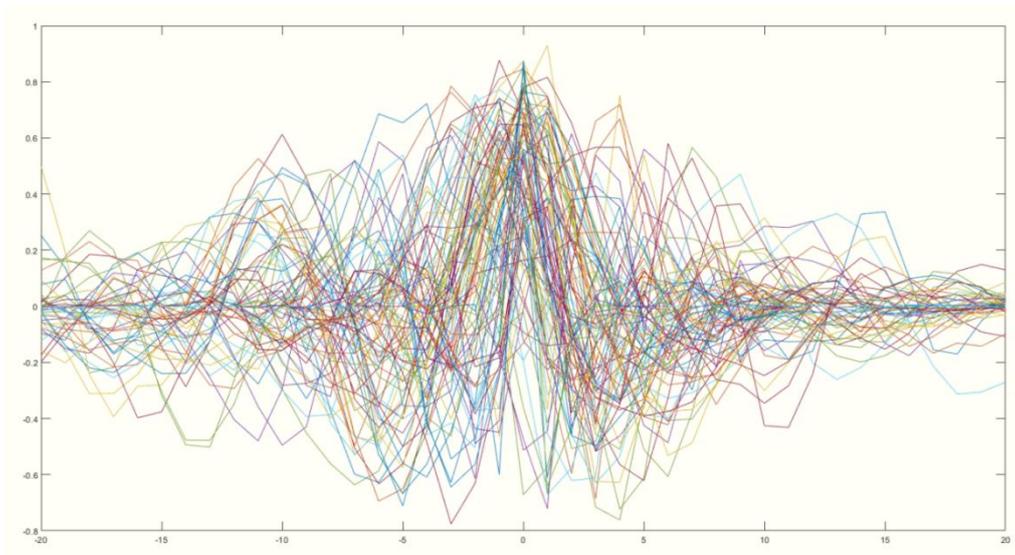


Figure 22 Second Method (T\_NE)

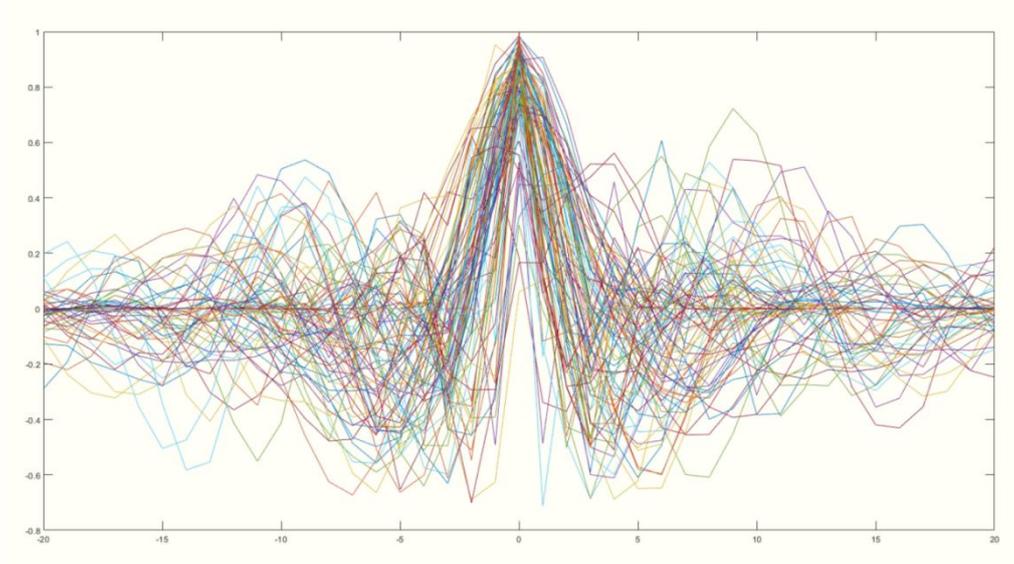


Figure 23 Third Method (NT\_E)

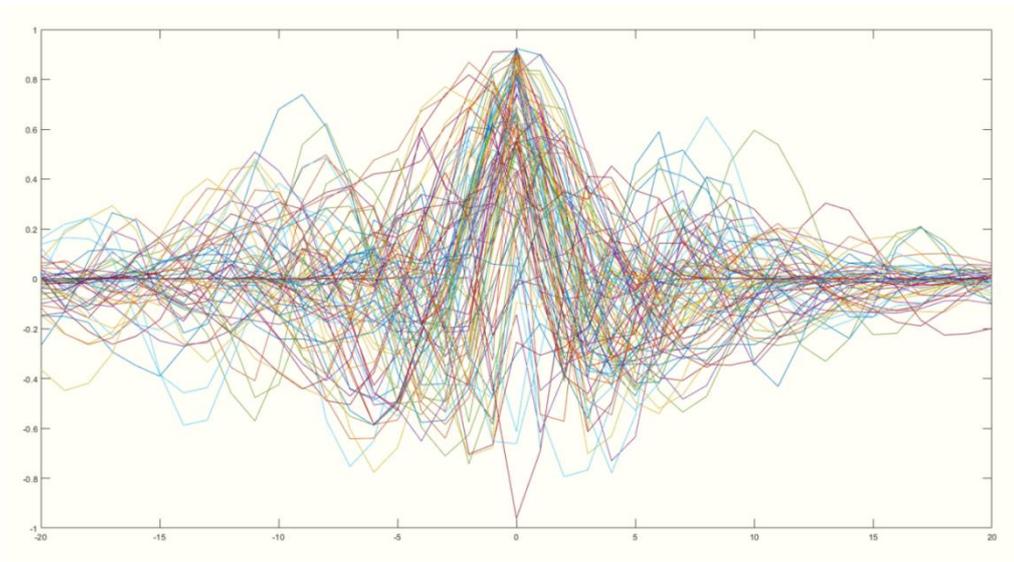


Figure 24 Fourth Method (T\_E)

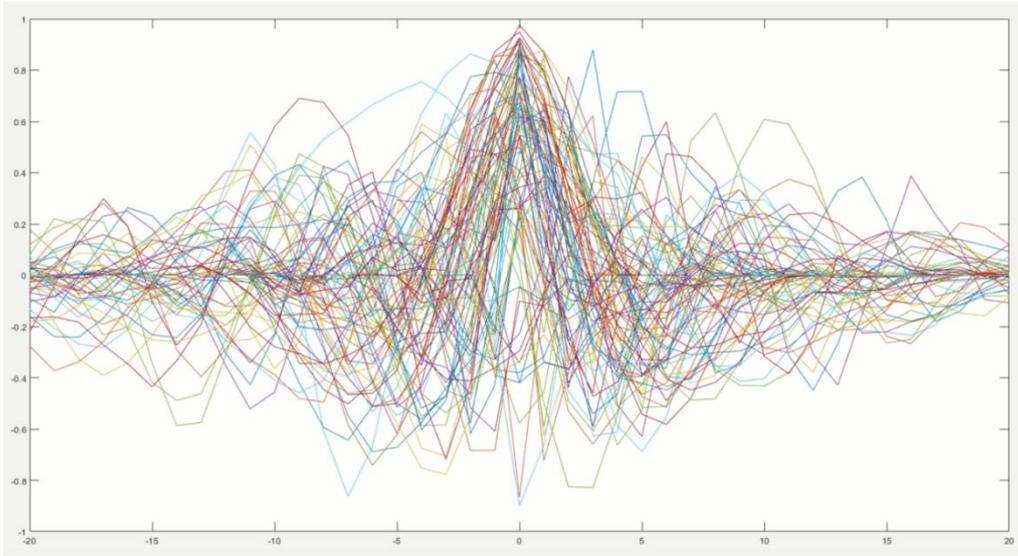


Figure 25 Fifth Method (T\_E\_O)

The average of all valid ID correlation results:

Method (Name)	Ave Correlation Result
First Method (NT_NE)	0.82
Second Method (T_NE)	0.47
Third Method (NT_E)	0.81
Fourth Method (T_E)	0.51
Fifth Method (T_E_O)	0.44

Table 5 Correlation Results for Ajuba and Myosin

We choose to use NT\_NE method to do the correlation analysis for “E-cad and Ajuba”dataset. The average correlation result is 0.88.

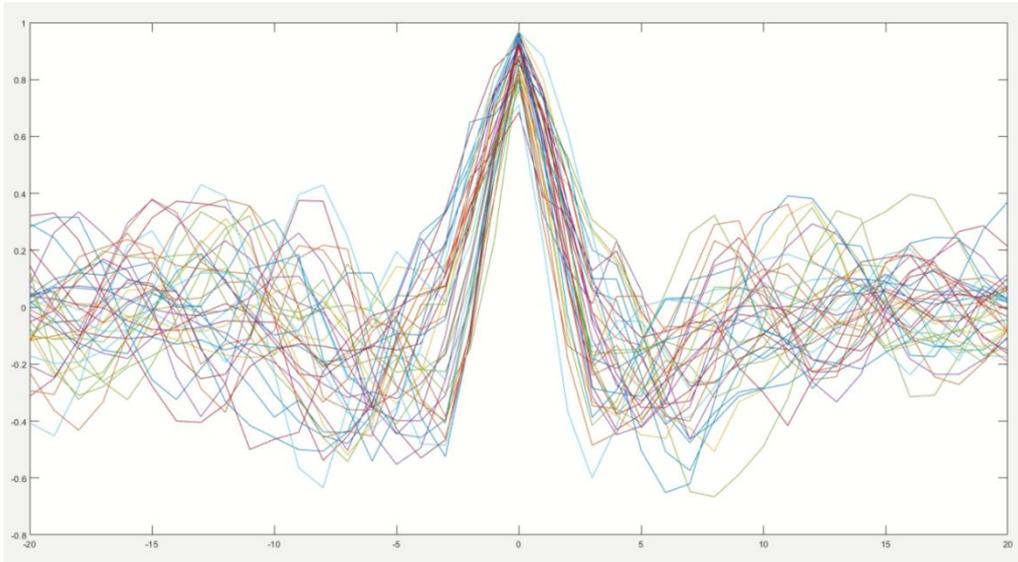


Figure 26 “Ecad & Ajuba” Correlation (NT\_NE)

We use both NT\_NE and NT\_E methods for “E-cad & Myosin” dataset. The average correlation result are 0.73 and 0.62, respectively.

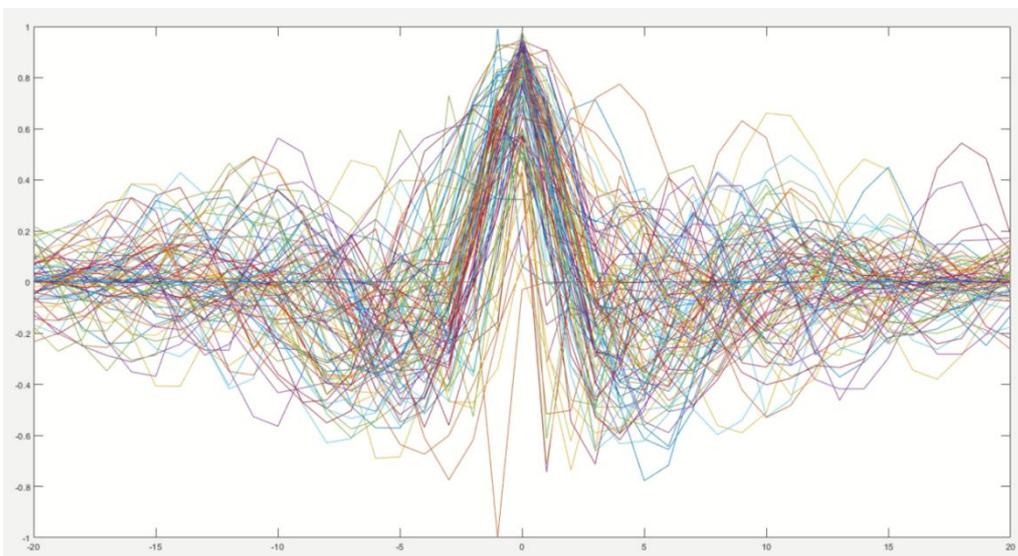


Figure 27 “Ecad & Myosin” Correlation (NT\_NE)

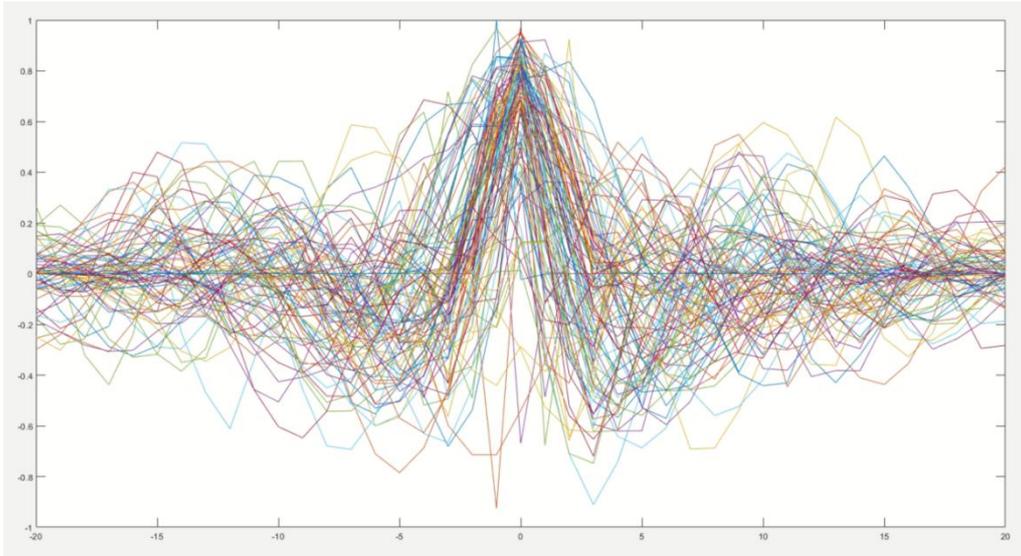


Figure 28 “Ecad & Myosin” Correlation (NT\_E)

## Chapter 6: Conclusions

In this thesis, we propose a framework to quantitatively analyze the correlation between two types of proteins based on the tracking results. The framework consist of three major steps: 1) sorting family trees of major objects by optimizing the relationship (fusion and split) between objects, and eliminating the interference of objects existed for a short time. 2) In the step of quantifying Myosin, Gaussian blur, binary dilation and moving average methods are applied to remove photobleaching, expanding masks, and smoothing the results, respectively. The correlation results of five different methods show that thresholding and opening are not necessary. The normalization by smooth function takes care of the thresholding already and is better at preserving the subtle changes. In our future work, this framework can be generalized and applied for quantification and analysis in other biological research.

## References

- [1] Schneider, Caroline A *et al.*, "NIH Image to ImageJ: 25 years of image analysis", *Nature methods*, vol. 9,7 (2012): 671-5. doi:10.1038/nmeth.2089.
- [2] Schindelin, Johannes *et al.*, "Fiji: an open-source platform for biological-image analysis", *Nature methods*, vol. 9,7 676-82. 28 Jun. 2012, doi:10.1038/nmeth.2019.
- [3] Carpenter, Anne E *et al.*, "CellProfiler: image analysis software for identifying and quantifying cell phenotypes", *Genome biology*, vol. 7,10 (2006): R100. doi:10.1186/gb-2006-7-10-r100.
- [4] Kamentsky, Lee *et al.*, "Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software", *Bioinformatics (Oxford, England)*, vol. 27,8 (2011): 1179-80. doi:10.1093/bioinformatics/btr095.
- [5] Hell, Stefan W., "Toward fluorescence nanoscopy," *Nature biotechnology*, vol. 21,11 (2003): 1347-55. doi:10.1038/nbt895.
- [6] Roth, Julian *et al.*, "Fast TIRF-SIM imaging of dynamic, low-fluorescent biological samples." *Biomedical optics express*, vol. 11,7 4008-4026. 26 Jun. 2020, doi:10.1364/BOE.391561.
- [7] Song, L *et al.*, "Photobleaching kinetics of fluorescein in quantitative fluorescence microscopy", *Biophysical journal*, vol. 68,6 (1995): 2588-600. doi:10.1016/S0006-3495(95)80442-X.
- [8] Yang, Fuxing *et al.*, "Cell segmentation, tracking, and mitosis detection using

- temporal context", International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 8,Pt 1 (2005): 302-9. doi:10.1007/11566465\_38.
- [9] J.R. Albani,, "Structure and Dynamics of Macromolecules: Absorption and Fluorescence Studies", Elsevier Science, 2004, Chapter 2 - Fluorescence: Principles and Observables, Editor(s): J.R. Albani, Pages 55-98, ISBN 9780444514493, doi:10.1016/B978-044451449-3/50002-2.
- [10]R.A. Haddad and A.N. Akansu, "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 39, pp 723-727, March 1991.
- [11]E. S. Gedraite and M. Hadad, "Investigation on the effect of a Gaussian Blur in image filtering and segmentation", Proceedings ELMAR-2011, 2011, pp. 393-396.
- [12]Arce, Gonzalo R., "Nonlinear Signal Processing: A Statistical Approach", New Jersey, 2005, USA: Wiley. ISBN 0-471-67624-1.
- [13]Arias-Castro, Ery; Donoho, David L., "Does median filtering truly preserve edges better than linear filtering", Annals of Statistics. 37 (3): 1172–2009, arXiv:math/0612422, Bibcode:2006math.....12422A, doi:10.1214/08-AOS604.
- [14]Pham, Nhu-An *et al.*, "Quantitative image analysis of immunohistochemical stains using a CMYK color model", Diagnostic pathology, vol. 2, Feb. 2007, doi:10.1186/1746-1596-2-8.
- [15]Prinyakupt, J., Pluempitiwiriwawej, C., "Segmentation of white blood cells and

- comparison of cell morphology by linear and naïve Bayes classifiers", *BioMed Eng OnLine* 14, 63 (2015). doi:10.1186/s12938-015-0037-1.
- [16] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002, doi: 10.1109/78.978374.
- [17] Y. Jiao *et al.*, "Automated quantification of white blood cells in light microscopy images of injured skeletal muscle", *Proc. IEEE 8th Annu. Computing and Communication Workshop and Conf*, 2018.
- [18] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Trans. Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, Jan. 1979.
- [19] J. Kittler and J. Illingworth, "On threshold selection using clustering criteria", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 5, pp. 652-655, Sept.-Oct. 1985, doi: 10.1109/TSMC.1985.6313443.
- [20] Jianzhuang *et al.*, "Automatic thresholding of gray-level pictures using two-dimension Otsu method". *Circuits and Systems, Conference Proceedings, China, 1991 International Conference on*: 325–327.
- [21] M. Khan, H. Nisar, N. Aun and P. Lo, "Iterative region based Otsu thresholding of brightfield microscopic images of activated sludge", in *Proc. IEEE EMBS Conf. Biomedical Engineering and Sciences*, 2016, pp. 533-538.
- [22] H. Sun *et al.*, "An improved OTSU's method for CT image boundary contour extraction", in *Proc. IEEE Int'l Conf. Imaging Systems and Techniques (IST)*,

2016, pp. 493-497.

[23]A. Jain and R. Gupta, "Gaussian filter threshold modulation for filtering flat and texture area of an image", International Conference on Advances in Computer Engineering and Applications, 2015, pp. 760-763, doi: 10.1109/ICACEA.2015.7164804.

[24]Edward R. Dougherty, "An Introduction to Morphological Image Processing", ISBN 0-8194-0845-X (1992).

[25]Booth, "Hydrologic Variability of the Cosumnes River Floodplain", San Francisco Estuary and Watershed Science, Volume 4, Issue 2, 2006.

[26]Ya-lun Chou, "tatistical Analysis", Holt International, 1975, ISBN 0-03-089422-0, section 17.9.

## Curriculum Vitae

Yuan Dao, B.S.
Department of Electrical and Computer Engineering University of Nevada, Las Vegas Las Vegas, NV 89119 Email: antonyyuan0604@gmail.com
Education
Beijing Jiaotong University, China B.S. 2020

Table 6 Curriculum Vitae