

POLITECNICO DI TORINO

Master of Science in Mathematical Engineering

Master's Degree Thesis

**3D Reconstruction of Experimental Images
of Molecular Sorting in Living Cells**



Supervisors

Prof. Andrea Gamba

Prof. Lamberto Rondoni

Candidate

Cristiana Turco

Academic Year 2017-2018

To my beloved mother, Ada

Abstract

Molecular sorting is a complex system by which eukaryotic cells sort specific proteins aggregating into localized domains, subsequently distilled into lipid vesicles and transported to appropriate intracellular destinations. The main difficulty in analyzing this process is related to its tridimensional evolution on micrometric intracellular membrane structures called endosomes. Despite being powerful tools, microscopy techniques have limit of resolution when working on this scale. Consequently, to overcome the instrumental barriers new methods are needed. The aim of this thesis work is to reconstruct the molecular sorting from experimental images obtained by means of the microscope. The whole project can be summarized in three main phases: image acquisition, preprocessing and identification of structures. I have collected the experimental data, using a confocal fluorescence microscope, at Molecular Biotechnology Center (MBC) of the Università degli Studi di Torino (UniTo). Subsequently, the images have been preprocessed using the deconvolution technique to remove the noise associated with the experimental setup. At this stage, the main issue was the poor resolution. To cope with this problem, I have developed a 3D reconstruction procedure, based on the a priori knowledge about endosomal and protein domain morphologies. This approach has proven to reach good results providing a faithful representation of the experimental images. The thesis is organized as follows. In the first chapter molecular sorting is introduced. Then, in the second chapter are presented the mathematical tools, while the third is devoted to the data acquisition and to the deconvolution technique. Subsequently, in the fourth

chapter the reconstruction procedure is explained. Finally, in the last two chapters, the main results obtained are discussed, presenting conclusions and future work.

Acknowledgements

First and foremost, I would like to thank my thesis supervisor Professor Andrea Gamba for his motivation, enthusiasm, and immense knowledge. Without his patient guide, this thesis project would have not been possible.

I am also grateful to Dr. Jean Piero Margaria for providing all the necessary means for the experiments. His preparation, talent and knowledge have been crucial in this work.

I would also like to acknowledge Professor Alfredo Brunstein for useful discussions and valuable suggestions.

Furthermore, I must express my very profound gratitude to my family members Francesco, Giulio, Alessandra, Francesco, Fabrizio and also to my dear friend Debora, for the support and continuous encouragement throughout my years of study.

Finally, I thank my loved Giuseppe to encourage and inspire me with his positivity.

This accomplishment would not have been possible without them.

Thank you.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Molecular Sorting	4
2	Mathematical Framework	12
2.1	Elements of Information Theory	12
2.2	Properties of Kullback-Leibler Divergence	17
2.3	Statistical Models	19
2.4	Maximum Likelihood and Kullback-Leibler Divergence	20
2.5	Alternative Forms for the Kullback-Leibler Divergence	23
3	Deconvolution of Experimental Images	25
3.1	Fluorescence Microscopy	27
3.2	Image formation	30
3.3	Deconvolution	37
4	3D Reconstruction of Experimental Images	40
4.1	Data Interpretation	40
4.2	Structure Recognition	42
4.2.1	Recognition of Endosome Structure	43
4.2.2	Recognition of Protein-Enriched Membrane Domains	46
4.2.3	Reconstruction Procedure	50

5	Results	53
5.1	Image Deconvolution	53
5.2	Endosome Reconstruction	57
5.3	Domain Reconstruction	61
6	Conclusions and Future Work	67
	Appendices	69
A	Acquisition of Experimental Images	70
B	3D Deconvolution Manual	72
B.1	Creating a PSF image	72
B.2	3D Deconvolution	73
C	Python Code for 3D Reconstruction	75
C.1	Read_Meta_images.py	77
C.2	Tools_fyve2x.py	82
C.3	Tools_gpf.py	85
C.4	Reconstruction.py	89
C.5	Final_Script	94

Chapter 1

Introduction

1.1 Motivation

The process of molecular sorting is an elaborate system whereby specific proteins and lipids are concentrated and distilled into nanometric lipid vesicles in eukaryotic cells [14, 20]. This complex process takes place on the plasma membrane, on inner membrane bodies called endosomes [22]. It is a crucial phenomenon of great interest for a different variety of biological phenomena. In particular, molecular sorting conduce to the traffic of vesicles, transporting lipids and proteins to the most disparate subcellular locations [21], contributing in this way to maintain their biochemical identity. However, still today this dynamics is not yet fully understood. For this reason, it is clear the great interest of biologists and physicists to investigate the phenomenon experimentally [6].

Molecular sorting can be observed in real time using modern fluorescence microscopy techniques, which are discussed in Chapter 3.

The experiments are carried out on cell cultures in vitro, in which the endosome membrane and the proteins are made visible with fluorescent markers. These experiment provide real time movies of the process.

The main difficulties are related to the three-dimensional nature of the phenomenon. In fact, the molecular factors can aggregate at each point of

the endosome membrane and the process generating the domains changes very quickly over time.

Besides, despite the continuous technological evolution, imaging tools have resolution limits linked to the light nature and to its diffraction. In fact, 3D visualization techniques are not yet widely used to observe phenomena of this type. For this reason, the biologists analysis are limited to observations carried on two-dimensional optical sections of the specimen.

To overcome the instrumental barriers new methods are needed.

The goal of the present work is to reconstruct the three-dimensional volume of the endosome and to identify the proteins domains on the lipid membrane, using experimental images obtained from the microscope.

The images collected are called Z-stacks and are described in detail in Chapter 3. They are data generated from a series of optical sections imaged at regular intervals along the Z axes, as in Figure 1.1.

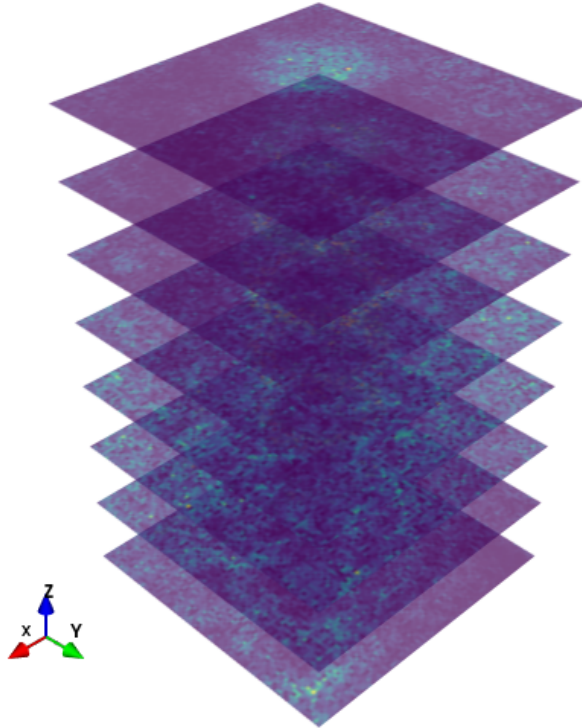


Figure 1.1: Example of Z-stack

Using the current microscope techniques, it is possible to observe only partially the volume into consideration. In fact, in our experiments, optimizing the acquisition parameters of the microscope, a resolution about 3 times lower in the z-direction compared to the xy ones is obtained. Consequently, the digital volume element of the images, the voxel, is strongly anisotropic, with dimensions of $\Delta z = 155 \text{ nm}$ and $\Delta x = \Delta y = 45 \text{ nm}$.

To cope with this technical problems, the strategy is to use the knowledge about the phenomenon. In fact, experimental observation suggest that the protein domains have mostly a circular morphology [25]. Furthermore, theoretical considerations about the circular shape has been linked to the presence of a linear tension, the 2D analogous of the surface tension observed in the physical process of phase separation [8, 9].

Using the mathematical tools of information theory, probability, statistics and optimization theory, I have developed a method able to reconstruct the endosome 3D volume and the molecular agglomerates on the lipid membrane.

To summarize, the aim of the thesis is to acquire information about the phenomenon, which can not be currently recorded due to the limit of the instrument optics, exploiting the a priori knowledge of molecular sorting.

1.2 Molecular Sorting

The cell is the fundamental structural and functional unit of living organisms. There are two different types of cells: *prokaryotic* and *eukaryotic*. Differently from a prokaryotic cells, a eukaryotic cell has a more complex structure, subdivided into more than a single intracellular compartments.

Several thousands of different types of eukaryotic cells have been identified. Over the time the evolution process allowed the specialization of eukaryotic cells to perform dedicated functions such as movement, energy storage and reproduction. Every cell belonging to this family has a plasma membrane separating the interior of the cell from the outside environment, called extracellular space. In addition, it is elaborately subdivided into functionally distinct, membrane-enclosed compartments, also named *organelles* [1]. The major intracellular compartments, common to all eukaryotic cells are shown in Figure 1.2.

The cells have a *nucleus* that contains the genome which is the principal site of DNA and RNA synthesis. The surrounding *cytoplasm* consists of the *cytosol* and the cytoplasmic organelles suspended in it. Among the most important, the *endoplasmic reticulum* (ER) is a transport network for molecules targeted for certain modifications and specific destinations. The ER has two forms: the *rough ER*, which has on its surface *ribosomes* secreting proteins into the ER, and the *smooth ER*, which lacks ribosomes.

Another fundamental organelle is the *Golgi apparatus*. Its primary function is to process and package macromolecules, such as proteins and lipids, from the ER dispatching them to various destinations.

A further organelle is the *mitochondria*, where respiration occurs. This process generates the energy for the cell by oxidative phosphorylation, using oxygen to release the energy stored in cellular nutrients.

The *lysosome* is the organelle containing digestive enzymes which degrade defunct or worn-out intracellular organelles, as well as macromolecules, particles, engulfed viruses or bacteria taken in from outside the cell by endocytosis. On the way to lysosomes, endocytosed material must first pass through a se-

ries of organelles called *endosomes*. The endocytosed molecules are sorted on the endosomes that subsequently send them to their final destination.

In general, each membrane-enclosed organelle performs the same set of basic functions in all types of eukaryotic cell.

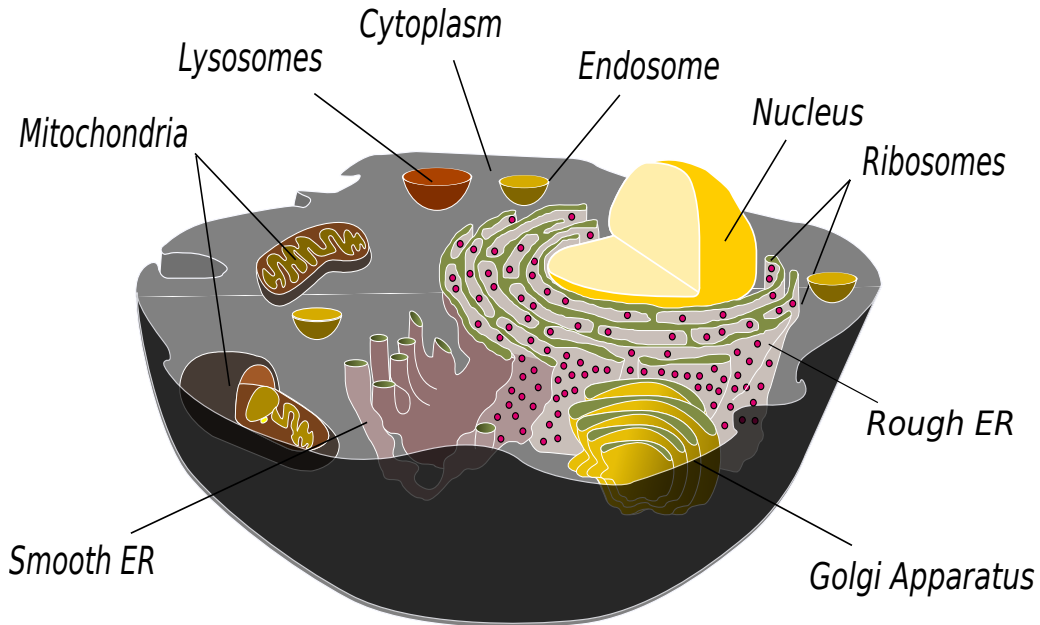


Figure 1.2: The major intracellular compartments of an animal cell. The cytosol (gray), ER, Golgi apparatus, nucleus, mitochondria, endosomes and lysosomes are distinct compartments isolated from the rest of the cell by a membrane.

Membranes are crucial to the life of the cell. As the cellular membrane encloses it, defines its boundaries and maintains the essential differences between the cytosol and the extracellular environment, similarly, inside eukaryotic cells, the membranes of the organelles maintain the characteristic differences between the contents of each organelle and the cytosol. Despite their differing functions, all biological membranes have a common general structure: each is a very thin film of lipid and protein molecules (Figure 1.3).

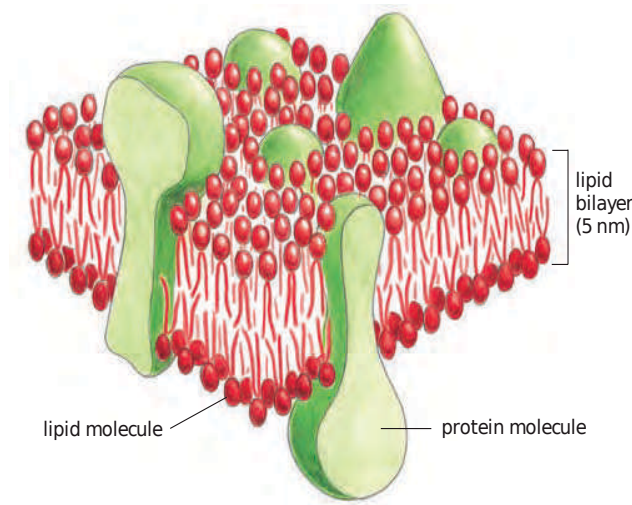


Figure 1.3: A schematic view of a cell membrane and the general disposition of its lipid and proteins [1].

Cell membranes are dynamic fluid structures, and most of their molecules move in the plane of the membrane. In fact, many vital biochemical processes take place in membranes or on their surfaces.

The lipid molecules are arranged as a continuous double layer of about 5 nm thick. This lipid bilayer provides the basic fluid structure of the membrane and serves as a relatively impermeable barrier to the passage of most water-soluble molecules.

Most membrane proteins span the lipid bilayer and mediate almost all of the other functions of the membrane, including the transport of specific molecules across it, and the catalysis of membrane-associated reactions.

For instance, the plasma membrane contains proteins which act as sensors of external signals, allowing the cell to change its behavior in response to environmental cues, including signals from other cells. The protein sensors, or receptors, transfer information—rather than molecules—across the membrane.

A simple example is given by cellular duplication. The starting structure of the cell is initially isotropic. In response to external signal the cell breaks

the membrane symmetry. At this stage is possible to observe the emergence of a daughter cell, and subsequently the duplication, as shown in Figure 1.4.

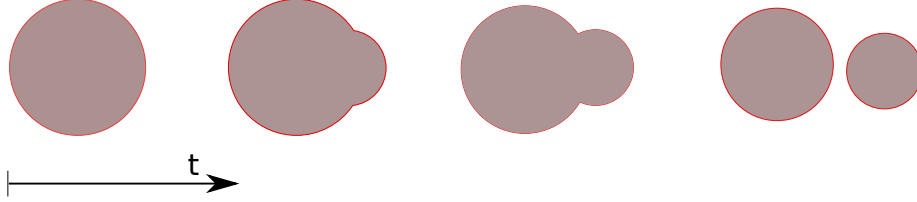


Figure 1.4: Duplication of the cell.

A crucial point is to understand in which part of the cell the germ nucleates. Specific membrane proteins aggregate into localized domains, deforming the membrane. In these accumulation regions the germ forms leading to the duplication process as graphically shown in Figure 1.5.

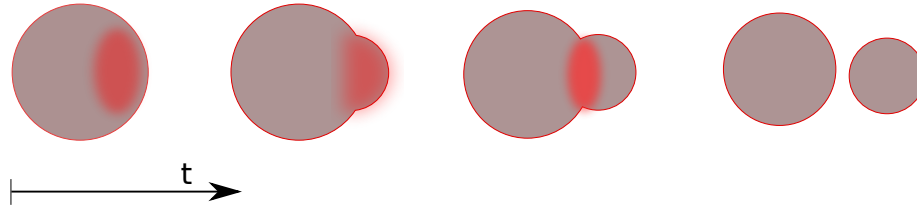


Figure 1.5: Molecular factors aggregation into localized domains leading to the cell duplication.

Experimental evidence of the aggregation phenomenon can be obtained by looking at the cells under a microscope. Molecular factors of interest are marked with fluorescent molecules allowing their visualization.

In [25] experiments have been conducted on yeast cells, in which two factors, Bem1 and Cdc3 were marked and followed over time. From these experimental images (Figure 1.6) it is evident that the molecules are initially spread on the cell membrane and subsequently aggregate into well-localized domains.

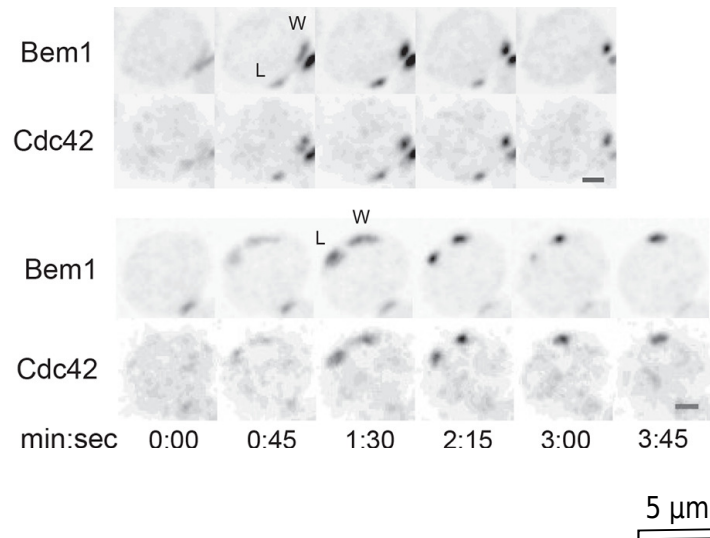


Figure 1.6: Example of aggregation of molecular factors on the membrane of a yeast cell [25].

Every membrane organelle contains its own characteristic set of enzymes and other specialized molecules making it unique. Each set of these molecular factors performs a different function.

In the intracellular space, subsets of molecules (proteins, reactants, ions) are concentrated to optimize the biochemical reactions in which they participate. Because the lipid bilayer of cell membranes is impermeable to most hydrophilic molecules, the membrane of an organelle must contain membrane transport proteins to import and export specific metabolites [1].

These proteins can move between organelles in different ways. The synthesis of all proteins begins on ribosomes in the cytosol. Their subsequent fate depends on the chemical composition, which can contain sorting signals directing their delivery from the cytosol to organelle surfaces.

There are different ways by which proteins move from one compartment to another. The most common way is the *vesicular transport*, in which small spherical transport vesicles, membrane-enclosed, transports proteins from one compartment to another.

The transport vesicles become loaded of molecules derived from a donor compartment as they distill and pinch off from its membrane; they discharge these proteins into a second compartment by fusing with the membrane enclosing that compartment, as shown in Figure 1.7.

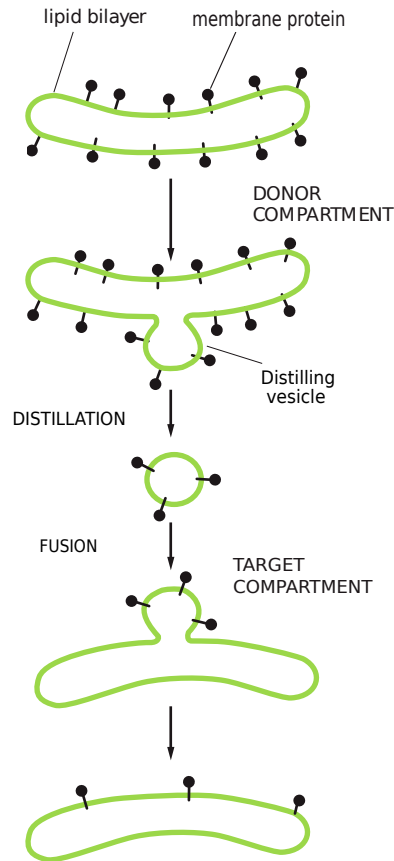


Figure 1.7: Transport vesicles are distillate from one compartment (donor) and fusing with another (target) compartment. In this process, proteins are transferred from one membrane to another.

These vesicles can take a great variety of pathways.

In each case to perform its function, the transport vesicles that buds from a compartment must be selective. Therefore, these must take up only the appropriate molecules and must fuse only with specific target membrane [1].

The main reason is that the cell has to maintain the molecular and functional differences between its compartments, preserving its special identity. Therefore, the molecules distilled from the donor compartment have to be sorted first.

The endosome is the organelle of the cell performing this task. The transport vesicles, loaded of mixed molecular factors, fuse with the endosome membrane as shown in Figure 1.8.

At this stage, the *molecular sorting* takes place: similar proteins aggregate on the membrane of the endosome by mutual affinity, creating homogeneous circular domains.

Once sorted, the proteins are distilled again into transport vesicles. These last, are then directed towards the target compartment, which can now recognize and accept them.

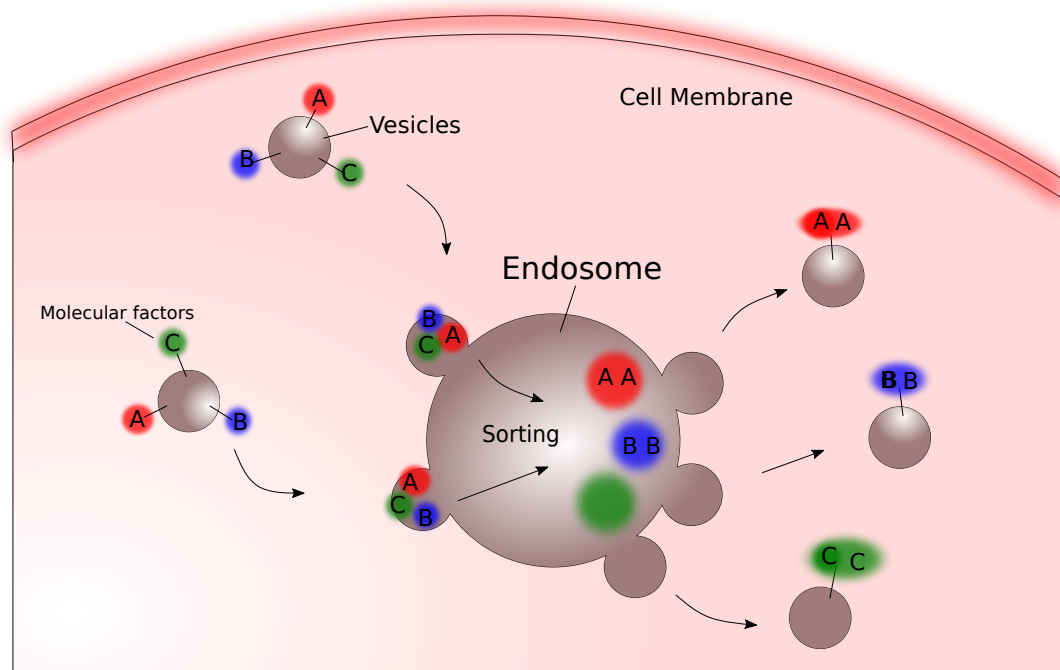


Figure 1.8: Molecular Sorting

Specifically, this happens in the case of endocytosis process, in which the molecules arriving from the extracellular space are internalized by means of transport vesicles derived from the cellular membrane. The same molecules are sorted on the endosomes and again distilled into vesicles. These can be now transported into the cytosol to be used in various biosynthetic processes, such as being delivered to the lysosome for degradation, to the plasma membrane to be released into the extracellular space or to other compartments to be further processed.

There are many different families of proteins taking part to molecular sorting. The most common is the Rab family, with over 60 members [1], such as Rab11, Rab4, Rab5 etc.

In particular, the sorting of Rab11 on the lipid membrane of the endosome will be considered in the present work. The aim is to give useful tools for further investigations on molecular sorting which is still today poorly understood.

Chapter 2

Mathematical Framework

In this chapter the main mathematical tools used in this thesis are exposed. In particular some definitions taken from information theory and statistics. These concepts will be very useful in the development of the 3D reconstruction presented in Chapter 4.

2.1 Elements of Information Theory

This section introduces the main concepts from information theory which has been a discipline only since the mid-1940s and covers a variety of theories that are fundamental to many of the sciences e.g. statistics, economics, physics, computer science etc.

Information theory was originally proposed by Claude E. Shannon in 1948 to find fundamental limits on signal processing and communication operations such as data compression, in a landmark paper entitled "A Mathematical Theory of Communication" [19]. Shannon's Information Theory definitely established the purely mathematical nature of entropy and relative entropy, in contrast to the previous identification by Boltzmann (1872) of his "H-functional" as the physical entropy [2].

The following statement is attributed to Shannon (Tribus and McIrvine 1971):

My greatest concern was what to call it. I thought of calling it “information”, but the word was overly used, so I decided to call it “uncertainty”. When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, “You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, nobody knows what entropy really is, so in a debate you will always have the advantage.”

The concepts of information theory are widely used in the following chapters of the thesis. In particular, the Kullback-Leibler divergence or relative entropy between two distribution of probability is defined and discussed. It will be convenient in the development of pattern recognition.

Let us to consider a discrete random variable X with a finite number of state x in the space of the states χ .

Let us to define the probability of the occurrence of an event $X = x$ as

$$p_X(x) = P(X = x) \quad (2.1)$$

Of course, the axioms of the probability are valid:

$$0 \leq p_X(x) \leq 1 \quad \text{and} \quad \sum_{x \in \chi} p_X(x) = 1 \quad (2.2)$$

How much information is received when it is observed $X = x$?

It depends on how probable the event in question is: if it is told that an highly improbable event has just occurred, one will have received more information than if it is told that some very likely event has just occurred, and if it is knew that the event was certain to happen e.g. $p_X(x) = 1$ one would receive no information [3].

Hence, one can model the amount of information with a monotonic function $h(x)$ that will depend on the probability distribution $p_X(x)$. To choose the form of $h(\cdot)$, let us consider two independent events x and y in χ . The information gained observing both should be the sum of each one of them. Instead, according to the independence of the events, the joint probability should be the product. Therefore the following must satisfy:

$$h(x, y) = h(x) + h(y)$$

$$p(x, y) = p_X(x)p_X(y)$$

From the last two relationships follows that $h(x)$ must be given by the logarithm of $p_X(x)$:

$$h(x) = -\log p_X(x). \quad (2.3)$$

This quantity is called **information gain**.

Properties

1. $h(x) = 0$ if $p_X(x) = 1$ for some $x \in \chi$;
2. $h(x) \geq 0$ for every $x \in \chi$ such that $0 \leq p_X(x) \leq 1$;
3. $h(x) > h(y)$ if $p_X(x) < p_X(y)$ with $x, y \in \chi$.

As we can see by the property 3 the selected function satisfies the desired requirement that a lower probability event x corresponds to a high information gain.

The choice of basis for the logarithm is arbitrary. For example, in information theory is used the logarithm in base 2 and then unit of $h(x)$ are bits. Instead, in statistics is used the natural logarithm with unit nats ('natural unit of information'). In the following the natural logarithm is used.

It is possible to state the following:

Definition 2.1. *Let X be a discrete random variable and p_X a probability distribution. The **entropy** of the random variable X is defined as:*

$$H(p) = - \sum_{x \in \chi} p_X(x) \ln p_X(x), \quad (2.4)$$

with $p_X(x) \ln p_X(x) = 0$ for all $x \in \chi$ such that $p_X(x) = 0$.

Notice that in case of $p_X(x) = 0$ for some $x \in \chi$, the value of the corresponding addend is taken to be 0, which is consistent with the limit:

$$\lim_{p \rightarrow 0^+} p \ln(p) = 0. \quad (2.5)$$

To better understand the entropy definition, let us to suppose that a sender wish to transmit the value of a random variable to a receiver, then the average amount of information received is given by the expectation with respect to the distribution p_X .

Therefore, the entropy can also be interpreted as

$$H(p) = \mathbb{E}_p[h] = \mathbb{E}_p[-\ln p_X(x)]. \quad (2.6)$$

Now, let us suppose that $p(x)$ and $q(x)$ are the probability that the random variable X be in the same state x under different conditions, which are described by the two mass distributions p and q .

Under these assumptions it follows:

Definition 2.2. *Let us define **relative entropy** or **Kullback-Leibler divergence** of the distribution q from p*

$$\begin{aligned} D_{KL}(p||q) &= - \sum_{x \in \chi} p(x) \ln q(x) - \left(- \sum_{x \in \chi} p(x) \ln p(x) \right) \\ &= - \sum_{x \in \chi} p(x) \ln \left(\frac{q(x)}{p(x)} \right) \\ &= \sum_{x \in \chi} p(x) \ln \left(\frac{p(x)}{q(x)} \right) \end{aligned} \quad (2.7)$$

with $p(x) \ln \frac{p(x)}{q(x)} = 0$ for all $x \in \chi$ such that $p(x) = 0$.

Imagine p as some unknown distribution and suppose that we have modelled this using an approximating distribution q . If we use q to construct a coding scheme for the purpose of transmitting values of X to a receiver, then the average additional amount of information required to specify the value of X as a result of using q instead of the true distribution p is given by the Kullback-Leibler divergence of q from p .

It is possible to extend the definitions of entropy and relative entropy to include distributions over continuous variables as follows:

Definition 2.3. *Let X be a continuous random variable and p a probability distribution. Let us define **entropy** of the random variable X*

$$H(p) = - \int p(x) \ln p(x) dx \quad (2.8)$$

with $p(x) \ln p(x) = 0$ for all x such that $p(x) = 0$.

Definition 2.4. *The **relative entropy** or **Kullback-Leibler divergence** of the probability distribution q from p is defined as*

$$D_{KL}(p||q) = \int p(x) \ln \left(\frac{p(x)}{q(x)} \right) dx \quad (2.9)$$

with $p(x) \ln \frac{p(x)}{q(x)} = 0$ for all x such that $p(x) = 0$.

2.2 Properties of Kullback-Leibler Divergence

1. $D_{KL}(p||q) \geq 0$ for any p and q , the equality is satisfied if and only if $p = q$ almost everywhere;

2. $D_{KL}(p||q)$ is *convex* in the pair of probability distributions (p, q) .

Let p_1, q_1 and p_2, q_2 be probability distributions over a random variable X and $\lambda \in (0, 1)$, define

$$p = \lambda p_1 + (1 - \lambda)p_2$$

$$q = \lambda q_1 + (1 - \lambda)q_2$$

$$\text{Then, } D_{KL}(p||q) \leq \lambda D_{KL}(p_1||q_1) + (1 - \lambda)D_{KL}(p_2||q_2).$$

To prove the properties we shall use the log-sum inequality.

Lemma 2.5. (*Log-sum Inequality*) If a_1, \dots, a_n and b_1, \dots, b_n are non-negative numbers, then

$$\sum_{i=1}^n a_i \ln \frac{a_i}{b_i} \geq \left(\sum_{i=1}^n a_i \right) \ln \left(\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \right)$$

with equality if and only if a_i and b_i are equal for all i .

Proof (*Property 1*). Using the log-sum inequality we can prove the positivity of the Kullback-Leibler divergence as follows:

$$D_{KL}(p||q) = \sum_{x \in \mathcal{X}} p(x) \ln \frac{p(x)}{q(x)} \geq \left(\sum_{x \in \mathcal{X}} p(x) \right) \left(\ln \frac{\sum_{x \in \mathcal{X}} p(x)}{\sum_{x \in \mathcal{X}} q(x)} \right) = 1 \ln 1 = 0 \quad (2.10)$$

□

Proof (*Property 2*). Let $a_1(x) = \lambda p_1(x)$, $a_2(x) = (1 - \lambda) p_2(x)$ and $b_1(x) = \lambda q_1(x)$, $b_2(x) = (1 - \lambda) q_2(x)$. Then:

$$\begin{aligned}
 D_{KL}(p||q) &= \sum_{x \in \mathcal{X}} (\lambda p_1(x) + (1 - \lambda) p_2(x)) \ln \frac{\lambda p_1(x) + (1 - \lambda) p_2(x)}{\lambda q_1(x) + (1 - \lambda) q_2(x)} \\
 &= \sum_{x \in \mathcal{X}} (a_1(x) + a_2(x)) \ln \frac{a_1(x) + a_2(x)}{b_1(x) + b_2(x)} \\
 &\leq \sum_{x \in \mathcal{X}} \left(a_1(x) \ln \frac{a_1(x)}{b_1(x)} + a_2(x) \ln \frac{a_2(x)}{b_2(x)} \right) \quad (\text{Using the logsum inequality}) \\
 &= \sum_{x \in \mathcal{X}} \left(\lambda p_1(x) \ln \frac{\lambda p_1(x)}{\lambda q_1(x)} + (1 - \lambda) p_2(x) \ln \frac{(1 - \lambda) p_2(x)}{(1 - \lambda) q_2(x)} \right) \\
 &= \lambda D(p_1||q_1) + (1 - \lambda) D(p_2||q_2)
 \end{aligned} \tag{2.11}$$

□

Remark

Notice that the Kullback-Leibler divergence is not a metric distance on the space of probability distributions.

In fact, a distance over the space X is a function $d : X \times X \longrightarrow \mathbb{R}$ that satisfy the following property for every x, y, z in X :

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$ (symmetry)
4. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality)

The Kullback-Leibler divergence $D_{KL}(\cdot||\cdot)$ satisfies the first two properties of distance but it is not symmetric - that is, $D_{KL}(p||q) \neq D_{KL}(q||p)$ - nor does it satisfy the triangle inequality.

Thus we can interpret the Kullback-Leibler divergence as a measure of the dissimilarity of the two distributions p and q [3].

2.3 Statistical Models

In this section are listed some definitions that will be useful to formalize in mathematical terms the problem of 3D reconstruction from experimental images.

Definition 2.6. *Let us define **random sample** the vector $\mathbf{X} = (X_1, \dots, X_n)$ of random variables representing the data.*

Definition 2.7. *Let $\mathbf{X} = (X_1, \dots, X_n)$ be random sample. A **statistical model** is defined as a family of densities of probability \mathbf{f} for \mathbf{X} , indexed by the parameter θ belonging to the parametric space $\Theta \subseteq \mathbb{R}^n$:*

$$\mathcal{F} = \{f(\cdot; \theta), \theta \in \Theta\} \quad (2.12)$$

Once obtained the data $\mathbf{x} = (x_1, \dots, x_n)$, may be useful to consider their density distribution as a function not of the data itself - which post-experimentally are not random variables but observed constants - but of the parameter θ [10].

Hence, let us consider the following:

Definition 2.8. *Given the observations $\mathbf{x} = (x_1, \dots, x_n)$, is defined **likelihood function**:*

$$\mathcal{L}(\theta; \mathbf{x}) = f(\mathbf{x}; \theta) \quad (2.13)$$

It is the density function of the observations with $\theta \in \Theta$

Definition 2.9. *Let us consider a parametric statistical model of parameter $\theta \in \Theta$. The **maximum likelihood estimate** is given by the $\hat{\theta}$ that maximize the likelihood function:*

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \mathcal{L}(\theta; \mathbf{x}) \quad (2.14)$$

2.4 Maximum Likelihood and Kullback-Leibler Divergence

Suppose that observations are generated from the distribution $p(x)$ whose functional form is unknown and we wish to model.

In particular, we want to find a description of $p(x)$ in terms of $q(x|\theta)$, which represents the probability of observing the realization x once the values of certain parameters $\theta = (\theta_1, \dots, \theta_{N_p})$ that define the model.

Let $\mathbf{X} = (X_1, \dots, X_N)$ be a vector of random variables is independent and identically distributed (i.i.d) from p and also that we have observed $\mathbf{x} = (x_1, \dots, x_N)$ realizations of such random sample. Then, the joint probability distribution is given by:

$$q(\mathbf{x}|\theta) = \prod_{i=1}^N q(x_i|\theta) \quad (2.15)$$

From the Bayes theorem we have that the probability of the parameters given the data, that is the posterior probability of the parameters is as follows:

$$q(\theta|\mathbf{x}) = \frac{q(\mathbf{x}|\theta) q(\theta)}{q(\mathbf{x})} \quad (2.16)$$

where:

- $q(\mathbf{x}|\theta)$ is the *likelihood function*;
- $q(\theta)$ is called a prior probabilities and expresses what is already known about the parameters before that the experiment is conducted. Here we assume absence of information about the value of the parameters. A standard choice, in this case, for the a prior distribution of parameters is to assume uniform probability.
- $q(\mathbf{x})$ is called evidence and it is a normalization constant, in fact:

$$\begin{aligned} q(\mathbf{x}) &= \int q(\mathbf{x})q(\theta|\mathbf{x}) d\theta \\ &= \int q(\theta)q(\mathbf{x}|\theta) d\theta \\ &= \int d\theta q(\theta)q(\mathbf{x}|\theta) = \text{const} \end{aligned}$$

where the last two steps are due to the fact that the parameters estimate is made at fixed model and therefore it constant.

In conclusion, assuming a uniform a prior distribution we simply have:

$$q(\theta|\mathbf{x}) \propto \mathcal{L}(\theta; \mathbf{x}) \quad (2.17)$$

which is that the posterior probability distribution of the parameters given the data is proportional to the likelihood function.

At this stage our objective is to find the combination of the parameters $\theta = (\theta_1 \dots, \theta_{N_p})$ that maximize the a posterior probability or equivalently the likelihood function.

As seen in (2.17) it is equivalent to find the maximum likelihood estimate.

Furthermore, considering that the argument maximizing a function is the same that maximize his logarithm since it is monotonous and increasing.

Therefore, we have the following:

$$\begin{aligned} \arg \max_{\theta \in \Theta} \log q(\theta|\mathbf{x}) &= \log \mathcal{L}(\theta; \mathbf{x}) \\ &\stackrel{(a)}{=} \log \prod_{i=1}^N q(x_i | \theta) \\ &\stackrel{(b)}{=} \sum_{i=1}^N \log q(x_i | \theta) \\ &\stackrel{(c)}{\approx} \int_{\mathcal{V}} p(x) \log q(x | \theta) dx \\ &= \int_{\mathcal{V}} p(x) \log \frac{q(x | \theta)}{p(x)} dx + \int_{\mathcal{V}} p(x) \log p(x) dx \\ &= - \int_{\mathcal{V}} p(x) \log \frac{p(x)}{q(x | \theta)} dx + \int_{\mathcal{V}} p(x) \log p(x) dx \\ &= -D_{KL}(p||q(\cdot | \theta)) + H(p) \end{aligned} \quad (2.18)$$

where (a) is given by (2.15); (b) by the property of the logarithm; (c) from the following observation.

Let us consider the expectation of a function under a probability distribution $p(x)$:

$$\mathbb{E}_p[f] = \int p(x)f(x) dx \quad (2.19)$$

If is given a finite number N of points drawn from the probability distribution $p(x)$, then the expectation can be approximated as a finite sum over these points:

$$\mathbb{E}_p[f] \approx \frac{1}{N} \sum_{i=1}^N f(x_n) \quad (2.20)$$

In conclusion, it should be noted that the second term of (2.18), the entropy, does not depend on θ .

Follows that, the problem to find the maximum likelihood estimate is equivalent to find the argument θ over the set of the possible parameters Θ minimizing the Kullback-Leibler divergence:

$$\arg \max_{\theta \in \Theta} \mathcal{L}(\theta; \mathbf{x}) = \arg \min_{\theta \in \Theta} D_{KL}(p||q(\cdot|\theta)). \quad (2.21)$$

2.5 Alternative Forms for the Kullback-Leibler Divergence

Since the Kullback-Leibler divergence is not symmetric, an optimization problem based on $D_{KL}(p||q)$ will exhibit a different behavior compared to one whose objective function is defined by $D_{KL}(q||p)$.

Then we can give the following definition:

Definition 2.10. *Let us define **M-projection** the distribution $q(x|\theta^*)$ where θ^* is such that:*

$$\arg \min_{\theta \in \Theta} D_{KL}(p||q(\cdot|\theta)) \quad (2.22)$$

Definition 2.11. *Let us define **I-projection** the distribution $q(x|\theta^*)$, where θ^* is such that:*

$$\arg \min_{\theta \in \Theta} D_{KL}(q(\cdot|\theta)||p) \quad (2.23)$$

The I-projection and the M-projection will exhibit different comportment as shown in Figure 2.1.

The M-projection will turn a distribution q which is a global approximation of the distribution p . This is due to the fact that values of x such that $p(x) > 0$ will have $q(x) > 0$, and typically $q(x)$ will stretch to cover all of $p(x)$ and will over-estimate the support of $p(x)$ [3] as in Figure 2.1 (a).

Conversely, the I-projection turn q will typically under-estimate the support of p and will lock onto one of its modes as show Figure 2.1 (b) and (c). This is due to the fact that $q(x)$ can be zero also when $p(x)$ have support different from zero. Hence, the I-projection turns a local approximation of p .

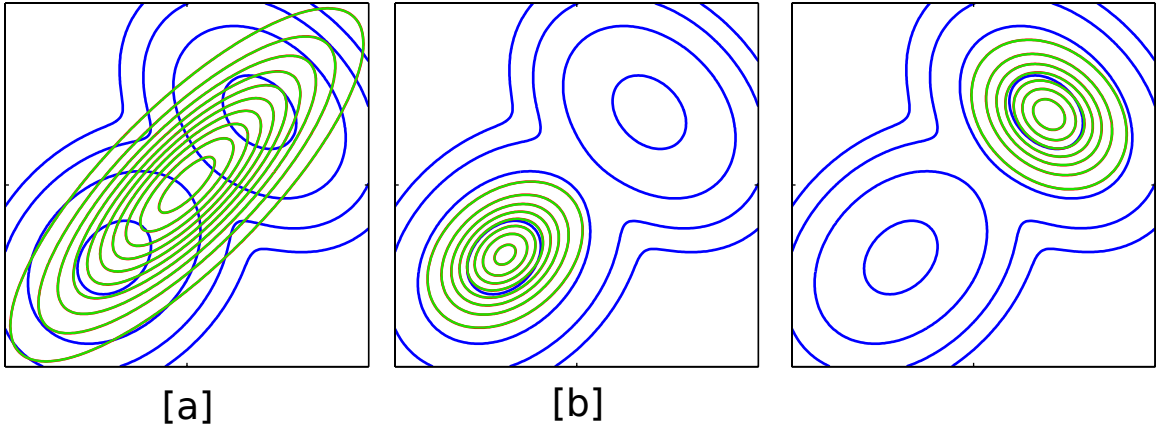


Figure 2.1: A comparison of the two alternative forms for the Kullback-Leibler divergence. (a) The blue contours show the distribution p given by a mixture of two Gaussians, and the green contours correspond to the single Gaussian distribution q that best approximates the first in the sense of minimizing the Kullback-Leibler divergence $KL(p||q)$. (b) As in (a) but now the green contours correspond to a Gaussian distribution q found by numerical minimization of the Kullback-Leibler divergence $KL(q||p)$. (c) As in (b) but showing a different local minimum of the Kullback-Leibler divergence.

Chapter 3

Deconvolution of Experimental Images

In modern cellular biology, images obtained from fluorescence microscopes are becoming a very important tool, as they are necessary for subsequent quantitative analysis.

Sub-cellular components and processes can now be visualized *in vivo*, both structurally and functionally. The observation of many biological processes relies on the ability to identify and locate specific proteins within their cellular environment. Cells are mostly transparent in their natural state, and the immense number of molecules constituting them are optically indistinguishable among each other. Consequently, the identification of a particular protein is a very complex task. However, a bright marker, attached to the protein of interest, could be a precise indicator of its position.

Many efforts have gone into finding suitable markers for this purpose, but only over the course of the past decade, with the advent of fluorescent proteins, this concept has been revolutionized. These biological markers have the crucial properties necessary for dynamic observations of living cells: they are essentially harmless to the organism, and can be attached to other proteins without impacting their function.

Therefore, the discovery of fluorescent markers suitable for in vivo use and the development of new techniques of microscopy have transformed the fluorescence microscope into a tool widely used in the field of quantitative analysis.

The 3D image of a biological specimen is acquired from a series of 2D images focusing the fluorescence microscope on different planes of the whole structure. In order to perform quantitative analysis on fluorescence images, a preliminary study of the image obtained with the microscope is necessary. In fact, microscopy images suffer of various distortions, due to multiple causes. The main reasons behind the alterations of the measured 3D image are the out-of-focus light, the spatial fluctuations of lighting in the sample, the Poisson noise caused by the emission of background photons.

The work presented in this chapter deals with the image restoration, which consists in the recovery of the image in its original form. We start from the definition of the mathematical model describing the physical process of the signal generation (Image). Subsequently, we present the images deconvolution as image restoration technique. This mathematical operation is applied to recover a signal degraded by a physical process. In particular, using an a known model (PSF) the deconvolution reconstructs the original characteristics of an image to increase its resolution, contrast and signal to noise ratio. Therefore, the images obtained experimentally with the fluorescence microscope, concerning the endosome and the Rab11, have been opportunely processed with this technique.

The aim is to remove possible distortions generated by the imaging system. At this stage, it is possible to perform the quantitative analysis presented in the next chapter.

3.1 Fluorescence Microscopy

The experimental images of the molecular sorting phenomenon on the endosome membrane have been obtained with a fluorescence confocal microscope (Leica SP8).

Fluorescence confocal microscopy is an optical technique widely used nowadays for the three-dimensional study of biological structures exploiting the physical phenomenon of fluorescence. It is the easiest way to observe the cellular structures: after labeled with a fluorescent dye, the biological specimen is illuminated at excitation wavelength and the fluorescence emission is used to form the image [17].

The three-dimensional images of the biological specimen in consideration are collected as a sequential optical sections. The *optical sectioning* of a biological specimen consists in a collection of images of parallel planes, moving the focus of the objective along an axis that generally coincides with the propagation axis of the light. To excite the molecules, confocal microscopy uses a very intense light source: the *laser*. The light emitted by the fluorophores excited by the laser is captured by the objective lens, it passes through the dichroic mirror and reaches the photomultiplier, which converts the light intensity into an electrical signal of proportional intensity (Figure 3.1).

Between the dichroic mirror and the photomultiplier, the light beam passes through the pinhole, which prevents most of the light coming from the out of focus areas to reach the photomultiplier.

To obtain the representation of an entire plane, the light beam is moved from point to point along the sample, so that the whole plane located at the desired depth is illuminated by the light beam according to a precise sequence. This process is called *scan*.

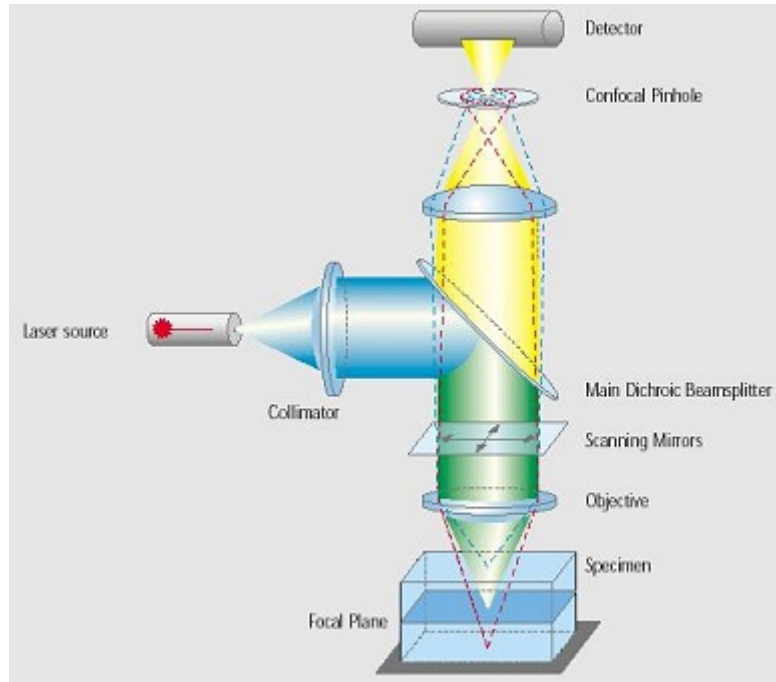


Figure 3.1: Basic confocal microscope structure

These values are used to construct the image: each point corresponds to a screen pixel and the light intensity of the point are represented by a grayscale value. The matching of all the pixels corresponding to points scanned by the laser beam in the sample give the final image. Moving the sample along the vertical axis after each scanning operation, it is possible to perform a series of successive ones corresponding to increasingly deep focal planes within the biological sample. These scans take the name of optical sections and their ordered overlap *Z-stack*. The *Z-stacks* allow to reconstruct the overall image of the entire scanned volume, in which all the planes are simultaneously in focus.

Hence we have information not only of the brightness of each point, but also related to its location in the biological specimen, ie its position in a plane and its depth. The location is defined by the three coordinates, (x, y, z) , called voxels, which are the three-dimensional equivalent of the pixels of a

two-dimensional image. An example of 3D acquisition of a cell that is created as a stack of 2D images by moving the focal plane through the specimen is shown in Figure 3.2.

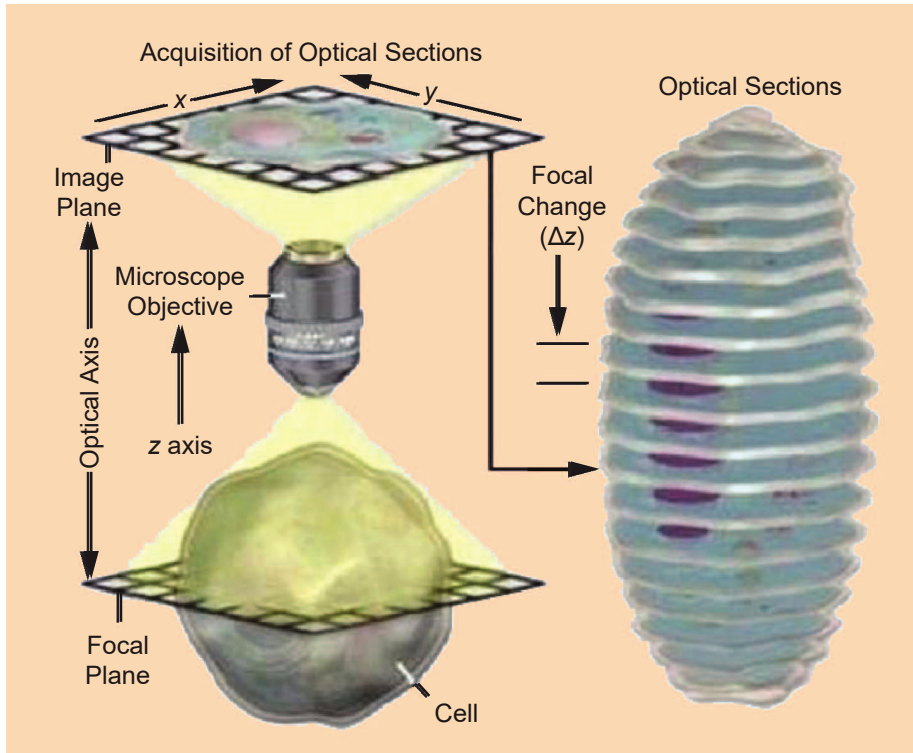


Figure 3.2: An example of the acquired image of a cell captured by a fluorescence microscope [18].

3.2 Image formation

The microscope is an imaging systems. As such, the image formation process can be described mathematically with the following equation.

$$g = T(f) \quad (3.1)$$

where g is the image recorded, f is the object under observation and T is the operator representing the imaging system, as shown in Figure 3.3.

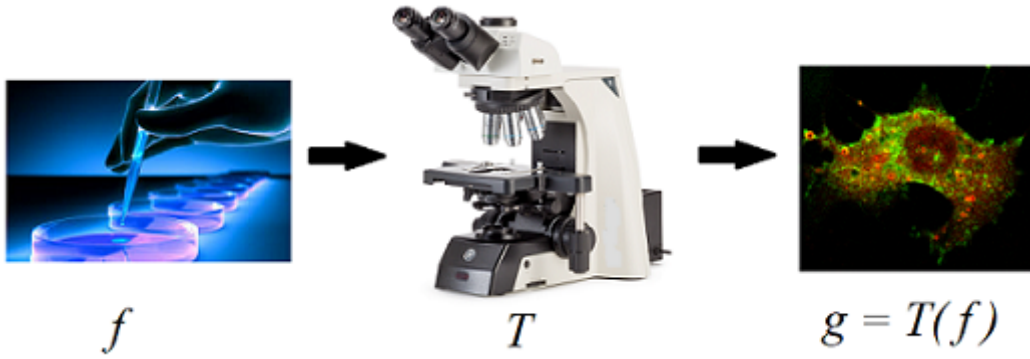


Figure 3.3: An example of the acquired image g as a result of the processing of the object f with the imaging system T .

In systems such as fluorescent microscopes, the image formation process is *linear* and *shift-invariant*. These two properties hold being most part of the process linear and being the response of the system independent from the position. Therefore, it is described using the Linear Shift-Invariant system theory.

Mathematically a point object is represented by a *Dirac delta distribution*.

Applying an useful property of the delta distribution we can write the object f as:

$$f(x, y, z) = \int \int \int_{\mathbb{R}^3} f(\varepsilon, \nu, \varsigma) \delta(x - \varepsilon, y - \nu, z - \varsigma) d\varepsilon d\nu d\varsigma.$$

In signal processing *impulse response* of a linear system is defined as the output of the system when an impulsive input is applied, i.e. a delta function.

Similarly, the *Point Spread Function* (PSF) is defined as the impulse response of an imaging system, which models how the imaging system represents a point object. In formula:

$$h = T(\delta).$$

In conclusion we can write the imaging equation in current context as:

$$g(x, y, z) = \int \int \int_{\mathbb{R}^3} f(\varepsilon, \nu, \varsigma) h(x - \varepsilon, y - \nu, z - \varsigma) d\varepsilon d\nu d\varsigma \quad (3.2)$$

The last relation is defined as the *convolution equation* between the original object and the Point Spread Function of the imaging system. It can also be rewrite as:

$$g = f * h \quad (3.3)$$

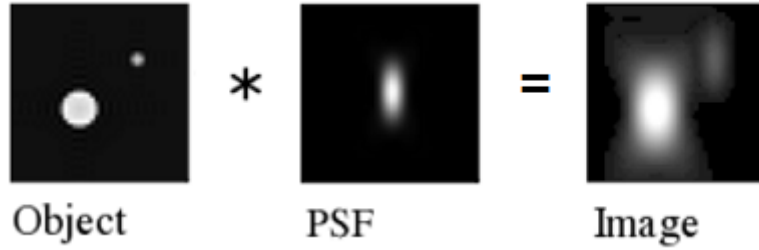


Figure 3.4: Convolution operation.

Sampling and Nyquist criterion

In order to be processed by a calculator a signal, such as an image, must be discrete.

Sampling is the process of converting a continuous signal into a numeric sequence [23]. The procedure is also called analog-to-digital conversion, or simply digitizing.

The *sampling density* is the number of recorded samples (i.e voxel which is the 3D equivalent of a pixel) per unit distance when converting from an analog signal to digital.

Thus, the larger is the sampling distance, i.e. the size of one voxel, the smaller is the sampling density.

The sampling density is a microscopic parameter describing the conditions of the image acquisition. It establishes a direct connection between one voxel in the Z-stacks and a real volume in the physical space.

The *Nyquist criterion* determines the minimal sampling density needed to capture all information from the microscope into the image. When the sampling distance is larger than the Nyquist distance, information about the image is lost.

The Nyquist critical sampling distance for a conventional fluorescence confocal microscope is given by:

$$\Delta_{x,y} = \frac{\lambda_{em}}{8n \sin \alpha}$$

$$\Delta_z = \frac{\lambda_{em}}{4n(1 - \cos \alpha)}$$

with n the Lens Refractive Index (usually 1,515 for immersion oil), α the half-aperture angle of the objective of the microscope, λ_{em} the Emission Wavelength of the fluorophore that we are observing, and $\Delta_{x,y}$, Δ_z the sampling distances in the lateral and axial direction respectively.

In the equation above α can be computed from $\alpha = \arcsin \frac{NA}{n}$ with NA the Numerical Aperture of the objective.

In conclusion, we have that a Z-stack is represented by a 3D matrix containing intensity values. Each element of this matrix represents a “volume

element,” or “voxel” whose physical dimensions are given by Δ_x , Δ_y and Δ_z calculated with the Nyquist criterion (i.e. Figure 3.5). Moreover, each intensity value represents a measured physical quantity of the object at the sample point.

Since all physical instruments have finite resolution, samples correspond with the average of the physical quantity over a volume element.

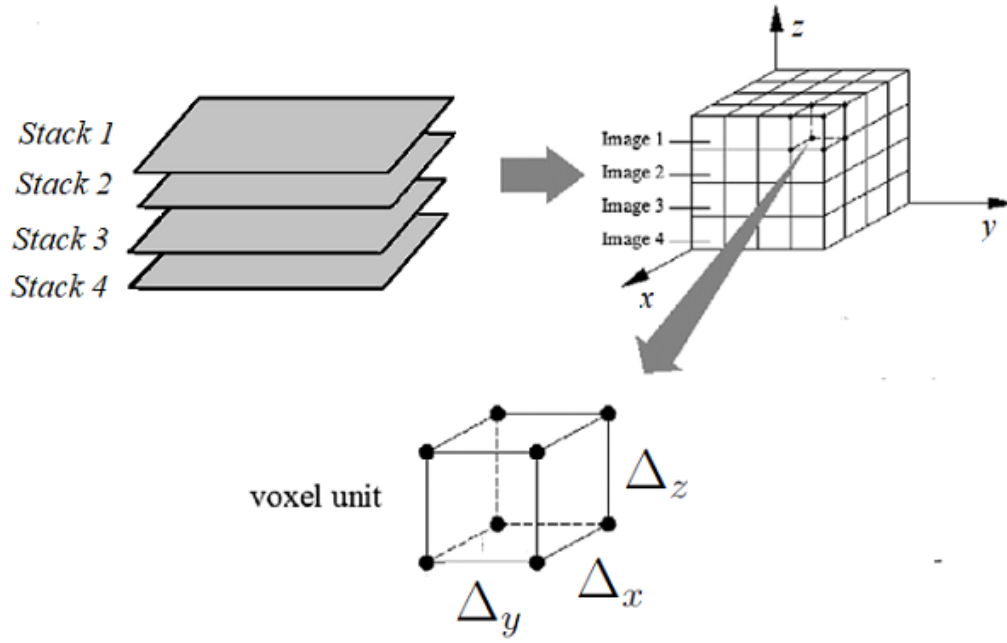


Figure 3.5: Graphical representation of a Z-stack as 3D matrix.

The Point Spread Function

The Point Spread Function (PSF) is the three-dimensional diffraction pattern of light emitted from an infinitely small point source in the specimen. As already seen in the previous section, it is considered the fundamental unit of an image in theoretical models of image formation. Therefore, the PSF is crucial to perform the deconvolution on experimental images. This function might be calculated theoretically for the used imaging arrangement, or alternatively might be directly measured. Both approaches are widely used in biological studies. The PSF can be experimentally measured using a fluorescent bead embedded in a gel that approximates an infinitely small point object in a homogeneous medium.

Nevertheless, these procedures require an high cost both in terms of time and economical resources. In fact, more experiments of images acquisition are necessary for a precise evaluation. Due to these drawbacks and to the lack of some equipment, we adopt a theoretically derived PSF. Indeed, the literature on PSF modeling is extensive and characterized by a large number of approximations with various degrees of accuracy.

A particular challenge in representing the PSF of a microscope is the reduced amount of information related to the exact design of the objectives.

As matter of fact, the only parameters usually known are the overall optical characteristics of the objective (i.e., magnification, numerical aperture, etc.) and the experimental conditions for which it is designed (i.e. refractive index of the immersion medium, working distance). Therefore, it is desirable a formulation of the PSF that relies only on these known parameters, accordingly treating the objective as a black box.

A significant contribution in this direction was made by Gibson and Lanni [11], Wolf and Richards [24, 15] and Born & Wolf [4].

The PSF of the first two models is shift invariant only in the lateral directions, while the last has this property also in axial direction. Hence, we adopt the Born & Wolf PSF model, which is defined by the following

equation:

$$h(x, y, z) = \left| C \int_0^1 J_0 \left[k \frac{NA}{n_i} \sqrt{x^2 + y^2} \rho \right]^{-\frac{1}{2} j k \rho^2 z \left(\frac{NA}{n_i} \right)^2} \rho d\rho \right|^2 \quad (3.4)$$

where:

- (x, y, z) denotes the coordinate of a point in the image plane.
- J_0 is the Bessel function of first kind of order zero.
- NA is the Numerical Aperture of the objective lens. This value is given by $NA = n_i \sin(\theta)$ where θ is one half of the angular aperture of the lens.
- n_i is the actual value of the refractive index of the immersion layer (i.e. the $n_{oil} = 1,515$).
- C is the normalization constant.
- k is the wavenumber in vacuum of the emitted light, $k = \frac{2\pi}{\lambda}$.

In Appendix B it is reported a guide which explains the procedure to generate the PSF.

Some examples of Z-stacks obtained with this model are presented in the Figure 3.6. Subsequently, these same Z-stacks are used to perform the deconvolution of the experimental images related to the molecular sorting.

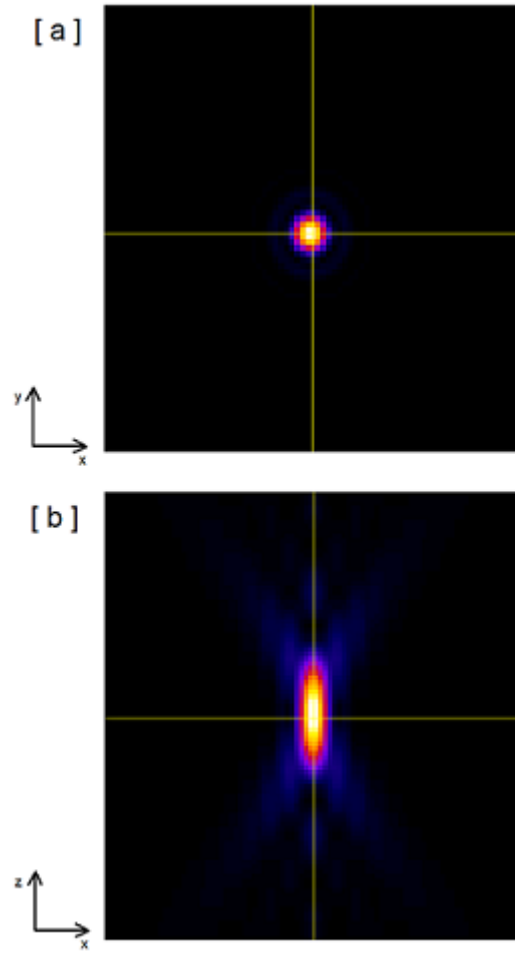


Figure 3.6: Theoretical PSF, generated with Born & Wolf model. [a] shown the plane xy and [b] xz

3.3 Deconvolution

Image Deconvolution is the mathematical operation aimed to reverse the effect of the convolution. Hence, it is used to reduce the distortions present in the Z-stacks due to the imaging system, restoring the effective object representation.

In modern biological research, deconvolution is becoming not only a fundamental, but almost a standard image processing step when analyzing small relevant details. For example, deconvolution can reveal hidden pertinent structures with the amelioration of contrast.

In Fluorescence Microscopy, much of the distortion comes from two basic physical limitations. First, the presence of out-of-focus light from multiple planes causing the blur effect. In an ideal microscope, only light from the focal plane would be allowed to reach the detector producing a clear image of the plane of the sample the microscope is focused on.

Unfortunately a microscope is not enough specific and also the light from sources outside to the focal plane reaches the detector; in a thick specimen there may be a significant amount of material, and so spurious signal.

As such, fluorescent images appear soft and diffused, losing much of the finer detail.

Secondly, the pinhole drastically reduces the amount of light detected by the photomultiplier and produces a grainy appearance in the image owing to the reduction in the signal compared to random variations recorded by the detector, causing then Poisson noise. Therefore, the acquired images differ from the true object since they are unavoidably affected by noise and convolution effects due to the optical system.

Hence, more realistic is to model the acquired image as follows:

$$g(x, y, z) = f(x, y, z) * h(x, y, z) + \epsilon(x, y, z) \quad (3.5)$$

Where ϵ represents the noise recorded.

One way to improve the images is numerically inverting the blurring and the noise processes applying the deconvolution, that mitigates the distortion

created by the microscope [18, 17] .

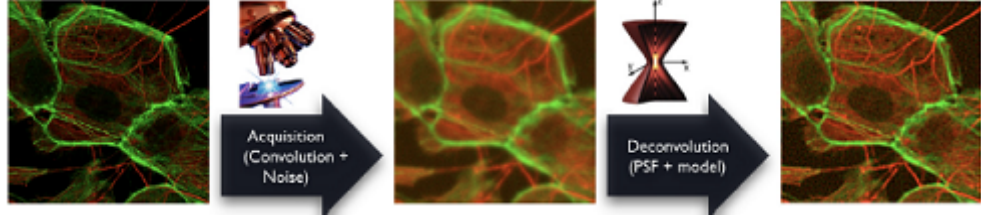


Figure 3.7: Principle of the deconvolution [17].

Several deconvolution methods have been proposed for 3D microscopy: Naive inverse filtering [17], is the simplest approach to deconvolution and consists in least-squares method; Tikhonov–Miller [12], Richardson–Lucy [7, 16] algorithms and many others achieve out-of-focus deblurring by iterative deconvolution.

A brief user manual of the software in which it is possible to perform the 3D deconvolution is shown in Appendix B.

The selection of the appropriate model depends on the type of image that we are analyze. As for our Z-stacks we choose to use the Richardson–Lucy method that give good results. Deconvoluted images are shown in the last chapter.

Richardson–Lucy Deconvolution Algorithm

The goal of the Richardson–Lucy (RL) deconvolution algorithm is therefore to obtain iteratively an estimate \hat{f} of the image, reversing the effect of noise and the blur created by the PSF , using the a prior knowledge of the image recorded g and the point spread function h .

The RL method is a maximum likelihood approach. The quantum nature of light leads to a Poisson modeling both the signal emitted by the object and the noise [18], as follows:

$$g(\mathbf{x}) = P(f(\mathbf{x}) * h(\mathbf{x})) + P(\epsilon(\mathbf{x})), \quad \text{with } \mathbf{x} = (x, y, z) \in \mathbb{R}^3 \quad (3.6)$$

where $g(\mathbf{x})$ is the number of photons measured in the detector in position \mathbf{x} , P indicate a Poisson distribution and $\epsilon(\mathbf{x})$ is the noise.

Both $P(f(\mathbf{x}) * h(\mathbf{x}))$ and $P(\epsilon(\mathbf{x}))$ are independent Poisson random variables and, hence, the measured output is a Poisson random variable $P(f(\mathbf{x}) * h(\mathbf{x}) + \epsilon(\mathbf{x}))$.

In order to obtain the maximum-likelihood estimate of f , the likelihood function for the Poisson imaging model (3.6) should be maximized. But maximizing the likelihood is equivalent to maximizing the log-likelihood, given by the functional:

$$J(f) = \sum_{\mathbf{x}} \{g(\mathbf{x}) \log f(\mathbf{x}) * h(\mathbf{x}) - f(\mathbf{x}) * h(\mathbf{x})\}.$$

A multiplicative gradient-based iterative search algorithm is used to maximize $J(f)$ and yields the following iterative formula:

$$\hat{f}_{i+1}(\mathbf{x}) = \left\{ \left[\frac{g(\mathbf{x})}{\hat{f}_i(\mathbf{x}) * h(\mathbf{x})} \right] * h(-\mathbf{x}) \right\} \times \hat{f}_i(\mathbf{x}),$$

where \hat{f}_i is the estimation of the object at i-th iteration.

Chapter 4

3D Reconstruction of Experimental Images

We aim to reconstruct the 3D volume of the endosome and identify the Rab11 protein domains over the membrane from experimental images.

The main problem of the Z-stacks is the poor spatial resolution. Furthermore, each one is composed by a limited amount of 2D images.

To cope with this problem, a 3D reconstruction procedure in this chapter have been developed. In particular, the crucial idea is to use the a priori knowledge about endosomal and protein domain morphologies.

Furthermore, information theory and statistics tools, introduced in the preceding chapter, are used to deal with the analysis.

4.1 Data Interpretation

In this section, we present the analytical formulation of data transformation. The aim is to prepare them for the formulation of the pattern recognition problem.

Let $\mathcal{V} \subseteq \mathbb{R}^3$ be the portion of space observed by the microscope.

In this volume a distribution of fluorescent molecules allows to detect the object under investigation.

The fluorescence collected by the microscope can be represented with the following continuous function:

$$\begin{aligned} f : \mathcal{V} \subseteq \mathbb{R}^3 &\longrightarrow \mathbb{R} \\ x &\longmapsto f(x) \end{aligned} \quad (4.1)$$

Hence, the empirical probability distribution of fluorescence is given by:

$$\begin{aligned} p : \mathcal{V} \subseteq \mathbb{R}^3 &\longrightarrow [0, 1] \\ x &\longmapsto p(x) = \frac{f(x)}{\int_{\mathcal{V}} f(x) dx} \end{aligned} \quad (4.2)$$

Consequently, $p(x)dx$ is interpretable as the probability to detect a fluorescent molecule in the element of volume dx centered in x .

As stated in the definition (4.1), the distribution describing the specimen in the volume $\mathcal{V} \subseteq \mathbb{R}^3$ is a continuous function.

However, after the sampling procedures, the signal become discrete: the data (Z-stacks) are 3D matrix reporting intensity values. Each value represents the mean fluorescence in the correspondent volume element (voxel) of physical sizes $\Delta x, \Delta y, \Delta z$ evaluated with the Nyquist criteria of sampling.

Hence, let $\tilde{\mathcal{V}}$ be the set of the voxels as reported in the following definition.

$$\tilde{\mathcal{V}} = \{v = (i, j, k), i \in \{0, \dots, N_x - 1\}, j \in \{0, \dots, N_y - 1\}, k \in \{0, \dots, N_z - 1\}\}$$

with N_x, N_y and N_z the number of pixels in the three directions.

The fluorescence can be defined as discrete function as follows:

$$\begin{aligned} \tilde{f} : \tilde{\mathcal{V}} &\longrightarrow \mathbb{R} \\ v &\longmapsto \tilde{f}(v) \end{aligned} \quad (4.3)$$

where:

$$\tilde{f}(i, j, k) = \frac{1}{\Delta x \Delta y \Delta z} \int_{k\Delta z}^{(k+1)\Delta z} \int_{j\Delta y}^{(j+1)\Delta y} \int_{i\Delta x}^{(i+1)\Delta x} f(x, y, z) dx dy dz \quad (4.4)$$

Therefore, we can define the empiric mass probability distribution of fluorescence for each point $v = (i, j, k) \in \tilde{\mathcal{V}}$, as reported below.

$$\begin{aligned} \tilde{p} : \tilde{\mathcal{V}} &\longrightarrow [0, 1] \\ v &\longmapsto \tilde{p}(v) = \frac{\tilde{f}(v)}{\sum_{v \in \tilde{\mathcal{V}}} \tilde{f}(v)} \end{aligned} \quad (4.5)$$

Being the function representative of a probability distribution the following relations hold:

$$\sum_{v \in \tilde{\mathcal{V}}} \tilde{p}(v) = 1$$

and

$$0 \leq \tilde{p}(v) \leq 1 \quad \forall v \in \tilde{\mathcal{V}}.$$

4.2 Structure Recognition

In this section, we present the analytical formulation of the optimization problems used to recognize the structures of interest.

As described in the second chapter, each Z-stack is composed by two channels. The first contains the signal related to Fyve2x, which is the marker of the endosome membrane. Instead, the signal of the second channel is referred to Gfp which labels the protein Rab11.

Starting from the fluorescence values of the two channels we define their empirical distributions, using the analytical formulations introduced in the previous paragraph.

At this point is useful to define the notation adopted along the whole dissertation to distinguish among the probability distribution of the different channels. In particular, the one related to Fyve2x is indicated with p_f , while p_g refers to Gfp.

4.2.1 Recognition of Endosome Structure

In this subsection we explain the procedures adopted, analyzing the first channel data, to identify the endosome structure. To achieve our objective we exploit the knowledge of the membrane geometrical properties. In fact, starting from the observations of the Z-stacks we can appreciate the endosome spherical symmetry, shown in Figure 4.1.

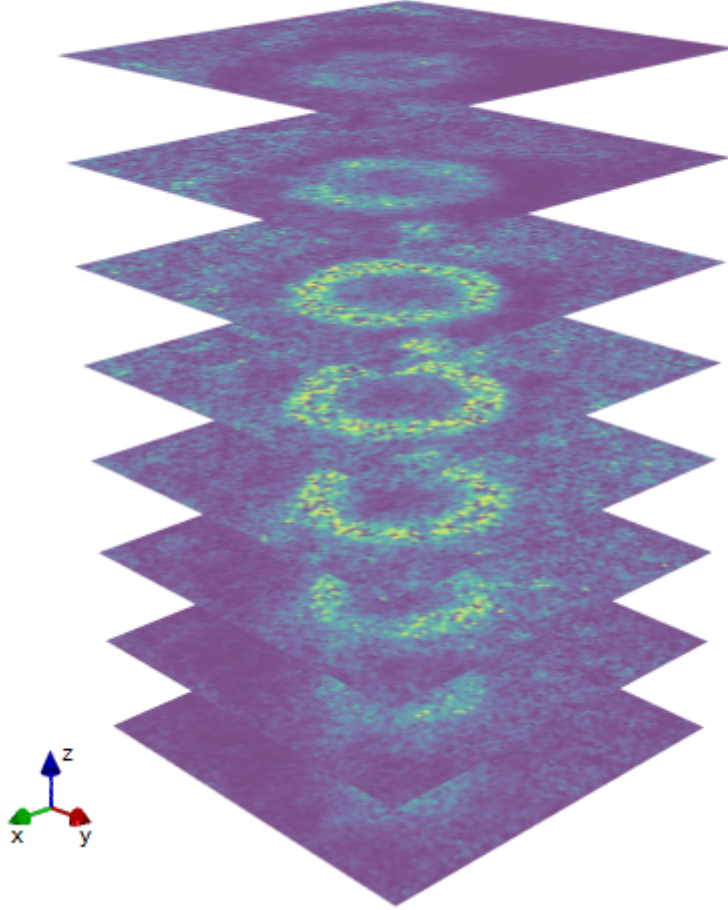


Figure 4.1: Experimental Z-stack of Fyve2x.

Consequently, we approximate the function p_f with a parametric distribution $q_f(\cdot|\theta)$, with θ belonging to the parametric space Θ . This parametric

function should be of the same type of the one reported in Figure 4.2.

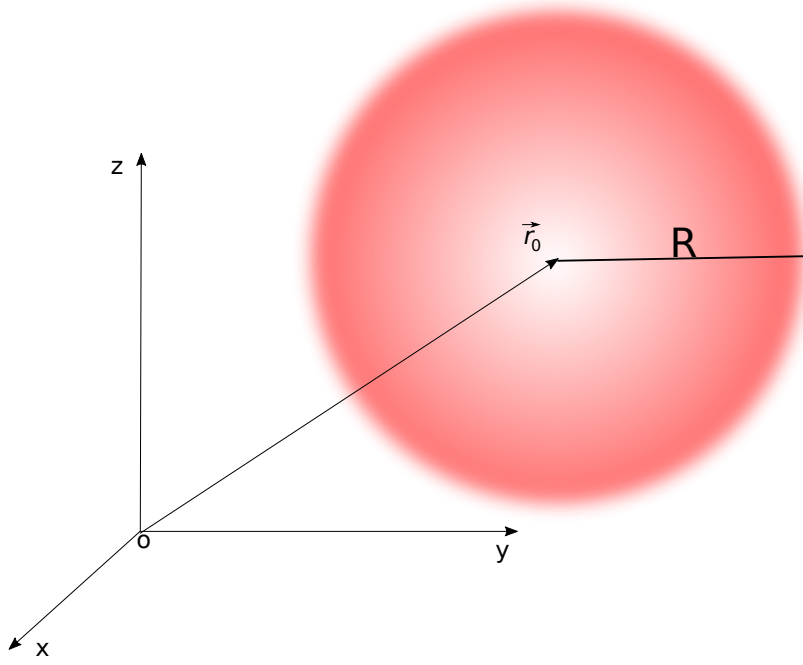


Figure 4.2: Ideal structure of the endosome membrane

In fact, being spherical the geometry of the membrane, $q_f(\cdot|\theta)$ is defined as an ideal distribution with parameters $\theta = (r_{0x}, r_{0y}, r_{0z}, R, \sigma) \in \mathbb{R}^5$, where:

- $\vec{r}_0 = (r_{0x}, r_{0y}, r_{0z})' \in \mathbb{R}^3$ is the vector identifying the position of sphere center;
- $R \in \mathbb{R}$ is the sphere radius;
- $\sigma \in \mathbb{R}$ identifies the dispersion of the fluorescence at different membrane radius.

Hence, the parametric distribution has the following explicit form:

$$q_f(v|\theta = (\vec{r}_0, R, \sigma)) = \frac{1}{C_f} \exp \left\{ -\frac{1}{2} \left(\frac{\|\vec{r}_0 - v\| - R}{\sigma} \right)^2 \right\}, \quad \forall v \in \tilde{\mathcal{V}} \quad (4.6)$$

where C_f is the normalization constant: $C_f = \sum_{v \in \tilde{\mathcal{V}}} q_f(v|\theta)$.

At this stage, the main issue is to find the proper value of the tuning parameters allowing the best fitting of the empirical distribution.

As stated in the Paragraph 2.4, having available the observations drawn from p_f , the value of the adjustable parameters could be derived with the maximum likelihood estimation.

However, as starting data, we have the empirical distribution rather than the observations. As demonstrated with the Equation (2.18), when a sufficient number of observations are available and under certain conditions, the maximum likelihood estimation is equivalent to find the argument minimizing the Kullback-Laibler divergence from the model to the empirical distribution.

The optimal fit of the data is given by the density function $q_f(\cdot|\theta^*)$, where θ^* is the solution of the following optimization problem:

$$\begin{aligned}
 & \arg \min_{\theta=(r_{0x}, r_{0y}, r_{0z}, R, \sigma) \in \mathbb{R}^5} D_{KL}(p_f(\cdot)||q_f(\cdot|\theta)) = \sum_{v \in \tilde{\mathcal{V}}} p_f(v) \log \left(\frac{p_f(v)}{q_f(v|\theta)} \right) \\
 & \text{subject to} \quad 0 \leq r_{0x} \leq N_x, \quad (C1), \\
 & \quad \quad \quad 0 \leq r_{0y} \leq N_y, \quad (C2), \\
 & \quad \quad \quad 0 \leq r_{0z} \leq N_z, \quad (C3), \\
 & \quad \quad \quad 0 \leq R \leq R_{max}, \quad (C4), \\
 & \quad \quad \quad 0 \leq \sigma \leq \sigma_{max} \quad (C5)
 \end{aligned} \tag{4.7}$$

As it is possible to observe in the optimization problem definition, we only impose simple bounds on the variables to constrain the solution. The three constraints (C1), (C2) and (C3) force the center of the endosome to be contained in the observation volume $\tilde{\mathcal{V}}$. On the other hand, (C4) and (C5), bound both radius and deviation to be positive and within the maximum limits.

4.2.2 Recognition of Protein-Enriched Membrane Domains

Following a procedure similar to the one of the previous paragraph, in this subsection we describe the logic used to identify the Rab11 domains, starting from the intensity values of the second channel of the Z-stacks.

Biological studies related to the molecular sorting highlighted the occurrence of the phenomenon on the membrane.

Hence, defining as $f_g(v)$ with $v \in \tilde{\mathcal{V}}$ the discrete function identifying the fluorescence distribution of Gfp in the Z-stacks, we eliminate the fluorescence values referred to zones outside the membrane.

To achieve this objective, we use the optimal solution of the problem (4.7).

In fact, being $\theta^* = (\tilde{r}_0^*, R^*, \sigma^*)$ the theoretical solution of the problem, $q_f(\cdot|\theta^*)$ is the function reconstructing the endosome membrane from the experimental images.

Hence, once known its functional form, we are able to identify with an indicator function the region referred to the membrane. Subsequently, using this function, we can localize the protein in the region of interest.

Analytically, we multiply f_g by the indicator function localizing the membrane, as reported in the following equation:

$$f_g^{(1)}(v) = f_g(v) \chi_{membrane}(v) \quad \forall v \in \tilde{\mathcal{V}} \quad (4.8)$$

where $\forall v \in \tilde{\mathcal{V}}$ the following definition holds:

$$\chi_{membrane}(v) := \begin{cases} 1 & \text{if } q_f(v|\theta^*) > \bar{q}_f(\cdot|\theta^*), \\ 0 & \text{if } otherwise. \end{cases} \quad (4.9)$$

in which $\bar{q}_f(\cdot|\theta^*)$ indicates the sample mean: $\bar{q}_f(\cdot|\theta^*) = \frac{1}{|\tilde{\mathcal{V}}|} \sum_{v \in \tilde{\mathcal{V}}} q_f(v|\theta^*)$.

Notice that the applied threshold have been selected heuristically.

Once normalized, the intensity function $f_g^{(1)}$ can be interpreted as an empirical distribution of Gfp molecules in the volume $\tilde{\mathcal{V}}$, and indicated as p_g .

Then, as for the first channel, we define a proper parametric function to fit the domains of Rab11. In this case, we exploit the a priori knowledge about the shape of the Rab11 domains, derived by biological and theoretical observations mentioned in Chapter 1.

In fact, it has been shown that the domains form is a circular calotte localized on the endosome membrane.

As stated in the introduction, our final purpose is to reconstruct the 3D image. To accomplish this task we need a parametric function, defined as the intersection between the membrane and a cone, as shown in Figure 4.3.

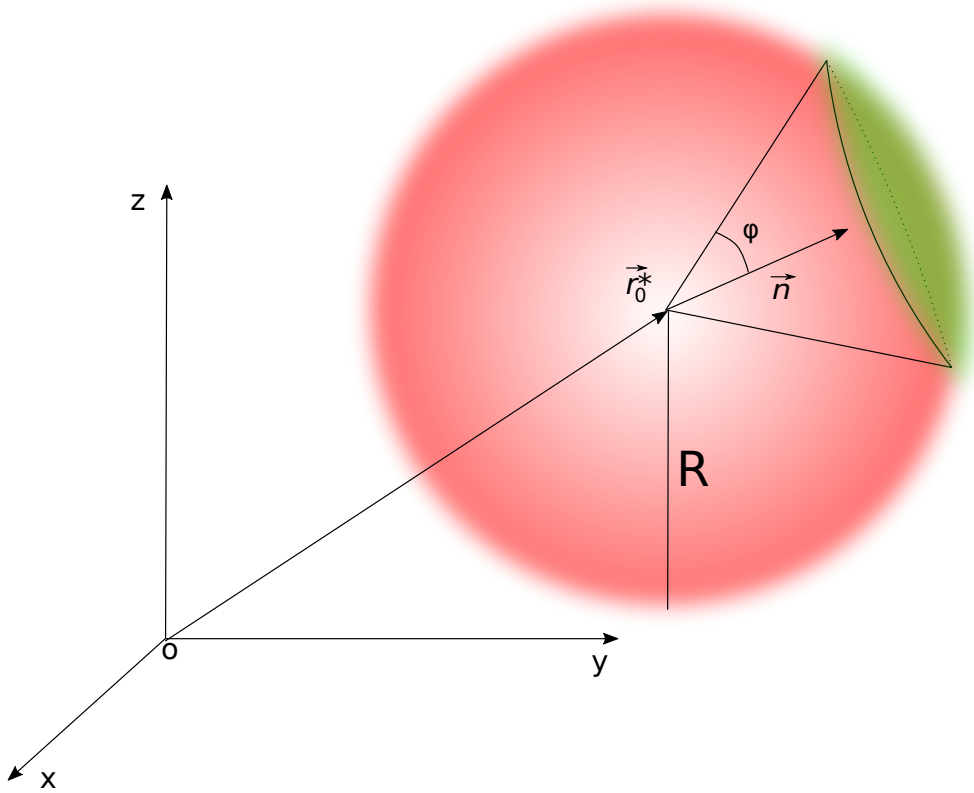


Figure 4.3: Ideal structure of the Rab11 protein agglomerate.

This implies that the vector of parameters identifying the model distribution is $\theta = (n_x, n_y, n_z, \varphi) \in \mathbb{R}^4$.

In particular:

- $\hat{n} = (n_x, n_y, n_z)' \in \mathbb{R}^3$ is the unit vector applied to the center \vec{r}_0^* of the sphere defining the orientation of the cone;
- φ is the cone angle.

A graphical visualization of the parameters is presented in Figure 4.3.

In order to construct the model distribution, we define the characteristic function identifying the cone as:

$$\chi_{cone}(v; \hat{n}, \varphi) := \frac{1}{\pi} \arctan(\hat{n} \cdot \hat{u} - \cos(\varphi)) + \frac{1}{2} \quad (4.10)$$

with

$$\hat{u} = \frac{v - \vec{r}_0^*}{\|v - \vec{r}_0^*\|} \quad (4.11)$$

Hence, we can define the parametric distribution as follows:

$$q_g(v | \theta = (\hat{n}, \varphi)) = \frac{1}{C_g} \chi_{membrane}(v) \chi_{cone}(v; \hat{n}, \varphi), \quad \forall v \in \tilde{\mathcal{V}} \quad (4.12)$$

where C_g is the normalization constant $C_g = \sum_{v \in \tilde{\mathcal{V}}} q_g(v | \theta)$ and $\chi_{membrane}$ is the characteristic function defined by the Equation (4.9).

At this point, as for the previous case, we need to find the proper tuning parameters of the distribution which better identify the domain.

The optimal solution $q_g(\cdot|\theta^*)$ is given by the argument θ^* minimizing the Kullback-Laibler divergence from the empirical distribution to the model, as follow:

$$\begin{aligned}
 \arg \min_{\theta=(n_x, n_y, n_z, \varphi) \in \mathbb{R}^4} \quad & D_{KL}(q_g(\cdot|\theta)||p_g(\cdot)) = \sum_{v \in \tilde{\mathcal{V}}} q_g(v|\theta) \log \left(\frac{q_g(v|\theta)}{p_g(v)} \right) \\
 \text{subject to} \quad & 0 \leq n_x \leq N_x, \quad (C1), \\
 & 0 \leq n_y \leq N_y, \quad (C2), \\
 & 0 \leq n_z \leq N_z, \quad (C3), \\
 & 0 \leq \varphi < 2\pi, \quad (C4)
 \end{aligned} \tag{4.13}$$

We impose simple bounds on the variables to constrain the solution. The three constraints (C1), (C2) e (C3) force the unit vector \hat{n} to be contained in the observation volume $\tilde{\mathcal{V}}$, while (C4) bound the angle to be in $[0, 2\pi)$.

4.2.3 Reconstruction Procedure

In this section the reconstruction algorithm is explained. The procedure is summarized in the following steps:

1. Evaluate the empirical distribution p_f
2. Define the distribution model of Fyve2x:

$$q_f(\cdot|\theta_f) \text{ with } \theta_f = (\vec{r}_0, R, \sigma)$$

3. Define the KL divergence $D_{KL}(p_f||q_f(\cdot|\theta_f))$
4. Solve the optimization problem

$$\arg \min_{\theta_f \in \mathbb{R}^5} D_{KL}(p_f(\cdot)||q_f(\cdot|\theta_f)) \mapsto \theta_f^*$$

5. Evaluate p_g (using the optimal distribution $q_f(\cdot|\theta_f^*)$) and set $p_g^{(0)} \leftarrow p_g$
6. Define the model distribution of Gfp (using the optimal distribution $q_f(\cdot|\theta_f^*)$):

$$q_g(\cdot|\theta_g) \text{ with } \theta_g = (\hat{n}, \varphi)$$

7. $i \leftarrow 0$
8. While $\varphi_i^* < \varphi_{min}$:

- 8.1. Solve the optimization problem

$$\arg \min_{\theta_g^{(i)} \in \mathbb{R}^4} D_{KL}(q_g(\cdot|\theta_g^{(i)})||p_g(\cdot)) \mapsto \theta_g^{(i)*}$$

- 8.2. Save the current solution in a list:

$$Domains_list[i] \leftarrow \theta_g^{(i)*}$$

- 8.3. Eliminate the current domain from the dataset and define the new distribution $p_g^{(i)}$

- 8.3. $i \leftarrow i + 1$

According to the observations presented in the Paragraph 2.5, is important to underline that two different optimization problem has been defined in the points 4 and 8.

Regarding the point 4, from the analytical formulation of the divergence $D_{KL}(p_f||q_f(\cdot|\theta_f))$, it follows that, if the empirical distribution p_f is different from zero, consequently also the model distribution $q_f(\cdot|\theta_f)$ is forced to be positive. Accordingly, solving the minimization problem, the optimal model distribution will turns a global approximation of p_f .

Contrarily, the definition of the diverge $D_{KL}(q_g(\cdot|\theta_g)||p_g)$ is adopted to define the minimization problem of the point 8. In this case, even if the empirical distribution p_g positive, the model distribution $q_g(\cdot|\theta_g)$ can be null.

As a matter of fact, if the p_g function is multimodal, the solution of the optimization problem 8.1 generates a model distribution, which gives a local approximation of the empirical one. Consequently, in this case, there is a value minimizing the operator for each of the mode of the empirical distribution. Our objective is to find all the models satisfying this criteria and consequently identifying the Rab11 domains.

The described process, as it can be noticed looking at the structure of the point 8, is iterative. At each iteration, we eliminate the fluorescence data referred to the domain identified using the optimal current solution. The iteration ends when the current optimal angle φ_i^* is below a certain threshold, selected based on the experimental images resolution.

It should be underlined that the proposed methodology is robust and general. In fact, it can be applied to reconstruct any number of domains presented in the experimental data.

Remark:

The explained procedure is the one leading to the best results. However, several strategies for domains reconstruction have been tested.

For example, another approach is to define the model function q_g as a linear combination of the used one. The main limitation of this approach is the huge set of parameters needed. In fact, in this way at each iterative step, the number of optimization parameters would have grow with a step of 4 at each iteration, since each cone is identified by a direction in the space and an angle.

Another idea was to use a mixture of trivariate normal distributions. However, also in this case, the main issue was related to the large amount of parameters needed to perform the estimation. In fact, in this way at each iterative step, the number of optimization parameters would have grow with a step of 9, since that to define a single trivariate normal distributions the means vector and the covariance matrix are needed.

For this reason we choose the the cone, which is the function with the lowest number of parameters and the elimination approach, since in this way at each iteration the number of parameters remains constant.

The code implementing our reconstruction algorithm has been developed in Python and is shown in the Appendix C. To find the solution of the optimization problems the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm has been adopted. This is a numerical optimization belonging to the family of quasi-Newton methods, used to solve large nonlinear optimization problems, which variables are constrained by simple bounds [5, 26].

Chapter 5

Results

In this chapter we report the main results obtained following the reconstruction process, presenting the main phases in three different paragraphs. In particular, the first shows the images obtained applying the 3D deconvolution algorithm, explained in the Paragraph 3.3. Instead, the second and third paragraphs report the results of the reconstruction referred respectively to the endosome membrane and of Rab11 domains.

5.1 Image Deconvolution

The results commented in this paragraph refer to the deconvolution of the experimental data. In Fig. 5.1 is shown the comparison among original images obtained with the microscope related to the endosome membrane and deconvoluted ones. The same comparison is reported in Fig. 5.2 for data relative to the protein. In both cases, the deconvolution registers good performances. In fact, as matter of example, in Fig. 5.3 we show the images before and after the deconvolution for both channels at a specific deep of focus. Comparing the intensity profile of the images, is interesting to observe, for deconvoluted data, the better identification of the membrane edges with a significant decrease of background noise. Hence, it is proven the benefit of the deconvolution for a successful quantitative analysis.

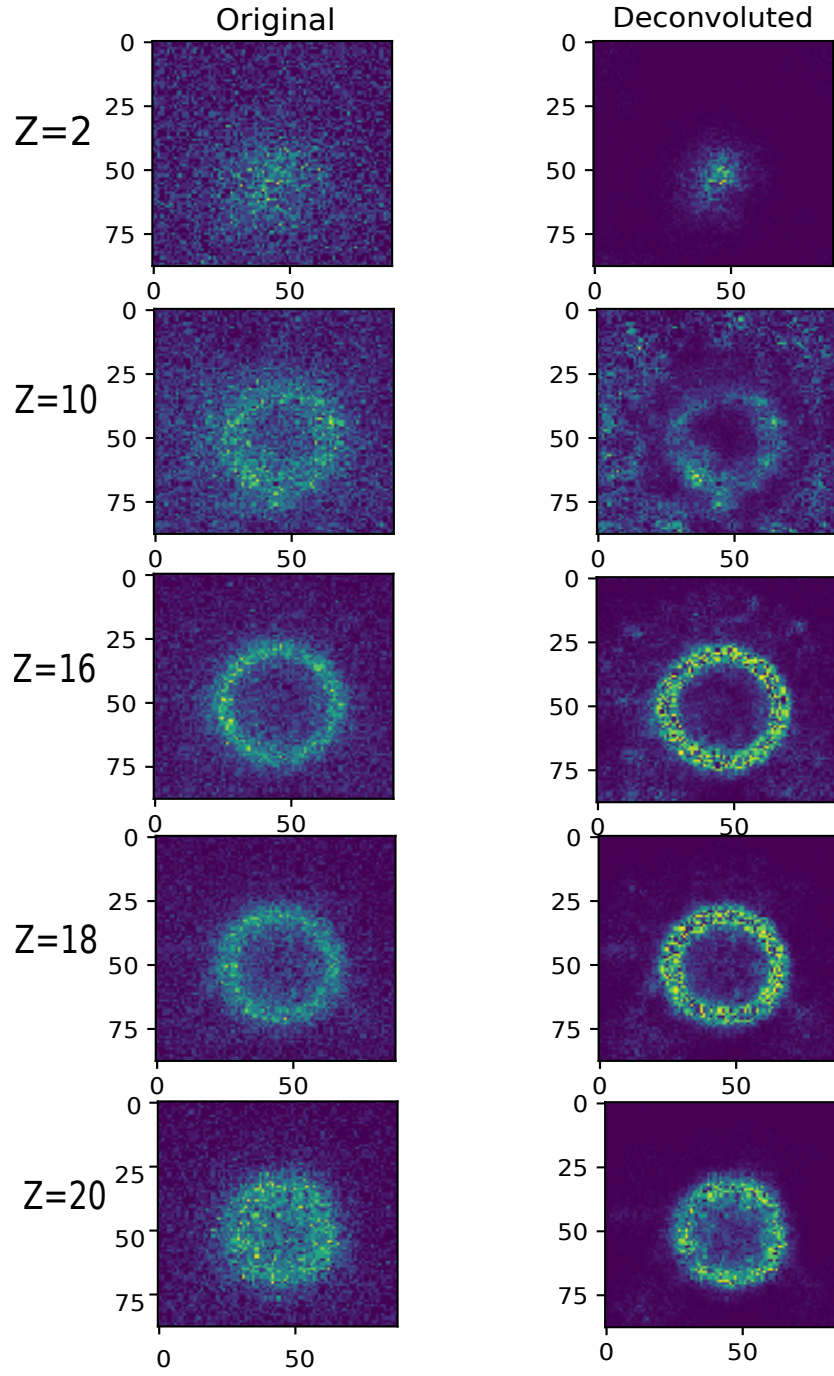


Figure 5.1: Example of deconvoluted Zstack of Fyve2x, representing the first channel of dataset. On the right side there are the original images, while on left side the deconvoluted ones. Each image corresponds to different depth of focus, indicated by the value of Z .

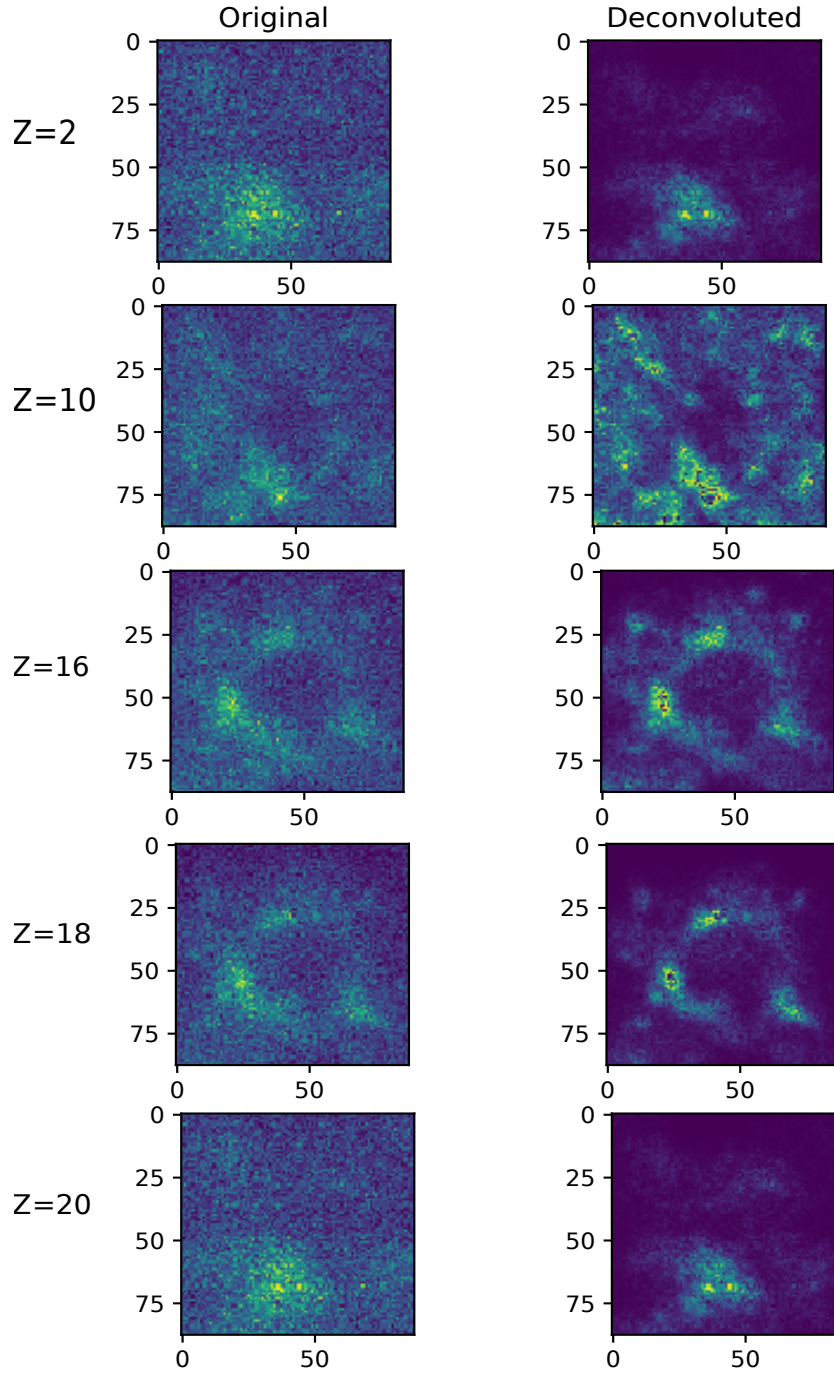


Figure 5.2: Example of deconvoluted Zstack of Gfp, corresponding to the second channel of dataset. On the right side there is the original image and on left side the deconvoluted one. Each image corresponds to different depth of focus, indicated by the value of Z.

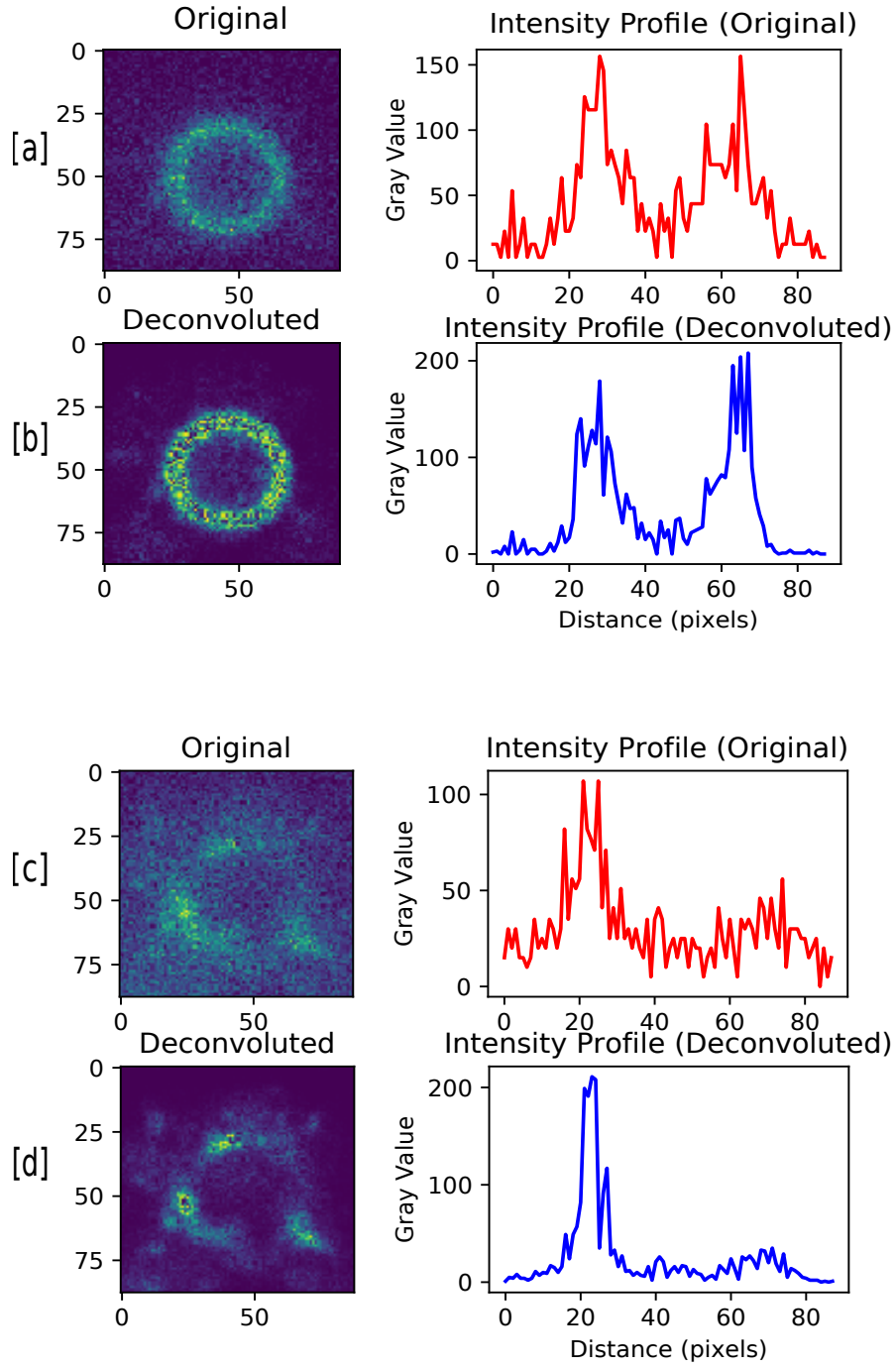


Figure 5.3: Intensity profiles of the stack $Z = 18$ to compare the signal of original image and of the deconvolved one. [a] and [b] refers to the first channel, while [c] and [d] to the second one.

5.2 Endosome Reconstruction

In this paragraph are shown the images of the membrane reconstruction. In Fig. 5.4, Fig. 5.5 and Fig. 5.6 we present in three different columns respectively the empirical data, the corresponding reconstruction and their difference. Are reported the images of the relevant slices of the entire Z-stack collection. In particular, to empathize the goodness of the obtained fitting, is worth to focus on the data difference. In fact, the reconstruction reproduces faithfully the membrane areas, ignoring the fluorescence of the surrounding zones. In conclusion, the significant results are collected in Fig. 5.7.

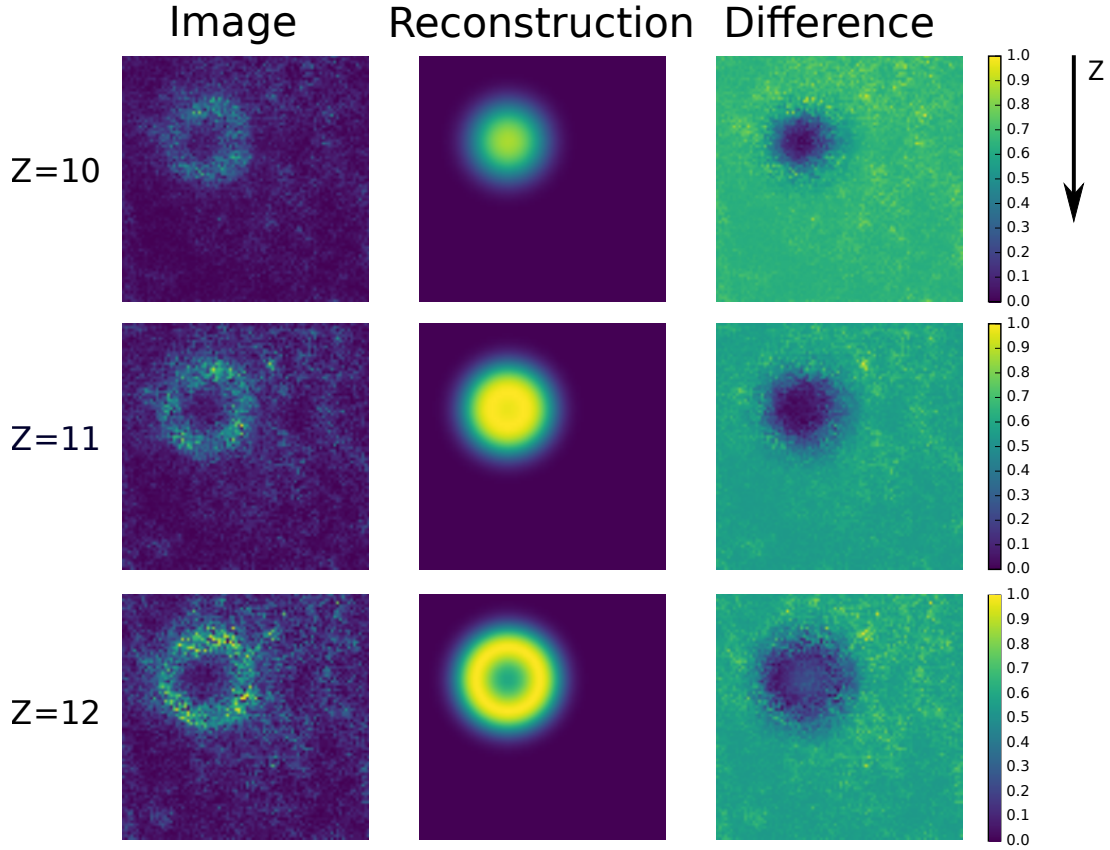


Figure 5.4: Results of the reconstruction procedure of the endosome membrane in $Z = 10, 11$, and 12 . From left to right, each column show respectively the starting image, the corresponding reconstruction and their difference.

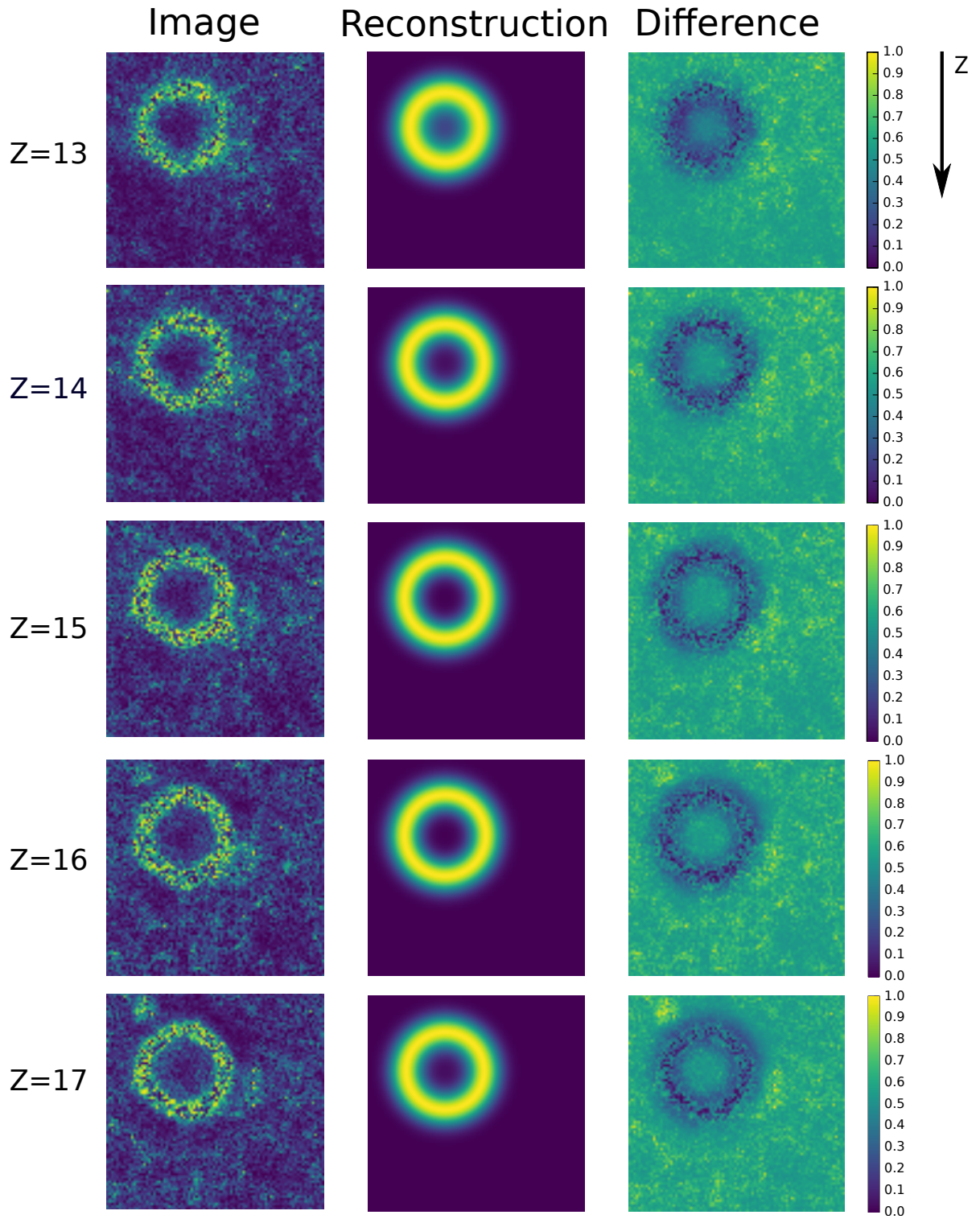


Figure 5.5: Results of the reconstruction procedure of the endosome membrane in $Z = 13, 14, 15, 16$ and 17 .⁵⁸ From left to right, each column show respectively the starting image, the corresponding reconstruction and their difference.

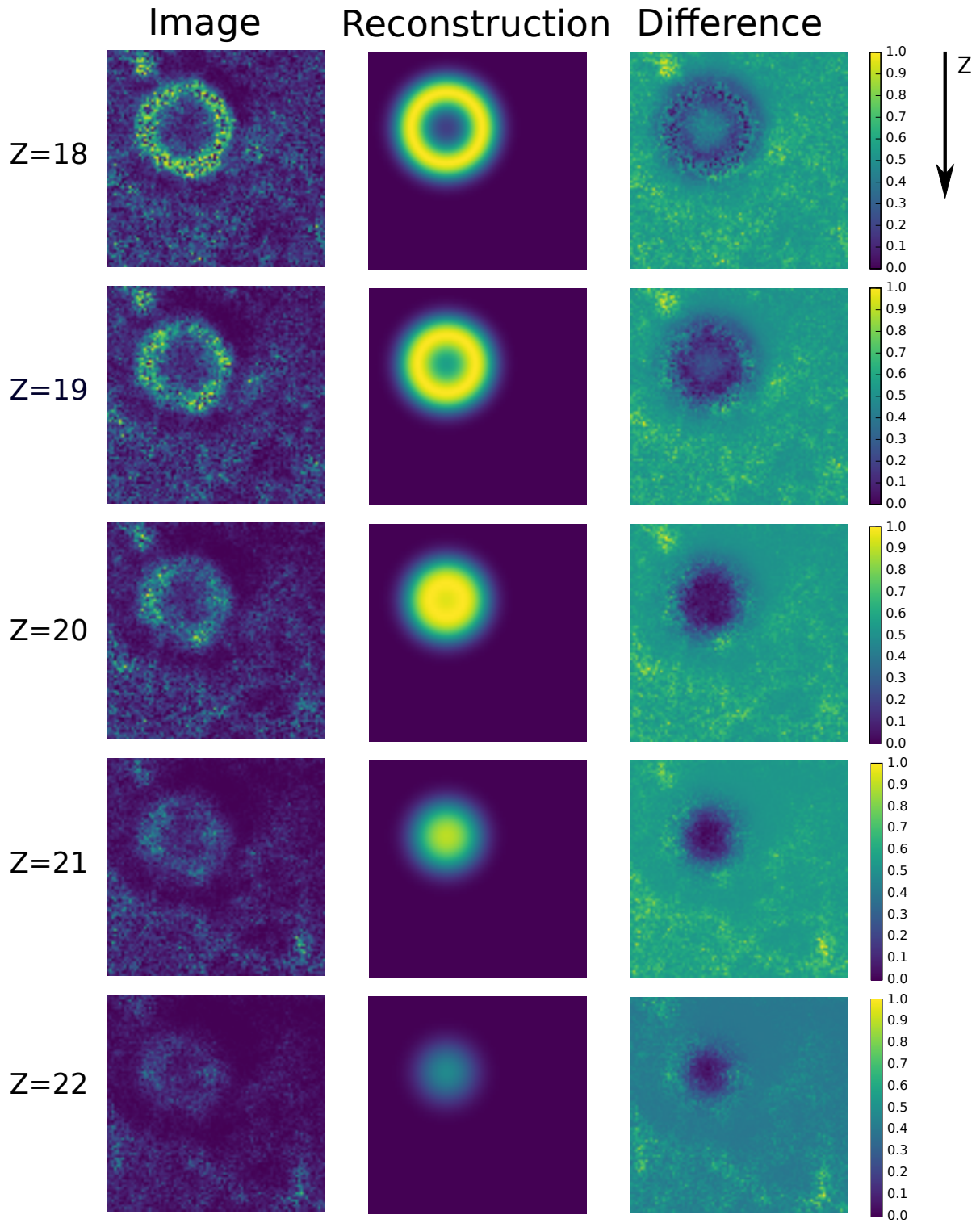


Figure 5.6: Results of the reconstruction procedure of the endosome membrane in $Z = 18, 19, 20, 21$ and 22 .⁵⁹ From left to right, each column show respectively the starting image, the corresponding reconstruction and their difference.

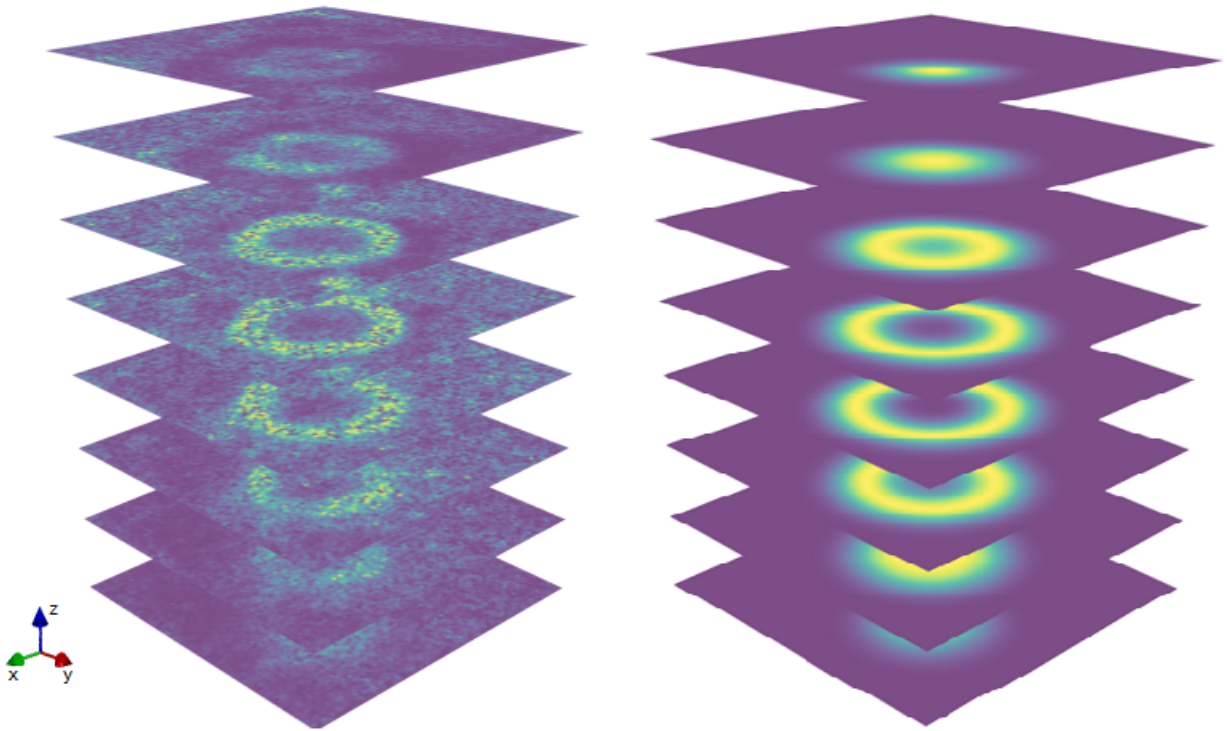


Figure 5.7: Summary of the results.

5.3 Domain Reconstruction

This last paragraph of the results section, deals with the domains identification and reconstruction. The results presented in the Fig. 5.8, Fig. 5.9 and Fig. 5.10 are organized by following. The first columns reports the original images after the application of the deconvolution pre-process. Instead, in the first column we show the membrane and protein colocalization obtained using the reconstructed membrane discussed in the previous paragraph. On the other hand, the third column is related with the reconstruction obtained with the proposed tool. Once more, to prove the quality of the approach, we present in the last column the difference among the experimental images processed and the reconstruction generated. The good fitting for the zones of interest confirm the value of the methodology. In conclusion, the significant results are collected in Fig. 5.11.

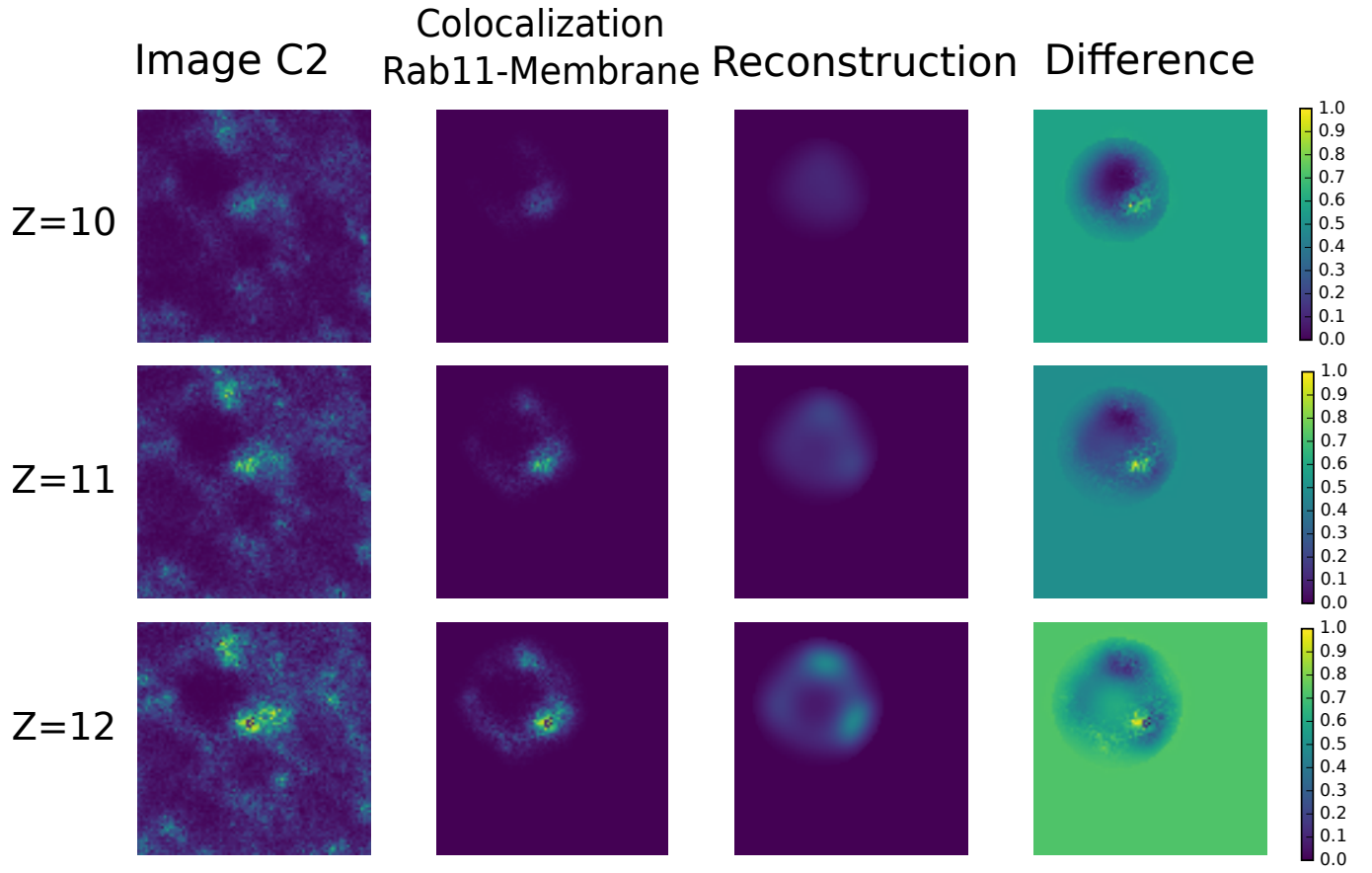


Figure 5.8: Comparison between original image and reconstructed in $Z = 10$, 11, and 12. From left to right, each column show respectively the starting image, the colocalization of protein and membrane, the reconstruction and in conclusion the difference between the last two.

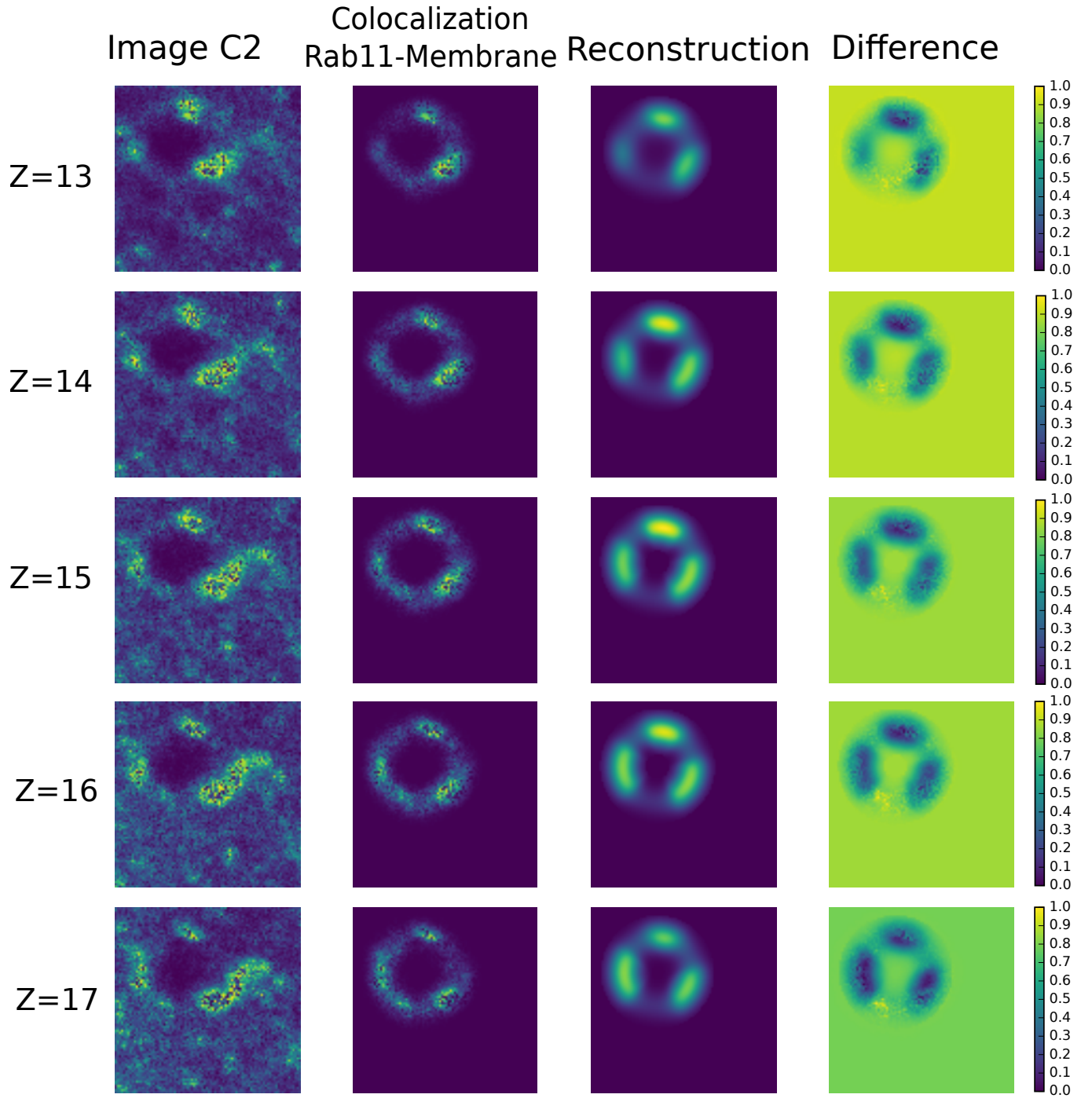


Figure 5.9: Comparison between original image and reconstructed in $Z = 13, 14, 15, 16$ and 17 . From left to right, each column show respectively the starting image, the colocalization of protein and membrane, the reconstruction and in conclusion the difference between the last two.

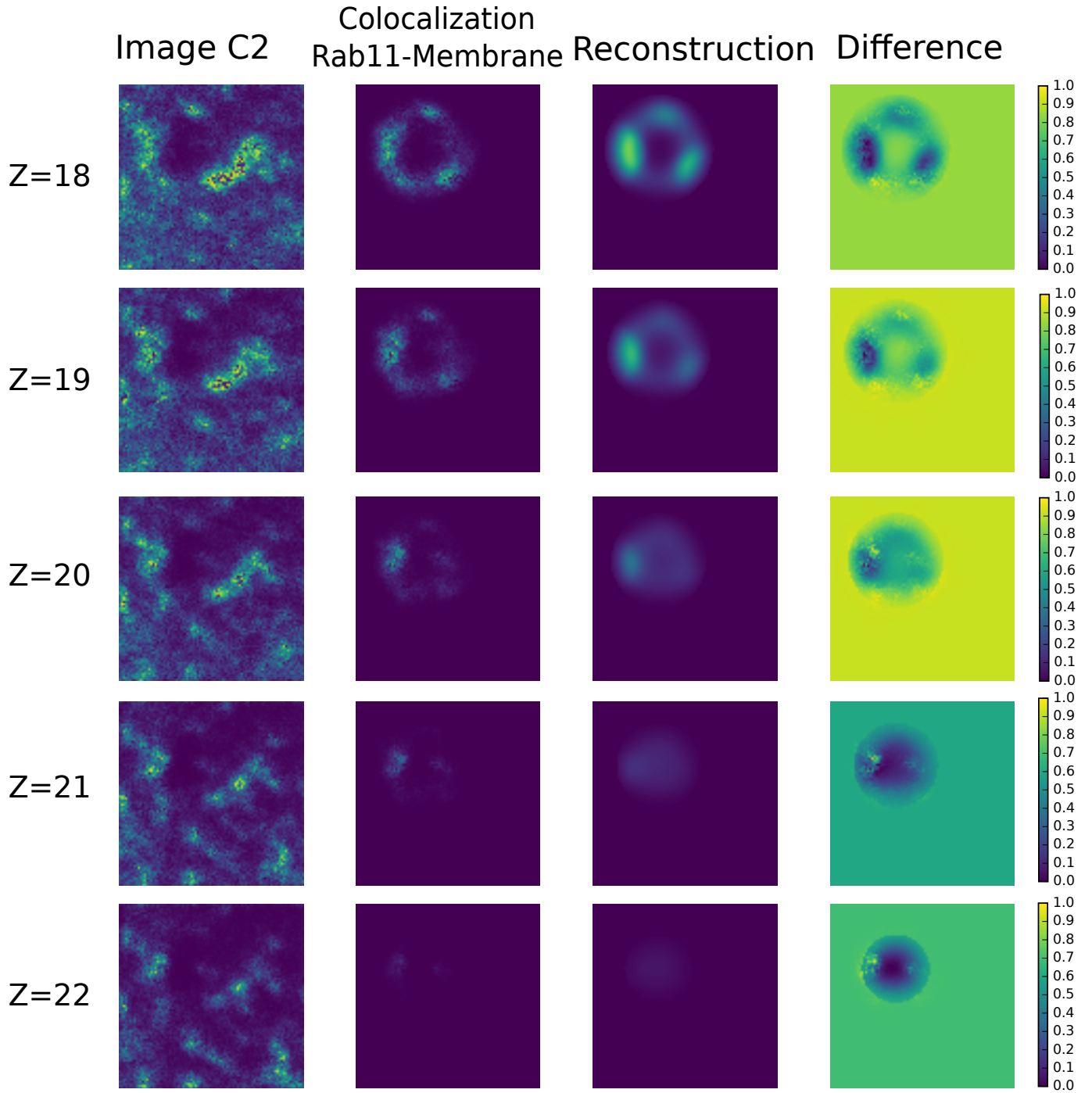


Figure 5.10: Comparison between original image and reconstructed in $Z = 18, 19, 20, 21$, and 22 . From left to right, each column show respectively the starting image, the colocalization of protein and membrane, the reconstruction and in conclusion the difference between the last two.

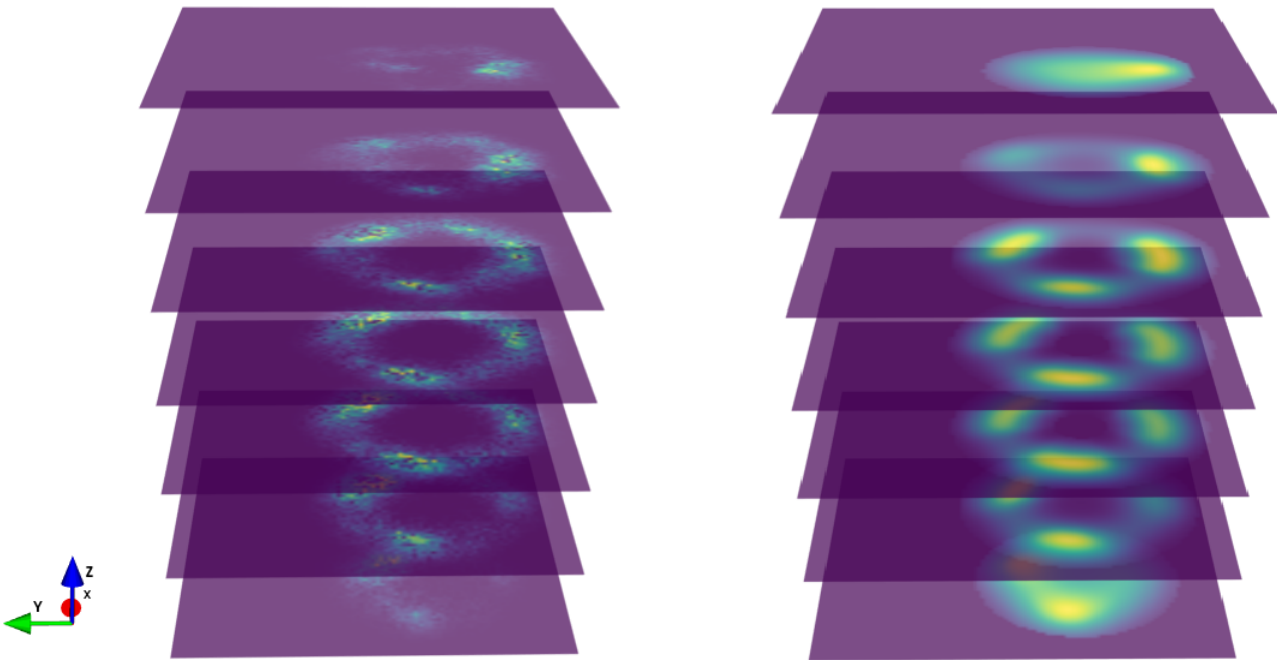


Figure 5.11: Summary of the results.

In the following figure (Fig. 5.12), are presented the 3D rendering of membrane and Rab11 domains from different point of view.

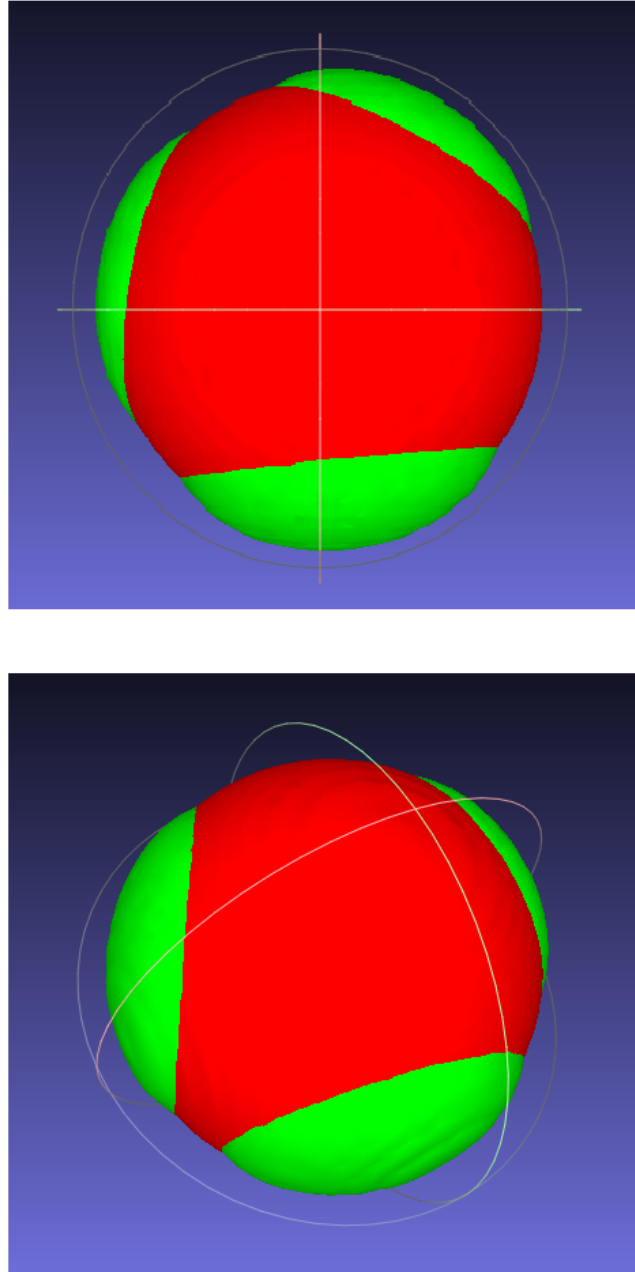


Figure 5.12: Reconstructed 3D volume

Chapter 6

Conclusions and Future Work

In this thesis work an experimental study related to eukaryotic cells have been performed.

In particular, the main focus has been the 3D reconstruction of molecular sorting from experimental images, aimed to allow a more complete understanding of the process.

The main contribution are related to the introduction of a methodology to accomplish this goal.

The proposed strategy has been implemented in Python.

A principal intuition, to obtain a good reconstruction of the images, has been the hypothesis related the morphologies of the structures of interest.

In particular, this assumption has compensated the reduced images resolution linked with the microscope limits.

The proposed approach is able, as well as to identify and reconstruct, to give physical informations related to size of the structure of interest for molecular sorting.

In addition, analyzing different Z-stack frames, allow the reconstruction of the whole process evolution over time. Consequently, this tool can be greatly contribute to investigate a large variety of phenomena related to living cells, which are still poorly understood.

The prospects are to adapt the procedure to quantitatively show some

phenomena. Such as, the domains subdivision, or contrarily, their aggregation. Other possible applications are the study of the domains kinetics, or the reconstruction of nucleation of the transport vesicles with the consequent detachment.

Appendices

Appendix A

Acquisition of Experimental Images

The acquired experimental images are obtained from living cells suitably labeled with fluorescent proteins.

These proteins bind to the structures we want to observe, allowing them to be traced and observed under a microscope.

Two fluorescent proteins are used: the Gfp and the Fyve2x which binds respectively to Rab11 and to the lipid membrane of the endosome.

The advantage of these fluorescent markers is that they bind to the subcellular structures of interest, without altering their normal functionality.

Live imaging of the endosomes was performed on a Leica SP8 confocal laser scanning microscope, equipped with 63X O2/oil immersion objective and with numerical aperture (NA) 1.40.

Hyd detectors allow the simultaneous detection of Gfp and Fyve2x.

Fluorescent dyes are sequentially imaged.

The Gfp is excited with a 458 nm laser line and imaged at 470 – 500 nm bandpass emission filters.

The Fyve2x instead is excited with the 568 nm Helium Neon laser line and imaged through to 580 – 650 nm bandpass emission filter.

Serial sections is acquired satisfying the Nyquist criteria for sampling and

processed.

The ROI has been chosen in order to contain the endosome and the surrounding intracellular region, sufficiently wide and homogeneous to be able to visualize moving vesicles.

Each image (Z-stack) consists of 28 dual-channel optical sections; the two channels corresponds to Gfp and Fyve2x respectively. Each optical section is an image of size 88×88 pixels.

In conclusion, the images are exported as a single TIFF file.

Appendix B

3D Deconvolution Manual

B.1 Creating a PSF image

In deconvolution the key element needed is the Point Spread Function (PSF) which defines how a given point of light is imaged by your camera and microscope. The PSF is notoriously tricky to calculate empirically, so we opted to create a theoretical PSF using the PSF Generator plug-in for Fiji.

You can install it from <http://bigwww.epfl.ch/algorithms/psfgenerator/>. Open this plugin from your Fiji menu:

`>Plugins >PSF Generator`

We need to have the following information to create a PSF:

- **The algorithm**, e.g. B&W;
- **Index of refraction**
- **Numerical aperture**
- **Wavelength** of the emission fluorescence of the fluorophore in nano-meters;
- **Pixelsize XY** in nano-meters;

- **Z-step** in in nano-meters;
- **SizeXYZ**.

When you have found and entered all of these numbers hit on this image and then select:

`File >Save as>.tiff`

from Fiji menu for later use.

B.2 3D Deconvolution

In 3D deconvolution a stack of images at different depths of focus are effectively cross referenced to eliminate blur from every layer. The algorithm used works in repeated iterations, to calculate successively more accurate images. This process is more computationally complex than can, especially where large images or large numbers of iterations are concerned, take a few hours to complete.

For 3D deconvolution you will need a stack of images (a single file containing the photographic slices at different depths). To begin, open the color image you wish to deconvolute in Fiji and then:

`>Image >Color >Split channels.`

You should now have different images, one for each channels. Save each of these separately to perform the 3D deconvolution process on each of them. Using the deconvolutionLab2 plug-in of Fiji we can perform our deconvolution operation.

You can install it from <http://bigwww.epfl.ch/deconvolution/>.

Open this plug-in from your Fiji menu:

`>Plugins >DeconvolutionLab2`

You have to select, for each channel separately:

- **Image file** Path to a single file;
- **PSF file** Path to the correspondent PSF previously generated;
- **Algorithm** Select the algorithm and set the parameters (for more details about it you can see the appendix B).

Press 'Run' to start the deconvolution and then save it.

Once that all channels have been deconvoluted using the process outlined above, you can merge them:

Image >Color >Merge Channels

Finally you can save your deconvoluted image.

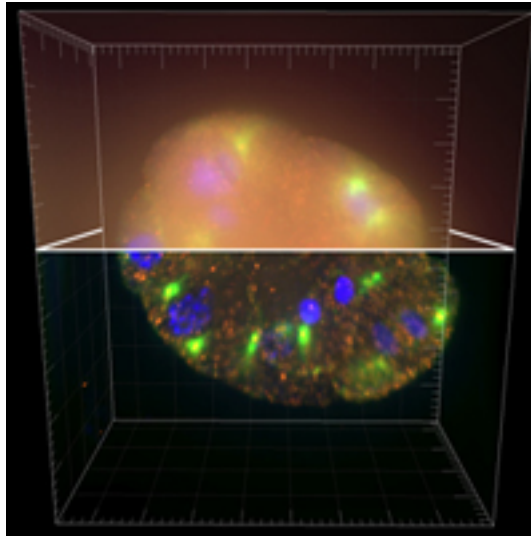


Figure B.1: Example of deconvolution on a dataset of three stacks of images of a *C. Elegans* embryo.

Appendix C

Python Code for 3D Reconstruction

In the following sections are reported the Python library containing the functions codes used to reconstruct the 3D volume from experimental images. In each of these functions is reported a brief comment to better describe them.

The code is structured as follows:

- **Read_Meta_images.py** In this library there are the definitions of the functions used to read the microscopy images. In particular, this code is able to read images using the metadata contained in the microscopy file (i.e. .liff, .tiff, etc...) which are stored following the Open Microscopy Environment (OME) schema [13]. Examples of metadata are: the number of pixels in the X, Y and Z directions, the temporal information T, the physical dimensions of the pixels, the information concerning the objective lens used, the numerical aperture (NA), the refractive index (RI), the information on the emission and excitation wavelength of the light used, the number of C channels present and many others. Typically a microscopy file contains several series and each series is a 3D movie. This last can be represented in the calculator as a 5-dimensional matrix: $[T, Z, C, X, Y]$, which values represent the pixel intensities. The data are saved in lists of length corresponding

to the number of series present in the file and each element of the list is a 5D matrix containing the pixel values.

- **Tools_fyve2x.py** In this library there are the python definitions employed to reconstruct the membrane of the endosome using the data of the first channel dataset. In particular, is possible to find the definition to evaluate the empirical distribution probability, the one evaluating the model function as in (4.6), and also the KL divergence. Furthermore, there is the function used to solve the optimization problem defined in (4.7). As stated in the dissertation, the optimization method selected is the 'L-BFGS-B' one. Starting from a random point, the algorithm search the solution moving in the feasible region until the objective function is minimized.
- **Tools_gpf.py** In this library there are the python functions used to reconstruct the Rab11 domains using the data of the second channel of the dataset. As for the previous case, there is the function evaluating the empirical distribution probability, the one that define model distribution (4.12) and the KL divergence. Even to perform this analysis, the method used to solve the optimization problem (4.13) is the 'L-BFGS-B'. In the minimization function, we are looking for a solution in the feasible region, starting from a random vector over the sphere identifying the endosome membrane and from a random value of the angle, representing the cone width.
- **Reconstruction.py** This is the main function reconstructing the entire volume from a single Z-stack. In this function, all the previously definitions are called to implement the procedure described in the Paragraph 4.2.3.
- **Final_Script** This script is shown for illustrative purposes. With the method defined in *Reconstruction.py* it is possible to reconstruct a 3D movie, simply giving in input the file name and indicating frames collection t we want to reproduce.

C.1 Read_Meta_images.py

```
from __future__ import print_function
import javabridge as jv
import bioformats
import numpy as np
from lxml import etree as etl
import sys

VM_STARTED = False
VM_KILLED = False

def start_jvm(max_heap_size='4G'):
    """
    Start the Java Virtual Machine.
    """
    global VM_STARTED
    jv.start_vm(class_path=bioformats.JARS, \
                max_heap_size=max_heap_size)
    VM_STARTED = True

def kill_jvm():
    """
    Kill the JVM.
    """
    global VM_KILLED
    jv.kill_vm()
    VM_KILLED = True
```

```
def jvm_error():
    raise RuntimeError("The_Java_Virtual_Machine_has_already_been"
                       "killed ,_and_cannot_be_restarted ."
                       "and_try_again .")

start_jvm(max_heap_size='4G')

def get_OMEXML(imagefile):
    if not VM_STARTED:
        start_jvm()
    if VM_KILLED:
        jvm_error()
    #get OME-XML and change the encoding to UTF-8
    omexml = bioformats.get_omexml_metadata(imagefile)
    omexml = omexml.encode('utf-8')
    return omexml

def getinfofromOMEXML(omexml, nodenames, ns):
    """
    This function can be used to read the most useful
    OME MetaInformation from the respective XML.
    Check for the correct namespace.
    """
    #get the root tree
    root = etl.fromstring(omexml)

#define the namespace in order to find the correct path
    NSMAP = {'mw': ns}
```

```

#construct the search string
if len(nodenames) >= 1:
    search = "./mw:" + nodenames[0]
if len(nodenames) >= 2:
    search = search + "/mw:" + nodenames[1]
if len(nodenames) >= 3:
    search = search + "/mw:" + nodenames[2]
#find all elements using the search string
out = root.findall(search, namespaces=NSMAP)
#create an empty list to store the dictionaries in
dictlist = []
for i in range(0, len(out)):
    #create the dictionary from key value
    dict = {}
    for k in range(0, len(out[i].attrib)):
        dict[out[i].keys()[k]] = out[i].values()[k]
    #add dictionary to the list
    dictlist.append(dict)
return dictlist

def get_relevant_metainfo(imagefile, \
    ns = 'http://www.openmicroscopy.org/Schemas/OME/2016-06'):
    if not VM_STARTED:
        start_jvm()
    if VM_KILLED:
        jvm_error()

#get the actual image reader
rdr = bioformats.get_image_reader(None, path=imagefile)
#Number of total series
totalseries = np.int(rdr.rdr.getSeriesCount())

```

```

omexml=get_OMEXML( imagefile )
Meta=[]
MetaInfo_list=[]
Meta=getinfofromOMEXML(omexml, [ 'Image', 'Pixels' ], ns)
for seriesID in range(0,totalseries):
    MetaInfo_list.append(Meta[seriesID])
return MetaInfo_list, totalseries

def get_image5d_multidim( imagefile , MetaInfo_list ):
    """
    This function read the image data series by series.
    Every series is stored inside a tuple as a 5D numpy array.
    The 5D array has the following dimension order: [T, Z, C, X, Y].
    """

    if not VM_STARTED:
        start_jvm()
    if VM_KILLED:
        jvm_error()
    rdr = bioformats.ImageReader(imagefile , perform_init=True)
    readstate = 'OK'
    readproblems = []
    # initialize the empty list that will hold all the 5D array
    images_list = []
    numseries = len(MetaInfo_list)
    for seriesID in range(0, numseries):
        current_sizeT = int(MetaInfo_list[seriesID][ 'SizeT' ])
        current_sizeZ = int(MetaInfo_list[seriesID][ 'SizeZ' ])
        current_sizeC = int(MetaInfo_list[seriesID][ 'SizeC' ])
        # read the XY dimension of the first series
        current_sizeX = int(MetaInfo_list[seriesID][ 'SizeX' ])
        current_sizeY = int(MetaInfo_list[seriesID][ 'SizeY' ])

```

```

newsize = [current_sizeT, current_sizeC, current_sizeX,
           current_sizeY, current_sizeZ]
# create the 5D numpy array
img5d = np.zeros(newsize, dtype=BF2NP_DTYPE
                  [rdr.rdr.getPixelType()])
# main loop to read the images from the data file
for timepoint in range(0, current_sizeT):
    for zplane in range(0, current_sizeZ):
        for channel in range(0, current_sizeC):
            try:
                img5d[timepoint, channel, :, :, zplane] =\
                    rdr.read(series=seriesID, c=channel, \

                #print(readstate)
            except:
                print('Problem_reading_data_into
.....

                readproblems = sys.exc_info()[1]

# store the 5D array inside a tuple
images_list.append(img5d)
# clear the array from memory
img5d = None
rdr.close()
return images_list, readstate

```

C.2 Tools_fyve2x.py

```
from __future__ import division
import numpy as np
from scipy.optimize import minimize

def normalization(M):
    """
    This function gives in input a 3D matrix of intensities ,
    returns  $p[x,y,z]$  which corresponds to the probability to
    observe a molecule in position  $(x,y,z)$ 
    """
    N=np.sum(M)
    if N != 0:
        p=M/N
    return p

def modulo(r0 , delta , rx , ry , rz):
    """
    This function returns the distances between
     $r=(rx,ry,rz)$  (points of the volume) and  $r0$  (fixed point).
    It takes into account the physical size of a voxel
    given by  $delta=(dx,dy,dz)$ .
    """
    mod=np.sqrt (((rx-r0[0])*delta[0])**2\
                  +((ry-r0[1])*delta[1])**2\
                  +((rz-r0[2])*delta[2])**2)
    return mod
```

```

def fun_fyve2x(mod, R, sigma):
    """
    Defines the function of the model that fit
    the endosome and evaluates it in each point of the volume.
    """
    f=(np.exp((-1/2)*((mod-R)/sigma)**2))
    return f

def div_fyve2x(theta ,p,delta ,rx ,ry ,rz ):
    """
    Given p and q probability distributions ,
    div_fyve2x evaluates the Kullback-Leibler divergence
    of q from p.

$$D_{kl} = p * \log(p/q).$$

    """
    r0 ,R,sigma=theta [:3] , theta [3] , theta [4]
    mod=modulo(r0 , delta , rx , ry , rz )
    C=np.sum(fun_fyve2x(mod, R, sigma))
    ind=np.nonzero(p)
    D = np.sum(p[ind]*np.log(p[ind]))\
        +np.sum(p*((np.log(C)-((-1/2)*((mod-R)/sigma)**2))))
    return D

def qmat_fyve2x(theta , delta , rx , ry , rz ):
    """
    This function evaluates the model matrix q
    given the values of the theta parameters.
    """
    r0 ,R,sigma=theta [:3] , theta [3] , theta [4]
    mod=modulo(r0 , delta , rx , ry , rz )
    C=np.sum(fun_fyve2x(mod, R, sigma))

```

```
q=(1/C)*fun_fyve2x(mod, R, sigma)
return q

def minimization(p,delta ,rx ,ry ,rz ,Size):
    """
    This function returns the argument (i.e. theta)
    minimizing the Kullback–Laibler divergence
    of q_theta from p.
    """
    Rmax=1000
    devmax=250
    theta0=np.array([np.random.uniform(0, Size[0]),\
                     np.random.uniform(0,Size[1]),\
                     np.random.uniform(0,Size[2]),\
                     np.random.uniform(0, Rmax),\
                     np.random.uniform(0, devmax)])
    arg=(p,delta ,rx ,ry ,rz)
    bnds=([0, Size[0]], [0, Size[1]], [0,Size[2]], [0,Rmax],[0,devmax])
    res = minimize(div_fyve2x,theta0,\
                   args=arg,method='L-BFGS-B',\
                   jac=None, bounds=bnds,\
                   tol=None, callback=None,\
                   options={'disp': None, 'maxcor': 10,\
                             'ftol': 2.220446049250313e-09, 'gtol': 1e-05,\
                             'eps': 1e-08, 'maxfun': 15000, 'maxiter': 15000,\
                             'iprint': -1, 'maxls': 20})
    return res
```


C.3 Tools_gpf.py

```
from __future__ import division
import numpy as np
from scipy.optimize import minimize

def normalization(M):
    """
    Normalization takes as input M, the matrix and
    returns the matrix p.
     $p[x,y,z]$  corresponds to the probability to observe
    a molecule in position  $(x,y,z)$ .
    """
    N=np.sum(M)
    if N != 0:
        p=M/N
    return p

def modulo(r0 , delta , rx , ry , rz ):
    """
    This function returns the distances between
     $r=(rx,ry,rz)$  (points of the volume) and  $r0$  (fixed point).
    It takes into account the physical size
    of a voxel given by  $\text{delta}=(dx,dy,dz)$ .
    """
    mod=np.sqrt(((rx-r0[0])*delta[0])**2\
        +((ry-r0[1])*delta[1])**2\
        +((rz-r0[2])*delta[2])**2)
    return mod
```

```
def modulou(r0 , delta , u):
```

```
    """
```

```
    Distance between the vectors u and r_0
```

```
    """
```

```
    mod=np.sqrt ((u[0]-r0[0]*delta[0])**2\
                  +(u[1]-r0[1]*delta[1])**2\
                  +(u[2]-r0[2]*delta[2])**2)
```

```
    return mod
```

```
def modulou(r0 , delta , u):
```

```
    """
```

```
    The function evaluates the distance between the u and r_0,  
taking into account the physical size delta=(dx,dy,dz).
```

```
    """
```

```
    mod=np.sqrt ((u[0]-r0[0]*delta[0])**2\
                  +(u[1]-r0[1]*delta[1])**2\
                  +(u[2]-r0[2]*delta[2])**2)
```

```
    return mod
```

```
def rand_unif_sphere(r0 ,R, delta):
```

```
    """
```

```
    The function generates a random point on a sphere of  
radius R and center r_0.
```

```
    """
```

```
    theta= 2. * np.pi * np.random.uniform(0,1)
    phi = np.arccos( 2*np.random.uniform(0,1)-1)
    x = r0[0]*delta[0] + R * np.cos(theta)* np.sin(phi)
    y = r0[1]*delta[1] + R *np.sin(theta)* np.sin(phi)
    z = r0[2]*delta[2] + R * np.cos(phi)
    p=np.array([x,y,z])
```

```

return p

def qmat_gfp(u,c,r0,mask,delta,rx,ry,rz):
    """
    The function evaluates the distribution
        model q for the Gfp channel.
    Data: r0,mask,delta,rx,ry,rz
    Parameters: ux,uy,uz,c
    """
    n=(u-r0*delta)/modulou(r0,delta,u)
    mod=modulor(r0,delta,rx,ry,rz)
    w=(( (rx-r0[0])*delta[0])/mod*n[0]\
        +((ry-r0[1])*delta[1])/mod*n[1]\
        +((rz-r0[2])*delta[2])/mod*n[2]-c)
    w0=0.1
    f0=1/np.pi*np.arctan(w/w0)+1/2
    f=f0*mask
    C=np.sum(f)
    return f/C

def div_gfp(theta,p,mask,delta,r0,rx,ry,rz):
    """
    The function evaluates the KL divergence of p from q,
    where p and q are the empirical and the model distribution.
     $D(q||p) = \sum(q \log(q/p))$ 
    """
    u,c=theta[:3],theta[3]
    q=qmat_gfp(u,c,r0,mask,delta,rx,ry,rz)
    ind=np.nonzero(q)
    D = np.sum(q[ind]*(np.log(q[ind]/p[ind])))

```

return D

```
def min_gfp(r0,R,delta ,p_mask,mask,rx,ry,rz,Size):
    """
    The function returns the value and the argument minimizing the
    KL divergence of p from q.
    The minimization algorithm takes in input objective
    function 'div_gfp' and the starting point theta0.
    Theta0 is random and uniformly distributed.
    """
    u0=rand_unif_sphere(r0,R,delta)
    theta0=np.array([u0[0],u0[1],u0[2],np.random.uniform(-1,1)])
    bnds=([0,Size[0]*delta[0]],[0,Size[1]*delta[1]],[
        0,Size[2]*delta[2]],[ -1,1])
    arg=(p_mask,mask,delta,r0,rx,ry,rz)
    res_gfp = minimize(div_gfp,theta0, args=arg,\
        method='L-BFGS-B',jac=None,\
        bounds=bnds, tol=None,\
        callback=None,\
        options={'disp': None, 'maxcor': 10,\
            'ftol': 2.220446049250313e-09,\
            'gtol': 1e-05, 'eps': 1e-08,\
            'maxfun': 15000,\
            'maxiter': 15000,\
            'iprint': -1,\
            'maxls': 20})
    return res_gfp.fun,res_gfp.x
```

C.4 Reconstruction.py

```
from __future__ import division
import numpy as np
import sys
sys.path.append('C:\Users\Path....')
import Read_Meta_Images as rmi
import tools_Fyve2x as tlf
import tools_Gfp as tlg
from scipy import ndimage

def reconstruction(imagefile, t):
    """
    The function takes in input the imagefile and the frame t.
    It returns:
    -p_fyve2x: the empirical distribution of Fyve2x in the volume;
    -theta_f: the argument minimizing  $D(p_{fyve2x}, qf(theta))$ ;
    -qf: the model distribution for Fyve2x, evaluated in theta_f;
    -p_gfp: the empirical distribution of Gfp in the volume;
    -theta_g_list: the list of the arguments minimizing the KL div.
    -qg: the model distribution for Gfp, evaluated in theta_g.
    """
    #Read the images in microscope format eg. .lif, .tiff etc
    #and save them in a list.
    #Each element of the list is a 5D-array:
    #a collection of T frame, each one with C channels and voxel XYZ.
    #e.g. List[Series][T, C, X, Y, Z]
    rmi.start_jvm(max_heap_size='4G')
    MetaInfo_list, totalseries = \
        rmi.get_relevant_metainfo(imagefile, \
            ns = 'http://www.openmicroscopy.org/Schemas/OME')
```

```

Img, readstate=rmi.get_image5d_multidim(imagefile , MetaInfo_list)
Img=np.uint8(Img)

#Save the frame t and split the two channels
M = Img[0][t,:,:,:]
M_fyve2x = M[0,:,:,:]
M_gfp = M[1,:,:,:]

#Physical sizes
delta = np.array([44.1867,44.1867,150.2039]) #mm

#size of the image
Size = M_fyve2x.shape

#voxel grid
x = np.arange(0 , Size[0] , 1)
y = np.arange(0 , Size[1] , 1)
z = np.arange(0 , Size[2] , 1)
rx , ry , rz = np.meshgrid(x, y, z)

#-----Fitting Fyve2x-----

#Probability matrix (Fyve2x)
p_fyve2x = tlf.normalization(M_fyve2x)

#gaussian filter (Fyve2x)
p_fyve2x_gaussian = tlf.normalization(\
    ndimage.gaussian_filter(p_fyve2x , 2))

#threshold (Fyve2x)
threshold_pf = (p_fyve2x_gaussian > \

```

```

        (np.mean(p_fyve2x_gaussian)+\
        2*np.std(p_fyve2x_gaussian)))
p_fyve2x_gaussian_tresh = p_fyve2x_gaussian*threshold_pf

#final probability matrix (Fyve2x)
p_fyve2x_final = tlf.normalization(p_fyve2x_gaussian_tresh)

#Minimize the divergence D(p_f||qf)
res_fyve2x = tlf.minimization(p_fyve2x_final,delta ,rx ,ry ,rz ,Size)

#Save the argument that minimize D(p_f||qf)
theta_f = res_fyve2x.x

#Evaluate the (optimal) model distribution (Fyve2x)
qf = tlf.qmat_fyve2x(theta_f,delta ,rx ,ry ,rz)

r0 = theta_f[:3]
R = theta_f[3]

#-----Fitting Gfp-----

#Probability matrix (Gfp)
p_gfp = tlg.normalization(M_gfp)

#gaussian filter (Gfp)
p_gfp_gaussian = ndimage.gaussian_filter(p_gfp, 5)

#mask (to ignore the fluorescence outside the membrane)
mask = (qf>(np.mean(qf))) + 1e-50

#threshold (Gfp)

```

```

threshold_pg = (p_gfp_gaussian > \
                (np.mean(p_gfp_gaussian)+ \
                 np.std(p_gfp_gaussian))) + 1e-50
p_gfp_gaussian_mask = threshold_pg * p_gfp_gaussian * mask

#final probability matrix (Gfp)
p_gfp_final = tlg.normalization(p_gfp_gaussian_mask)

q_gfp_list=[]
theta_g_list=[]
current_theta_g=np.array([0,0,0,0])
p_current= p_gfp_final
npatch=-1

while current_theta_g[3]<0.9 and npatch<10:
    #find 10 min of the divergence D(q_theta || p_gfp)
    #and save them in a list
    list_res_gfp = []
    for i in range(0,10):
        list_res_gfp.append(tlg.min_gfp(r0,R,delta, \
                                         p_current,mask,rx,ry,rz,Size))

        #sort the list with respect to the divergence
    list_res_gfp.sort()

    #save the argument that minimize D(qg_theta || p_gfp)
    current_theta_g=list_res_gfp[0][1]
    theta_g_list.append(current_theta_g)

    #evaluate the (optimal) model distribution (Gfp)
    q_gfp_list.append(tlg.qmat_gfp(current_theta_g[:3], \

```



```
        current_theta_g[3], r0, mask, delta, rx, ry, rz))
current_q = tlg.normalization(sum(q_gfp_list))
elim = current_q < np.mean(current_q) + np.std(current_q)
p_current = (p_gfp_final * elim) + 1e-50
npatch=npatch + 1
qg=tlg.normalization(sum(q_gfp_list[:-1]))
return(p_fyve2x, qf, theta_f, p_gfp, qg, theta_g_list)
```

C.5 Final_Script

```
from __future__ import division
import numpy as np
import sys
sys.path.append( 'path...' )
import Read_Meta_Images as rmi
import tools_Fyve2x as tlf
import tools_Gfp as tlg
import reconstruction as rec

imagefile='D_RL30_2018_07_19_clus_rab11gfp_fyve2x.lif_-_Series013.tif'

#reconstruction of frames 1,2,...,44

res_reconstruction_frames=[]
for t in range (1,45):
    res_reconstruction_frames = rec.reconstruction(imagefile,t)
```

Bibliography

- [1] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell, Sixth Edition*. Taylor & Francis Group, 2014.
- [2] François Bavaud. Information theory, relative entropy and statistics. In *Formal Theories of Information*, pages 54–78. Springer Berlin Heidelberg, 2009.
- [3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [4] Max Born, Emil Wolf, A. B. Bhatia, P. C. Clemmow, D. Gabor, A. R. Stokes, A. M. Taylor, P. A. Wayman, and W. L. Wilcock. *Principles of Optics*. Cambridge University Press, 1999.
- [5] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyu Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, sep 1995.
- [6] Carlo Cosimo Campa, Jean Piero Margaria, Abhishek Derle, Marco Del Giudice, Maria Chiara De Santis, Luca Gozzelino, Francesca Copperi, Carla Bosia, and Emilio Hirsch. Rab11 activity and PtdIns(3)p turnover removes recycling cargo from endosomes. *Nature Chemical Biology*, 14(8):801–810, jun 2018.
- [7] Nicolas Dey, Laure Blanc-Feraud, Christophe Zimmer, Pascal Roux, Zvi Kam, Jean-Christophe Olivo-Marin, and Josiane Zerubia. Richard-

- son–lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microscopy Research and Technique*, 69(4):260–266, 2006.
- [8] A. Gamba, A. de Candia, S. Di Talia, A. Coniglio, F. Bussolino, and G. Serini. Diffusion-limited phase separation in eukaryotic chemotaxis. *Proceedings of the National Academy of Sciences*, 102(47):16927–16932, nov 2005.
- [9] A. Gamba, I. Kolokolov, V. Lebedev, and G. Ortenzi. Patch coalescence as a mechanism for eukaryotic directional sensing. *Physical Review Letters*, 99(15), oct 2007.
- [10] Mauro Gasparini. *Modelli probabilistici e statistici*. 2014.
- [11] Sarah Frisken Gibson and Frederick Lanni. Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *Journal of the Optical Society of America A*, 8(10):1601–1613, oct 1991.
- [12] H.T.M. van der Voort G.M.P. van Kempen and L.J. van Vliet. A quantitative comparison of two restoration methods as applied to confocal microscopy. *Proceedings of ASCI’96, 2nd Annual Conference of the Advanced School for Computing and Imaging*, pages 196–201,, 1996.
- [13] Ilya G Goldberg, Chris Allan, Jean-Marie Burel, Doug Creager, Andrea Falconi, Harry Hochheiser, Josiah Johnston, Jeff Mellen, Peter K Sorger, and Jason R Swedlow. The open microscopy environment (ome) data model and xml file: open tools for informatics and quantitative analysis in biological imaging. *Genome Biology*, 6(5):R47, 2005.
- [14] Ira Mellman and W. James Nelson. Coordinated protein sorting, targeting and distribution in polarized cells. *Nature Reviews Molecular Cell Biology*, 9(11):833–845, nov 2008.

-
- [15] B. Richards and E. Wolf. Electromagnetic diffraction in optical systems. II. structure of the image field in an aplanatic system. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 253(1274):358–379, dec 1959.
- [16] William Hadley Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55, January 1972.
- [17] Daniel Sage, Laurène Donati, Ferréol Soulez, Denis Fortun, Guillaume Schmit, Arne Seitz, Romain Guet, Cédric Vonesch, and Michael Unser. DeconvolutionLab2: An open-source software for deconvolution microscopy. *Methods*, 115:28–41, feb 2017.
- [18] P. Sarder and A. Nehorai. Deconvolution methods for 3-d fluorescence microscopy images. *IEEE Signal Processing Magazine*, 23(3):32–45, may 2006.
- [19] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, jul 1948.
- [20] Sara Sigismund, Stefano Confalonieri, Andrea Ciliberto, Simona Polo, Giorgio Scita, and Pier Paolo Di Fiore. Endocytosis and signaling: Cell logistics shape the eukaryotic cell plan. *Physiological Reviews*, 92(1):273–366, jan 2012.
- [21] Birte Sönnichsen, Stefano De Renzis, Erik Nielsen, Jens Rietdorf, and Marino Zerial. Distinct membrane domains on endosomes in the recycling pathway visualized by multicolor imaging of rab4, rab5, and rab11. *The Journal of Cell Biology*, 149(4):901–914, may 2000.
- [22] Linton M. Traub. Common principles in clathrin-mediated sorting at the golgi and the plasma membrane. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1744(3):415–437, jul 2005.

-
- [23] Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. *Foundations of Signal Processing*. Cambridge University Press, sep 2014.
- [24] E. Wolf. Electromagnetic diffraction in optical systems. i. an integral representation of the image field. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 253(1274):349–357, dec 1959.
- [25] Chi-Fang Wu, Jian-Geng Chiou, Maria Minakova, Benjamin Woods, Denis Tsygankov, Trevin R Zyla, Natasha S Savage, Timothy C Elston, and Daniel J Lew. Role of competition between polarity sites in establishing a unique front. *eLife*, 4, nov 2015.
- [26] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, dec 1997.