



SAPIENZA
Università di Roma
Facoltà di Farmacia e Medicina

Ph.D in
MORPHOGENESIS AND TISSUE ENGINEERING

XXXI CICLO (2015 – 2019)
(A.A. 2017/2018)

**A FRAMEWORK FOR IMAGE-BASED, AUTOMATED,
MULTILEVEL ANALYSIS OF THE CYTOSKELETAL
MORPHOLOGY**

Ph.D. Student
Garima Verma

Tutor 
Prof./Dr. Mauro Grigioni
Co-Tutor
Prof./Dr. Chaitanya Athale

Coordinator
Prof. Antonio Musaro

Garima Verma

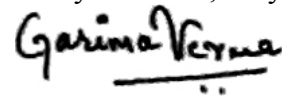
With the deepest gratitude, this thesis is dedicated to my parents
and my brother

Pag II

Declaration

I, Garima Verma, hereby declare that this Ph.D. thesis titled “A framework for image-based, automated, multilevel analysis of the cytoskeletal morphology” is the record of the original work done by me under the guidance of Tutor Prof. Mauro Grigioni, Dirigente Ricerca Bioingegneria presso Istituto Superiore di Sanità, Rome, Italy, and Co-Tutor Prof. Chaitanya Athale, Associate professor, Indian Institute of Science Education and Research, Pune, India. To the best of my knowledge, this work has not formed the basis for the award of any degree/diploma/certificate or a similar award, to any candidate in any University.

Dr. Garima Verma
Sapienza University of Rome, Italy



Garima Verma

Acknowledgments

I want to express my sincere appreciation to my supervisor Dr. Mauro Grigioni who has inspired and taught me many things and gave extensive useful suggestions during my doctoral degree program. At this present moment of accomplishment, my deep special thanks goes to him for his big support, encouragement as well as his guidance.

I am extremely grateful to my co-supervisor Prof. Dr. Chaitanya Athale at Indian Institute of Science Education and Research, Pune, India, who guided me to accomplish my research work for 6 months in the last year of my doctoral degree program. Without his motivation and kind support as well as suggestions this Ph.D. would not be achieved.

Many heartfelt thanks are owned to my doctoral committee members Prof. Mariano Bizzarri for his inspiration, valuable suggestions and advice.

I thankfully appreciate to Dr. Alessandro Palombo and Giuseppe D'Avenio for their valuable suggestions, guidance, and huge support.

I expand my thanks to my Lab mates and all my colleagues at Sapienza University of Rome (Italy), Istituto Superiore di Sanità, Roma, Italy and Indian Institute of Science Education and Research, Pune, India.

I would also like to give tremendous thanks and appreciation to all of my family and my friends, who helped me a lot with their moral support and valuable help. Besides, all of them, my deep sense of gratitude towards all of the people who knowingly and unknowingly helped me a lot in the successful completion of my Doctorate Thesis.

Index

ABSTRACT	1
SOMMARIO	3
1. Introduction and Background	5
1.1 System Biology	8
1.2 Cytoskeleton-I-beams of cell Biology	10
1.3 Measurable parameters of cytoskeleton and system	16
1.4 Quantitative methods to measure cytoskeleton parameters ...	17
1.5 Tools to analyze cytoskeleton morphology	24
1.6 Image analysis.....	27
1.7 Filopodyan- image analysis open-source software	28
1.8 Neuron J – Image J plugin	28
2. Aims	29
3. Results	32
3.1 A confocal based quantitative analysis in the cytoskeleton architecture of MCF-7 cells	32
3.2 Application of bioinformatics to analyse confocal image	32
3.3 Pilot study for cytoskeleton analysis.....	33
3.4 Validation of Image J software for the analysis of Cytospace Experiment	40
3.5 Analysis for identifying cytospace data through matlab.....	41
3.6 Validation of Matlab programming language for the analysis of Cytospace Experiment	53
3.7 Identification of analysis of neurite	56
3.8 Neurite data analysis	64
3.9 Neurite results produce from cell profiler.....	68
3.10 Neurite changing in different image in different time interval	76
3.11 Programming for detecting the neurites.....	78
3.12 Comparison of the total neurites length under various time series data	80
3.13 Detection of neurites from time period 0 to 2 hrs by Neuron J	82

3.14 Detection of neurites in PC12 cells from time period 0-11 hrs by Matlab and neuron J	85
3.15 Further detection of the neurite using Neuron J along with their logistic curves	86
3.16 Length of single neurite measured by Image J	98
3.17 Length of single neurite measured by Filopodyan.....	100
3.18 Validation of a tools and softwares for identification and analysis of Neurites	102
4. Material and Methods	103
4.1 The experimental sample of cytospace	103
4.2 Pipeline for the analysis of cells and Tubulin by MATLAB	105
4.3 Method for the analysis of neurites.....	118
4.4 Dataset of neurites.....	122
4.5 The experimental algorithms and protocols.....	123
Conclusion	130
Appendix A	131
References	138
List of publications	150

ABSTRACT

Aim: The aim of this thesis is to create an integrated bioinformatic framework for the quantitative assessment of cytoskeleton (CSK) morphology. In this research, we demonstrated the validity of this approach by applying computational biology in two cases: Cytospace images and neurite images. I used these 2 dataset of images as reference dataset to build the computational biology framework and to verify its functionality. To accomplish this, I have analyzed Cytospace optical and confocal microscope images by Image J and MATLAB programming.

I have analyzed the neurite number and length of PC 12 cells by using the following software and tools: (a) custom made tool MATLAB, (b) Cell profiler, (c) Image J, and (d) Filopodyan. Outside this goal of CSK analysis I have also analysed the microcalcification and parenchyma images through my framework to verify the capability of the framework to operate on a different scenario.

Methods: We have developed various algorithms and protocols for the analysis of nuclei, tubulin and microtubules. Each component of the cytoskeleton plays a very essential role in understanding the behaviour of the cell. Moreover, the confocal images were processed properly to extract the information regarding the cytoskelton.

Finally, we also studied PC 12 cells and selected / designed useful algorithms to analyze changes in the number and length of neurites.

Results: In the first part of this thesis we presented an in-silico model constructed by using measurable parameters obtained from microscope images of cells. We use image analysis software Image J to identify cell shape parameters – including surface area, roundness, fractal dimension, such as entropy and coherency. In

the second part of thesis, we also analysed area, perimeter, major and minor axis, circularity, the solidity of the nuclei and tubulin by MATLAB programming. In the third part of the thesis, we analyzed the dynamics of neurites by developing MATLAB scripts, using also other analysis tools such as Image J, Neuron J.

Conclusion: I have build a computational framework to analyse in quantitative manner images from optical and confocal microscopy.

During the first two years of PhD course I select, develop and integrated bioinformatics protocols and algorithms to define optimized operational pipelines. I used two datasets of image as reference to tuning the computational pipeline.

In the last period of phd course I can apply my optimised framework on microscopy images for detection and analysis of neurites. By means of the application of the framework was possible reduce considerable time of biologist to analyse the images.

SOMMARIO

Obiettivo: Lo scopo di questa tesi è quello di creare un framework bioinformatico integrato per la valutazione quantitativa della morfologia del citoscheletro (CSK). In questa ricerca, abbiamo dimostrato la validità di questo approccio applicando la biologia computazionale alle immagini di due casi sperimentali. Ho usato questo set di dati di immagini come set di dati di riferimento per creare il framework di biologia computazionale e verificarne la funzionalità. A tale scopo, ho analizzato alcune immagini da microscopia ottica e confocale del progetto CYTOSPACE mediante la programmazione di Image J e MATLAB.

Da immagini provenienti da un altro esperimento ho analizzato il numero e la lunghezza dei neuriti delle cellule PC 12 utilizzando i seguenti software e tools: (a) MATLAB, (b) Cell Profiler, (c) Image J e (d) Filopodyan. Al di fuori dell'analisi CSK ho anche analizzato le immagini di microcalcificazione e parenchima (da mammografie provenienti da un altro esperimento) attraverso il mio framework per verificare la capacità del framework di operare su uno scenario diverso.

Metodi: abbiamo sviluppato vari algoritmi e protocolli per l'analisi di nuclei, tubulina e microtubuli. Ogni componente del citoscheletro svolge un ruolo molto essenziale nella comprensione del comportamento della cellula. Le immagini confocali sono state opportunamente elaborate per estrarre le informazioni relative al citoscheletro.

Infine, abbiamo anche applicato il framework computazionale ad immagini di cellule PC 12, al fine di individuare e misurare le variazioni dinamiche dei neuriti in termini di numero e lunghezza.

Risultati: Nella prima parte di questa tesi abbiamo presentato un modello in silico costruito utilizzando parametri misurabili ottenuti da immagini di cellule acquisite tramite microscopia ottica e

confocale. Abbiamo usato il software di analisi delle immagini Image J per identificare i parametri di forma delle cellule, tra cui area superficiale, rotondità, dimensione frattale, entropia e coerenza. Nella seconda parte della tesi, abbiamo anche analizzato area, perimetro, asse maggiore e minore, circolarità, solidità dei nuclei e della tubulina mediante codice appositamente sviluppato in MATLAB. Nella terza parte della tesi, abbiamo analizzato la dinamica dei neuriti sviluppando script MATLAB, usando anche altri strumenti di analisi come Image J, Neuron J.

Conclusione: ho costruito un framework computazionale per analizzare in modo quantitativo le immagini provenienti da microscopia ottica e confocale.

Durante i primi due anni del corso di dottorato ho selezionato, sviluppato e integrato protocolli e algoritmi bioinformatici per definire pipeline operative ottimizzate. Ho usato due set di dati di immagini come riferimento per ottimizzare la pipeline computazionale.

Nell'ultimo periodo del corso di dottorato ho applicato il framework ottimizzato su immagini al microscopio per il rilevamento e l'analisi dei neuriti. Tramite l'applicazione del framework è stato possibile ridurre considerevolmente i tempi del biologo per analizzare le immagini.

Chapter 1

1. Introduction and Background

The adult human body contains an estimated $3.72 \cdot 10^{13}$ cells [1] that can be grouped into about 200 specialized cell types making [2] the complex tissue architecture that forms 60 organs [3].

Cells are the building blocks of the human body, and are responsible for taking the nutrients from the food and converting those nutrients into energy. Furthermore, with the developments in molecular biology, several molecular components and mechanisms have been revealed by which embryos are patterned; therefore, require cells to acquire different fates and tissue adopt their defined shapes. In particular, careful experimentations and quantifications, mathematical and computational modeling are increasing day by day to formulate the assumptions and to test the hypothesis, although a lack of appropriate methods and tools to represent multiscale multicellular systems hampers these studies.

In a normal living cell, all activities originate in the cytoskeleton which plays very important active roles. The cytoskeleton study is a very dynamic area of research from the past 70 years. The dynamic 3D structure popularly know as “cytoskeleton” is a system consisting of three main kinds of protein filaments crucial for the cell shape, division, and function within all three domains of life [4]. Important questions have since opened up at the interface of cell biology, system biology, molecular biology, tissue morphogenesis, and computational biology. Can we identify physiological changes and characterize the cellular biophysical properties such as density, connectivity, solidity, stiffness in drug treatments? Which methods and techniques we can use to represent the large image dataset of the cytoskeleton? Using tools and ideas from the computational biology and system biology approach, this interface can be understood.

On the one hand, the new techniques are required to manipulate and visualize these large image datasets of the cytoskeleton. Upcoming technologies such as high throughput screening [5,6] gene expression analysis [7,8] quantitative image analysis [9–11] are essential to produce the high-resolution quantitative data on these dynamics across the spatiotemporal scales. On the other hand, new methods for the mathematical and computational modeling are much needed to investigate the consequences of such a feedback; generally, the behavior of the complex dynamics systems goes far beyond the human intuition. The large amounts of quantitative data that are being acquired are prompting scientists to new hypothesis explaining the complex biological processes. Therefore, expressing these hypothesis regarding to the rules and equations which are simulated by the computer allows to investigate the non-intuitive consequences of the complex interactions and feedback loops. In addition, quantitative comparison of the simulation results with the large data enables one to narrow down the set of the possible hypothesis to a few plausible one that will able to tested experimentally (fig 1) .

Dynamic models are normally ordinary differential equations of the genetic regulatory networks which can allow us to know the cell fate decisions of multipotent cells and how their trajectories through gene expression space are constrained by the interaction between genes [12]. Spatial models such as reaction-diffusion models enable us to reveal how the morphogen gradients influenced these decisions and moreover how the patterned gradient can spontaneously be aroused through the interaction between diffusible morphogens [13]. Finally, cell-based models in which the cells are introduced as discrete and motile agents, can allow us to grasp how specific tissue shapes emerge from the mechanical interactions in between the cells [14].

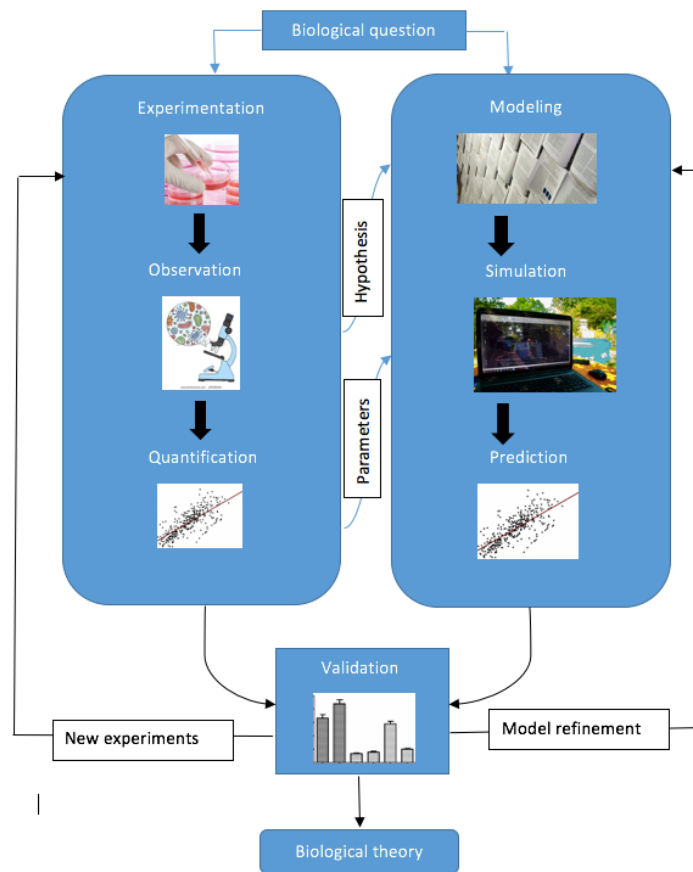


Figure 1: The representation of the computational modeling in quantitative biology. To answer in a unified way for the specific biological questions, observations and quantification of wet lab experimentations are followed by the construction of the mathematical and computational models that are expressed through hypotheses and assumptions. Moreover, these kinds of models are basically based upon first principles and/ or on hypothesis which derived from the new experiments and observations. When the quantification of the large experimental data is used for the parameterization, simulations to generate predictions that can be used directly for the statistical validation against experimental data. The validation may be imposed model refinement or suggest the data acquisition of new experimental data. Through the immense repetitions between the workflow of experimentation, quantification, and modeling which

increase the confidence in the validity of the model assumptions leads to the formation of the new biological theory.

In this thesis, a framework is introduced, which facilitates the exploration of such an interplay. Conceptually, this framework is based upon the integration of the methods to tackle some specific questions on the interactions between the cytoskeleton complexity and morphogenesis.

1.1 System Biology

System Biology (SB) deals with the living systems and the functions of the system that is in the functional context, for example – free energy transduction, or formation of the building blocks for the biosynthesis. However, the biological systems have evolved such that these functions can take place under a great variety of conditions: gene expression or even evolutionary changes. Cellular function and its capability to adapt are dependent on the interactions between the components of the system. Therefore, system biology used to understand the system behavior based on the interactions and characteristics of the cellular components and their microenvironment [15]. System biology is an approach that is widely classified into two types: a) Pragmatic system biology b) System – theoretic biology.

Pragmatic system biology reveals the use of large scale molecular interactions along with the omics approach, which is aiming at building the multidisciplinary network by implementing mathematical modeling. It is helpful to understand the system behavior based on the interactions and characteristics of the cellular components and their microenvironments [16]. System biology involves various applications of experimental, theoretical, and modeling techniques with the goal to study of living organisms at all levels, from molecular, through cellular to the behavioral perspectives. Various developments in this field make it possible to understand determine and perform multidimensional analysis of modeling and design of the cytoskeleton architecture from

experimental data and system-level analysis [17]. Furthermore, the computational models are used widely to build and make improvements based on the analysis and comparison with the data experiments [18]. A large number of ongoing research and technological developments are making it possible to manipulate the great challenge of exploration. Clearly, morphogenesis and phenotypic differentiation are space and time-dependent processes: [19] morphological plasticity, rather than being the result of genetic “adaptation”, and shows the influence of external physio-chemical parameters on any material system and is, therefore, an inherent, inevitable property of the organism [20].

System biology (SB) is a discipline designed to promote an integrated approach focussed on the redefinition of the main idea of a biological system, understood as an integrated unit of observable entities, interconnected by quantifiable relationships capable of expressing a biological function that has, within certain limits, apparent autonomy. The model at the center of observation is again the cell and cellular populations - along with its micro-environment - since in biology subordinate levels to this are incapable of meeting the indispensable requirements of the living: replication, evolution, and differentiation. SB's approach tends to promote an organic vision that is capable to combine complexity prospects (both bottom-up and top-down), which are not immediately assimilable to classical concepts of deterministic causality but, as in the physics approach, to the concept of probabilistic causality.

SB believes that the flow of info that directs the observed processes depends on the concurrence of molecular signals (DNA, proteins, lipids, etc.) and physical (system state parameters: forces, thermodynamics constraints) interrelating with a dynamic not linear. This makes it necessary to have data on which statistical assessments can be conducted. This is how made possible by high throughput techniques (genomics, proteomics, metabolomics, fractal analysis, etc.) that can provide large amounts of information. Biological organisms are the major entities that maintain their existence through the mutual interactions of its parts [21], i.e., they

are systems [22]. Therefore, this view has become the dynamic force for an approach to study the complex biological systems, which is known as system biology. While the system biology approach has a major influence on the field of molecular biology and cellular biology, it has the main impact on developmental biology. However, providing the current focus on the quantification and dynamic modeling in order to understand the developing systems of the cytoskeleton, the role of the system biology is becoming essential to the developmental biology.

In this thesis, system biology is regarded to as a support discipline in the computational strategies to gain insight into the phenotype of the original and treated exposed cells. System biology is used for the quantitative collection, analysis, and integration of cancer cells [23]. System biology heavily relies on the mathematics of dynamical system theory as well as computational simulation methods in order to thoroughly understand the dynamics of the system and its emergent properties that arise from the nonlinear interactions [24]. Hence, within system biology, different methodologies are applied to produce high- throughput data that can be used to understand cellular phenotypes. In particular, SB highlights the detailed quantification of the experimental data in order to make the quantitative computational models that can be able to provide the testable predictions and able to validate these models against the experimental data. Owing to this fact, the integration of the huge amount of the data generated is the key challenge that system biology is facing in order to understand the establishment of a holistic view of the functioning of the cell.

1.2 Cytoskeleton – I- beams of the cells

In the earliest 1900s, several biologists started to explore the amazing deformations that some cells could undergo. They have assumed that the mechanical properties of those cells generated by a specific scaffolding of subcellular components, which is commonly known as the cytoskeleton by Paul Wintrebert in 1931.

One of the main discoveries in this regard was the characterization of the three-dimensional protein network, called cytoskeleton, which supports the various cell compartments and connects them mechanically and functionally one to another. The cytoskeleton is made up of a network of filamentous proteins that can dynamically polymerize and depolymerize. This network consists three types of structural elements that have different dimension and different molecular nature: microtubules (having a thickness of 25 nm and formed by the combination of alpha and beta-tubulin molecules), intermediate filaments, (which have a thickness of 8-10 nm and can be formed from various types of filamentous proteins such as cytokerite, vimentina, desmin, and laminae) and microfilaments (having a thickness of 5 nm and are formed by the association of G-actin molecules). One of the significant functions cited for the cytoskeleton was that of mechanical support of the cell: it was clear from the very beginning that the cytoskeleton elements were to ensure the maintenance of the cellular form in coherence with its differentiated state, to support and maintain core position and suspend the various organelles within the cytoplasm. However, the static view of the cytoskeleton that the study of cellular biology has profoundly and rapidly changed, pointing out how the cytoskeleton actually acts, and above all, to link the various cell compartments and thus virtually participate in active mode to most of the functions performed by the cells.

The cytoskeleton is composed of multiple strands of 3 main components microtubules, intermediate filaments, and microfilaments [25]. Each of these polymers has very different geometrical, biochemical, and mechanical properties. But some of the cell have more specialized cytoskeleton filaments [26]

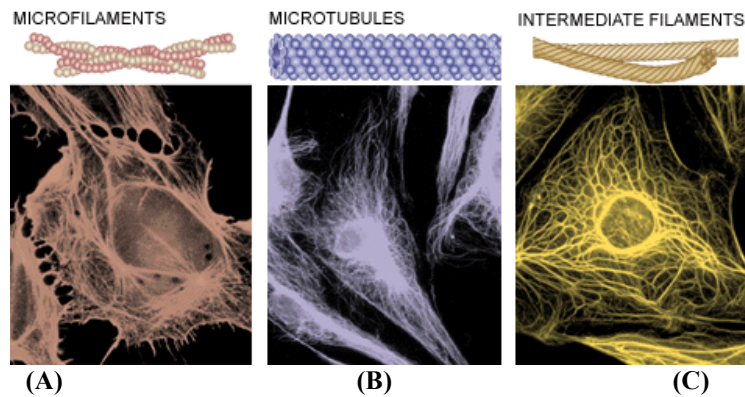


Figure 2: Image of cytoskeletal components (A) microfilaments, (B) microtubules, (C) intermediate fibers

A systematic picture of a typical cell with its constituents is presented in Figure 3 along with a photo of a fibroblast cell with cytoskeleton components observed. In figure 3, the numerous components have their own responsibilities. For example, the main intracellular entity cell governing not only the mechanical behavior but also the cell shape, in a very active manner, which is an interconnected network of polymerized filaments called cytoskeleton.

Interestingly, the cytoskeleton is a cellular scaffolding that is bathed into the cytoplasm, and it is responsible for the structural integrity and mechanical force generation into the living cells. It also provides the cell with mechanical stability; it also helps in connecting the different parts by serving as tracks for the transportation of the proteins and vesicles and also directs the intracellular organization through the positioning of the organelles. Moreover, the cytoskeleton also plays a crucial role when cells rearrange their internal components as they grow, divide or adapt to changing the circumstances. Therefore, these processes are accompanied and driven by the rearrangement of the cytoskeleton.

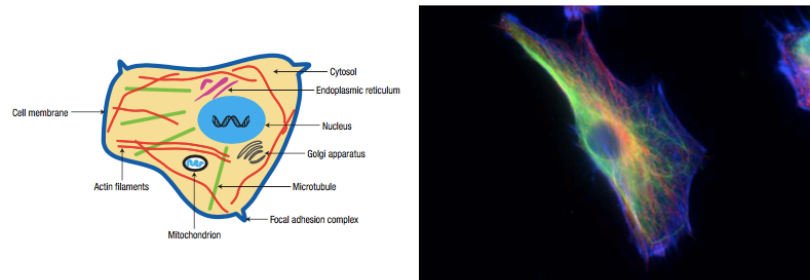


Figure 3: A) The demonstration of a cell with its prominent constituents reproduced from [27] with permission from Macmillan Publishers Ltd. B) Fibroblast cell is visualized through the immunofluorescence staining under the microscope as - Zeiss AxioVert 40 CFL microscope, using Zeiss AxioCAM MRm digital camera as well as AxioVision software (Carl Zeiss AG) [28] the cytoskeleton networks are colored as blue for actin, green for intermediate filaments (vimentin) and red for microtubules.

Functionally, the cytoskeleton helps to maintain the cell shape, and also crucial in the movement of cellular components, segregation of chromosomes during mitosis and also in the development of a sensing network for mechanotransduction. The eukaryotic cytoskeleton is made up of three basic types of filaments, along with their associated proteins. Cytoskeleton filaments are interconnected with each other, and their functions are coordinated by hundreds of associated cytoskeleton accessory proteins [29]

The functions of the various components of the cytoskeleton classically can be schematized as follows:

- Microtubules are the third type of the cytoskeleton filaments, which is produced from the tubulin into a hollow cylinder with an outer diameter of roughly $25\mu\text{m}$ and is, therefore, a far higher bending stiffness than that of either actin or intermediate filaments [30]. The persistence length is so far more, about 6mm [30,31]. Along with the high bending stiffness and the rod-like appearance, they are also useful in supporting the creation of the long slender structure. Microtubules are often found in an arrangement in which

they radiate from the center of the cell [2]. They are very highly dynamic, allowing for the remodeling and adaptation of the cell structure [30].

Microtubules are liable to vesicular traffic and mostly to the movements of all diaphragm organelles; form the meiotic and mitotic cell melt, as well as the "midbodies" characteristic of the late phases of cell division; Also represent the structural and functional support of membrane expansions with movement such as cilia and flagella. Microtubules are assembled into hollow tubes to both sides of protofilaments [32]. The term tubulin is a microtubule subunit, which is called as an α/β heterodimer [33].

- Intermediate filament [34] are responsible for maintaining the cellular form in coherence with the differentiated state of the cell, as well as maintaining the shape and position of the nucleus [35,36].; They are also involved in the development of intercellular junctions and cell junctions and the extracellular matrix, thus providing the important chemical and physical information about the microenvironment in which they are immersed. The different kinds of Intermediate filaments share a similar type of basic characteristics and provide tensile strength of the cell [37]. Intermediate filaments are the family of various protein of which the expression is depending upon the type of the cell, and its functions such as endothelial cells express different types of keratin and vimentin is outstanding into the mesenchymal cells such as fibroblasts [38]. Furthermore, IFs are more resistant to high salt concentrations and much more stable than the actin filament or microtubules [30,39]. Generally, cancer cells have been observed to exhibit a disrupted intermediate filament network, which is collapsed around the nucleus along with the different mechanical properties.

- Microfilaments are located at the cortical cell cytoplasm level and are involved both in the creation of joining cellular structures and in cell migration as they are responsible for the formation of membrane expansions (filopods and lamellipods) necessary for the cell

movement; microfilaments also mechanically support real-world expansion such as microvilli that have absorbing or exchange function with the outside environment but which can function as "mechanical sensors" when exposed to fluids; finally, this type of cytoskeleton is involved in the plasma membrane deformation processes associated with the formation of endocytosis and exocytosis vesicles and the formation of cytoplasmic bridges [40,41].

While the previous scheme suggests a clear separation of functions between the various cytoskeletal components, it is good to keep in mind that the dynamic modification of the structure of one of the types of filaments of the cytoskeleton necessarily leads to a modification, albeit partial, of the set of other components: we can say that elements of the cytoskeleton act as an integrated system. For example, during mitosis, the shape of the cell changes radically, though the mitotic fusing is substantially formed by microtubules, while the cellular division also observes the change in the tonal filament and microfilament concept. The same considerations can be extended to cellular motion: although we know classically that filopods and lamellipods are related to changes in the actinic cytoskeleton, we cannot generate cell migration without profoundly altering its adhesive shape and shape, thus involving all cytoskeletal components in this event. Even vesicular traffic is an exemplary example of the integrated action of the cytoskeleton: the vesicles are transported on "tracks" of microtubules, but fusion with the outer membrane or with the internal membrane compartments also depends on the arrangement of the other cytoskeleton components. Finally, it is now known that elements of the cytoskeleton are involved in "signal transduction" within the cell, and therefore contribute substantially to modifying not only adhesive/migratory but also metabolic/functional behaviour. Signal transduction was initially studied regarding the stimulation of hormone cells or growth factors. However, it is now clear that cytoskeletal elements actively participate in this function and can activate it autonomously

in response to signals from the "mechanical" variation of the micro-environment.

It is clear that in order to study the structure of this three-dimensional network of protein filaments, it is necessary to evaluate "in situ" the use of protocols to maintain the structural integrity of the cell. This was possible thanks to the acquisition of cellular fixation techniques, which are the basis of most optical and electronic microscopy techniques. These techniques include the use of substances (in most cases, alcohols or aldehydes) that allow the structure of the molecules that make up the cells to be maintained, in conditions as representative as possible of what is happening in vivo.

1.3 Measurable parameters of cytoskeleton system

Image analysis approaches were used to measure a cytokeratin filament network visualized by immunofluorescence and confocal microscopy. There are two methods as follows that are used to segment sets of curvilinear objects.

1. **Mesh approach** – This approach is based upon the classical methods of the mathematical morphology, takes into consideration of global network topology.
2. **Filament approach (novel)** – This approach is based upon the individual element morphology.

Both methods, as well as their combination, will allow the calculation of several features at two levels of geometry: Global (network topology) and Local (filament morphology) [42].

Topology: It is known as the study of geometrical properties as well as the spatial relations which are unaffected through a continuous change of shape or the size of the figures. Generally, the physical layout of the network is called topology [43].

Network Topology: The collection of all the network elements can't be itself constitute a network, however, there must be something equally real to make it function as a the whole. The way

in which the elements are “wired together” must also be specified in some rigorous manner. This pattern of the connections is called the network’s topology.

Table 1: The table representing the parameters to describe the topology and morphology of the Cytoskeleton [42]

Global parameters (Network Topology)	Local parameters (Filament Topology)
Network (degree) connectivity, Number of connection points (Ncp). Connecting index (Ic).	Filament (s) morphology, Filament shape, Polarity
Network density, Total length of detected filaments (Lf), Number of regions (Nr).	Length
The mesh structure, Areas of regions (Ar), Mean length of filaments, enclosing a region (Lr), Mean elongation factor (Ef).	Thickness, mean length of filaments enclosing a region ($2x Lf/Nr$)

1.4 Quantitative methods to measure cytoskeleton parameters

The possible survey methods for analyzing the structures so far described are based upon various microscopy technologies, each featuring specific strengths. In particular, we can highlight optical microscopy, confocal, and electronic microscopy.

1.4.1 Optical Microscopy

Optical microscopy [44] is a technique of observation capable of producing magnified images of objects or details of them, too small to be observed with a naked eye; The instrument used to perform

this technique takes the name of an optical microscope. The simple microscope is a converging lens between the eye and the object to be observed so that the latter is in the intermediate position between the focal plane and the lens itself. Under such conditions, the lens provides a straightforward and magnified virtual image of the object, even if it is at a distance from the lower eye to the next point. The maximum magnification obtainable with a single lens is 8-10 times (8x-10x).

Larger magnifications can be obtained thanks to the composite microscope, a system of two convergent lenses, respectively, lens and eyepiece. The object to be observed is placed in front of the lens that provides a real, upside-down, and enlarged image. This image is dropped at an appropriate distance in front of the eyepiece, which provides another image, virtual, magnified, and oriented in the same direction. 100x magnification is only possible with the technique of "immersion in oil bath."

In the cell culture study, optical microscopy is used at various enlargements to qualitatively and quantitatively evaluate the morphology of cells and their displacement, as it appears on the focus plane. Small magnifications are initially used to test cell culture, count, cluster, or aggregation display for the "broad-spectrum" statistical significance of the under-investigation phenomenon. As the magnification and resolution power increases, it focuses on the specific regions of interest (ROI) that the researcher feels useful to visualize the expected effects due to a specific treatment of the biological sample as compared to the controlled, untreated sample. The overall, qualitative and sometimes quantitative evaluations are closely linked to the researcher's ability and experience.

1.4.2 Immunofluorescence microscopy

Immunofluorescence microscopy [45] is generally used to define the cellular distribution of both the soluble and structural proteins. It is an essential technique for the determination of the distribution of the proteins within the cells. It is also useful to extract the information

on the location and function of proteins that are not well understood. One of the major application of immunofluorescence microscopy is the fixation conditions beneficial to prepare the cells for the antibody which truly preserves the in vivo distribution of the antigen of interest [46] Another application of the immunofluorescence microscopy is to visualize subcellular localization of proteins in the coat morphogenesis and chromosome as in packing throughout the development of sporulation in *Bacillus subtilis* [47].

1.4.3 Confocal Microscopy

The contemporary evolution of optical fluorescence microscopy and molecular biology has allowed to develop immuno-marking techniques which, preceded by fixation protocols, allow "in situ" observation of the distribution of molecules within the cells. Specifically, these experimental protocols provide for the use of specific antibodies that identify molecules of interest within the cells, even when they are part of cellular structures whose dimensions are below the optical microscope resolution power. A detailed study of the cytoskeleton plan would allow us to have not only information about the cell structure but also about the functional activities the cell itself is carrying out. A significant boost to the study of "in situ" cellular biology has been possible thanks to the development of confocal microscopy [48]: a system that uses a microscope capable of gathering information at various depth levels and giving a three-dimensional reconstruction of the analyzed fluorescent tracer. However, this technology is not used in laboratory practice to the utmost of its potential. Indeed, there is always a compromise between the detail that can be obtained for a reconstruction that involves the acquisition of a great number of sequential images and the time required for such acquisition and processing. These studies allow to perform morphological evaluations on a very small sample and on interpretative bases that may be partly subjective and may change from experimenter to experimenter. Hence a quantitative and qualitative analysis of the entire cell population is virtually never carried out, losing valuable

information that could give us new perspectives for cellular behavioral knowledge. Hence, it is necessary to develop new innovative technologies to optimize the use of the various microscopes and analyze simultaneously the information derived from a large number of different images and possibly to link the observations made using the different types of microscopy.

The Confocal Laser Scanning Microscopy (CLSM) is an optical microscopic technique that was applied to generate high resolution two dimensional and three-dimensional images, projections as well as three dimensional of autofluorescent objects. The application is widely distributed and related in such disciplines like petroleum geology, marine geology, archaeology, cell biology, etc. [49]. CLSM has low-cost computer data processing and imaging system with the availability of laser light sources. It is popular in the field of life sciences and semiconductor device technology. CLSM is also useful in characterizing the amorphous structure and also identify the gas-solid coexistence in deionized suspensions [50].

The figure 4 showing the principle for confocal microscopy. The CLSM using the fluorescence mode can collect the emissions originating from the interior of a biological sample, and hence CLSM has been extensively used to map intracellular mechanisms. When antibodies were used to stain the specific biological structures, fluorescent confocal microscopy provides a powerful way to simultaneously map the distribution of the different cellular components. The cells here were imaged using a confocal (Leica DMIR2) microscope.

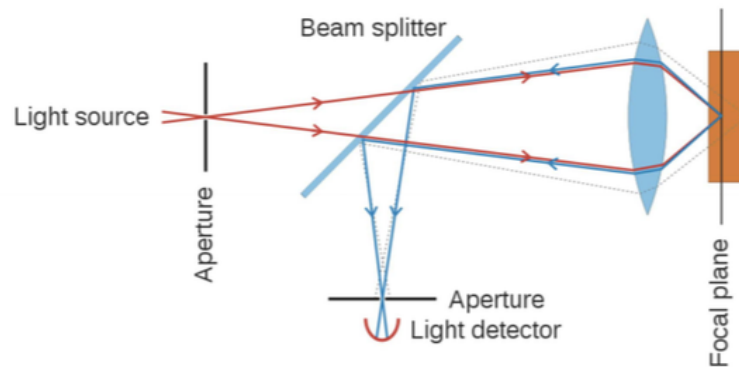


Figure 4: The figure presents the principle of confocal microscopy.

1.4.4 Electronic microscopy

Electronic microscopy [51] was certainly the first technology to give structural knowledge about the existence of the cytoskeleton since cytoskeletal filaments all have a caliber less than the resolution power of a conventional optical microscope (0.2 microns). In particular, if electronic scanning microscopy (SEM) has provided very detailed information on cell surface characteristics (filopods, lamellips, cilia, flagellas, microvilli, etc.), electronic transmission microscopy (TEM) due to its extraordinary resolution power (0.2 nm) allowed for the first observation of cytoskeleton elements.

If the cytoskeletal morphological/ultrastructural study was possible thanks to electronic microscopy, knowledge about the nature of the structural proteins of the cytoskeleton was possible thanks to molecular biology. Conventional molecular biology techniques, however, require the lysis of the cells that they want to study and, therefore, the loss of structural integrity of the same, so having a "proteomic" analysis of a cell population prevents, by chance, the possibility of simultaneously perform morphological observations on the same cell population.

TEM is a principal technology in the field of cell biology to make several contributions for over 50 years. TEM is valuable in the analysis of the cellular components, for example, cytoskeleton, membrane systems, organelles, cilia, and flagella, as well as several specialized structures in differentiated cells, for example, microvilli and synaptonemal complex [52]. TEM is also useful in the study of viral and bacterial infections and understanding their knowledge [53]. TEM was used to visualize nanocrystals (NCs), which is able to find in the crystallization drops [54].

1.4.5 Atomic Force Microscopy

Atomic Force Microscopy (AFM) [55] is a greatly involving technique for the exploration of the biological systems at the nanoscale. Biological samples can be viewed with AFM in two of the imaging modes: first is contact mode, and the second is tapping mode. AFM has been used for a long time in many scientific fields because of the study of the surface structure at nanometer to sub-angstrom resolution in an ambient and liquid environment. The application of AFM has a unique place in pharmaceutical research. Also, AFM has several advantages over SEM/TEM for the characterization of nanostructures. It also presents three-dimensional data so that it is possible to measure the height of the nanoparticles quantitatively. One of the characteristics of AFM is that it is a very cost-effective microscope for nanoscale imaging [56].

TABLE 2:

Novel techniques to study cytoskeleton are as follows:

S.No.	Visualization Techniques	Major contributions	Advantage	References
1.	Atomic force microscopy (AFM)	It is used in material science and widely used applications in biological sciences.	Used to probing the extracellular matrices and the mechanical properties of the cell including receptor-ligand interactions.	[57]
2.	Transmission Electron Microscope (TEM)	It is used to study the biological structures and to understand how the images formed.	Used to study the minute structures by taking the electron micrographs of thin sections.	[58–60]
3.	Immunofluorescence microscopy.	It is used in modern biology and medicine developed by Coons et al in 1950.	Used to develop the tissues and analyze the distribution of proteins and small biological and non-biological molecules.	[61–63]

4.	Confocal Laser scanning microscopy	It is used to produce blur-free images of thick specimens at various depths	Images are taken point by point and reconstructed with a computer, rather than projected through an eyepiece.	[64,65]
----	------------------------------------	---	---	---------

1.5 Tools to analyze Cytoskeleton morphology

1.5.1 Cell profiler to quantify and identify the cells parameters

Cell profiler is designed for biologists. It is used for the quantitative analysis of biological images. It is free and open-source software for the study of cellular functions. It is a flexible software tool and measures the cell size, shape, intensity and texture. Cell profiler can be downloaded from the website <http://www.cellprofiler.org>. It runs on various platforms Windows, Mac, and Unix. Cell profiler is a freely available image analysis software that can able to handle hundreds of thousands of images. In cell profiler, the analysis takes place the concept of a pipeline of individual modules for the image analysis. Each module works in a sequential order to construct a pipeline. Many modules are automatic, but the software adjusts each module appropriately [66].

1.5.2 Image J – the technical advance tool

Rasband develops image J software. It is one of the main scientific image analysis programs. To facilitate more advanced and precise results, the program provides plugins, which is necessary to add functionality and diverse visualization. This program is having large

bio formats used in over 30, 000 laboratories. In this way, image J is a very popular research tool helpful in imaging processing and can run in any web browser [67].

Moreover, both NIH image and image J used to keep the program very simple with no complex user interface. When we open image J, the single toolbar appears, and from this straightforward interface, all of the proficiencies of the Image J can be found and used. There are over 500 plugins that consist of a wide range of functions accessible on the Image J website. Most of the plugins are now distributed along with the core of the Image J [67].

Image J is called an image analysis program that is widely used in the biological sciences and beyond it. Owing to its ease of use, recordable macros, and extensible plug-in architecture, image J appreciates contributions from non-programmers, unprofessional programmers, and professional developers alike. Image J is a very powerful platform for image processing, which is developed by Wayne Rasband at the National Institute of Health (NIH). Since when its initial release in 1997, it has proven the paramount in many of the scientific endeavors and projects, and particularly those which come inside the life sciences. There are so many existing extensions – plugins, macros, and scripts for the image j application, which have proven very useful to the user community [68].

Interestingly, image J software is an open software platform that had a significant impact on the life of sciences and continues to do. Image J is growing significantly because of freely available and its lively and helpful user community. There are several scientists, interested biologists, technical assistants, students, and scientific staff who use Image J on a daily basis, and can exchange knowledge through a dedicated mailing list. The uses of image J is ranging from data visualization and teaching to the advanced image processing as well as statistical analysis. From the technical point of view, image J provides a very useful interface along with the function to load, display and save images, the basic image processing functionality like- convolution filters and extension mechanism which includes support for macros and plugins [69].

1.5.3 ICY TOOL

Icy is an open community platform software for bioimage informatics. It offers the software resources to quantify, visualize, and annotate the bioimaging data. Icy comprises a flexible framework that is designed around a plugin architecture; it also offers rich data structure for the biological images and output data, enhanced data visualization. Icy is open-source software that can address several issues by providing a comprehensive framework for the extensive reproducible research in two bioimage informatics.

Moreover, the icy internal structure has a core layer that provides core functionality a plug-in layer which can implement the application-oriented modules and a plug-in management engine that can ensure plug-in interoperability. Plugin development takes advantage of a rich application programming interface (API) that can facilitate the development of the specific algorithms, augmented visualization tools, a database that can access and other software components. This software uses the Bio format library to read more than 100 different file types. Also, icy uses the internal data format and offers over 80 different functions for plug-in to access the image data. Icy is bundled with a set of bioimage informatics tools, which is released on April 2011, having more than 100 plug-ins published online [70].

1.5.4 MATLAB

MATLAB is a fourth-generation high-level programming language developed by Mathworks. MATLAB allows many functions like matrix manipulations, plotting of function of the data, implementation of the algorithms, creating the user interfaces, interfacing the programs written in the other languages such as C, C++, Java and FORTRAN. MATLAB provides an interactive environment for the iterative exploration, designing and problem-solving, In addition, it also provides the built-in tools for the visualization of the data and tools for creating custom plots.

MATLAB nowadays is very much used as a computational tool in the field of science and engineering as well as physics, chemistry, math field [71].

Moreover, MATLAB is a high- performance language used for technical computing. However, it integrates for the computation, visualization, and programming environment. Although it is a modern programming language environment that comprises sophisticated data structures, consists of built-in editing and debugging tools, it also supports object-oriented programming. Hence, these factors make the MATLAB an outstanding tool for research and teaching. MATLAB consists of many advantages as compared to conventional computer languages like C, Fortran to solve the technical problems. Interestingly, this software package has been commercially available since the past from 1984 and which is now been considered as a standard tool in most of the universities and industries worldwide. MATLAB is also known for as powerful built-in routines, which enable a very wide variety of computations. MATLAB consists of its specific applications collected in packages, which are called the toolboxes. There are several toolboxes such as signal processing, control theory, optimization, simulation, symbolic computation and other fields of engineering and applied sciences [72]. MATLAB is available on UNIX, PC, and MAC platform. We can get Matlab by double-clicking on the Matlab icon or by typing Matlab on Unix on the command line. We can also get access to the MATLAB command line, which is denoted by “>>”.

1.6 Image analysis

Image analysis is a very critical phase in most of the medical and biological applications and automated image analysis has become the central tool for such applications in recent times. A number of the commercial high content cellular imaging systems include software (Imaris – 3D/4D image visualization and analysis software, Solis software, Metamorph and Infinity can be used to characterize neuronal projections such as axon and dendrites[73]. In image analysis, one of the most crucial step is image segmentation [74].

Segmentation subdivides an image into its components. It also differentiates the object of interest from the background. Moreover, global thresholding or thresholding is an essential and quick method for characterizing the image region which has the same grey value [74].

1.7 Filopodyan – image analysis open-source software

Filopodyan or **Filopodia Dynamics Analysis** is a program developed in R and Fiji [75]. It is an image analysis open-source software used for automated segmentation and filopodia analysis which quickly annotates the huge number of dynamic filopodia. Furthermore, it also used to measure the filopodia fluorescence along with the full generation and automation of the large dataset of the fluorescence intensity at the sites of membrane proximal while formation and at the tip.

Filopodia are the thin membrane protrusion which acts as an antenna for the cell to probe the surrounding environment. Moreover, filopodia are commonly found embedded within or protruding from the lamellipodia at the free front of the migratory tissue sheets. Filopodia have a very important mechanical and sensory role in the movable cells.

Filopodyan performs great on high-magnification, high-quality images in the cell types along with the clearly differentiated filopodia, in 2D time series. Filopodyan is used for detecting the neurites as well as the filopodia [75].

1.8 Neuron J – image J plugin

The neurite tracing method is fully manual delineation and was tested with four users who repetitively traced the chosen neuronal processes in 20 fluorescence microscopy images of the cells inside the culture. By analogy with the Image J, the authors called the tool ‘Neuron J’. It provides functions to label and tracing, to load and store the tracings, to calculate and display the dimension of each and every tracing and perform statistics on the measurement of all the particular tracings[76].

Chapter 2

2.1 Aims

Our first aim is to build an integrated framework for the quantitative assessment of cytoskeleton (CSK) morphology. That task implies that we should represent an in-silico model constructed by using measurable parameters obtained from cells imaging. This process is likely a translation from a ‘biological phenotype’ to an ‘electronic phenotype’. First of all, the quality of images provided by currently available technologies (confocal, optical and electronic microscopy) must be optimized in order to allow a reliable extraction of useful data. Currently, quantitative morphological appraisal of cells is usually restricted to only a few parameters (surface area, diameter), thus leading to a ‘reductionistic’ approach in grasping the morphological configuration. This approach is mainly disappointing when we are facing non-Euclidean structures – like ‘natural’ objects – that should be more realistically represent by fractal-based geometry. Therefore, we adopt a Systems Biology approach aimed at grasping the whole configuration of the living cell. This innovative model requires a great deal of integrated measures in order to identify the ‘observables’ (dependent as well as independent) that can exhaustively describe the dynamics of the system. By considering the cytoskeleton as the main driver of the cell shape, we assume to posit a correlation among CSK structure and cell shape parameters. Thus, we identified the following parameters describing CSK architecture: length, density, thickness, branching and hubs. Cell shape parameters belong to the following: roundness, solidity, fractal dimension, lacunarity, surface area, aspect ratio. Overall, those measures will be integrated in order to grasp the complexity of the cell morphology. To do that the cell morphology will be quantitatively assessed by means of fractal analysis, while CSK architecture will be evaluated by means of

coherency analysis. Coherency, indeed, has been proven to greatly differ from different cell phenotypes and it is currently deemed reliable parameters in following cell phenotypic transitions. These parameters will be included in an integrated and comprehensive in silico model for which specific protocols and algorithms will be defined. The in silico model will be used experimentally to measure the changes that occur in cell morphology after exposure to a specific treatment.

My second aim is to apply bioinformatic approach to evaluate in quantitative manner data from some experiment led by lab research team. I receive data about two experiments, cytospace experiments and the experiment of the breast microcalcification. In the last year of the PhD I receive data of the neurite.

From the various surveys of the literature, we have noticed that the earth's gravity is one of the major factors limiting of the cell size. It also influences the diversity of the cell types and size throughout in the biological evolution. In one of the literature [77]. Pavel E. Morez ask the question that what is the role of gravity in the life of the cell? Cells have evolved in the earth's field of gravity and it is very interesting to study how gravity affects the cells on their size, variety, division rate and so on. With the beginning of the space travel in the recent years which is connected with the weightless environments, it is very essential to know more about the effects of both the reduced as well as the increased (more than 1g) gravity on the development and growth of the cellular and multicellular systems [77].

The third aim is to develop the image analysis algorithms through Matlab to describe the cytoskeleton network. The quantitative analysis of the microtubules and nuclei pattern was performed by using various images – analysis methods. Indeed, the aim is to identify the modifications in the CSK network.

Moreover, in our experiment, we have preliminary combined confocal and optical microscopy image analysis to improve the quantitative analysis of cytoskeleton architecture of the breast cancer cell line (MCF – 7).

The fourth aim is that the neurite numbers and length measurement of PC 12 cells were analyzed using the following software tools : (a) custom made tool MATLAB (b) Cell profiler (c) manual workflow in image analysis and (d) Filopodyan. We have shown some algorithms which are useful for analyzing the changes in the neurite and length. A comparison of the different software and tools suggests identifying the length of neurites can be achieved using image analysis.

Chapter 3

3. Results

In this chapter, we present the results achieved with our framework on the given dataset. We have worked upon the dataset of the cytospace experiments and neurites.

The complete analysis is taking place into 2 parts: analysis by image J and analysis by MATLAB. In the third part of the thesis, the neurite analysis takes place.

We now begin the analysis of the dataset with the flow of pipeline in the framework and showing how the results produced by image J.

3.1 A confocal – based quantitative analysis in the cytoskeleton architecture of MCF -7 Cells:

MCF – 7 is a breast cancer cell line derived from pleural effusion in 1970. The cell line is displaying 85 chromosomes. These cells tend to grow the colonies. MCF – 7 cells are capable of forming tumors in nude mice. These cell lines can easily grow in soft agar and having low to moderate activity. Hence, MCF – 7 cells have a low invasive capability in vitro.

A great variety of cellular alterations has been observed during the microgravity in the MCF – 7 cells.

The result section is divided into three parts – in the first part the cytoskeleton is analyzed by IMAGE J and in the second part, the cytoskeleton data is analyzed by MATLAB programming. The results are differentiating from one another. And in third part, the neurites data is analyzed by different tools and softwares.

3.2 Application of bioinformatics to analyse confocal image:

To explore the CSK morphology I received a preliminary data set from cytospace experiment.

A total of three experimental conditions were approached to participate in the cells, one subjected to Treated and Untreated with melatonin on ground; another subjected to treated and untreated with melatonin in space, and cells in space without melatonin at docking time. To build an integrated framework from the cytoskeleton image series we followed a step-by-step procedure: we split the color channels of the cytoskeleton confocal images to separate microtubules and nuclei. From the image J, we refine and adjust the threshold by the top and bottom slider. We calculate the threshold of each confocal image. Thus, projecting and applying the measurement properties to analyze the area, shape, descriptors, perimeter, mean, median, standard deviation. This results in a description of significant changes of each cytoskeleton nuclei at different conditions on the ground as well as in space and docking. The steps will be repeated for all the images of the nuclei and microtubules separately.

We identify the following parameters describing CSK architecture: length, density, thickness of microtubules. Descriptive nuclei shape parameters that belong to the following: roundness, solidity, fractal dimension, lacunarity, surface area, aspect ratio. Thus, to define the given parameters which are significantly higher or lower than predictable by chance, analyses were performed within all cytoskeleton structure. Differences were detected for ground, space, or docking nuclei experimental conditions.

3.3 Pilot Study for cytoskeleton analysis

We apply computational analysis to measure CSK morphology variation among the controls, docking, and space experimental conditions.

The objective of this study phase is not to investigate the real effect of microgravity on cells, but to develop a calculation model that takes into account all the parameters of interest. On the other hand, the images received are only a small set of images relating to the

experiment, representative from a point of view of image analysis but not sufficient to be able to formulate any scientific hypothesis on the possible biological meaning.

Here, I have put the formula to calculate each parameter.

For area of the cytoskeleton as: area of total number nuclei/area of total number microtubules.

Area of cytoskeleton (A1) = $a1/a2$

A1 = Area of the nuclei/Area of the microtubule

This formula is applied to calculate each parameter

For the roundness of the cytoskeleton as: roundness of nuclei/roundness of microtubules.

Roundness of cytoskeleton (R1) = $r1/r2$

R1 = roundness of the total number of nuclei/roundness of total number of microtubules

And so on for other parameters as well...

3.3.1 Formula for the measurement of the cytoskeleton

- Area: The selected area was measured and calculated in square pixels in calibrated square units (e.g., mm^2 , μm^2 , etc.).
Note: To measure the area of the cells the function in the image J as: analyze---set scale parameter was used to spatially calibrate the image.
- Roundness: With this parameter, we have calculated the roundness of the nuclei, which gives the value directly relative to the aspect ratio and circularity.
 $4 * [\text{area}] / [\pi * [\text{Majoraxis}]^2]$
- Perimeter: The measured length of the outside boundary of the selection.

- **Circularity:** It was well defined as the ratio of the area and perimeter with the value 1.0, which is, signify the perfect circle. As the value is reached to 0.0, it shows an increased elongated shape. All the values might not be valid for the few particles.

$$4\pi * [\text{area}] / [\text{perimeter}]^2$$

- **Solidity:** Through this parameter, we are measuring the convex area selected by the specific command.

$$\text{Solidity: } [\text{Area}] / [\text{Convex area}]^2$$

- **Aspect ratio:** In this parameter, the aspect ratio of the particle fitted in the ellipse.

$$\text{AR} = [\text{Major axis}] / [\text{Minor axis}]$$

In this, the fitted ellipse is selected as the major and minor axis which are observed in the nuclei.

3.3.2 MCF -7 Cytoskeleton results calculated by Image J Software:

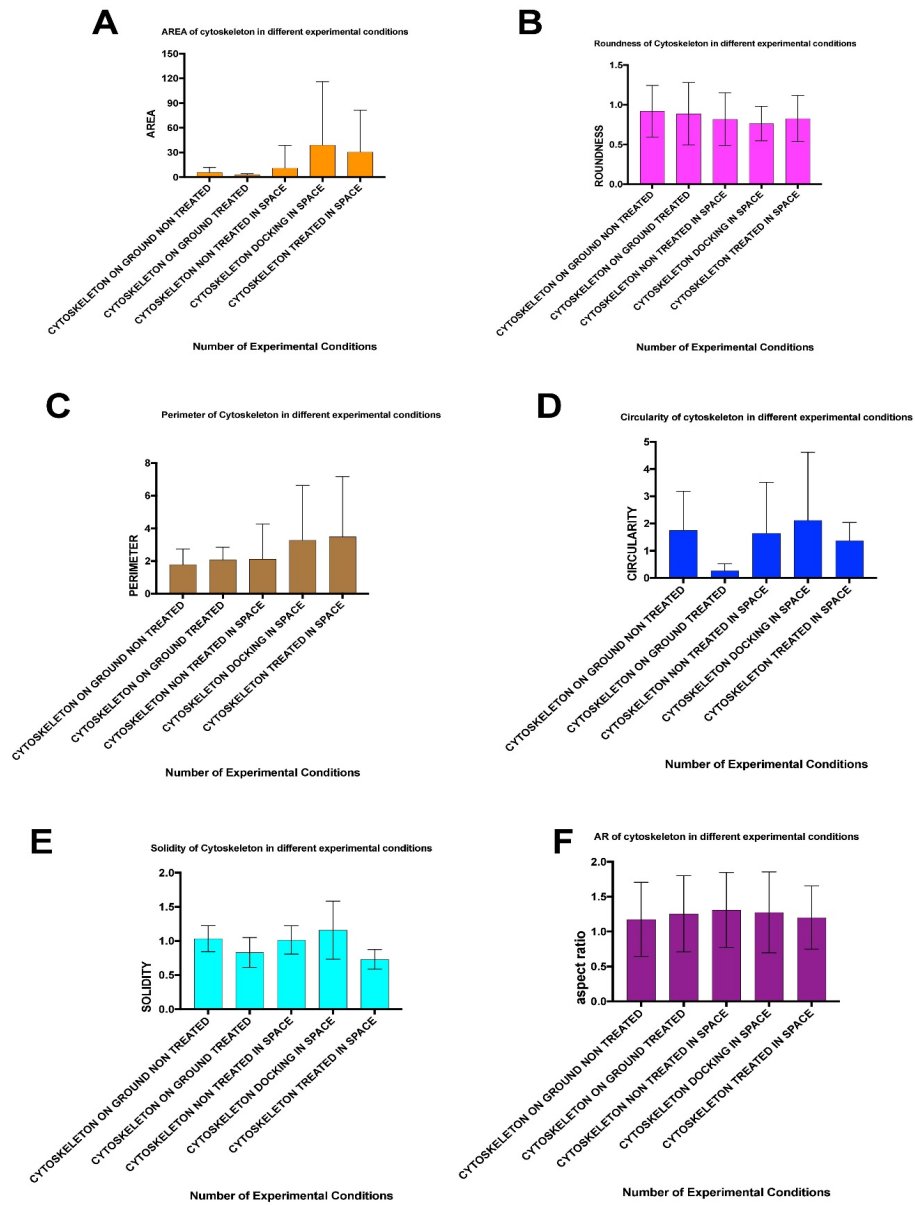


Figure 5: The Confocal microscopy analysis of the MCF -7 cytoskeleton structure. Qualitative analysis of MCF-7 cells: All bars representing the mean and standard

deviation. Mean and standard deviation are shown for each experimental group. (A) The cytoskeleton area (area of total number of nuclei/area of total number of microtubules) was calculated by Image J. (B) The cytoskeleton roundness (roundness of total number of nuclei/roundness of total number of microtubules) was calculated by Image J. (C) The cytoskeleton perimeter (perimeter of total number of nuclei/perimeter of total number of microtubules) was calculated by Image J. (D) The cytoskeleton circularity (circularity of total number of nuclei/circularity of total number of microtubules) was calculated by Image J. (E) The cytoskeleton solidity (solidity of total number of nuclei/solidity of total number of microtubules) was calculated by Image J. (F) The cytoskeleton aspect ratio (aspect ratio of total number of nuclei/aspect ratio of total number of microtubules) was calculated by Image J.

3.3.3 Discussion:

Our analysis providing the new horizons into the diversity of cell morphology and suggest the different experimental approaches for future study. Besides, coherency is being defined through the structure tensor, which evaluates the local orientation in a small region of a confocal image.

In figure 5 (A) showing the area of the cytoskeleton in different experimental conditions i.e., on the ground, in space and during docking with mean and standard deviation. Figure 5 Image (B) showing the roundness of cytoskeleton including nuclei and microtubules in three different experimental conditions. Image (C) showing the perimeter of the cytoskeleton in three different conditions. Image (D) showing the circularity of the cytoskeleton. Image (E) and (F) showing the solidity and aspect ratio of the cytoskeleton in all three experimental conditions which is not so much changing according to the graphical representation.

Fractal Dimensions: Fractal is a never-ending pattern. These are infinitely complex patterns that are self-similar across different scales. However, they are used to create by repeating a simple process over and over in an ongoing feedback loop. Moreover, fractals are very complex, and they can be made by repeating a

simple process[78]. It is well known as a non-regular geometric shape that contains the same degree of non-regularity on all scales [79]. Fractal dimension was known to be firstly introduced by Mandelbrot. Fractal dimension usually describes the shape and appearance of an object, which consists of the property of self-similarity. The fractal dimension of many objects is usually calculated by using the concept of self-similarity[80].

The fractal dimension (FD) is a real number which is used to define the geometric complexity of a fractal. One of the ways used to quantify FD is the box-counting method. It is used to covering of the image with the non-overlapping grid which is made up of the boxes and then counting how many boxes of the grid were covering our image. After that, this method is repeated by iterating the process using the finer grid. Hence, in the end, the pattern of $N(r)$ changes with r which will be obtained, as represented in the graph. By using a logarithmic scale, a linear regression model will be used to fit the line. The FD value is given by the slope of this line. However, the input can be a 1D segment, a 2D image, 3D volume. In particular, smaller squares will pick up more detail, which will give a better approximation of the shape ($N(r)$ squares of the side length r). In this sense, the more irregular surface, the higher the value will be.

Box counting dimension is the typically used method which is in the characterization of irregular sets of the science.

We have performed our analysis with a standard box-counting method using FracLac which is based upon the multiple origins of the ROI. With the use of the box-counting method, we have defined the largest and the smallest box sizes.

$$FD = \lim_{\varepsilon \rightarrow 0} \left[1 - \frac{\log [L_{\varepsilon}(C)]}{\log \varepsilon} \right],$$

Here, C = considered curve, L = Length of the curve C , ε = length of the segment used as unit to calculate L .

Along with this, we have processed the images properly and calculate the linear relationship when plotting the graph $\log(N(s))$

versus $\log(1/s)$ in order to extract the fractal dimension with the minimum error, here s is the linear size of the square or cubic mesh. Moreover, the box-counting algorithm thus counts the number $N(r)$ for different values of the r and the plot of the log of the number $N(r)$ versus the log of the actual box size r . Also, the value of the box-counting dimension that is D is estimated from the Richardson's plot best fitting curve slope [81].

$$-D = \lim_{r \rightarrow 0} \frac{\log N(r)}{\log r}$$

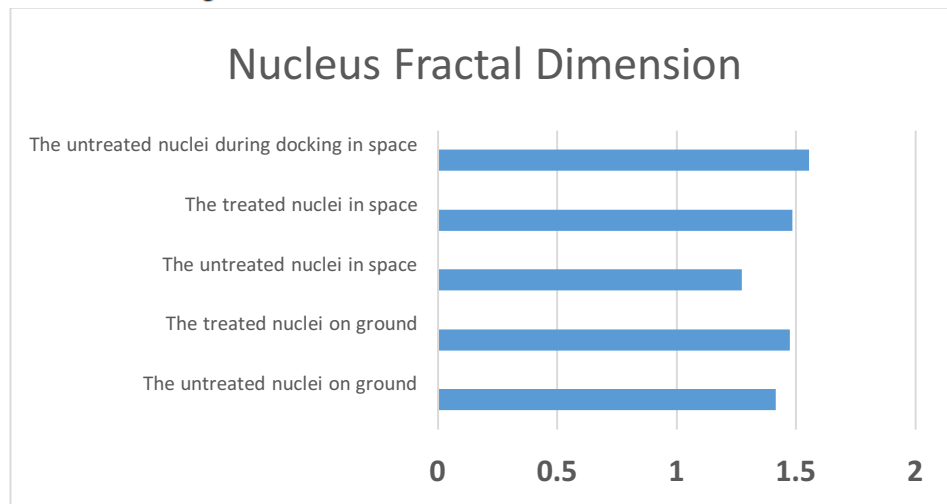


Figure 6: The fractal dimension of MCF-7 nuclei. The graph is showing the mean values of fractal dimension experimented on ground and space obtained by calculating from image J plugin Frac Lac. All the data are expressed as the mean of 5 different experiments.

Discussion

It is commonly recognized that available laboratory data are not fully exploited. Even though the biologist's insight is always extremely useful in assessing the relevance of cell samples in different experimental conditions, nevertheless the need for more

objective and repeatable assessments is becoming urgent, given the possibility to automatize – at least partially – the analysis of experiments.

In light of these considerations, at our laboratory, the decision was taken to fit the lab with an informatics infrastructure for data exchange and processing. In doing so, it has been considered that the lab personnel generally had no particular skill or previous experience in information engineering, so one essential requirement for managing a quantitative, computer-based approach in image analysis was to make available easily usable software tools, preferably from open source developers.

The easy accessibility of data stemming from multiple sources is an important factor in driving characteristic parameters, to be fed into suitable algorithms.

Confocal microscopy images are tremendously useful for quantitative analysis of experiments and model construction. Visualization of the cytoskeleton structure is useful to understand cell motility, stiffness and more generally the cell phenotype. Cytoskeleton properties can be investigated, among other methods, by the calculation of the fractal dimension, starting from microscopy data. Such an analysis has been performed on confocal images, to observe the changes of cell properties due to the experimental protocol. Clear meaningful relationships are identified from the data experimented on different treatment conditions and are assessed graphically.

3.4 Validation of Image J software for the analysis of cytoskeleton architecture

The key challenge faced by scientists nowadays is using image J is to determine the morphological parameters for the cytoskeleton.

Therefore, we have verified the performance of Image J in 2D by using confocal images of MCF-7 cells. The image groups were classified as follows :

Data is divided into 3 parts [82]: 1. experimented on ground (with and without melatonin) 2. experimented in space (with and without melatonin) 3. experimented in space during docking (without melatonin). For each image, we have used our algorithm to generate the results defining the correlation between the cell shape parameters above and tubulin parameters as shown in figures above. We have also found that the fractal dimension is a good estimate for the cytoskeleton analysis.

In the existing literature, researchers have already used semiautomatic methods for such an analysis. In Masiello MG et al., the authors used Image J to analyze single cell of MDA-MB-231 cells to obtain single cell shape parameters [83].

Hence, the empirical results obtained using the automatic algorithms developed by us are used to describe the shape-morphological parameters of various nuclei and tubulin which are exposed to microgravity. The role of each parameter is easy to understand, along with their biological interpretation. In contrast to the existing methods, the results obtained using our algorithm are enough to detect the most important aspects that are related to the cytoskeleton architecture. The stepwise algorithm is shown in the next chapter.

To validate our algorithm we compare our results with other methods which are able to measure the image features quantitatively [84]. We find that our algorithm is able to identify not only the microtubule and cell shape parameters but it is also able to improve the image processing that helps in identifying the cytoskeleton morphology.

In conclusion, our algorithm can be quite useful to researchers who are working on cytoskeleton research.

3.5 Analysis for identifying cytospace data through Matlab programming

As we have already analyzed the cytospace data by image J in the previous section, hence in this section we will be more specific to

elaborate the morphological parameters of the cytospace data to explain the data information.

Here, we considered the cytoskeleton data (which includes nuclei and microtubules) experimented into 3 types of forms for this work [82] confocal imaging data experimented on ground (non melatonin treated and melatonin treated), confocal imaging data experimented on space (non melatonin treated and melatonin treated) and during docking in space, confocal imaging data in RPM and during docking in RPM [85]. Here, the samples which are used during docking whether in space or in RPM is untreated with melatonin [82].

Random positioning Machine: A random positioning machine (RPM) is a laboratory instrument that is widely used to provide the continuous random change in orientation relative to the gravity vector of an accommodated (biological) experiment. Moreover, the use of the RPM can generate the effects of the comparable to the effects of true microgravity when it changes in the direction are faster than the objects (cell and tubulin) response time to the gravity. However, in RPM microscopy most of the flight experiments the analysis of any microgravity effects is usually done after the fixation of the specimens. It would be very relevant to monitor the experiment during the actual microgravity [85]

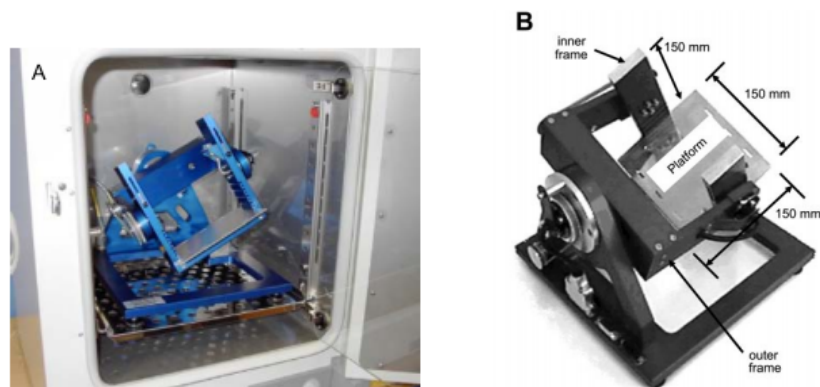


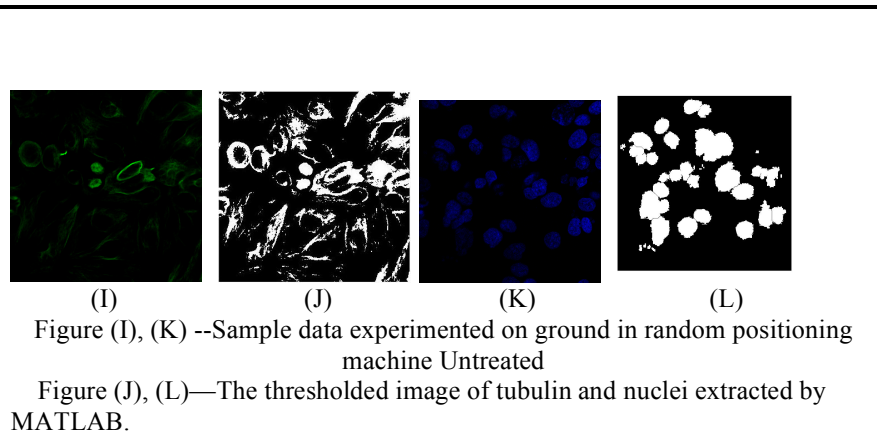
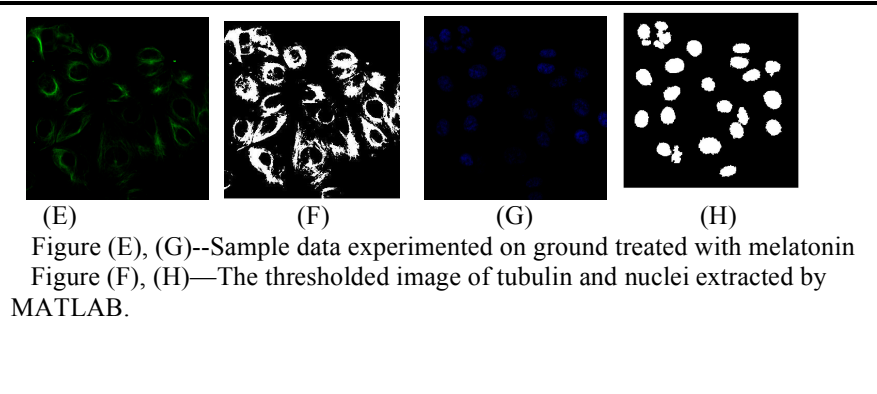
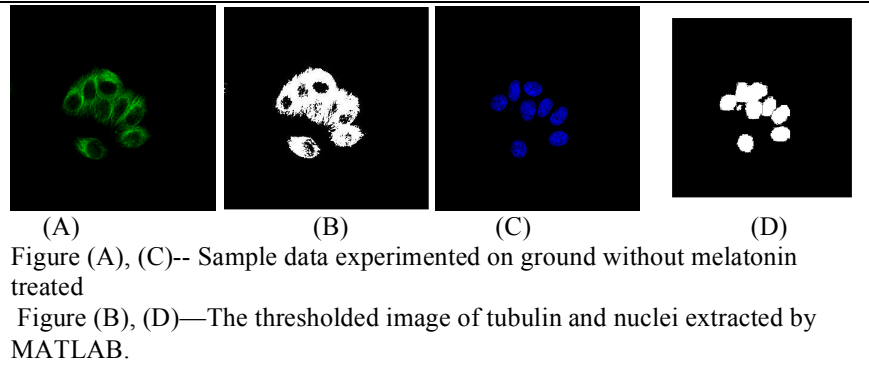
Figure 7: figure (A) representing the desktop RPM which is placed in a standard incubator [85] Figure (B) representing the RPM which is made up of two motors

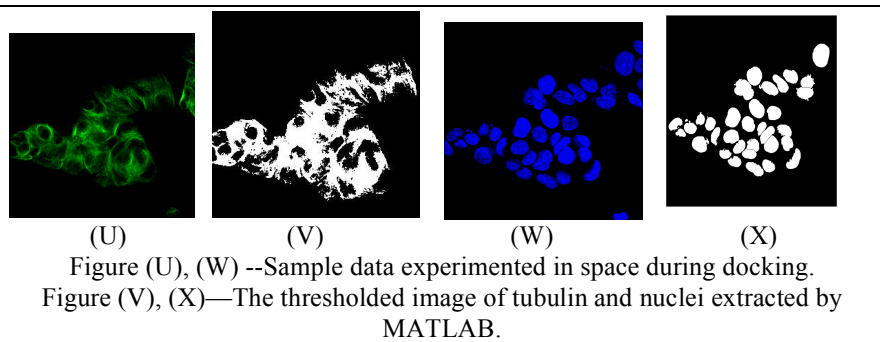
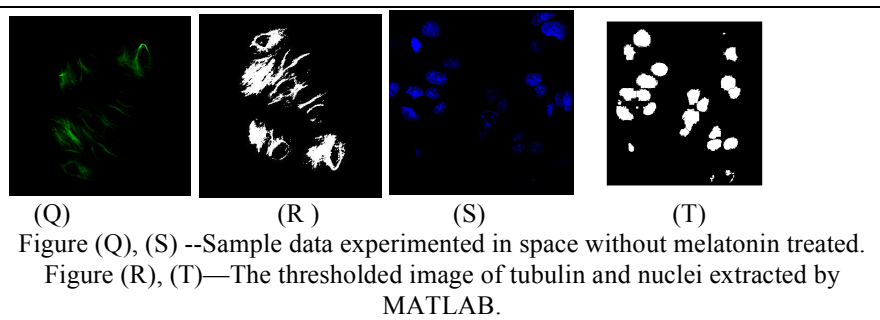
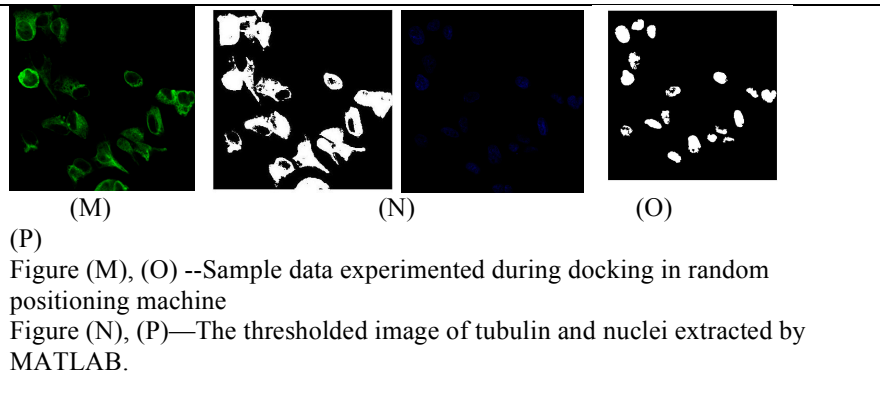
independently control the rotation of the outer and inner frames. The total size of the RPM is 30*30*30 cm [86]

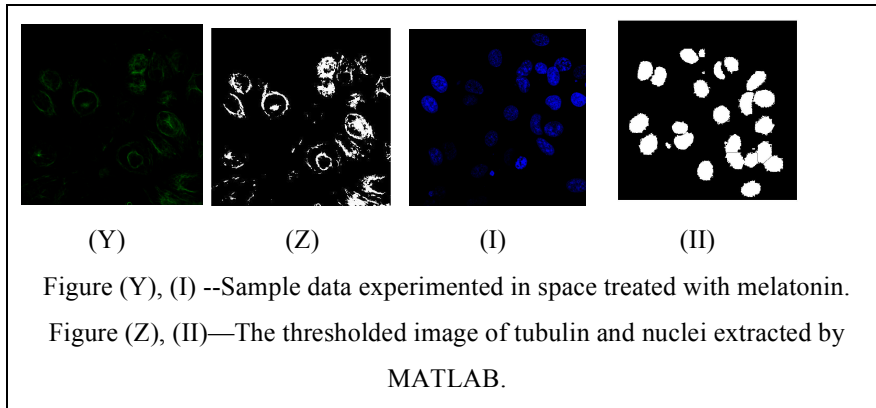
A desktop Random positioning machine is usually defined by Huijser and is manufactured by Fokker space which is used to stimulate microgravity. As shown in figure 7 above (B) the dimensions of the RPM are generally 30*30*30 cm with the inner and outer frames which are independently controlled by two different motors [86].

Here we have been used for our experiment the random positioning machine (RPM; desktop RPM, Dutch Space, Leiden, the Netherlands), It is a particular kind of 3D clinostat. It contains of two independently rotating frames. One frame which is positioned inside the other giving a very complex net change of orientation to a biological sample mounted in the middle. The degree of microgravity simulation basically depends on the angular speed and on the inclination of the disk. However, these tools do not actually remove the gravity, but it is a micro weight simulator based on the principle of “gravity-vector averaging”: it allows us to apply a 1 g stimulus omnidirectionally rather than unidirectionally and the sum of the gravitational force vectors tends to zero. The effects caused by the RPM are comparable to those of the real microgravity, provided that the direction changes are faster than the response time of the system to the gravity field. Further, the desktop RPM we used has been positioned within an incubator (in order to maintain the temperature, CO₂, and humidity levels) and is connected to the control console through standard electric cables [83,87].

Figures show the sample images of every experimental data to showcase the data.







3.5.1 Dataset of Cytospace for tubulin by Matlab analysis

In this section, I analyze the data set of the cytospace with the MATLAB algorithm. Measuring the tubulin from the image data set is very challenging for various reasons. The main issue is the adjustment of thresholding. But with the help of MATLAB, I have tried to solve the issue.

The sequence of confocal images which are processed is 48. All the images have been segmented by our algorithm and then compared with the morphological parameters by the algorithm described in the next chapter of this thesis - materials and methods along with the nuclei.

Table 3: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 3 sample data at the threshold value 5.

	Confocal images of the tubulin	No. of tubulin	Mean and standard deviation values of the parameters in the images at gray level 5					
			Area	Perimeter	Major axis	Minor Axis	Entropy of the image	Circularity
images experimented on ground without treated with melatonin	Image1	21	12956.57 ± 9436.43	523.97 ± 234.81	150.27 ± 58.84	104.30 ± 41.58	0.8260	2.58 ± 1.01
	Image2	39	9102.89 ± 9580.49	440.34 ± 302.36	117.81 ± 66.32	80.85 ± 49.43	0.9234	2.74 ± 0.97
	Image3	9	13530 ± 3020.21	534.48 ± 355.61	150.62 ± 95.44	98.42 ± 47.54	0.4587	2.76 ± 0.77

Table 4: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 4 sample data at the threshold value 40.

	Confocal images of the tubulin	No. of tubulin	Mean and standard deviation values of the parameters in the images at gray level 40					
			Area	Perimeter	Major axis	Minor Axis	Entropy of the image	Circularity
images experimented on ground treated with melatonin	Image1	34	1519.88± 1553.00	224.23± 156.78	72.40± 44.14	28.32± 16.40	0.4231	4.78±2.45
	Image2	25	2561.32± 2561.73	287.80± 208.66	75.11± 41.82	40.16± 26.21	0.2482	3.59±1.59
	Image3	39	1316.74± 1240.69	210.76± 113.04	66.05± 31.64	29.14± 15.83	0.2820	4.01±2.26
	Image4	44	2070.36± 1888.44	242.95± 145.33	64.93± 34.05	39.57± 20.09	0.5006	3.21±1.23

Table 5: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 3 sample data at the threshold value 10.

	Confocal images of the tubulin		Mean and standard deviation values of the parameters in the images at gray level 10					
			No. of tubulin	Area	Perimeter	Major axis	Minor Axis	Entropy of the image
images experimented on ground without the treatment of melatonin in RPM	Image1	31	659.90± 6333.61	177.23± 84.09	65.29± 48.71	36.74± 28.01	0.4738	4.90±2.09
	Image2	39	455.84± 568.1852	170.61± 106.54	50.49± 34.32	26.67± 15.39	0.3617	6.29±2.7
	Image3	15	346.933± 295.32	126.32± 52.98	66.65± 31.18	33.18± 24.10	0.264	4.371±1.97

Table 6: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 3 sample data at the different threshold value mentioned.

	Confocal images of the tubulin		Mean and standard deviation values of the parameters in the images						
			No. of tubulin	Area	Perimeter	Major axis	Minor Axis	Entropy of the image	Circularity
images experimented during docking in RPM	Image1	23	1687± 1376.67	215.66± 109.77	84.35± 57.60	55.24± 40.54	0.6127	2.70± 1.26	5
	Image2	13	11919.07± 9695.72	548.49± 353.63	138.30± 86.35	89.42± 49.47	0.6042	2.28± 0.98	5
	Image3	9	1861.88± 1417.82	243.25± 106.93	129.76± 123.94	81.97± 85.62	0.3545	2.94± 0.94	15

Table 7: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 3 sample data at the different threshold value mentioned.

	Confocal images of the tubulin		Mean and standard deviation values of the parameters in the images						
			No. of tubulin	Area	Perimeter	Major axis	Minor Axis	Entropy of the image	Circularity
images experimented in space without melatonin treated	Image1	20	660.5± 410.044	164.17± 79.26	62.91± 39.28	38.32± 22.89	0.2967	3.82± 2.47	10
	Image2	21	664.47± 669.21	177.31± 111.45	67.19± 36.53	31.56± 21.38	0.3108	4.53± 1.75	10
	Image3	21	598.95± 328.46	181.41± 78.69	68.18± 47.79	41.21± 26.69	0.4336	4.65± 2.02	13

Table 8: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 3 sample data at the different threshold value mentioned.

	Confocal images of the tubulin	No. of tubulin	Mean and standard deviation values of the parameters in the images						
			Area	Perimeter	Major axis	Minor Axis	Entropy of the image	Circularity	Threshold value
Images experimented in space during docking	Image1	14	2105.78± 3165.10	261.90± 227.00	108.14± 80.70	58.55± 44.45	0.5600	3.47±1.30	5
	Image2	22	1055.63± 1081.47	181.62± 119.55	81.24± 50.44	58.45± 37.28	0.4386	2.96±1.31	10
	Image3	23	1342.65± 1364.43	230.39± 142.26	78.50± 52.68	43.52± 30.57	0.5408	3.66±1.71	10

Table 9: This table summarizing the type of the cytospace tubulin data (mean and standard deviation) extract from the MATLAB algorithm. Table summarizing the values of morphological parameters of the tubulin image on 3 sample data at the different threshold value mentioned.

	Confocal images of the tubulin		Mean and standard deviation values of the parameters in the images						
			No. of tubulin	Area	Perimeter	Major axis	Minor Axis	Entropy of the image	Circularity
Images experimented in space treated with melatonin	Image1	27	611.40± 459.54	177.63± 82.00	76.54± 54.63	40.93± 29.34	0.3485	4.67± 2.25	10
	Image2	24	390.66± 353.51	140.65± 81.88	49.80± 23.67	25.92± 12.61	0.2273	4.50± 1.95	30

3.6 Validation of MATLAB algorithm for the analysis of cytoskeleton architecture

We have performed the validation studies for the identification of cytoskeleton architecture using MATLAB. We have investigated and developed the algorithms that are needed to measure the cytoskeleton morphology.

Literature survey was done in this regard to see if MATLAB based tools have been used previously to identify and investigate the architecture of cytoskeleton. We found there are some matlab based

softwares like : Code J mapper , Fiberapp, and Cytospectre which are able to track and analyse the cellular information.

In Code J mapper MATLAB is used for extracting the centerline location of the cytoskeleton filaments from the confocal microscopy image analysis. It is able to compute local connectivity maps, as well as several important filament network-related quantities such as: curvature, orientation, as well as local filament topology [88].

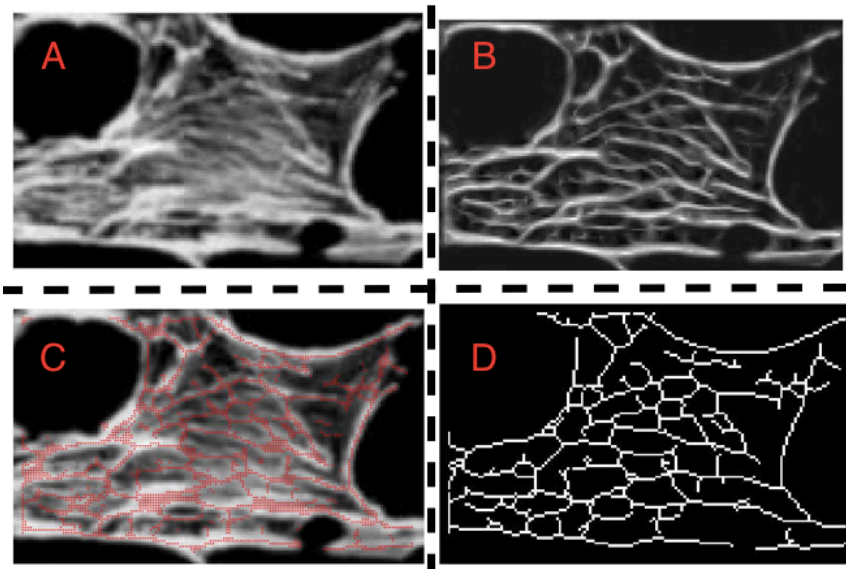


Figure 8: (A) Original image (B) Representing the enhanced image after matched filtering with an artificial array of filament (C) The image representing the localized filament centerline points. (D) The image representing the minimum spanning tree in the connectivity matrix ST1.

Another Matlab based program Fiber app is able track the linear and circular fiber-like objects. It also has the capability to define the segments of heterogeneous stiffness along with their contours. In addition to these, Fiber app is also able to simulate the “insilico” statistical data as well as the artificial images of the fiber-like objects which serves as a powerful tool for the prediction and rationalization of the experimental results .

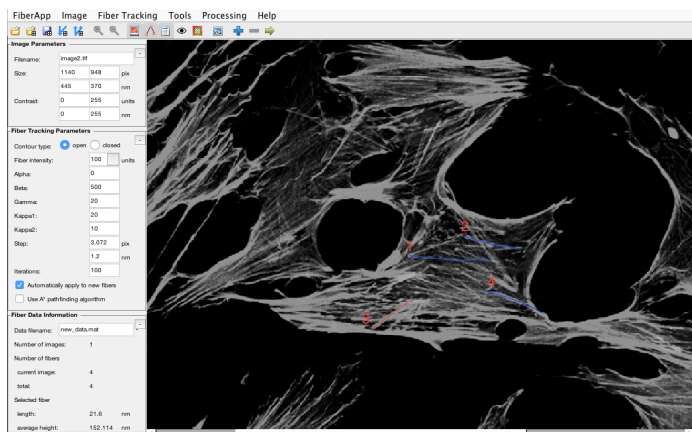


Figure 9: Fiberapp software screenshot of the analysis and measurement of the filaments. The Confocal microscopy image was selected for tracking the contours of fibers.

Cytospectre is another MATLAB based application that can be used for the spectral analysis of microscopy images. Spectral analysis allows the users to obtain information regarding the orientation and size distribution of the targets within the images [89].

By the analysis of these MATLAB based softwares we found that they are limited capabilities to perform the morphological analysis of nuclei and tubulin. Hence, we wrote our own MATLAB scripts for tracking the nuclei and tubulin in confocal images. This work proved useful for tracking the morphological features like – area, circularity, major and minor axis etc. The details of the MATLAB based algorithms are provided in the next chapter and results obtained as discussed above.

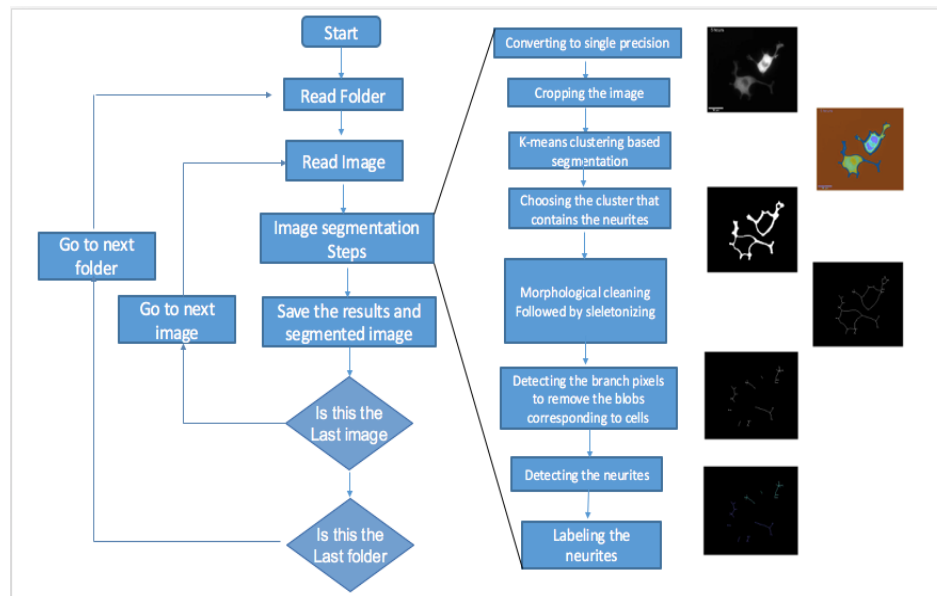
3.7 Identification and analysis of neurites

3.7.1 Collection of the confocal 2D images for the Image Processing

Images were processed on a Mac workstation with a 1.6 GHz Intel Core i5 along with the memory of 8 Gigabyte 1600 MHz DDR3. A custom-made MATLAB program (MATLAB R2017, MathWorks) was used. The program has been created in such a way that it can read, process and analyze different bio formats i.e., .tiff, .jpg, and other MATLAB readable file formats for the access. MATLAB is an essential tool used for the image processing and analysis of the neurites. In this section, we are describing the MATLAB outcomes and results to measure neurites parameters.

The counting of the neurites is the major task to do. They play a very important role in the processing of the neurons. Neurites can be identified by the eye using the fluorescence data and can be manually annotated in the fluorescence data. Neurites which include axons and dendrites are distinctive features of the neurons, which allows the maintenance and formulation of the nervous system. Axons and dendrites are collectively called neurites which are the neuronal projections that are used to create the functional connections (synapses) between the neurons. Consequently, the development of these projections is necessary for the creation of the functional nervous system.

3.7.2 Flowchart of the neurite detection in Pheochromocytoma cells 12 by MATLAB



3.7.3 MATLAB program for the measurement of the neurites in the time-series data

The sample images are shown above in the figures. A couple of 8-bit images (480*380 pixels), 8-bit with 167 K, in tiff format, is available for each hour of the experiment. These images are the input for the MATLAB code.

The program basically works in three main phases;

- reading the image folder
- processing the raw image by inbuilt functions in MATLAB
- saving the extracted neurites and the measurements

3.7.3.1 Reading the image folder

MATLAB treats the images as 2D matrices whose elements are the values of the image pixels. Our algorithm for detecting the neurites works only for grayscale images, the input is a true-color image or

any types rather than the grayscale, it has to be converted to grayscale before running the code.

The code starts by reading the folder containing our dataset. The command `data_folder='./datasets_of_ebi_protein'` sets the name of the data set folder, which in our case is named 'datasets_of_ebi_protein'. Then `filepattern = fullfile(data_folder, '*dataset')` command is used to set the path to the data folder. Then we use `dd=dir(filepattern)` to read the contents of the folder specified by the path already set. This is followed by creating a directory called 'results' by the command `mkdir([data_folder filesep 'results'])`.

3.7.3.2.Processing the input images by starting the loop

We employ a for-loop for reading the folders from 3hrs to 11hrs. The code reads the folder names using the command `folder_name = dd(i).name`. To store the results, a directory having the same name as the current folder is created inside the result folder command `mkdir([data_folder filesep 'results' filesep folder_name])`. We are using `current_folder = [data_folder filesep folder_name]` for naming the current folder according to the loop (value of the i). Then the code reads the images inside the above folder by the command `images=dir([current_folder filesep '*.tif'])`.

Then a second loop is run for reading the images inside the current folder one by one. The image names are retrieved using the command `current_image=images(j).name`. `imname = current_image(1:end-4)` is the command use to removing the last four characters from the name i.e. '.tif' is removed from the 'im122.tif'. For making the folder for the current image inside the directory, we use `mkdir([data_folder filesep 'results' filesep folder_name filesep imname])`. The next step of the algorithm is to read the individual files and convert them to single precision for further processing. The command used for this is `BW=im2single(imread([current_folder filesep current_image]))`. We obtain the size of the image using `[nrow,ncol]=size(BW)`, and crop

the image slightly using the command as $BW = imcrop(BW, [5\ 5\ ncol-10\ nrow-10])$. Here, we crop the image slightly because there are some stray pixels on the border of each image so to remove them, we use the command `crop`.

3.7.3.3 Selecting the cluster containing the neurites

For segmenting the neurites, we use k-means algorithm to cluster the image intensity values in to six clusters ($k=6$) using the command $[L, Centers] = imsegkmeans(BW, numclust)$. Then, the cluster containing the neurites is selected. We have chosen `numclust = 6` because only by this value the neurites are selected in the image. If we select 3 to 5 values the neurites are not selected by lower than 6 values. The value of `numclust` is determined experimentally. Typically, the number of clusters should be low, as increasing the cluster size significantly increases the computational requirements. For the kind of images that we have in the dataset, `numclust=6` gives good enough segmentation for the algorithm to work. A value less than this leads to overlapping segments in the cell.

Detecting the cluster containing neurites is done by choosing the cluster that encircles the second-highest area (the cluster corresponding to the highest area represents the background).

We use `bwareaopen` to remove the small objects from the image and here we choose the value 100 because it works fine. We have experimented with different values and chosen the best performing one. We use the command `nnz` for converting the image in which the small elements are removed to the number of nonzero matrix elements.

The following code snippet does this:

```
numclust=6;
[L,Centers] = imsegkmeans(BW,numclust);
B1 = labeloverlay(BW,L);
th=[];
```

```
for k=1:numclust % Choosing the cluster that
contains the neurites
    temp_mask = L==k;
    BWn = imfill(temp_mask,'holes');
    BWn=bwareaopen(BWn,100);
    an = nnz(BWn);
    gg=[an k];
    th=[th;gg];
end
[mm,ii]=max(th(:,1));
th(ii,1)=0;
[mm,ii]=max(th(:,1)); %Choosing the second
highest
kf=th(ii,2);
mask = L==kf; %This contains the neurites
```

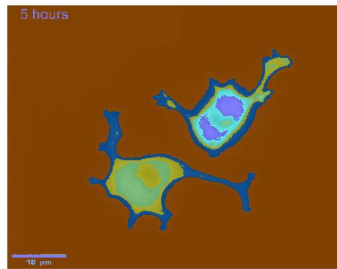


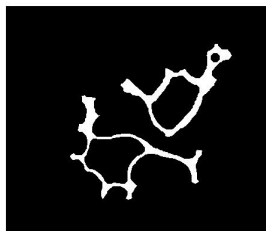
Figure 10: The figure is presenting the two cells at 5 hrs time period on which k means clustering has been applied.

3.7.3.4 Skeletonized the segmented neurites

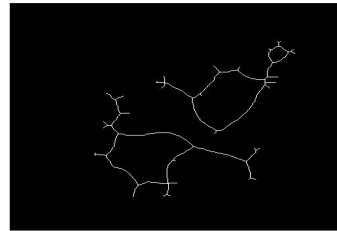
In this section, we describe the basic morphological operations performed on the logical/BW image obtained after extracting the cluster corresponding to the neurites. We close the holes in the image, remove the unconnected, smaller components and smooth the image. Erode the image to get the skeletonized version of the image using the command $BW3 = bwmorph(mask, 'skel', Inf)$. These operations are performed in the following part of the code:

```
se = strel('disk',8);
```

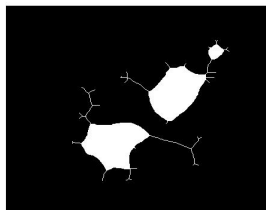
```
mask=imclose(mask,se);    %Cleaning the
image..fill the gaps
mask=bwareaopen(mask,200);
[L11,n1] = bwlabel(mask,8);
if n1==1
    break;    % Aborting the processing if the
images are connected
end
BW3 = bwmorph(mask,'skel',Inf); % Skeletonize
the image
BW4 = imfill(BW3,'holes');
```



(a)



(b)



(c)

Figure 11: (a) Image consisting of clear borders and defining the small structuring elements as neurites (b) Skeletonized image (c) Image containing the 2 cells in which holes are filled.

3.7.3.5 Detecting and isolating the neurites

For detecting the branches corresponding to the neurites, we detect the branch points. These are the points from where the neurites branch-out from the cell body. These are the points having three

connected pixels in the skeletonized image. Then, these points are removed. This results in a binary image where the neurites are no longer connected to the blob corresponding to the cell body. The blobs in the image are removed in this way, leaving behind the segmented neurites.

Here, `bwlookup` runs on the matrix and stops where the condition satisfies and `makelut` is used to create the lookup table for the use with `bwlookup`. It is a utility function to help us to construct the lookup table. Using lookup table with `bwlookup` to perform the filtering.

Below is the corresponding code snippet:

```
                                %Detecting Neurites
                                %Detecting start of branches
                                bp=bwlookup(BW3, makelut(@(x) sum(x(:))>=4 &
x(5)==1,3)); %branch      points...
                                bpImage = imdilate(BW3, true(3));
                                BW5=BW4&~bp;
                                % Removing the blob...
                                BW6=bwareaopen(BW5,1500);
                                BW7=imsubtract(BW4,BW6);
                                BW7=bwareaopen(BW7,10);
                                BW7 = bwmorph(BW7, 'skel', Inf);
                                %%
                                L2=immultiply(L1,BW7);
                                %
                                [yy1 xx1] = find(L2==1);
                                [yy2 xx2] = find(L2==2);
                                figure, imshow(1.15-BW)
                                hold on
                                plot(xx1,yy1, 'r.', 'MarkerSize', 10)
                                plot(xx2,yy2, 'b.', 'MarkerSize', 10)
                                hold off
                                saveas(gcf,[data_folder filesep 'results'
filesep folder_name...          filesep imname filesep
current_image]);
```

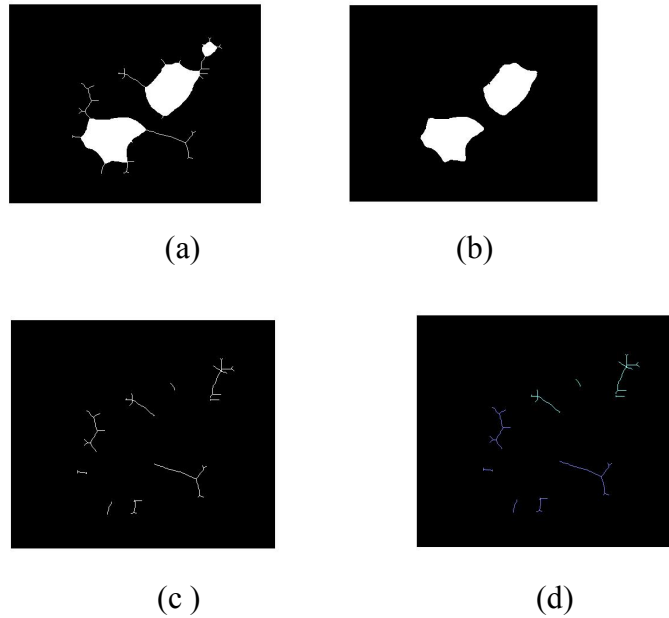


Figure 12 : (a) Detecting the branch points in the image (b) Cells in which removing the smaller objects fewer than 1500 (c) Skeletonized the neurites in the image (d) labeled image consisting the different color of neurites for cell 1 and cell 2.

3.7.3.6 Saving the cell data in the text file and saving the .mat file

Once the neurites are segmented, they are overlaid with two different colors, depending on which of the two cells they belong to. Then measurements are made to get the lengths of the neurites, and the overlaid image and the measurements are saved in the results directory.

The corresponding code snippet is:

```
header11 = 'Total Neurites detected= ';  
header12 = num2str(length(cell_1_data));
```

```
        fid=fopen([data_folder filesep 'results'
filesep folder_name filesep imname filesep
'Cell_1_data.txt'], 'w');
        fprintf(fid, [ header11 ' ' header12 '\r\n']);
        fprintf(fid, [ 'Lengths : ' '\r\n']);
        fprintf(fid, '%f \r\n', A );
        fclose(fid);
        %
        tempmask2=L2==2;
        cell_2_data=regionprops(tempmask2, 'area');
        B=zeros(1,length(cell_2_data));
        for kk2=1:length(cell_2_data)
            B(kk2)=cell_2_data(kk2).Area;
        end

        header21 = 'Total Neurites detected = ';
        header22 = num2str(length(cell_2_data));

        fid=fopen([data_folder filesep 'results'
filesep folder_name filesep imname filesep
'Cell_2_data.txt'], 'w');
        fprintf(fid, [ header21 ' ' header22 '\r\n']);
        fprintf(fid, [ 'Lengths : ' '\r\n']);
        fprintf(fid, '%f \r\n', B );
        fclose(fid);
        % Saving as .mat file
        save([data_folder filesep 'results' filesep
folder_name filesep imname filesep imname
'.mat'], 'cell_1_data', 'cell_2_data');
```

3.8 Neurite Data Analysis

3.8.1 Logistic curve analysis

The growth curves are the tool used for the study to describe how the variable increases over a particular time interval, until it approaches its saturation value. Logistic growth occurs when the neurite growth rapidly increases in size until it reaches a certain point called carrying capacity. The carrying capacity act as a moderating force in the growth rate by slowing it when the resource

become limited and stopping the growth once it has been reached [90].

The logistic growth curve is an S-shaped curve that can be used to model the function that increases gradually at first, more rapidly in the middle of the growth period and slowly at the end, leveling off at a maximum value after some period of time.

It is one of the simplest S-shaped growth curves. It is based upon the hypothesis that when these varying influences do not produce extraordinary changes, the population would probably follow the growth curve characteristics of living things within limited space and with limited economy opportunity.

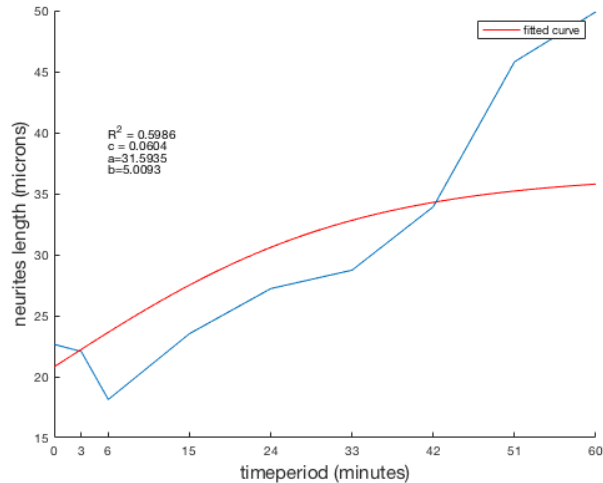
The biggest difference is that the line in the logistic growth graph changes direction and begins to level off as it near the carrying capacity. That means the main difference between exponential and logistic growth curve is that logistic growth takes into account carrying capacity. GP, IP, SP are the points on curve growth point, inflection point, and saturation point.

All data analysis and plotting were performed using MATLAB 2019b (MathWorks Inc., USA). Fitting of custom functions was performed using a non-linear equation or the Trust-Region method, implemented in the Curve Fitting Toolbox (ver. 3.5) of MATLAB [91].

The equation is used for the logistic growth model as follows:

$$y = b + \frac{a}{1 + e^{-cx}}$$

In this equation, a,b,c are the fit parameters. Here, a, b, c are the coefficients. And the values of a, b, c are 31.5935, 5.0093, 0.0604. One of the figure shown below for the fitting curve as:



In this figure, the Y-axis is shown as the neurite length in microns and X-axis is shown as time period in minutes along with the fitted curve in which R^2 value is 0.5986, a value is 0.0604, b value is 5.0093 and c value is 0.0604. To obtain the fitting curve we have done with the MATLAB program.

The graph above is showing the growth rate changes over time. The graph increases from left to right, but the growth rate only increases until it reaches its point of maximum growth rate, at which the rate of increase decreases.

In this work, the logistic model have been applied to describe the growth of neurites in PC 12 cells. The study results have shown that the logistic model is better in predicting the growth of neurites. Therefore, here my goal is to analyse the general model, by using the the MATLAB program. In conclusion, the analysis of logistic growth model provide the mathematical framework for better understanding of neurite growth and the neurite growth which is affected by different labelled in the culture medium. It is concluded that the logistic function model is an appropriate description of the experimental data.

Below is the corresponding code snippet:

```
dataset = xlsread('Q9HOQ7_second_cell.xlsx');
x=dataset(:,1);
y=dataset(:,2);
figure (1),.....
    hold on
plot(x,y)

xticks([0 3 6 15 24 33 42 51 60])

a1 = 1;
c1 = 0;
b1 = min(y);

s =
fitoptions('Method','NonlinearLeastSquares',...
    'Startpoint', [ b1 a1 , c1] , 'Lower', [ 0 1
0 ] , 'Upper', [ Inf ] );

f1 = fittype( ' b + (a./(1 + exp(-c.*x) ))' ,
'coefficients', { 'b' , 'a' , 'c'} , 'options',
s);

[cf,s1,t] = fit( x , y , f1 )
    figure(1),...
        plot(cf )
        xlabel('timeperiod (minutes)', 'FontSize',15)
ylabel ('neurites length
(microns)', 'FontSize',15)
    [ cf.b, cf.a , cf.c , s1.rsquare ]
    text(6,40, 'R^2 = 0.5986')
    text( 6, 39 , 'c = 0.0604')
    text(6,38, 'a=31.5935')
    text(6,37, 'b=5.0093')
```

3.8.2 Significance of the Logistic Curve

- The correlation between the physical parameters was obtained by using the statistical data analysis tools.
- By implementing Non linear least squares to the data we obtain the results which were compared to the actual values available for validation.
- Logistic models are very useful in studies of medical sciences and biological sciences.
- The logistic growth curves (S-shaped) is a popular model for studying and forecasting future changes.

3.9 Neurites Results produced from cell profiler (Stepwise)

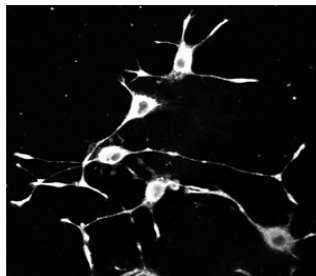
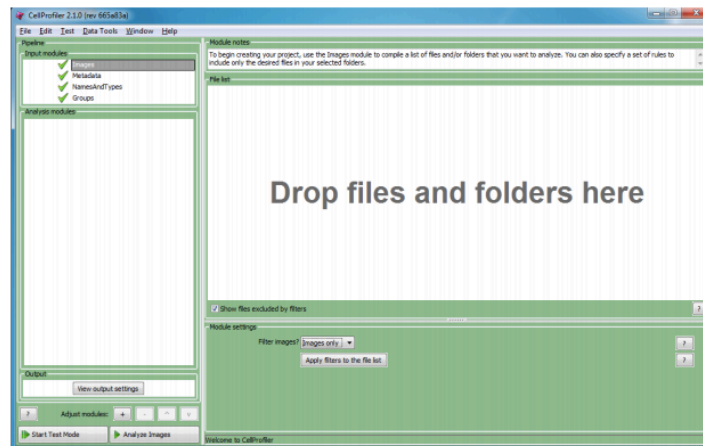


Figure 13: coiled-coil domain containing 8, CCDC8 Image used as an input in the cell profiler to produce the results.

Step 1:

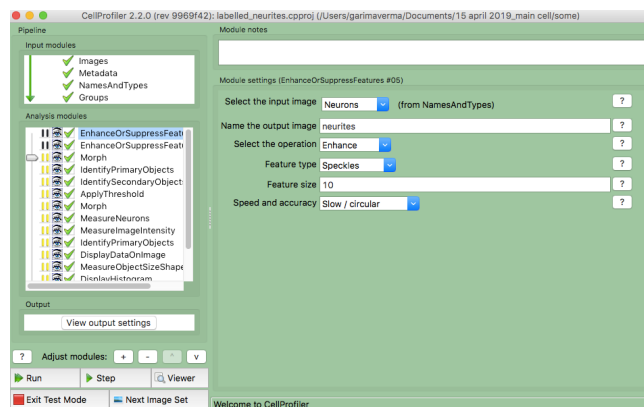


Input the CCDC8 image by image input module in cell profiler, we can drag and drop individual neurite file into this panel.

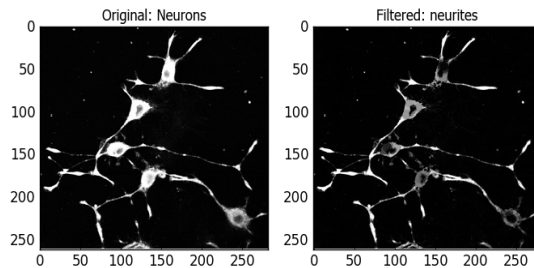
Step 2:

Enhance or Suppress features

It is used to increase or enhance the contrast of the neurites. By setting the module as shown in (a) we can get the result output as shown in (b) below:



(a)



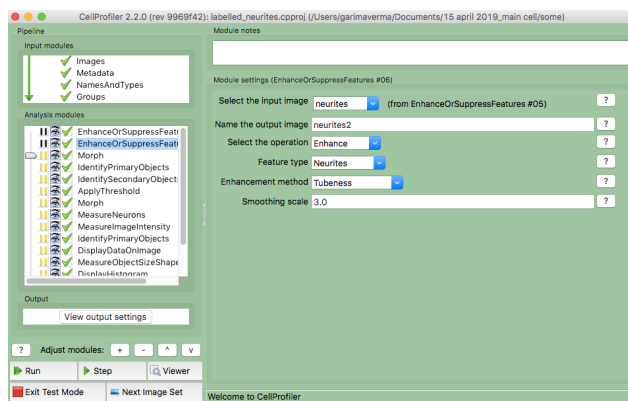
(b)

Step 3

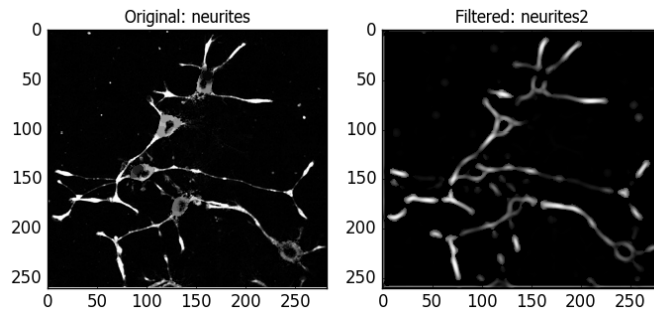
Enhance or Suppress features:

Using the previous contrasting image labeled as “neurites”, in this step again Enhance or suppress features module is used to suppress the intensity of certain pixels relative to the rest of the image by changing the module setting.

By setting the module as shown in (a) we can get the result output as shown in (b) below



(a)



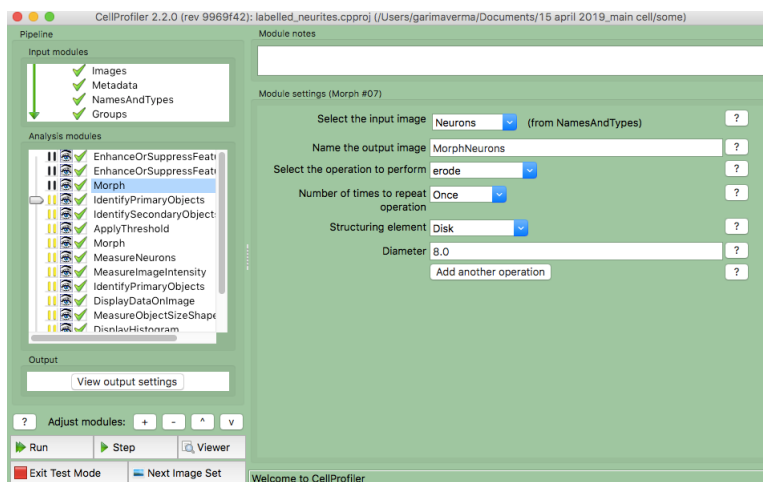
(b)

Step 4

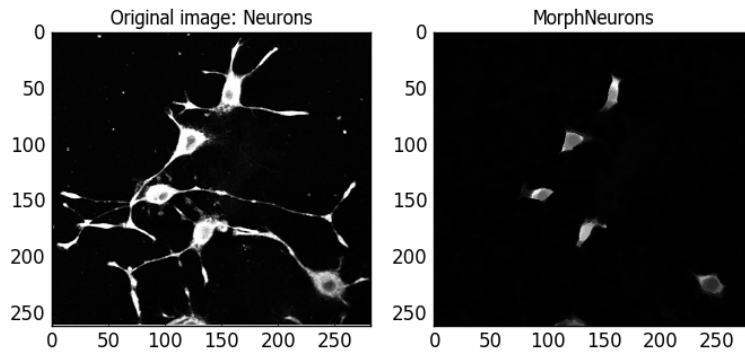
Morph:

Morph used the skeleton module to generate a skeletonized image. It usually performs the low-level morphological operations on binary or grayscale images.

By setting the module for morph as shown in (a) we get the result output as shown in (b) below



(a)

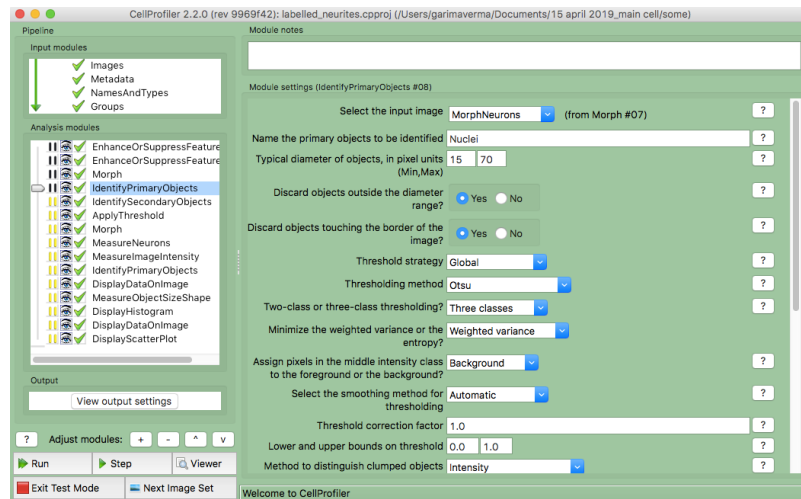


(b)

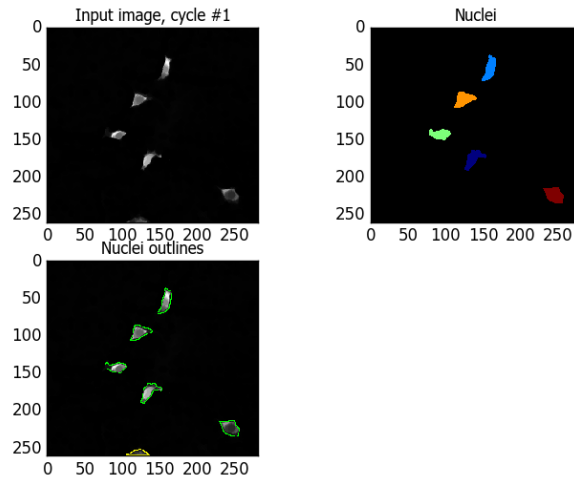
Step 5 Identify the primary objects

Identifies the nucleus.

By setting the module for identification of primary objects as shown in (a) we get the result output as shown in (b) below



(a)



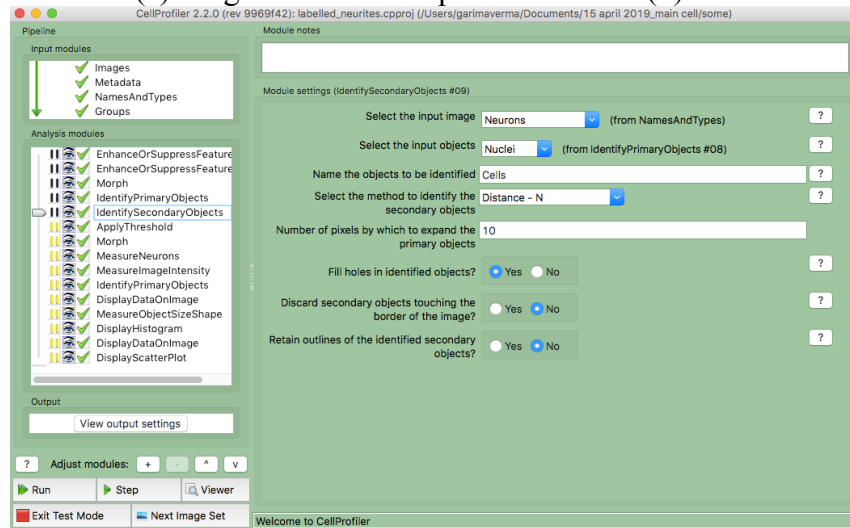
(b)

Step 6

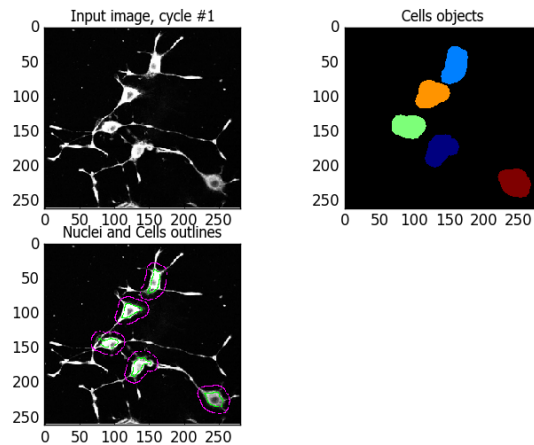
Identify secondary objects

Identifies the cell body.

By setting the module for identification of secondary objects as shown in (a) we get the result output as shown in (b)



(a)



(b)

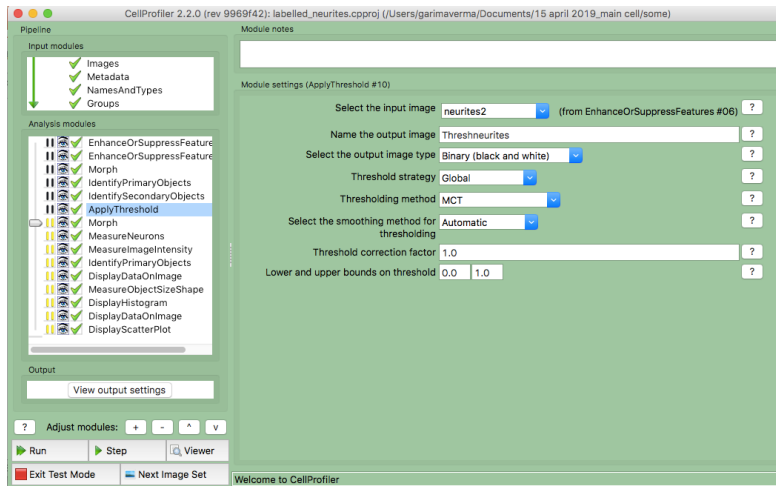
Step 7

Apply threshold

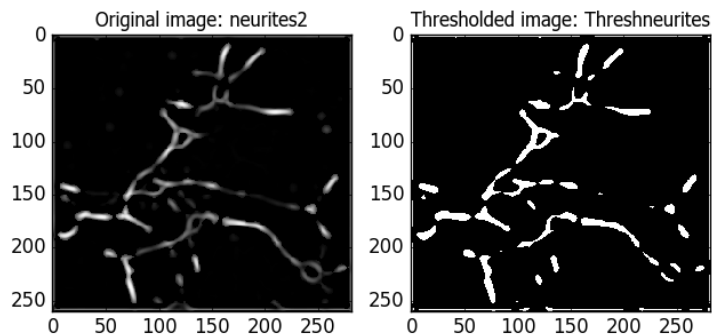
It is used to create a binary image.

Here, we have used the second “neurites2” enhance and suppress image as shown in step 3 to produce the thresholding result

By setting the module for identification threshold as shown in (a) we get the result output as shown in (b)



(a)

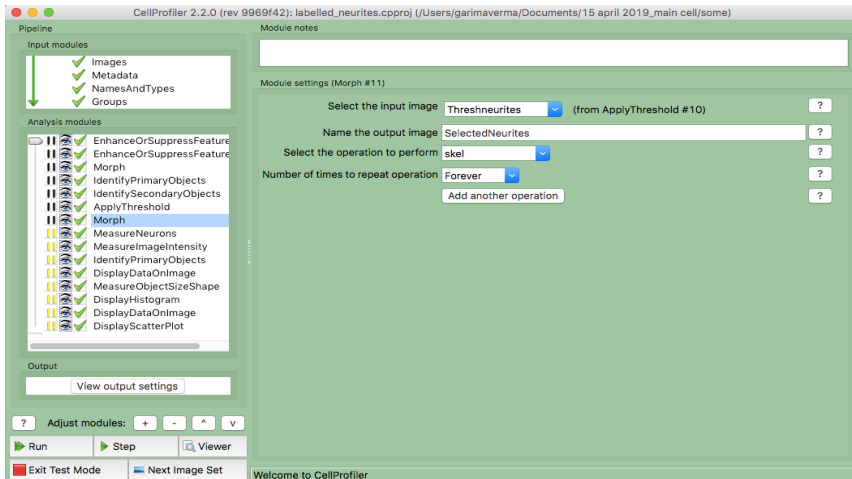


(b)

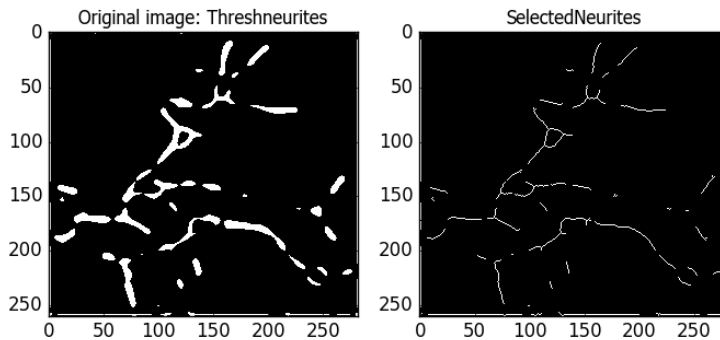
Step 8 Morph

Morph used the skeleton module to generate a skeletonized image. It usually performs the low-level morphological operations on binary or grayscale images. Here, we are using the threshold image to produce the output shown in figures.

By setting the module for morph as shown in (a) we get the result output as shown in (b)



(a)



(b)

3.10 Neurites changing in different image at different time interval:

From the figure below we can see that the neurites are changing at different time intervals in both the cells, i.e first and second cells. The images were taken from the database of neurites as <http://neurite.embl.de/> The length of neurites are having different length at each time period taken from the database.

I have studied this neurite analysis by different tools and techniques. Focusing on the detection as well as measurement. I have used different tools like MATLAB, Filopodyan, cell profiler, neuron J. And then I studied different techniques to detect these neurites by the tools discussed previously. Once I detect the neurites from the tools like MATLAB, Filopodyan, Cell profiler, Neuron J and then I compared them, and I got the results of different neurite length and count.

Due to the great number of images obtained from the neurite database, and performing their analysis was a very challenging task. The central objective of this work was done both by manual as well as automatic analysis. Manual work is done by the tools – Cell Profiler, Neuron J and other image J plugins. Automated work is done by MATLAB and *Filopodyan*.

The comparison of the results obtained with Filopodyan and Neuron J shows that individual length measurement is highly fluctuating. Hence, I have tried to show some differences in the time series data by proposing the analysis from different tools and techniques.

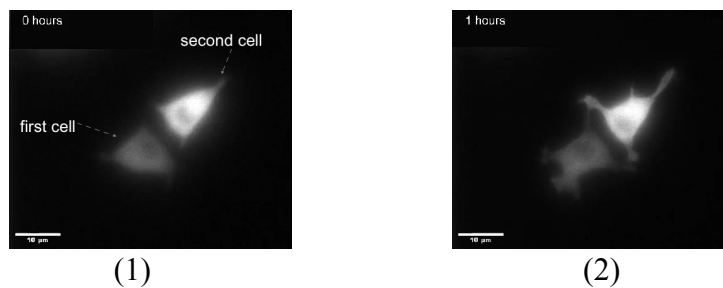
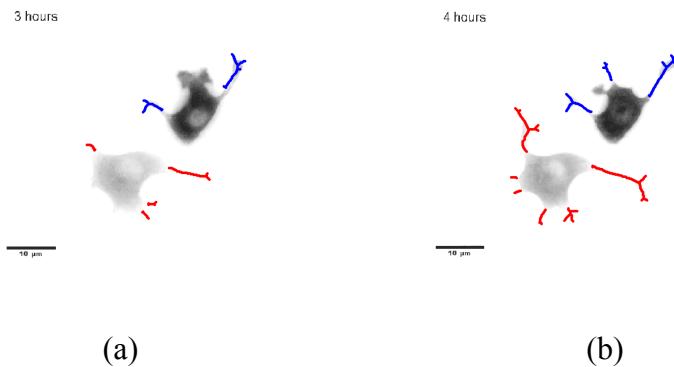
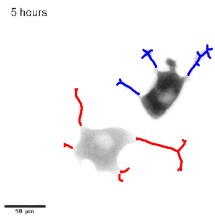


Figure 14: Sample images of the EB 1 protein (GFP tagged) labeled PC12 cells from time period 0 to 11 hours. Scale bars are 10µm. Couple of images from the database are used for the image analysis by different software.

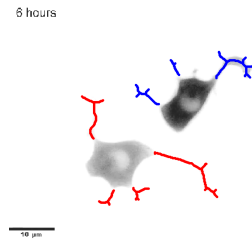
3.11 Programming for detecting the neurites

For the detection and counting of the neurites, I am using the MATLAB inbuilt functions. I am using the segmentation process for the detection of neurites as the MATLAB function `imsegkmeans` clustering. Below are shown the results about the detection and number of neurites detected from the raw image. The process of detecting is done by the morphological functions inbuilt in MATLAB. As for the study of these neurites in the raw images, we are facing some problems in the dataset of 0,1,2 hrs time series data. In the dataset of 0,1,2 the two cells are overlapping with each other. Hence the results are not satisfactory for the dataset 0-2 hrs. So, for the analysis of 0,1,2 hrs time series dataset I have used Neuron J plugin in Image J. As for the dataset from 3 to 11 hours we have used MATLAB. MATLAB helps us to produce the results from 3-11 hrs time series dataset.

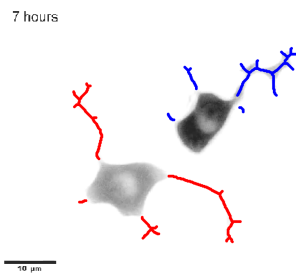




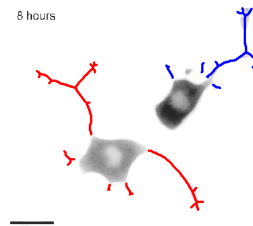
(c)



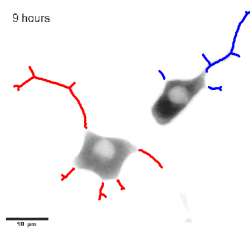
(d)



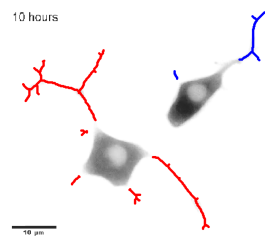
(e)



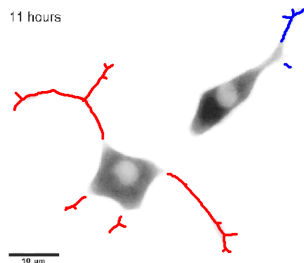
(f)



(g)



(h)



(i)

Figure 15: Morphology of the cells: Analysis of the neurites counting and length measurements in two separate data set, i.e first and second cell: detection of length and counts of neurites at different time periods given below. The first cell represented in red color neurite and second cell represented in blue color neurites in each image, a) 3 hrs (b) 4 hrs (c) 5hrs (d) 6hrs (e) 7hrs (f) 8hrs (g) 9hrs (h) 10hrs (i) 11hrs. All scale bars are at 10 μm .

With the help of a script written in MATLAB, neurite detection from the time period 3-11hrs was successful. It helps us to know the measurement of the neurites at different time. The plot is shown below in section 3.13. In this plot, the average length of the first cell neurites is 45.32 microns and the average length of the second cell is 24.07 microns. Here in the result above MATLAB helps to segment the neurites from the cell body. We need to measure only the neurites which is attached with the cell body so that we know the count and measurement of the neurites.

3.12 Comparison of the total neurites' length under various time-series data

The length of neurites changes found from the tables were plotted in a graph to find out the distinctive pattern of changes in the neurite time series data. The MATLAB program is able to detect the neurites and count the number of neurites along with their length

measurements of the time series data from 3 to 11 hrs. In figure 14 we have compared the two sets of cells i.e first and second cells. Furthermore, we have plotted the sum of the lengths of neurites at each particular time period for the first and second cells.

The results are different in the first and second cells because the number of neurites is different at different time periods. We have measured each and every neurite at different time intervals. We have represented the first and second cell in figure 14 in the first image. The first and second cell are not equally treated. I have chosen first cell and second cell according to the measurement. The first cell I have chosen because I have measured the neurites of first cell firstly. And the latter second cell, I have measured the neurites after the first cell. I have named first and second cell randomly. Neurite measurement of the first cell is done firstly, and the neurite measurement of the second cell is done afterward.

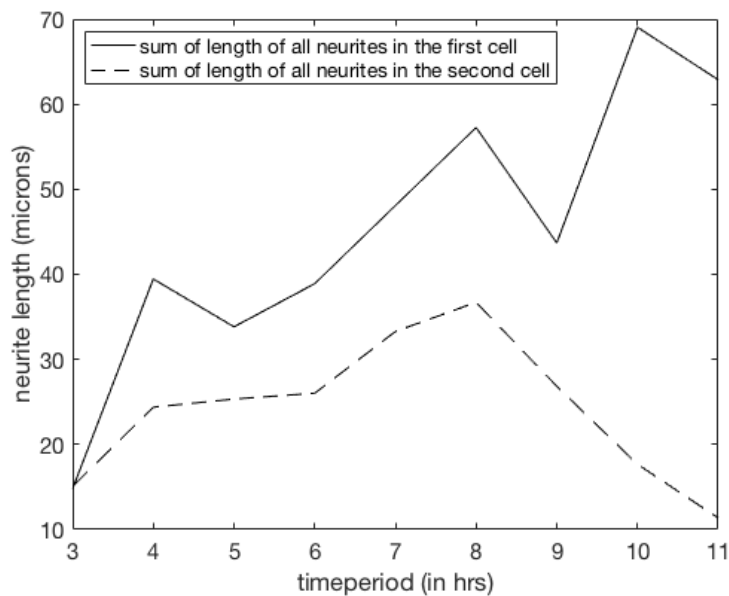


Figure 16 : Total length of neurites at each time period from 3-11 hours is presenting in two different sets of cells. Sum of total neurites length at each time period.

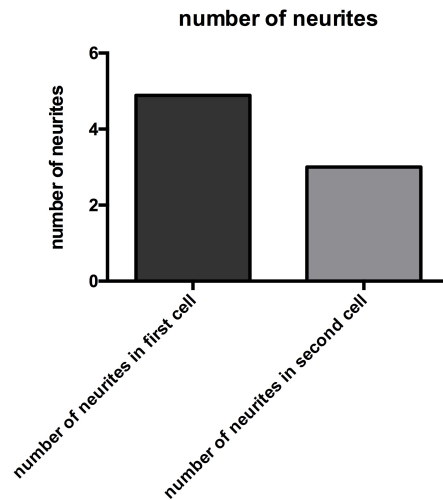


Figure 17: The histogram for counting total number of neurites in two sets of cells the time series data ranges from 3 to 11 hrs. The dataset presenting the cumulative sum of the neurites from 3 to 11 hrs.

3.13 Detection of neurites from time period 0 to 2 hrs by Neuron J

Neuron J is a plugin of Image j to enable the analysis and tracing of the image structures which are elongated, like neuronal processes. The Image J plugin neuron J can able to handle only 2D images of 8bit grayscale / indexed color. Here we are presenting the comparison of neurite dataset from 0 to 2 hrs in the first cell and second cell.

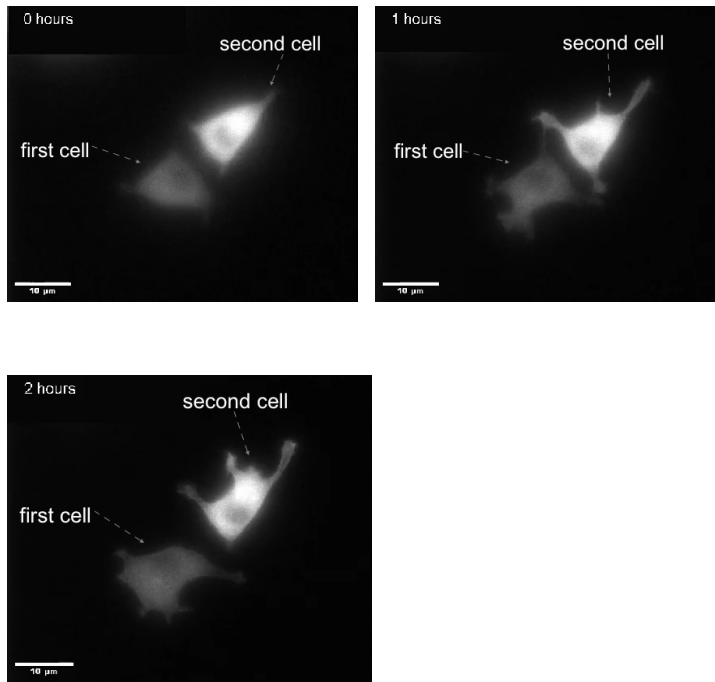
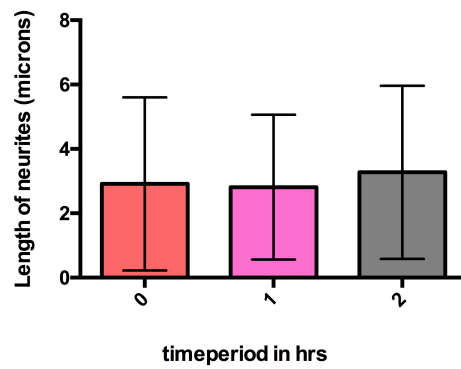


Figure 18 (a) Representative images of the first cell and second cell dataset from 0-2hrs time period. Scale bar is 10μm



(a)

Figure 19: The histogram plot of total mean length of neurites in first cell data series at different time intervals from 0 to 2 hrs. Mean \pm SD

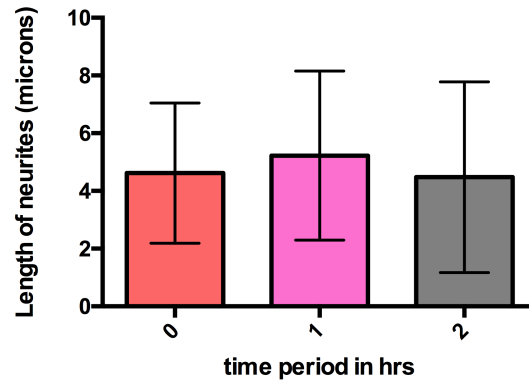


Figure 20: The histogram plot of total mean length of neurites in second cell data series at different time intervals from 0 to 2 hrs. Mean \pm SD

The significant differences of the dataset between 0-2 hrs in the first and second cell is non-significant done by one-way ANOVA. There is no significant difference in between the first and second cell at different time intervals.

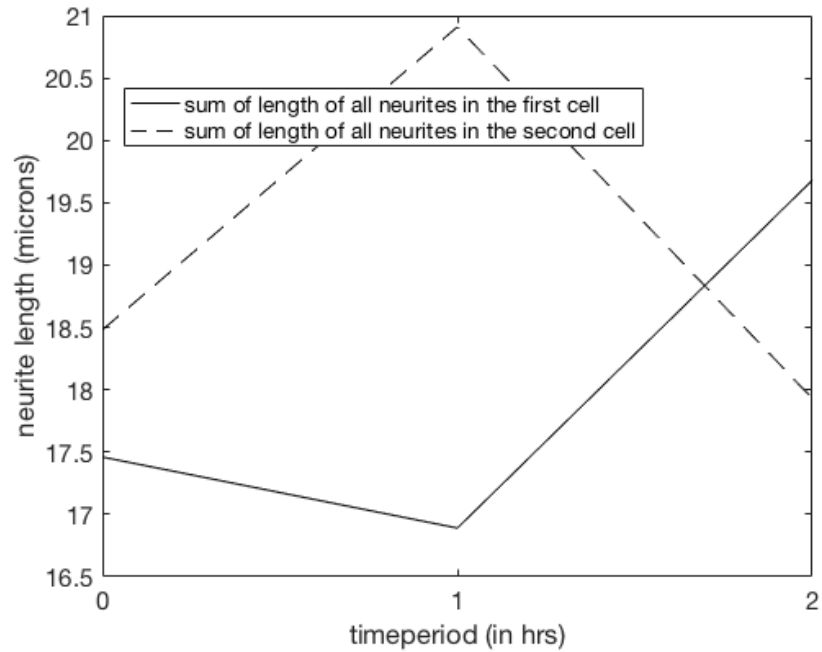


Figure 21: The comparison of the sum of length of neurites in first and second cell at 0-2 hrs time period. Running line representing the sum of total length of neurites in first cell and dotted line representing the sum of total length of neurites at each time period 0-2 hrs in second cell.

3.14 Detection of neurite in PC 12 cells from time period 0-11hrs by MATLAB and neuron J

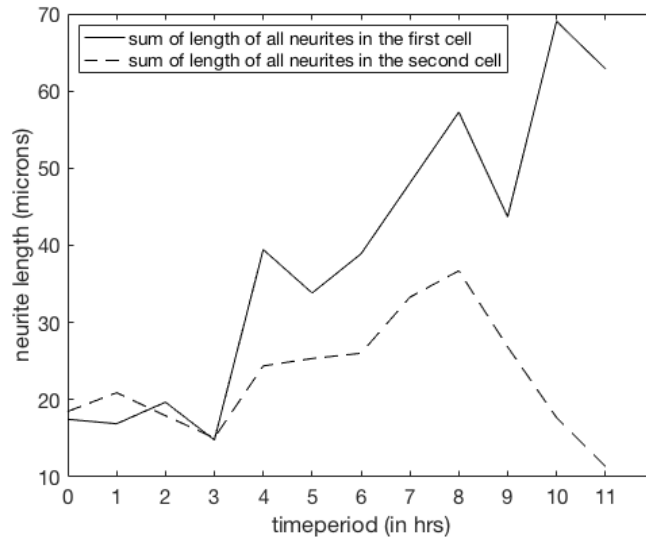


Figure 22: The dataset presentation of sum of length of neurites in the set of first and second cell. Running line representing the total length of neurites in first cell and dotted line representing the sum of length of second cell. In this dataset, the neurite dataset 3-11 hrs is measured by MATLAB and 0-2 hrs dataset is measured by neuron J.

Interestingly, we have studied here the counting and measuring the data of neurites from figure 22 and present the sum of length of neurites in both the cell first and second cell measured from both Neuron J and MATLAB. The dataset of neurites from 0-2hrs is measured by Neuron J and 3-11 hrs neurite dataset is measured by MATLAB segmentation. Both of the measurement is joined together and shown as in figure 22

3.15 Further detection of the neurites using neuron J along with their logistic curves

We have taken the neurite database of PC 12 cells which are sorted in the subcellular localization in vitro cells. In the following,

different proteins have been considered to analyze the neurites outgrowth by the Image J plugin called neuron J.

3.15.1 First, we have used the KIF3C protein in PC 12 cell data from the site http://neurite.embl.de/html/kinesin-like_protein_kif3c.html. The time-lapse data is used as O14782 c.mov which is converted into the images by ffmpeg software and then our analysis took place on those images. A sample of the images is shown below.

In particular, there are 2 cells in the images which are separately analyzed at different time intervals from 0-8 hours. The cell in which the neurites are firstly measured is named as first cell and the cell in which the neurites are measured latter is named as second cell. One of the images is shown below as figure 23(A):

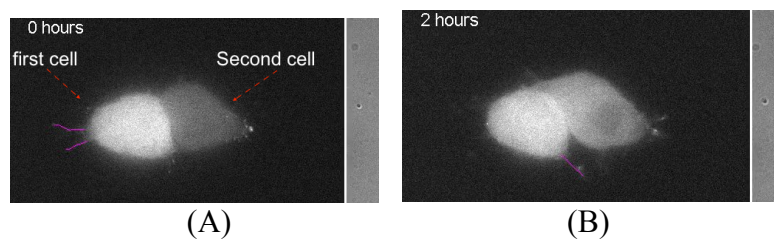


Figure 23: Images with detected neurites by neuron J of KIF3C of PC 12 cells in first cell at different time periods (A) 0 hours time period (B) 2 hours time period Scale bar is 10 μ m.

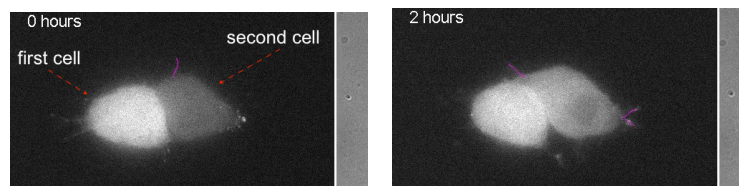


Figure 24: Images with detected neurites by neuron J of KIF3C of PC 12 cells in second cell at different time period (A) 0 hours time period (B) 2 hours time period Scale bar is 10 μ m.

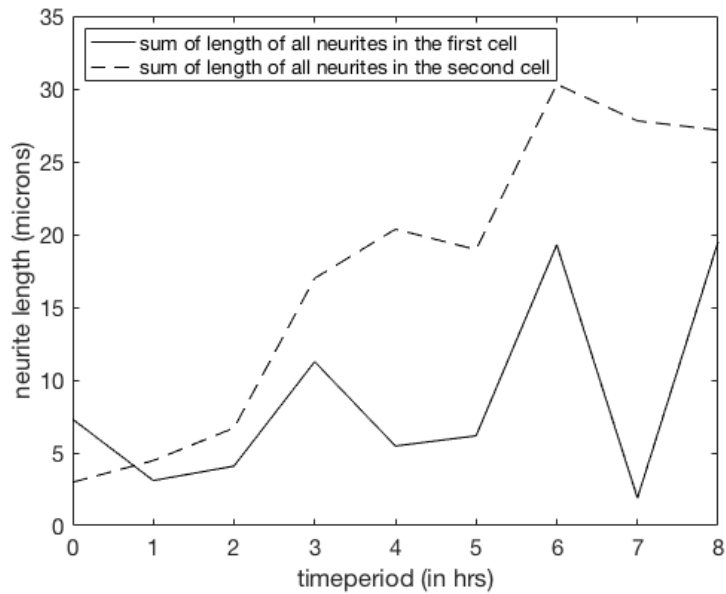


Figure 25: Graphical representation of the sum of the length of all the neurites at the particular time period from 0-8 hrs in KIF3C proteins in PC12 cells. The measurement is according to the neurite length measured at the different time intervals in KIF3C protein in PC 12 cells.

3.15.1.1 Logistic curve analysis of the KIF3C in PC 12 cells :

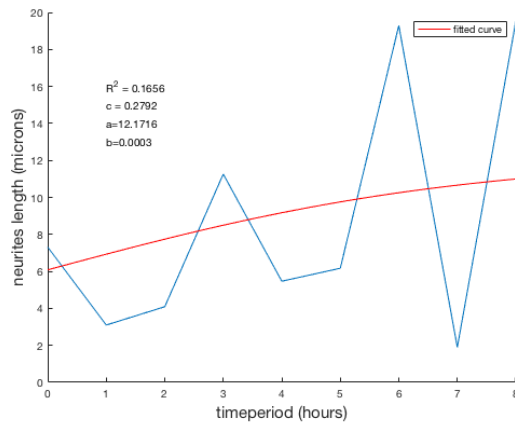


Figure 26: Graphical representation of the logistic curve of neurite length of first cell values from 0-8 hrs time period. Y-axis values are in microns and X is the time period in hrs. Red line is presenting the fitted curve in which R^2 value is 0.1656.

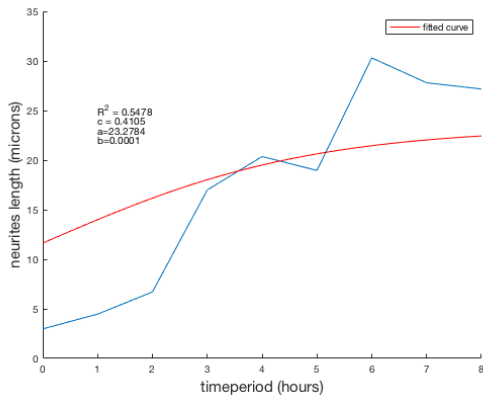


Figure 27: Graphical representation of the logistic curve of neurite length of second cell values from 0-8 hrs time period. Y-axis values are in microns and X is the time period in hrs. Red line is presenting the fitted curve in which R^2 value is 0.5478.

3.15.2 **EB-1:** Another analysis is taken place by taking EB1 protein (gfp tagged) labeled PC 12 cells for the analysis from the web link http://neurite.embl.de/html/microtubule-associated_protein2.html. It belongs to the RP/EB family member 1 and is known as microtubule-associated protein.

The time-lapse data is used as EB1 1.mov which is converted into the number of images by ffmpeg software and then our analysis took place on those images. In particular, there are 2 cells in the images which are separately analyzed named as first and second cell changing their morphology at different time intervals from 0-11 hours. One of the image is shown below:

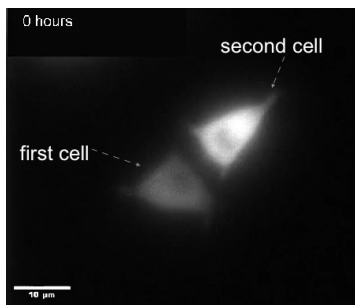
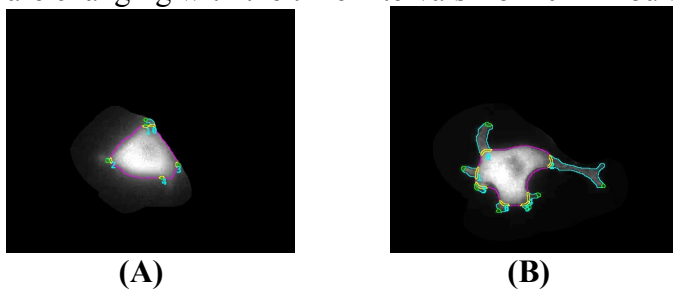
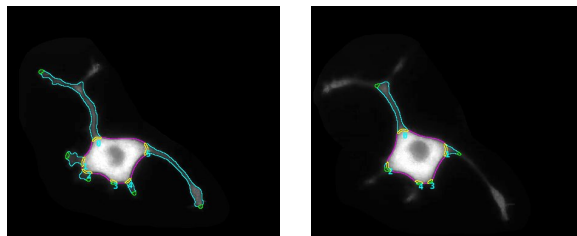


Figure 28: Sample Image showing the first cell and second cell at 0 hours.

Below are some images shown in which the morphology of the cells are changing with the time intervals from 0-11 hours.

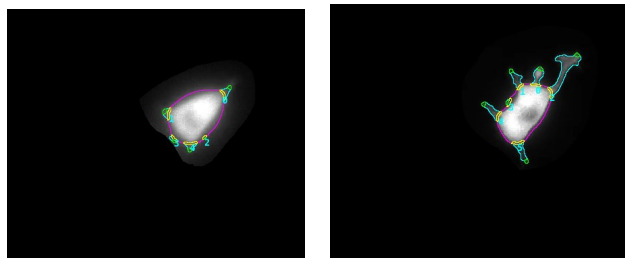




(C)

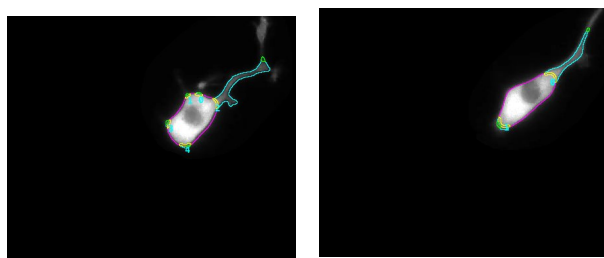
(D)

Figure 29: Images with detected neurites by Filopodyan in microtubule-associated protein of PC 12 cells in the first cell (A) Neurite detected in PC 12 cells expressing microtubule-associated protein in first cell at 0 hours time period (B) Neurite detected in PC 12 cells expressing microtubule-associated protein in first cell at 4 hours time period (C) Neurite detected in PC 12 cells expressing microtubule-associated protein in first cell at 8 hours time period (D) Neurite detected in PC 12 cells expressing microtubule-associated protein in first cell at 11 hours time period. Scale bar is 10 μ m



(A)

(B)



(C)

(D)

Figure 30: Sample Images with detected neurites by Filopodyan (image J plugin) in microtubule-associated protein of PC 12 cells in second cell dataset (A) Neurite

detected in PC 12 cells expressing microtubule-associated protein in second cell at 0 hours time period (B) Neurite detected in PC 12 cells expressing microtubule-associated protein in second cell at 4 hours time period (C) Neurite detected in PC 12 cells expressing microtubule-associated protein in second cell at 8 hours (D) Neurite detected in PC 12 cells expressing microtubule-associated protein in second cell at 11 hours. Scale bar is 10µm

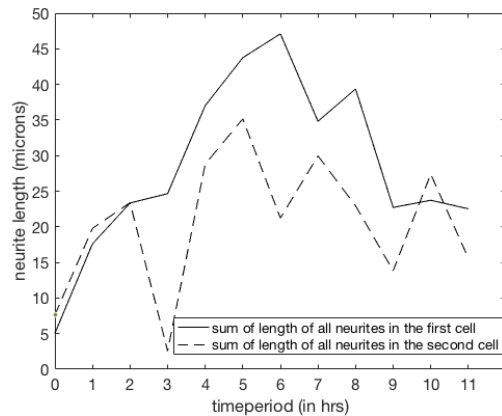


Figure 31: Dataset representation for the sum of length of all neurites at particular time period from 0-11 hrs in PC12 cells expressing membrane-associated protein.

3.15.2.1 Logistic curve analysis of EB-1 microtubule-associated protein in PC 12 cells:

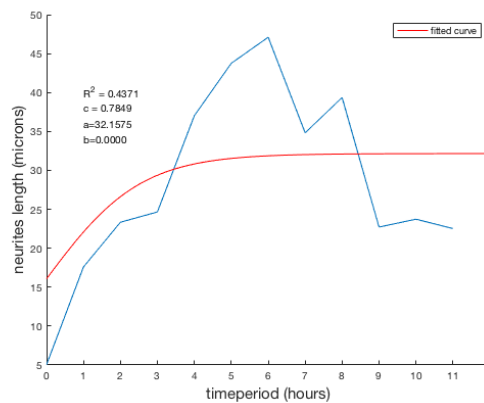


Figure 32: Graphical representation of the logistic curve of neurite length of first cell values from 0-11 hrs time period. Y-axis values are in microns and X is the

time period in hrs. Red line is presenting the fitted curve in which R^2 value is 0.4371.

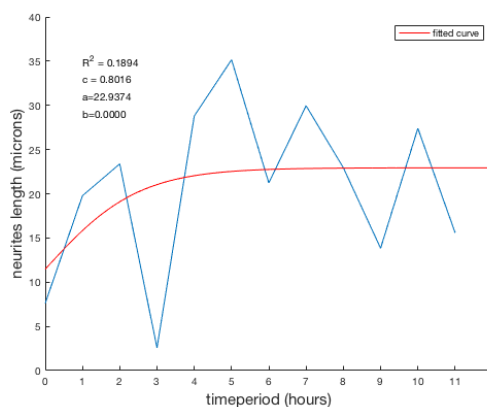


Figure 33: Graphical representation of the logistic curve of neurite length of second cell values from 0-11 hrs time period. Y axis values are in microns and X is the time period in hrs. Red line is presenting the fitted curve in which R^2 value is 0.1894.

3.15.3 Tubulin alpha 2, TUBA 2: Tubulin is known as the major constituent of microtubules. It is a protein that is localized in PC12 cells inside the microtubules. We have taken the data of tubulin alpha 2, TUBA 2 from the neurite database website for the analysis of the neurites http://neurite.embl.de/html/tubulin_alpha_2_tuba2.html.

The time-lapse data is used as tubulin 1.mov which is converted into the number of images by ffmpeg software and then our analysis took place on those images. Below some of the images are shown in which neurite has been detected:

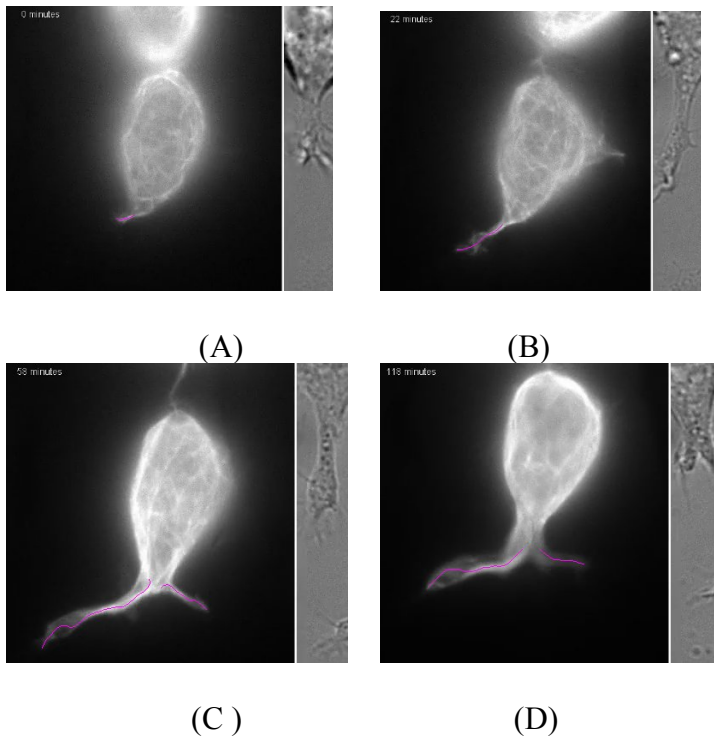


Figure 34: Images with detected neurites in tubulin alpha 2, TUBA 2 of PC 12 cells at different time period (A) 0 minutes time period (B) 22 minutes time period (C) 58 minutes (D) 118 minutes. Scale bar is 5 μ m

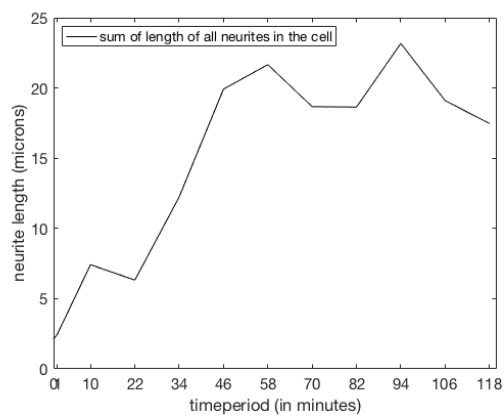


Figure 35 : Graphical representation of sum of length of all the neurites, by calculating the sum of number of neurites at each particular time period which is differentiated in minutes as : 0, 1, 10, 22, 34, 46, 58, 70, 82, 94, 106, 118 in tubulin alpha 2, TUBA 2 of PC 12 cells. TUBA 2 of PC12 cells contain only one cell including neurites measured by neuron J.

3.15.3.1 Logistic curve analysis of tubulin alpha 2, TUBA 2 in PC 12 cells:

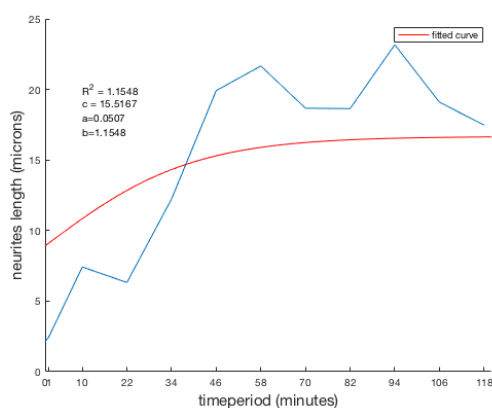


Figure 36: Graphical representation of the logistic curve of neurite length of the cell values at 0, 1,10, 22, 34, 46, 58, 70, 82, 94, 106, 118 minutes time period. X axis values are in microns and Y is the time period in minutes. Red line is presenting the fitted curve in which R^2 value is 1.1548.

3.15.4 Hypothetical protein Q9H0Q7 : It is a protein that is localized in PC12 cells inside the cytoskeleton - microtubules. We have taken the data of hypothetical protein from the neurite database website for the analysis of the neurites http://neurite.embl.de/html/hypothetical_protein_q9h0q7.html. The time-lapse data is used as Q9H0Q7 3.mov which is converted into the number of images by ffmpeg software and then our analysis

took place on those images. In particular, there are 2 cells in the images which are separately analyzed named as first and second cell changing the morphology at different time intervals from 0, 3, 6, 15, 24, 33, 42, 51, 60 minutes. Some of the images are shown below:

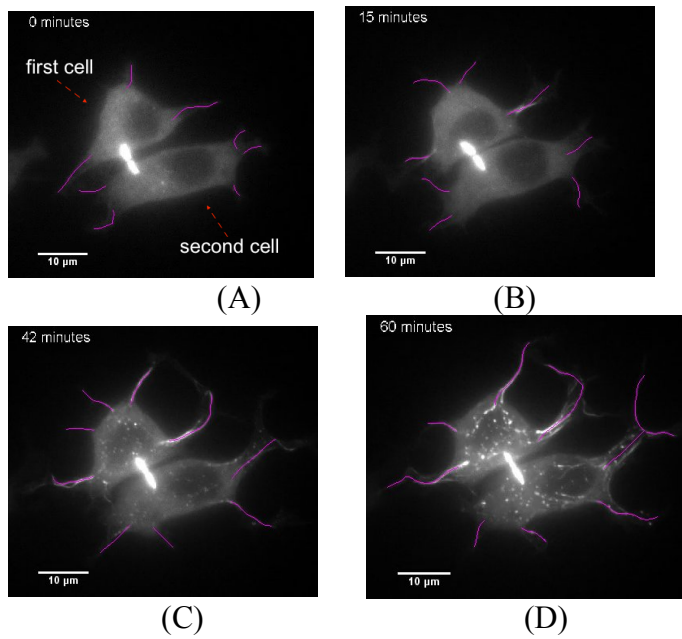


Figure 37: Images with detected neurites in hypothetical protein Q9H0Q7 of PC 12 cells in first and second cells (A) 0 minutes time period (B) 15 minutes time period (C) 42 minutes time period (D) 60 minutes time period. Scale bar is 10µm

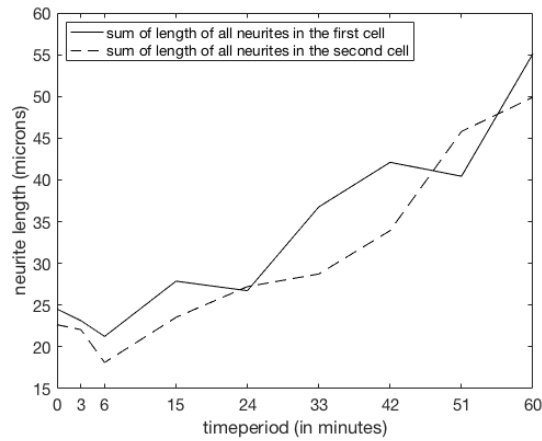
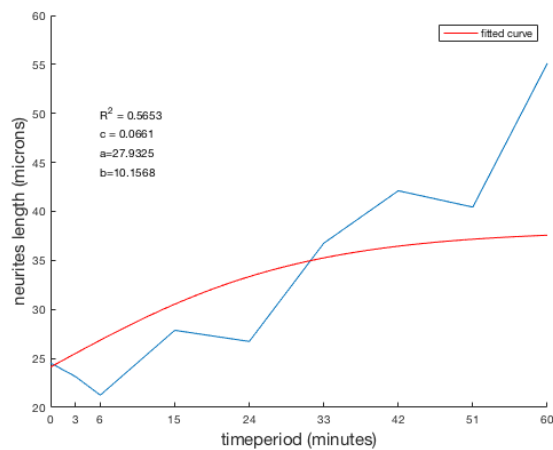
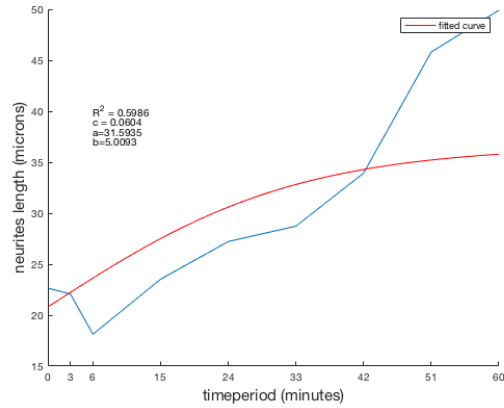


Figure 38: Graphical representation of the sum of length of all the neurites, at each particular time period of first and second cell values which is differentiated in minutes as : 0, 3, 6, 15, 24, 33, 42, 51, 60 in hypothetical protein Q9H0Q7 of PC 12 cells.

3.15.4.1 Logistic curve analysis of hypothetical protein Q9H0Q7 in PC 12 cells:



(A)



(B)

Figure 39: Graphical representation of the logistic curve of neurite length of the cell values at 0, 3, 6, 15, 24, 33, 42, 51, 60 minutes time period. Y axis values are in microns and X is the time period in minutes. Red line is presenting the fitted curve (A) logistic curve of the first cell values with the R^2 value as 0.5653 (B) logistic curve of the second cell values with the R^2 value as 0.5986.

3.16 length of single neurite measured by Image J

In this current study, we studied the length of single neurite of PC 12 cell which is manually detected by software image j as shown in the figure below:

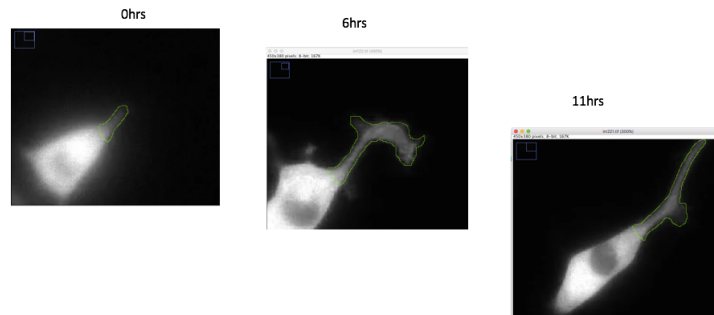


Figure 40: The representation of single neurite length at position 0 measured by Image J at different time intervals

Here, I have opened the image in image J and then measured each neurite length by selecting the neurite by freehand line in image J and then I have click on the analyze—measure to measure the neurite length.

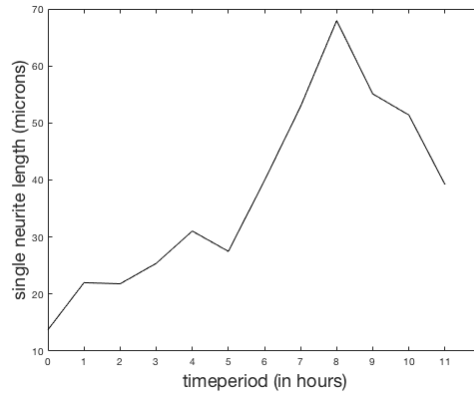


Figure 41: The length of single neurite at the position 0 neurites measured by image J from time period 0-11 hrs.

3.17 length of single neurite measured by Filopodyan

In this study, we have demonstrated that the length of single neurite of eb-1 protein analyzed from image J and filopodyan have different results as shown in figure 41 and figure 43.

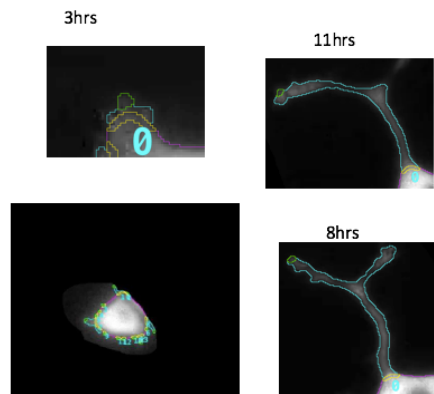


figure 42: The figure representing the single neurite at position 0. The length of neurite at different time interval from 0-11 hrs.

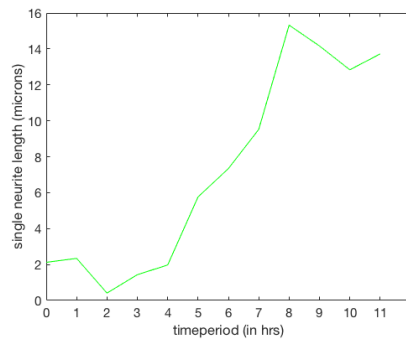


figure 43: The single neurite length at position 0 is measured by filopodyan at different time intervals from 0-11 hrs.

By focusing on the neurite analysis we also plotted the profile graph at time period 0 of eb1-1 protein as shown in figure 44, with this plot

profile we can display the 2D graphical plot for the intensities of the pixel as a line within the image. Here in this image below representing the x axis as the distance along the line and y axis is representing the pixel intensity.

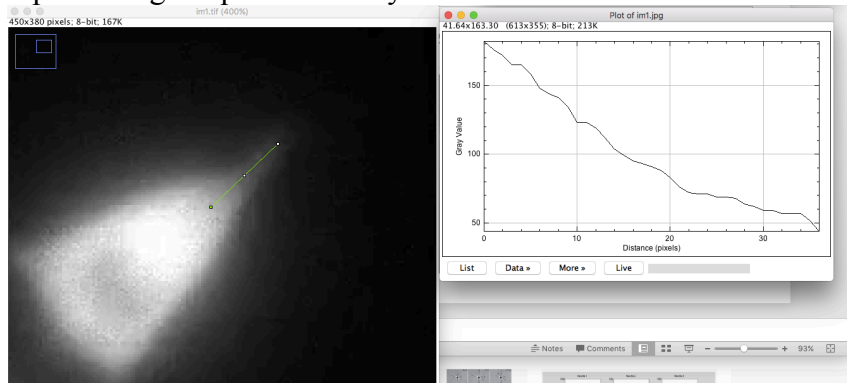


figure 44: the plot profile graph at time period 0 of eb1-1 protein. At the left representing the image of eb1-1 cell with the line measurement of the neurite. At the right the representation of the plot profile.

In the current study, the result tells us that the single neurite measured by Image J at different time periods showing the different fluctuations as compared to the single neurite measured by Filopodyan as shown in figure 41 and 43. The results from Image J and filopodyan of single neurite measurement at position 0 is showing difference. On the other hand, the length of neurite measured by Image J is decreasing after 8hrs as compared to the length of neurite measured by Filopodyan which is also showing the same result after 8 hrs. There is as such no much difference between the two measurements. Further, the plot profile is formed by Image J as shown in figure 44 which is representing the intensities of the pixels. As shown in figure 41 the neurites are changing from time to time at different time intervals at position 0 when measured by Image J the neurites are having different lengths at different time intervals. Whereas when measured by Filopodyan as shown in figure 43 it is showing the different length at position 0 at different time interval which shows the different structures of neurites.

3.18 Validation of a tools and softwares for identification and analysis of Neurites

We have used Image J plugin Neuron J which is a fully manual tracing method to measure the length of neurites.

The disadvantage of this method is that it still requires the user interaction, which allows the analysis of only a manageable number of neurites. On the other hand, Filopodyan – another image J plugin can work without user interaction but it is useful only for counting of the neurites present in the image and measuring their lengths.

Furthermore, We have also written the MATLAB scripts to extract the neurite information from the images. Interestingly, we were able to extract the neurite information like – length, number, and correlation between the shape of neurites as we can see in the results above.

There are other existing softwares which are useful for neurite outgrowth quantification as discussed in literature [92].

One of the interesting software is Neuron Cyto which is fully automated and helpful in visualizing the neurite length, complexity, quantity of soma and soma size.

In conclusion, we presented various softwares which are useful for the segmentation of neurites in images based on quantitative analysis. The neurites analysis which is described in previous section allows both automated segmentation and tracking of images to obtain the neurite results. Properties such as length, count, and other changes which can be easily extracted from the images.

Compared to other softwares our analysis can be expand by using the softwares like Neuron Cyto.

Chapter 4

4. Materials and Methods

4.1 The experimental sample of Cytospace

Overview

The experiment Cell Shape and Expression (CYTOSPACE) is one of the nine scientific research and technology projects selected, which is developed and managed by the Italian Space Agency (ASI) to be part of the Italian scientific complement for the Futura Mission to the International Space Station (ISS). The experiment, designed by the Cytospace Team and led by Kayser Italia, was carried out on the International Space Station (ISS) by Samantha Cristoforetti, starting from April 16th, 2015.

The experiment CYTOSPACE aims for the first time to create an experimental model capable of highlighting the relationship between microgravity, cell shape, and gene expression. In particular, cytospace experiment will investigate the *in vitro* effects exerted by microgravity and melatonin exposure on breast carcinoma cell line MCF-7 by exploring morphological, cytoskeletal and genetic changes with respect to parallel on ground experiments.

CYTOSPACE experiment has been carried out autonomously inside Kayser Italia's Experiment Unit, which accommodates a culture chamber, five reservoirs filled with chemicals, a fluidic system through which fluid displacement takes place and a microcontroller board. The experiment unit was accommodated into the Kayser Italia's Experiment Container (KIC-SL), designed to fit into the KUBIK incubator facility in the Columbus Lab on the ISS.



4.1.1 Cytospace Data analyzed by MATLAB

I have received a big set of confocal images like these as below shown into the figure with all the cells and tubulin separately. Data is being divided into 3 parts [82]: 1. experimented on ground (with and without melatonin) 2. experimented on ground in RPM without melatonin and during docking in RPM without melatonin. 3. experimented in space (with and without melatonin) and during docking without melatonin [82].

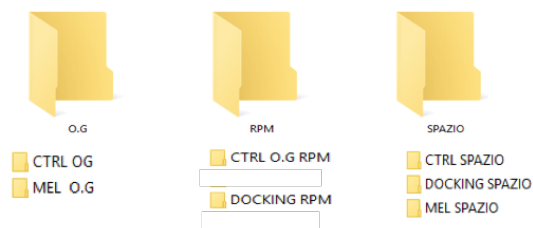
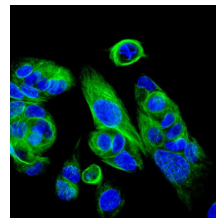
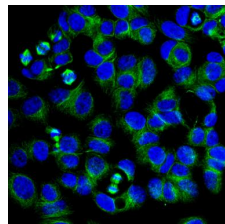


Figure 45: The figure showing the folders of the data of CYTOSPACE.

EACH FOLDER CONSISTS IMAGES LIKE THESE BELOW:



CTRL on ground

E1 docking space

4.2 Pipeline for the analysis of nuclei and tubulin by Matlab

- The chosen image is firstly converted into the grayscale;
- Threshold has been applied on the grayscale image;
- morphologically holes are filled with the data (nuclei, tubulin);
- nuclei/tubulin from the borders are cleared from the image;
- Reducing the noise from averaging filter and median filter
- Removing the small objects from the binary image;
- A watershed segmentation is performed on the binary image
- Calculation of entropy has been done, Number of objects counted, morphological parameter (area, perimeter, minor axis, major axis, circularity, solidity) have been calculated.

4.2.1 Image Processing

The use of the computer algorithms is to enhance the digital images of the nuclei and microtubules in order to prepare them for the morphological analysis the main aim of this chapter. Thus, MATLAB is a great programming language for technical computing is used. There are basic MATLAB distribution which can be expanded by adding a range of toolboxes for the various applications. The particular toolbox which is interest to us as the

image processing toolbox. This software provides us a comprehensive set of reference – standard algorithms as well as graphical tools for image processing, analysis, visualization, and algorithmic development. Hence, in this section, the MATLAB program has been aimed to process the three types of experimental data experimented on ground and in space to extract the nuclei and tubulin. The MATLAB function and the resulting algorithms are discussed below.

4.2.2 MATLAB program for the cytospace data

MATLAB program for measurement of Area, Perimeter, Major axis, and Minor axis and Entropy of the nuclei/Tubulin in all 3 experimental data (on the ground, in space and docking).

The sample images are shown above in the figures. Like this, we have a couple of images in every experimental data in a .tiff type image of size (1024*1024); 8 bit with 1MB. These are the raw image is the input for the MATLAB programming.

The program basically works in three main phases;

- reading the raw image
- processing the raw image
- then returning the extracted nuclei for the measurement

4.2.2.1 Reading the image

In the MATLAB software, the images are considered as matrices whose elements are the pixel value of the image. Moreover, if the input is the true color image or any other type rather than the grayscale, it has to be converted to grayscale before running the code.

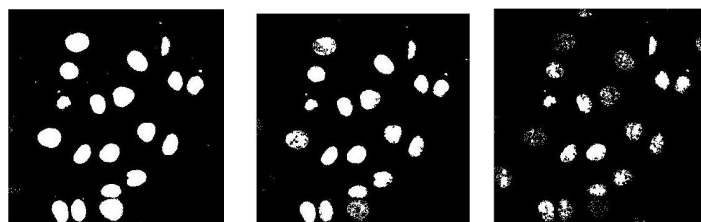
Therefore, to read the image the code is $I = \text{imread}(\text{filename.jpg})$, reads the grayscale image from the graphics file into the matrix I. Moreover, if the file contains a grayscale, I is an M-by-N array. Here, M refers to the number of rows and N is the number of columns respectively.

We use the function `rgb2gray` in the Matlab to convert the true color image into the gray-scale. However, this function is first of all

converts the RGB image into the HSL color space. As just like the RGB color space, in the HSL, every pixel has three values. Unlike the RGB, these values don't represent the individual basic color components, they stand for the hue, saturation and luminance. The first two are used to measure the color and the third one is used to measure the intensity. The conversion from RGB to HSL is simply linear transformation- multiplying the vector of the color values by 3*3 regular matrix. Hence, this function keeps only the value of the luminance in the resulting gray. Subsequently, we linearly scale the values of the pixels in Gray into the range of the [0, 1]. so that the maximum value is present in Gray is 1 , the minimum value is 0.

4.2.2.2 Thresholding the image

The next step is the image binarization after choosing the appropriate threshold value. MATLAB has the threshold function, which can thresholds an image of any data type, by using the general syntax $I_2 = I > threshold$. Where I is the raw image and further my idea is to extract the blue channel at a given threshold. Because the blue channel consists of the nuclei. All pixels under the threshold belongs to the object. By setting this threshold the raw image is displayed as the binarized image. Moreover, thresholding operation is used to convert a grayscale image into a binary image. A binary image is consisting of 2 colors whether it is black ('0') or white ('1'). A suitable threshold should be selected in order to separate the objects from the background image.



Threshold at 10

Threshold at 15

Threshold at 20

Figure 46: By setting the threshold at 10, 15, 20 gives us the following the three results respectively of the nuclei.



Threshold at 10

Threshold at 15

Threshold at 20

Figure47: By setting the threshold at 10, 15, 20 gives us the following the three results respectively of the tubulin.

Here, at the low thresholds, we have lost several pixels of the nuclei. As increasing the threshold value, more and more pixels are fall under it.

4.2.2.3 Removing particles

After thresholding, we need to remove the impurities and particles within nuclei which can strongly affect the quantitative analysis on the cell boundaries and hence it should be removed. To this aim, the region filling operation should be used. The function *imfill*, $I = imfill(I2, 'holes')$, which identifies the pixels constituting holes within the connected components in the binary image and fills them. Moreover, a hole is a set of connected background pixels that can't be reached by filling in the background from the edge of the image. The resulting image is shown in the figure

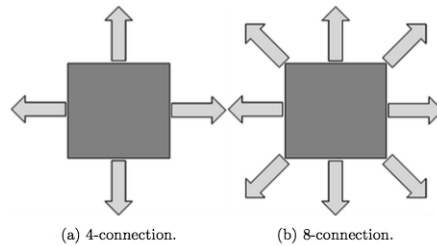
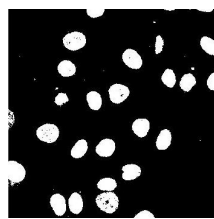


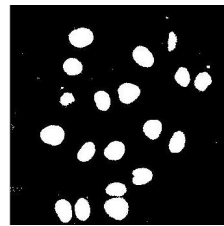
Figure 48: this figure representing the concept of connectedness.

4.2.2.4 Clearing the border components

The image processing function $I_4 = \text{imclearborder}(I_3)$; is used to remove the connected objects of a binary image that touches any image border. This function is used to clear the image border. It tends to reduce the overall intensity level in addition to suppressing border structure. The nuclei of the interest has been successfully segmented by thresholding. Further, the objects which are connected to the border of the image has been removed by using the `imclearborder` function. The connectivity in the `imclearborder` function has been set to 4 to remove the diagonal connections [93].



(A)



(B)

Figure 49 (A) representing the thresholded image without removing border components as nuclei whereas figure (B) representing the image removing the border components as nuclei.

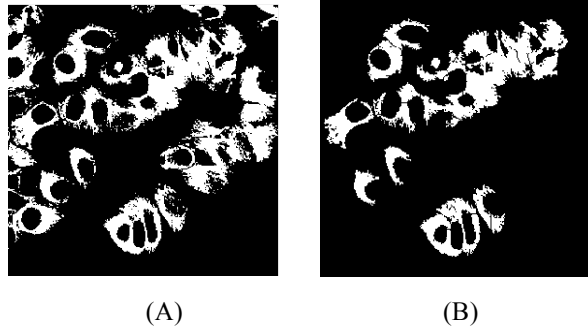


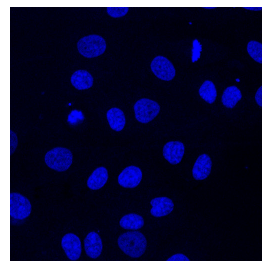
Figure 50 (A) representing the thresholded image without removing border components as tubulin whereas figure (B) representing the image removing the border components as tubulin.

4.2.2.5 Removal of the noise by averaging filter and median filter

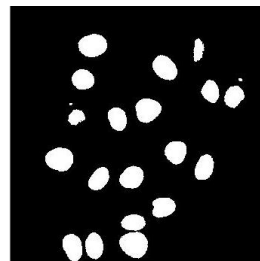
Now after clearing the border components next step is to filter the noisy image through averaging filter. Therefore, we are applying filters as average filter. Median filter is used to filter the noisy image, $[J]. K_{median} = medfilt2(I4)$; is used as a median filter to remove the noise from the binarized image.

4.2.2.6 Particle removing step from the processed image

If there is any small particle still left remained inadvertently after the removal of the noise from the median filter, `bwareaopen` function is used to eliminate it. $I5 = bwareaopen(K_{median}, 100)$; this function helps to removes from a binary image all connected components (objects/nuclei/microtubules) that have fewer than 100 pixels, and then produces another binary image, $I5$. The default connectivity is 8 for the two dimensions.

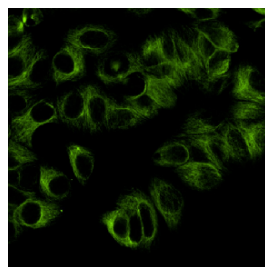


(a) Original nuclei

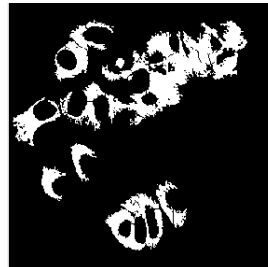


(b) Extracted nuclei

Figure51 : Initial image vs processed image.



(a) Original tubulin



(b) Extracted tubulin

Figure52: Initial image vs processed image.

4.2.2.7 Image Segmentation by watershed algorithm

Segmentation is seemed to be one of the key issues in modern medical image analysis which is enabling in the numerous clinical applications. However, segmentation is the process of assigning the labels into the pixels in 2D images or inside the voxels of 3D images. The procedure is that the image is split into the segments which are also known as regions or areas. Further, in the medical imaging segmentation is very essential for the quantification of the data and for the 3D visualization of the appropriate image data. Moreover, for clinical purposes, the segmentation techniques with the use of MRI has been widely used in the monitoring of the brain infarctions, as well as brain tumors.

Watershed transform is been said to be the most useful image segmentation [94]. The watershed lines are effectively divided into the individual catchment basins in the raw or gradient image. The methodologies of the image segmentation is usually based upon the watershed transform algorithm which has been developed and improved from the past decade. The term watershed is referred to as the ridge which divides the areas drained by different river systems. Hence, a catchment basin is area usually draining into the river or the reservoir [94]. Thus the watershed transformation in a mathematical morphology is known as a very powerful tool for image segmentation [95]. The watershed transform applies to the gray-scale in the image processing in a way that can be used to solve a variety of image segmentation problems. Here, I am presenting the watershed algorithm applied on the grayscale images below:

4.2.2.8 Watershed algorithm

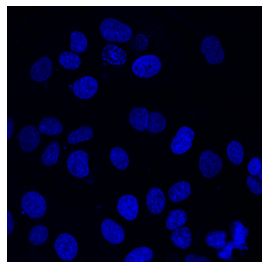
After removal of noise, the next step is to run the watershed algorithm for those nuclei which are joined with each other as shown in figure 50. We will use the watershed algorithm on the filtered image which is produced from the section above 4.2.2.6

Hence the first code to be implemented is $I6 = watershed(I5)$; used to calculate the watershed of the processed image.

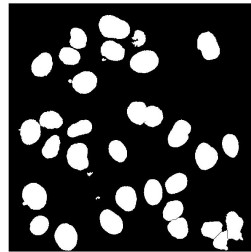
$Lrgb = label2rgb(I6)$; used to calculate the label image, $imshow(imfuse(I5,Lrgb))$; $axis([10 175 15 155])$, this code is used to fuse the two images together and then zooming in into one particular blob. Next, we use $bw2 = \sim bwareaopen(\sim I5, 10)$; to remove very small dots. It is also used to removes them in the foreground, that's why we have complemented the image before as well as after the calling $bwareaopen$. Then, $D = -bwdist(\sim I5)$; $Ld = watershed(D)$; requires to compute the watershed transform of D. Then, $bw2 = I5$; $bw2(Ld == 0) = 0$; used to convert the watershed ridge lines, in white and correspond to $Ld == 0$. After that, by using these 2 functions the ridge lines to segment the binary image by changing the corresponding pixels into the background. Then, the

watershed-based segmentation methods is used to filter out little local minima by using `imextendedmin` and then later can modify the distance by transforming so that no other minima would occur at the filtered-out locations. This is known as “minima imposition” which can be implemented via the function `imimposemin`. `mask = imextendedmin(D, 2);`

`Ld2 = watershed(D2); bw3 = 15; bw3(Ld2 == 0) = 0; imshow(bw3);` and then finally to modify the distance transform which only has minima at the desired locations. the final image we get is shown in the figure below:

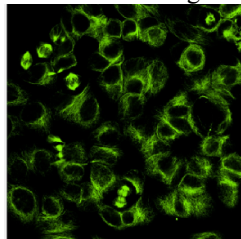


(a) Original nuclei

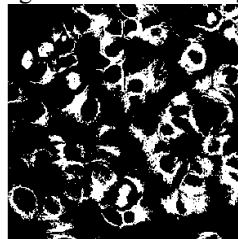


(b) watersheded nuclei

Figure 53: Initial image vs watershed image.



(a) Original tubulin



(b) watersheded tubulin

Figure 54: Initial image vs watershed image.

4.2.2.9 Determination of the entropy of nuclei and tubulin

Entropy can be defined as the measure of the information content. To achieve the high compression ratio, entropy should be in low values. In image processing, the entropy E is known as the amount of information which is present in an image in other terms, it is defined as the measure of randomness. It is defined by [96,97].

Here, Entropy is defined as:

$$H(x) = \sum_{i=1}^n P(x_i) \log_b(p(x_i))$$

although (x_i) is representing the probability mass function of the outcome x_i . (i.e H is predicted as the entropy of the grayscale image, p is known as the probability of the given intensity value in the image, for example: if $b = 2$, then we will get the unit of entropy as in bits) [98].

Logically defining entropy is that it is the measure of disorder, a neighborhood should be taken into account, which is popularly known as local entropy [99]. In MATLAB command entropy can be find out by the command as `entropy(image)`, which computes the value (single value) of the logical array or the grayscale image. Therefore, here we used, $J = \text{entropy}(bw3)$; which returns J, a scalar value representing the entropy of the grayscale image `bw3`.

4.2.2.10 Determining the number of nuclei and tubulin, area, perimeter, major axis, and minor axis:

The task of counting the nuclei by MATLAB program of the cytospace data has been approached. Although there are different specific techniques which differ. This type of geometric features can be done in MATLAB by using specific commands.

The cell and tubulin count function which is used in the binary image to count each individual nuclei/tubulin. We have the cytospace data to differentiate the neuclei from tubulin which were in close proximity from one to another. `bwconncomp(image,conn)` function in MATLAB can label each and every cell within the image. Moreover, this `bwconncomp` function also allow us for the usage of

the region props function which can be provided as the information about each cell's and tubulins- area, perimeter, major axis, and minor axis.

Hence, these morphological parameters were found to be calculated by the use of command regionprops (cc, area, perimeter, majoraxis, minor axis) operation in which 'cc' is representing the stack of the connected components within the binary image. The stack, 'cc', is found to be throughout the use of the bwconncomp (bw3,conn) command, in which bw3 was the image and 'conn' is the connectivity. Thus, the cell area values were further overlaid on the original image by using each of the cell's center of the mass as a plot point.

The command which is used to count the number of connected objects and number of objects in the watersheded image we use $cc = bwconncomp(bw3, 8); n = cc.NumObjects;$

4.2.2.11 Determining the number of circularities and solidity of the neuclei and tubulin in the image:

In addition to the cell area, perimeter, major and minor axis. cell circularities and solidity is another method to describe the cell morphology. Neuclei that are quantified by the function 'cell circularity' are utilized to differentiate the ratio between the length and the width of the cell therefore in order to calculate the cell circularity value. The command which is used to extract the circularity data is 'circularities=perimeter.^2./(4*pi*filledarea)'. This will help to obtain the cell behavior which is considered as the most informative parameter of the cell. Thus, circularity is calculated via $circularity = 4\pi \text{ area}/\text{perimeter}^2$ [100,101]. This measure, however, indicates the perimeter of a perfect circle at the value of 1.0 and the decrease from this value would reflect the less circular in shape [102].

Another morphological parameter as a shape descriptor of the cell which is used to describe the structure of the cell is the 'Solidity'. Solidity is computed as the ratio between the object of the area and

the convex envelope area

$$Sol = A/A_c$$

Here, A is the area of the cell, A_c is the proportion of the pixels in the convex hull that are also in the region.

4.2.3.1 Basic process of tubulin image analysis

Data of confocal tubulin image:

The color images are used three channels (red, green and blue) (RGB) to produce the overall composite. The RGB images require a 3D array to convey the extra color information. The first plane in the third dimension is presenting the red pixel intensities, the second plane presenting the green pixel intensities and the third plane presenting the blue pixel intensities. In this analysis, we have to convert the RGB image to grayscale to perform the tubulin analysis to utilize the green channel alone to enhance the contrast between the background and the features such as tubulin. There are so many processing and measurement tools, which are written to operate on the greyscale images, and this may need to be extracted from RGB color image.

Furthermore, a binary image is one which contains only black and white pixels. The image consists of a binary array, typically of 0's and 1's. Images of any type which may be converted to this format for the processing or analysis.

Image processing

The image processing task is used to transform the grey values of the pixels. There are basically three mechanisms by which it can be done. a) in its most simple form of the pixels the grey values are changed without any processing of the surrounding or neighborhood pixel values. b) Neighborhood processing is used to incorporate the values of the pixels in a small neighborhood around each pixel in question. c) Finally, the transforms are used to more complex and involve the manipulation of the entire image so that the pixels values

are represented in a different but equivalent form. This might allow for the more efficient and powerful processing before the image is reverted to its original mode of representation.

Thresholding

Thresholding is used to allow the separation of an image into separate components by converting it into the binary image. This usually involves the image being separated into the white or black pixels on the basis of whether their intensity value is more or less than a certain threshold level. The process of the thresholding may be particularly useful to remove the unnecessary detail or variations and the highlight detail that is of interest. For the analysis of most of the tubulin images through Matlab, I have taken different threshold value because the pixels of green tubulin in binary and grey-level converted image are having varying levels of the background illumination and by taking different thresholding values in MATLAB I have separate the tubulin data from the whole RGB image.

4.2.3.2 Algorithm

The algorithm is summarized as shown in the figure below to extract the tubulin from the confocal images.

The detailed algorithm of tubulin is the same as I have done for the Neuclei through Matlab programming therefore here we are discussing the results.

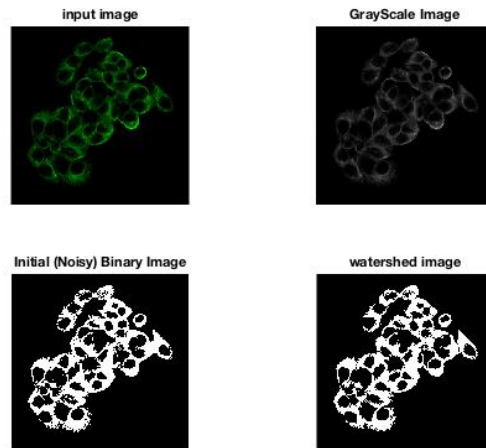


Figure 55: The figure is divided into 4 parts as (i) input image (ii) Confocal image is converted into grayscale image (iii) noise removal from the image (iv) watershed segmentation is used for the overlapping tubulin fibers.

4.3 Methods for the analysis of neurites

4.3.1 MATLAB Platform

For our study, Image processing has been done particularly on the MATLAB platform because of its various features which are exclusively found in MATLAB. With the help of MATLAB, we have measured the neurites. First of all, we have set the path of the data set in the MATLAB environment. Then, running a for-loop to read each folder in the dataset. Next step is to run a sub for-loop to read each image in the current folder one by one. We have used k-means clustering to segment the image into different clusters according to pixel intensity. Then, by detecting the cluster that contains neurites, by choosing the cluster that encircles the second-highest area (the cluster corresponding to the highest area represents the background). Actually, the background has a remarkably

uniform intensity and texture. Instead, the foreground is composed of nuclei bodies in which different segments have different intensity values. This results in the fact that the foreground is itself divided into many more clusters, while the background is represented by a single cluster because of its homogeneous intensity. By performing erosion to skeletonize the image and then detecting the branch points. These points represent the points where the neurites are attached to the nuclei body.

Finally segmenting the neurites and putting on a color overlay. And then, at last, saving the results as text and mat files in the result folder.

4.3.2 Image J plugins – Filopodyan and neuron J

4.3.2.1 Filopodyan

Filopodyan is developed as an image analysis open-source software for the analysis and for the automated segmentation of filopodia's which can also annotate to track a huge number of the dynamic filopodia's. The Filopodyan software is written in R and Fiji which is publicly accessible at <https://github.com/gurdon-institute/Filopodyan>.

The method to analyze the neurite with Filopodyan as follows:

- The chosen image is firstly open into the Fiji software.
- By free-hand line outline the particular cell for measuring the neurite.
- Removing the outside things from the edit scroll down menu “clear the outside”.
- Measuring the selected cell neurite by the plugin *Filopodyan* automatically.
- Calculation of the length of the neurites will be seen in the selected spreadsheet resulted from the *Filopodyan* plugin

A typical image from the neurite database was loaded into FIJI. With the help of the freehand line tool, we outline the particular cell for measuring the neurite. Then after cleaning the other cell, we will measure the selected cell from Filopodyan. Next, we will select Filopodyan from the plugins in FIJI. The pixel size which is set as 1 pixel, thresholds are chosen from the drop-down menu and the ED iterations, base back frames, and LoG sigma are set. The program then runs the neurites are selected in the images automatically. The length of each neurite in the table pops up once the Filopodyan filter is applied.

4.3.2.2 Neuron J

Neuron J is a very well-known image J plugin which helps us for the quantification and tracing of the elongated image structure. This program is able to handle one image at one time [103]. The following method to detect the neurites from the neuron J image j plugin:

First of all, the image is selected from the neurite database and then opens the neuron J plugin inside the image j software. For this purpose, we use the command as plugin—neuron J. Then next is to open the image in neuron J plugin by following command as neuron J---load image. Then manually trace all the neurites with the help of function inside the neuron J plugin as “Add tracing”. Then by adding all the tracing, we need to measure the tracing with “measure tracing” this will measure the length of neurites.

The pipeline to perform the morphological analysis by opting the image from neurite database with Neuron J is as follows:

- The chosen image is firstly open into image J software.
- By selecting the neuron J plugin menu of the image J. The neuron J toolbar will replace the image J toolbar.
- Then open the image to be measured for neurite detection.

- Manually load the neurites tracings and then measure the tracing of the neurites by the “Measure tracing” the option inside the neuron J plugin menu.
- Neuron J will display the tracing measurements as a length of neurites.

4.3.2.3 Cell profiler

The selected images were processed by adding modules to make a pipeline. When the pipeline is structured by organizing the modules through the setting option, then we can run our structured pipeline to produce the results. The modules are shown in the figure below as a menu of “analysis modules” which is a pipeline to analyze the neurites.

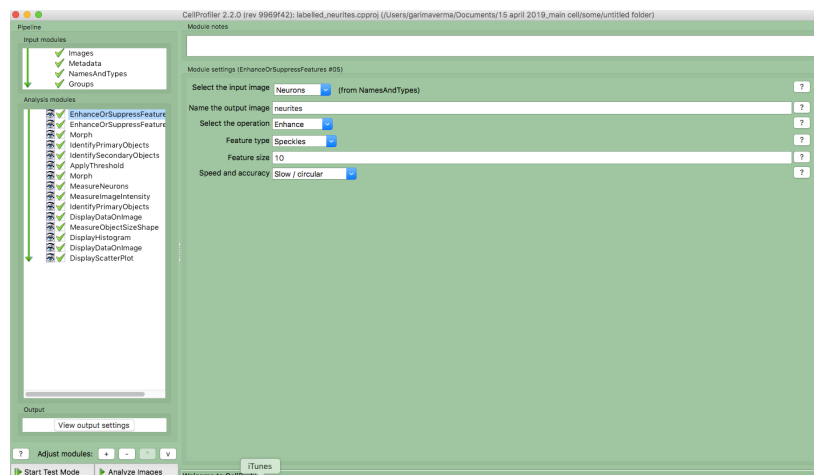


Figure56 : Screenshot of the cell profiler software showing the analysis modules which is used as pipeline for producing the results by cell profiler.

4.4 Dataset of neurites

The dataset of neurites from a published report by Laketa et al., [104] has been taken from the http://neurite.embl.de/html/microtubule-associated_protein2.html. We have analyzed the neurites from the neurite database that were labeled for EB-1, KIF3C, Tubulin alpha 2 (TUBA 2), the major constituent of the microtubules and the hypothetical protein Q9H0Q7 which is localized to the cytoskeleton of PC12 cells. The time-lapse data is used as eb1-1.mov, KIF3C.mov, Tubulin alpha 2.mov, Hypothetical protein Q9H0Q7.mov which are converted into the number of images by ffmpeg software and then used for the analysis.

4.5 The experimental Algorithms and protocols

1. The following experimental protocol for performing the quantitative morphological analysis of the data measured by Image J represents an example of procedure among a big set of realized ones [105]:

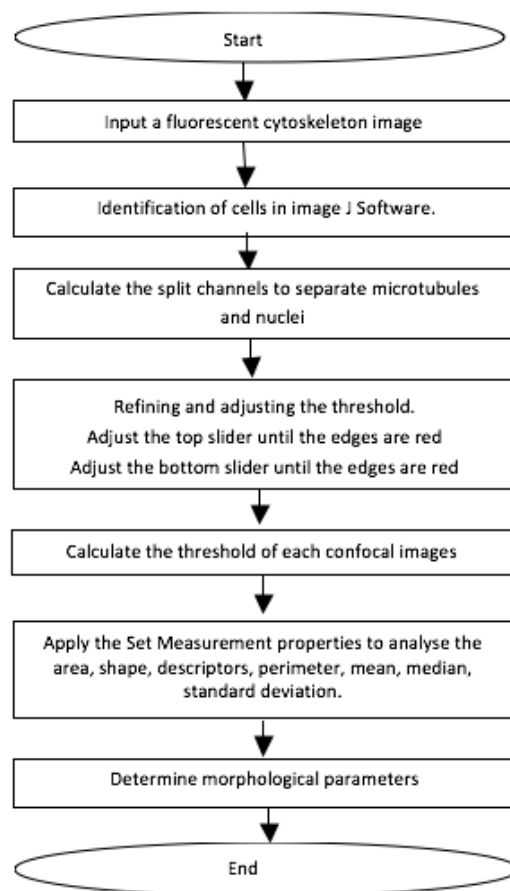
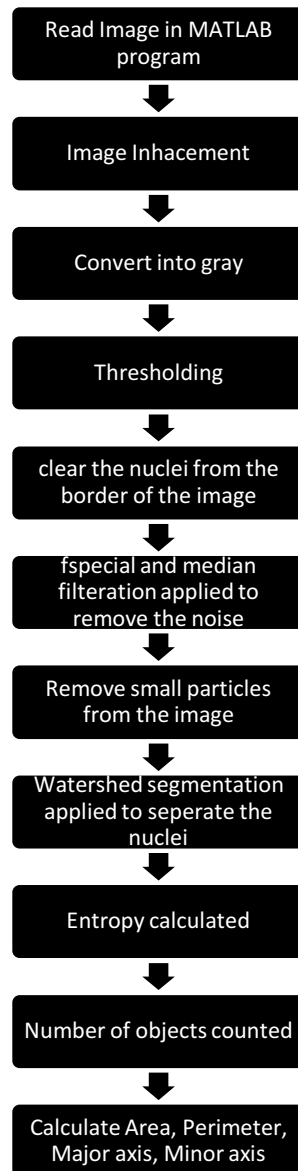
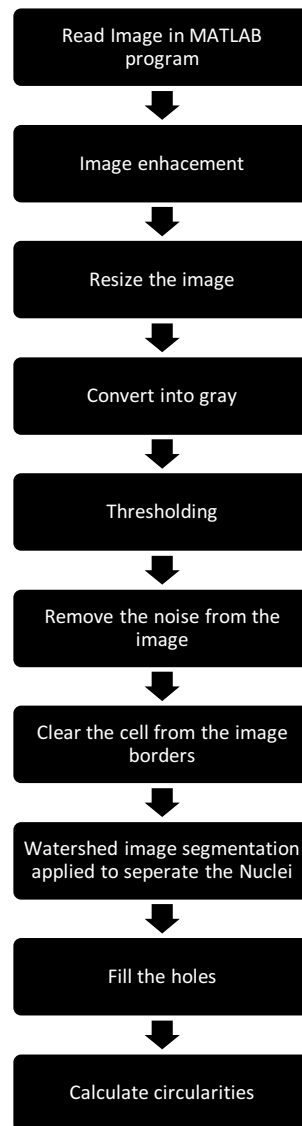


Figure 57 : Flowchart of thresholding and morphometric analysis of nuclei and tubulin from confocal microscopy images of cytoskeleton.

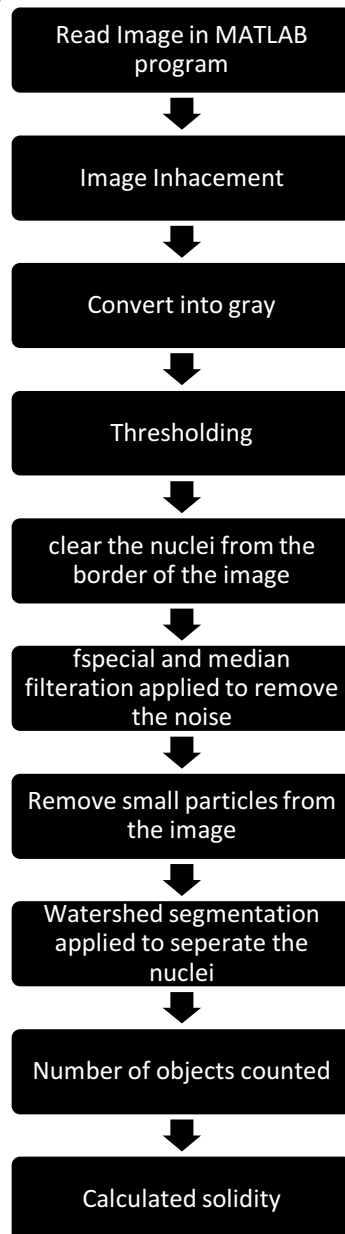
2. The algorithm for calculating area, perimeter, major axis, minor axis and entropy of the nuclei through Matlab program



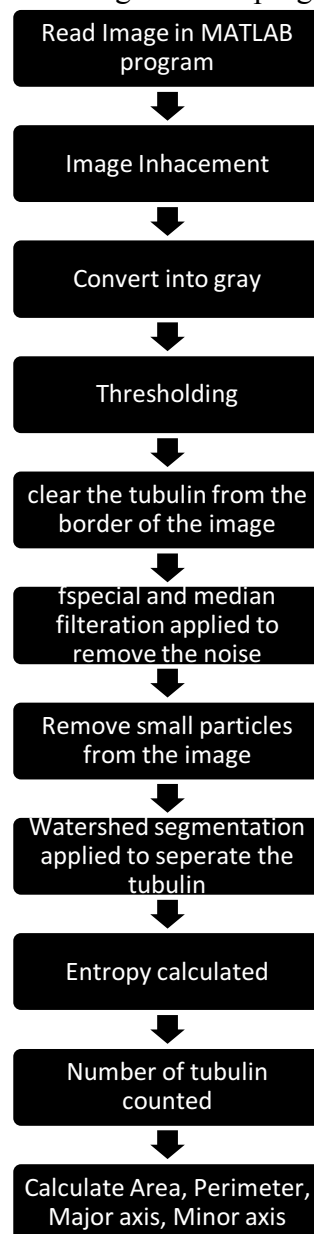
3. The algorithm for calculating circularity of the nuclei through Matlab program



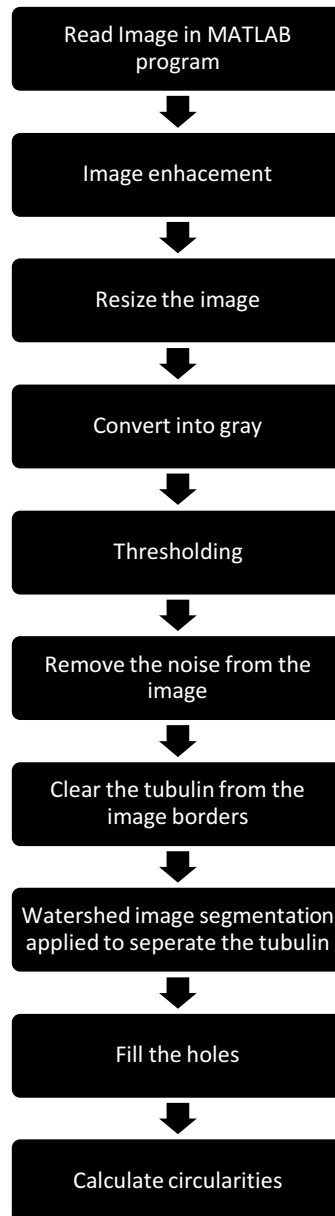
4. The algorithm for calculating solidity of the nuclei through Matlab program



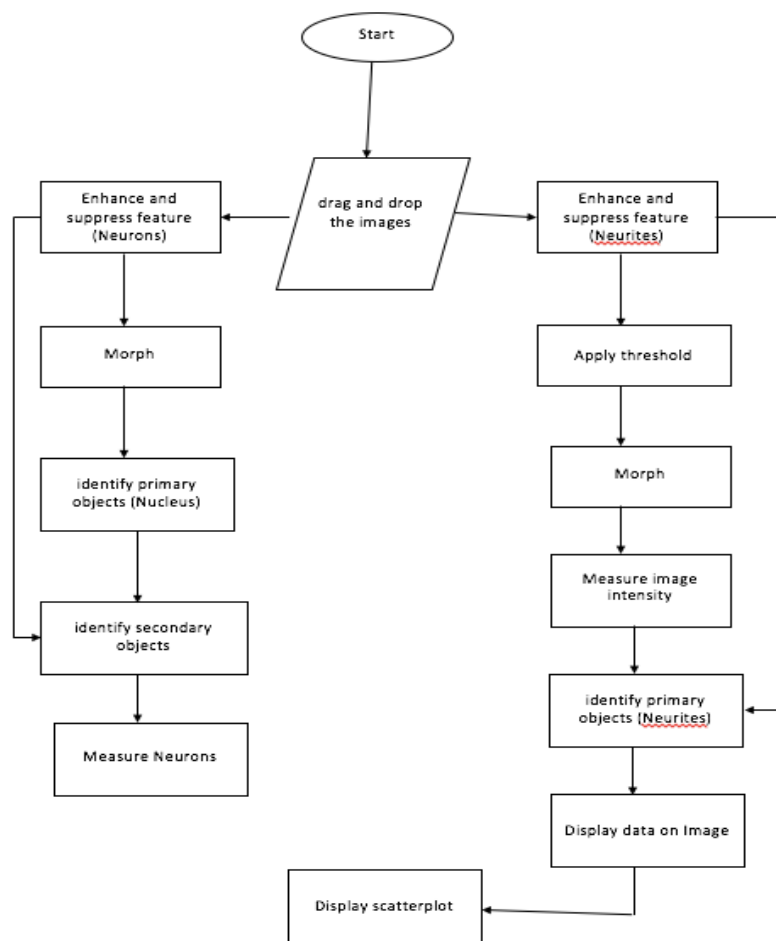
5. The algorithm for calculating area, perimeter, major axis, minor axis of the tubulin through Matlab program



6. The algorithm for calculating circularities of the tubulin through Matlab program



7. Workflow of the cell profiler by which neurites are detected in the images taken from neurite database www.neurite.embl.de



CONCLUSION

The main task of this PhD course was to build a workframe to use bioinformatics tools to support biologists. For this reason many types of software have been taken into consideration and a careful selection has been made. Furthermore, original code was developed to improve the analytical capacity in specific cases. We then identified 2 uses cases as an application example. A work of benchmarking needs instead the availability of many uses cases, each representative of an application class. A benchmarking work could be the natural consequence of the work done so far and could be a topic for an upcoming post doctoral course.

I applied my computerized methods to study the images of slides with samples of cell cultures under simulated microgravity conditions (via RPM – Random Positioning Machine) and the real exposure which came from the CYTOSPACE experiment conducted by a team of our laboratory on April 2015. The calculations have been done for evaluating the CSK architecture and density of fibers (tubulin, Microtubules, Intermediate filaments). Furthermore, we have analyzed the difference between the neurites images taken from the neurite database and examined the results by comparing (a) custom-built code using the MATLAB image processing toolbox (b) Neuron J (c) Cell profiler pipeline (d) Filopodyan. The MATLAB code for studying neurites has used K-means clustering to segment the image based on the intensity value and we manually select the cluster that contains neurite. Finally, we use some morphological operations to segment the neurites. The fitting of a logistic curve to evaluate the growth rate (r) of neurites is a first step towards a more comprehensive quantification and automation of the neurite analysis. Hence in the future, the analysis of the neurite images could be automated to measure the network properties and correlation with the cell shape.

Appendix A:

MATLAB CODES FOR THE DETECTION OF CYTOSKELETON (NUCLEI, TUBULIN) ANALYSIS

```
#Image processing of the Nuclei
a=imread('~/Desktop/spacemel.jpg');
>> R= a (:,:,1);
>> G= a (:,:,2);
>> B= a (:,:,3);
>> imshow(B,'InitialMagnification','fit');
>> imshow(G,'InitialMagnification','fit');
>> imshow(B,'InitialMagnification','fit');
>> B2 = imclearborder (B);
>> imshow(B2,'InitialMagnification','fit');
>> Kaverage = filter2(fspecial('average',3),B2)/255;
>> figure
>> imshow(Kaverage,'InitialMagnification','fit');
>> Kmedian =medfilt2(B2);
>> imshow(Kmedian,'InitialMagnification','fit');
>> C= imnoise (Kmedian,'gaussian',0.005);
>> D= wiener2(C,[5 5]);
>> imshow(D,'InitialMagnification','fit');
>> imshow(C,'InitialMagnification','fit');
>> figure ,
>> imshow(D,'InitialMagnification','fit');
>> threshold = 90;
>> E=D>threshold;
>> imshow(E,'InitialMagnification','fit');
>> F= imfill(E,'holes');
>> imshow(F,'InitialMagnification','fit');
>> L=watershed(F);
>> imshow(L,'InitialMagnification','fit');
>> Lrgb=label2rgb(L);
>> imshow(Lrgb,'InitialMagnification','fit');
>> imshow(imfuse(F, Lrgb))
>> axis([10 175 15 155])
>> G= ~bwareaopen(~F,10);
>> imshow(G,'InitialMagnification','fit');
>> H=-bwdist(~bw);
Undefined function or variable 'bw'.

>> H=-bwdist(~F);
>> imshow(H,'InitialMagnification','fit');
>> Ld = watershed(H);
>> imshow (label2rgb(Ld))
In imshow (line 332)
>> G=F;
>> G(Ld == 0) = 0;
>> imshow(G,'InitialMagnification','fit');
```

Garima Verma

```
>> mask = imextendedmin (H,2);
>> imshowpair(F,mask,'blend')
> In images.internal.initSize (line 71)
  In imshow (line 332)
    In imshowpair (line 126)
>> D2=imimposemin(H,mask);
>> Ld2=watershed(D2);
>> bw3=F;
>> bw3(Ld2 == 0)= 0;
>> imshow(bw3,'InitialMagnification','fit');
```

#Counting the nuclei and calculating the density of the Nuclei

```
I=imread('nuclei.jpg');
b=imopen(I,ones(5,5));
figure;imshow(b,'InitialMagnification','fit');
d=bwareaopen(b,1000);
cc=bwconncomp(d);
objects_per_unit_area = cc.NumObjects/(size(d,1) * size(d,2));
```

#To set the contrast in the image

```
>> clear
>> i=imread('image_epi.jpg');
>> h1=figure;
>> imshow('image_epi.jpg');
>> h2=figure;
>> imcontrast(h1);
```

#to set the contrast in the tubulin image

```
a=imread('Tubulin.jpg');
imcontrast
h1=figure;
imshow('tubulin.jpg');
h2=figure;
imshow('tubulin.jpg');
imcontrast(h1);
```

#Removing the noise from the image

```
c=imclearborder(b);
imshow(c);
d=imadjust(c);
figure, imshow(d);
Kmedian=medfilt2(d);
imshow(Kmedian);
e=imnoise(Kmedian,'gaussian',0.005);
f=wiener2(e,[5 5]);
imshow(f);
```

```
#to crop the image
```

```
figure,imshow(I,(1:100, :, : ))  
figure,imshow(I,(1:100, : ))
```

```
#to save the nuclei in the red channel for  
posterior usage
```

```
imwrite(R,'hela.red','jpeg')  
imwrite(R,'hela.red,jpeg')  
imwrite(R,'hela.red.jpeg','Quality',100)
```

```
the imtool allow us to interactively explore an image  
imtool(I);  
imtool('hela')  
imtool(I,'InitialMagnification',200)
```

```
BW= imread('acell.tif');  
imshow(BW)  
bwarea(BW)
```

```
#to find area perimeter minor and major axis  
of the nuclei
```

```
% clc;  
% close all;  
% clear all;  
I =imread('image1.jpg');  
I =rgb2gray(I);  
Threshold = 10;  
I2 = I>Threshold;  
imshow(I2,'InitialMagnification','fit');  
I3=imfill(I2,'holes');  
imshow(I2,'InitialMagnification','fit');  
I4=imclearborder(I3);  
imshow(I4,'InitialMagnification','fit');  
Kaverage = filter2(fspecial('average',3),I4)/255;  
figure;  
imshow(Kaverage,'InitialMagnification','fit');  
Kmedian = medfilt2(I4); figure;  
imshow(Kmedian,'InitialMagnification','fit');  
I5 = bwareaopen(Kmedian, 100);  
imshow (I5,'InitialMagnification','fit');  
I6 = watershed (I5);  
Lrgb = label2rgb(I6);  
imshow(Lrgb,'InitialMagnification','fit');  
imshow(imfuse(I5,Lrgb));  
axis([10 175 15 155])  
bw2 = ~bwareaopen(~I5, 10);  
imshow(bw2,'InitialMagnification','fit');  
D= -bwdist(~I5);  
imshow(D,[]);  
Ld = watershed(D);  
imshow (label2rgb(Ld));  
bw2=I5;  
bw2(Ld == 0) = 0;
```

Garima Verma

```
imshow(bw2)
mask = imextendedmin(D, 2);
imshowpair(I5,mask,'blend');
D2 = imimposemin(D,mask);
Ld2 = watershed(D2);
bw3 = I5;
bw3(Ld2 == 0) = 0;
imshow(bw3);

cc= bwconncomp(bw3, 8);
n= cc.NumObjects;

Area = zeros(n,1);
Perimeter = zeros(n,1);
Majoraxis = zeros(n,1);
Minoraxis = zeros(n,1);
K=
regionprops(cc, 'Area', 'Perimeter', 'MajorAxisLength', 'MinorAxisLength');

for i=1:n
    Area(i) = K(i).Area;
    Perimeter(i) = K(i).Perimeter;
    Majoraxis(i)=K(i).MajorAxisLength;
    Minoraxis(i)=K(i).MinorAxisLength;
end
graindata(1, 1) = mean(Area);
graindata(2, 1) = mean(Perimeter);
graindata(3, 1) = mean(Majoraxis);
graindata(4, 1) = mean(Minoraxis);
```

#To find the circularities of the image nuclei

```
I = imread('image4.jpg');
I = imresize(I, [500,500]);
subplot(2,2,1)
imshow(I)
title('input image')
grayimage = rgb2gray(I);
subplot(2,2,2)
imshow(grayimage)
title('GrayScale Image')
BinaryImage = imbinarize(grayimage);
subplot(2,2,3)
imshow(BinaryImage)
title('Initial (Noisy) Binary Image')
BinaryImage = bwareaopen(BinaryImage, 100);
subplot(2,2,4)
imshow(BinaryImage)
title('Cleaned Binary Image')
% BinaryImage2 = imclearborder(BinaryImage);
% subplot(3,2,1)
% imshow(BinaryImage2);
% title('Clear border Binary Image')
```

```
[LabeledImage, Numberofobjects] = bwlabel(BinaryImage);
Stats = regionprops('Table',
LabeledImage, 'FilledArea', 'Perimeter');
FilledImage = imfill(BinaryImage, 'holes');
Boundaries = bwboundaries(FilledImage);
Perimeter = [Stats.Perimeter];
FilledArea = [Stats.FilledArea];
Circularities= Perimeter.^2./(4*pi*FilledArea);
```

#calculating the contrast and homogeneity

```
I =imread('image4.jpg');
I2 =rgb2gray(I);
% imshow(I2);
glcm = graycomatrix(I2, 'offset',[2 0;0 2]);
stats = graycoprops(glcm,{'contrast','homogeneity'})
```

#To calculate the solidity of the Nuclei the code will run

```
I =imread('clearnuclie.jpg');
imshow(I)
% I2 =rgb2gray(I);
% % imshow(I2);
% glcm = graycomatrix(I2, 'offset',[2 0;0 2]);
% stats = graycoprops(glcm,{'contrast','homogeneity'})
BW = im2bw(I,graythresh(I));
L= bwlabel(BW);
figure, imshow(L)
S = regionprops(L, {'Solidity'});
Solidity = [S. Solidity];
figure; imshow(ismember(L, find(Solidity >= 0.9)))
figure; imshow(ismember(L, find(Solidity <= 0.25)))
```

#program for calculating circularities of Nuclei and tubulin

```
I = imread ('image1.jpg');
I = imresize(I, [500,500]);
subplot(2,2,1)
imshow (I)
title('input image')
grayimage = rgb2gray(I);
subplot(2,2,2)
imshow(grayimage)
title('GrayScale Image')
% BinaryImage = imbinarize(grayimage);
Threshold=10;
BinaryImage=grayimage>Threshold;
subplot(2,2,3)
imshow(BinaryImage)
title('Initial (Noisy) Binary Image')
BinaryImage = bwareaopen(BinaryImage, 100);
BinaryImage = imclearborder(BinaryImage);
subplot(2,2,4)
```

```
imshow(BinaryImage)
title('Cleaned Binary Image')
% BinaryImage2 = imclearborder(BinaryImage);
% subplot(2,3,[3,2])
% imshow(BinaryImage2);
% title('Clear border Binary Image')
J= entropy(BinaryImage);
[LabeledImage, Numberofobjects] = bwlabel(BinaryImage);
Stats = regionprops('Table',
LabeledImage, 'FilledArea', 'Perimeter');
FilledImage = imfill(BinaryImage, 'holes');
Boundaries = bwboundaries(FilledImage);
Perimeter = [Stats.Perimeter];
FilledArea = [Stats.FilledArea];
Circularities= Perimeter.^2./(4*pi*FilledArea);
```

#program with the water shed algorithm

```
I = imread ('image1.jpg');
I = imresize(I, [500,500]);
subplot(2,2,1)
imshow (I)
title('input image')
grayimage = rgb2gray(I);
subplot(2,2,2)
imshow(grayimage)
title('GrayScale Image')
% BinaryImage = imbinarize(grayimage);
Threshold=10;
BinaryImage=grayimage>Threshold;
subplot(2,2,3)
imshow(BinaryImage)
title('Initial (Noisy) Binary Image')
BinaryImage = bwareaopen(BinaryImage, 100);
BinaryImage = imclearborder(BinaryImage);
subplot(2,2,4)
imshow(BinaryImage)
title('Cleaned Binary Image')
% BinaryImage2 = imclearborder(BinaryImage);
% subplot(2,3,[3,2])
% imshow(BinaryImage2);
% title('Clear border Binary Image')
I6 = watershed (BinaryImage);
Lrgb = label2rgb(I6);
imshow(imfuse(BinaryImage,Lrgb));
axis([10 175 15 155])
bw2 = ~bwareaopen(~BinaryImage, 10);
D= -bwdist(~BinaryImage);
imshow(D, []);
Ld = watershed(D);
imshow (label2rgb(Ld));
bw2=BinaryImage;
bw2(Ld == 0) = 0;
```

```
mask = imextendedmin(D, 2);
% imshowpair(BinaryImage,mask,'blend');
D2 = imimposemin(D,mask);
Ld2 = watershed(D2);
bw3 = BinaryImage;
bw3(Ld2 == 0) = 0;
imshow(bw3);
J= entropy(BinaryImage);
[LabeledImage, Numberofobjects] = bwlabel(bw3);
Stats = regionprops('Table',
LabeledImage, 'FilledArea', 'Perimeter');
FilledImage = imfill(bw3, 'holes');
Boundaries = bwboundaries(FilledImage);
Perimeter = [Stats.Perimeter];
FilledArea = [Stats.FilledArea];
Circularities= Perimeter.^2./(4*pi*FilledArea);
```

References

- [1] Bianconi E, Piovesan A, Facchin F, Beraudi A, Casadei R, Frabetti F, et al. An estimation of the number of cells in the human body. *Ann Hum Biol* 2013;40:463–71. <https://doi.org/10.3109/03014460.2013.807878>.
- [2] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P. *Molecular Biology of the Cell*. 4th ed. Garland Science; 2002.
- [3] Allen WE. Terminologia anatomica: international anatomical terminology and Terminologia Histologica: International Terms for Human Cytology and Histology. *J Anat* 2009;215:221. https://doi.org/10.1111/j.1469-7580.2009.1093_1.x.
- [4] The evolution of the cytoskeleton | JCB n.d. <http://jcb.rupress.org/content/194/4/513> (accessed April 28, 2018).
- [5] Carpenter AE. Image-based chemical screening. *Nat Chem Biol* 2007;3:461–5. <https://doi.org/10.1038/nchembio.2007.15>.
- [6] Inglese J, Johnson RL, Simeonov A, Xia M, Zheng W, Austin CP, et al. High-throughput screening assays for the identification of chemical probes. *Nat Chem Biol* 2007;3:466–79. <https://doi.org/10.1038/nchembio.2007.17>.
- [7] Lewandoski M. Conditional control of gene expression in the mouse. *Nat Rev Genet* 2001;2:743–55. <https://doi.org/10.1038/35093537>.
- [8] Corre G, Stockholm D, Arnaud O, Kaneko G, Viñuelas J, Yamagata Y, et al. Stochastic Fluctuations and Distributed Control of Gene Expression Impact Cellular Memory. *PLoS ONE* 2014;9. <https://doi.org/10.1371/journal.pone.0115574>.
- [9] Oberholzer M, Östreicher M, Christen H, Brühlmann M. Methods in quantitative image analysis. *Histochem Cell Biol* 1996;105:333–55. <https://doi.org/10.1007/BF01463655>.
- [10] Breuer D, Ivakov A, Sampathkumar A, Hollandt F, Persson S, Nikoloski Z. Quantitative analyses of the plant cytoskeleton

- reveal underlying organizational principles. *J R Soc Interface* 2014;11:20140362. <https://doi.org/10.1098/rsif.2014.0362>.
- [11] Eliceiri KW, Berthold MR, Goldberg IG, Ibáñez L, Manjunath BS, Martone ME, et al. Biological imaging software tools. *Nat Methods* 2012;9:697–710. <https://doi.org/10.1038/nmeth.2084>.
- [12] Ferrell JE. Bistability, bifurcations, and Waddington’s epigenetic landscape. *Curr Biol CB* 2012;22:R458-466. <https://doi.org/10.1016/j.cub.2012.03.045>.
- [13] Kondo S, Miura T. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science* 2010;329:1616–20. <https://doi.org/10.1126/science.1179047>.
- [14] Lecuit T. “Developmental mechanics”: Cellular patterns controlled by adhesion, cortical tension and cell division. *HFSP J* 2008;2:72–8. <https://doi.org/10.2976/1.2896332>.
- [15] Alberghina L, Westerhoff HV, editors. *Systems Biology: Definitions and Perspectives*. Berlin Heidelberg: Springer-Verlag; 2005.
- [16] Bizzarri M, Palombo A, Cucina A. Theoretical aspects of Systems Biology. *Prog Biophys Mol Biol* 2013;112:33–43. <https://doi.org/10.1016/j.pbiomolbio.2013.03.019>.
- [17] Press TM. *Foundations of Systems Biology*. MIT Press n.d. <https://mitpress.mit.edu/books/foundations-systems-biology> (accessed May 25, 2019).
- [18] Mesarović MD. Systems Theory and Biology—View of a Theoretician. In: Mesarović MD, editor. *Syst. Theory Biol.*, Springer Berlin Heidelberg; 1968, p. 59–87.
- [19] Nelson CM, Bissell MJ. Of extracellular matrix, scaffolds, and signaling: tissue architecture regulates development, homeostasis, and cancer. *Annu Rev Cell Dev Biol* 2006;22:287–309. <https://doi.org/10.1146/annurev.cellbio.22.010305.104315>.
- [20] Newman SA, Forgacs G, Muller GB. Before programs: the physical origination of multicellular forms. *Int J Dev Biol* 2006;50:289–99. <https://doi.org/10.1387/ijdb.052049sn>.

- [21] Kohl P, Noble D. Systems biology and the virtual physiological human. *Mol Syst Biol* 2009;5:292. <https://doi.org/10.1038/msb.2009.51>.
- [22] Bertalanffy LV. *General System Theory: Foundations, Development, Applications*. Revised edition. New York: George Braziller Inc.; 1969.
- [23] Rejniak KA, editor. *Systems Biology of Tumor Microenvironment: Quantitative Modeling and Simulations*. Springer International Publishing; 2016.
- [24] Kitano H. Computational systems biology. *Nature* 2002;420:206–10. <https://doi.org/10.1038/nature01254>.
- [25] Fife CM, McCarroll JA, Kavallaris M. Movers and shakers: cell cytoskeleton in cancer metastasis. *Br J Pharmacol* 2014;171:5507–23. <https://doi.org/10.1111/bph.12704>.
- [26] Shaevitz J, Nørrelykke S. The cytoskeleton: I-beams of the cell. *Phys Today* 2010;63:60–1. <https://doi.org/10.1063/1.3326997>.
- [27] Bao G, Suresh S. Cell and molecular mechanics of biological materials. *Nat Mater* 2003;2:715–25. <https://doi.org/10.1038/nmat1001>.
- [28] Rathje L-SZ, Nordgren N, Pettersson T, Rönnlund D, Widengren J, Aspenström P, et al. Oncogenes induce a vimentin filament collapse mediated by HDAC6 that is linked to cell stiffness. *Proc Natl Acad Sci* 2014;111:1515–20. <https://doi.org/10.1073/pnas.1300238111>.
- [29] Hughes-Fulford M. Function of the cytoskeleton in gravisensing during spaceflight. *Adv Space Res Off J Comm Space Res COSPAR* 2003;32:1585–93. [https://doi.org/10.1016/S0273-1177\(03\)90399-1](https://doi.org/10.1016/S0273-1177(03)90399-1).
- [30] Mofrad MRK, Kamm R, editors. *Cytoskeletal Mechanics: Models and Measurements in Cell Mechanics*. 1 edition. Cambridge ; New York: Cambridge University Press; 2006.
- [31] Gittes F, Mickey B, Nettleton J, Howard J. Flexural rigidity of microtubules and actin filaments measured from thermal

- fluctuations in shape. *J Cell Biol* 1993;120:923–34. <https://doi.org/10.1083/jcb.120.4.923>.
- [32] Wade RH. On and around microtubules: an overview. *Mol Biotechnol* 2009;43:177–91. <https://doi.org/10.1007/s12033-009-9193-5>.
- [33] Ludueña RF. Multiple forms of tubulin: different gene products and covalent modifications. *Int Rev Cytol* 1998;178:207–75.
- [34] Herrmann H, Strelkov SV. History and phylogeny of intermediate filaments: now in insects. *BMC Biol* 2011;9:16. <https://doi.org/10.1186/1741-7007-9-16>.
- [35] Toivola DM, Tao G-Z, Habtezion A, Liao J, Omary MB. Cellular integrity plus: organelle-related and protein-targeting functions of intermediate filaments. *Trends Cell Biol* 2005;15:608–17. <https://doi.org/10.1016/j.tcb.2005.09.004>.
- [36] Capetanaki Y, Bloch RJ, Kouloumenta A, Mavroidis M, Psarras S. Muscle intermediate filaments and their links to membranes and membranous organelles. *Exp Cell Res* 2007;313:2063–76. <https://doi.org/10.1016/j.yexcr.2007.03.033>.
- [37] Ehler E, Perriard J-C. Cardiomyocyte Cytoskeleton and Myofibrillogenesis in Healthy and Diseased Heart. *Heart Fail Rev* 2000;5:259–69. <https://doi.org/10.1023/A:1009861504264>.
- [38] Herrmann H, Bär H, Kreplak L, Strelkov SV, Aebi U. Intermediate filaments: from cell architecture to nanomechanics. *Nat Rev Mol Cell Biol* 2007;8:562–73. <https://doi.org/10.1038/nrm2197>.
- [39] Herrmann H, Bär H, Kreplak L, Strelkov SV, Aebi U. Intermediate filaments: from cell architecture to nanomechanics. *Nat Rev Mol Cell Biol* 2007;8:562–73. <https://doi.org/10.1038/nrm2197>.
- [40] dos Remedios CG, Chhabra D, Kekic M, Dedova IV, Tsubakihara M, Berry DA, et al. Actin binding proteins:

- regulation of cytoskeletal microfilaments. *Physiol Rev* 2003;83:433–73. <https://doi.org/10.1152/physrev.00026.2002>.
- [41] Quintá HR, Galigniana NM, Erlejman AG, Lagadari M, Pilipuk GP, Galigniana MD. Management of cytoskeleton architecture by molecular chaperones and immunophilins. *Cell Signal* 2011;23:1907–20. <https://doi.org/10.1016/j.cellsig.2011.07.023>.
- [42] Portet S, Vassy J, Beil M, Millot G, Hebbache A, Rigaut JP, et al. Quantitative analysis of cytokeratin network topology in the MCF7 cell line. *Cytometry* 1999;35:203–13.
- [43] Bergeron B. *Bioinformatics Computing*. 1st ed. Prentice Hall.; 2002.
- [44] *Optical Microscopy - 1st Edition* n.d. <https://www.elsevier.com/books/optical-microscopy/herman/978-0-08-057139-3> (accessed May 26, 2019).
- [45] Thorn K. A quick guide to light microscopy in cell biology. *Mol Biol Cell* 2016;27:219–22. <https://doi.org/10.1091/mbc.E15-02-0088>.
- [46] Melan MA, Sluder G. Redistribution and differential extraction of soluble proteins in permeabilized cultured cells. Implications for immunofluorescence microscopy. *J Cell Sci* 1992;101 (Pt 4):731–43.
- [47] Pogliano K, Harry E, Losick R. Visualization of the subcellular location of sporulation proteins in *Bacillus subtilis* using immunofluorescence microscopy. *Mol Microbiol* 1995;18:459–70.
- [48] Pawley J, editor. *Handbook of Biological Confocal Microscopy*. 3rd ed. Springer US; 2006.
- [49] Kus J. Application of confocal laser-scanning microscopy (CLSM) to autofluorescent organic and mineral matter in peat, coals and siliciclastic sedimentary rocks — A qualitative approach. *Int J Coal Geol* 2015;137:1–18. <https://doi.org/10.1016/j.coal.2014.10.014>.

- [50] Tata BVR, Raj B. Confocal laser scanning microscopy: Applications in material science and technology. *Bull Mater Sci* 1998;21:263–78. <https://doi.org/10.1007/BF02744951>.
- [51] Electron Microscopy: Principles and Fundamentals. WileyCom n.d. <https://www.wiley.com/en-cf/Electron+Microscopy%3A+Principles+and+Fundamentals-p-9783527614554> (accessed May 26, 2019).
- [52] Winey M, Meehl JB, O’Toole ET, Giddings TH. Conventional transmission electron microscopy. *Mol Biol Cell* 2014;25:319–23. <https://doi.org/10.1091/mbc.E12-12-0863>.
- [53] Curry A, Appleton H, Dowsett B. Application of transmission electron microscopy to the clinical study of viral and bacterial infections: present and future. *Micron Oxf Engl* 1993 2006;37:91–106. <https://doi.org/10.1016/j.micron.2005.10.001>.
- [54] Stevenson HP, Makhov AM, Calero M, Edwards AL, Zeldin OB, Mathews II, et al. Use of transmission electron microscopy to identify nanocrystals of challenging protein targets. *Proc Natl Acad Sci U S A* 2014;111:8470–5. <https://doi.org/10.1073/pnas.1400240111>.
- [55] Atomic Force Microscopy: Understanding Basic Modes and Advanced Applications. WileyCom n.d. <https://www.wiley.com/en-us/Atomic+Force+Microscopy%3A+Understanding+Basic+Modes+and+Advanced+Applications-p-9780470638828> (accessed May 26, 2019).
- [56] Yun MV and K. Study of Atomic Force Microscopy in Pharmaceutical and Biopharmaceutical Interactions - A Mini Review. *Curr Pharm Anal* 2009. <http://www.eurekaselect.com/69746/article> (accessed May 26, 2019).
- [57] Last JA, Russell P, Nealey PF, Murphy CJ. The Applications of Atomic Force Microscopy to Vision Science. *Invest Ophthalmol Vis Sci* 2010;51:6083–94. <https://doi.org/10.1167/iovs.10-5470>.

- [58] Braet F, Kalle WH, De Zanger RB, De Grooth BG, Raap AK, Tanke HJ, et al. Comparative atomic force and scanning electron microscopy: an investigation on fenestrated endothelial cells in vitro. *J Microsc* 1996;181:10–7.
- [59] Braet F, de Zanger R, Seynaeve C, Baekeland M, Wisse E. A comparative atomic force microscopy study on living skin fibroblasts and liver endothelial cells. *J Electron Microsc (Tokyo)* 2001;50:283–90. <https://doi.org/10.1093/jmicro/50.4.283>.
- [60] Wilson SM, Bacic A. Preparation of plant cells for transmission electron microscopy to optimize immunogold labeling of carbohydrate and protein epitopes. *Nat Protoc* 2012;7:1716–27. <https://doi.org/10.1038/nprot.2012.096>.
- [61] Osborn M, Webster RE, Weber K. Individual microtubules viewed by immunofluorescence and electron microscopy in the same PtK2 cell. *J Cell Biol* 1978;77:R27–34. <https://doi.org/10.1083/jcb.77.3.r27>.
- [62] Borek F. The fluorescent antibody method in medical and biological research. *Bull World Health Organ* 1961;24:249–56.
- [63] Weber K, Rathke PC, Osborn M. Cytoplasmic microtubular images in glutaraldehyde-fixed tissue culture cells by electron microscopy and by immunofluorescence microscopy. *Proc Natl Acad Sci U S A* 1978;75:1820–4.
- [64] Le Dévédec SE, Geverts B, de Bont H, Yan K, Verbeek FJ, Houtsmuller AB, et al. The residence time of focal adhesion kinase (FAK) and paxillin at focal adhesions in renal epithelial cells is determined by adhesion size, strength and life cycle status. *J Cell Sci* 2012;125:4498–506. <https://doi.org/10.1242/jcs.104273>.
- [65] Severin S, Gaits-Iacovoni F, Allart S, Gratacap M-P, Payrastra B. A confocal-based morphometric analysis shows a functional crosstalk between the actin filament system and microtubules in thrombin-stimulated platelets. *J Thromb Haemost JTH* 2013;11:183–6. <https://doi.org/10.1111/jth.12053>.

- [66] Carpenter AE, Jones TR, Lamprecht MR, Clarke C, Kang IH, Friman O, et al. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol* 2006;7:R100. <https://doi.org/10.1186/gb-2006-7-10-r100>.
- [67] Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of Image Analysis. *Nat Methods* 2012;9:671–5.
- [68] Rueden CT, Schindelin J, Hiner MC, DeZonia BE, Walter AE, Arena ET, et al. ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics* 2017;18:529. <https://doi.org/10.1186/s12859-017-1934-z>.
- [69] Schindelin J, Rueden CT, Hiner MC, Eliceiri KW. The ImageJ ecosystem: an open platform for biomedical image analysis. *Mol Reprod Dev* 2015;82:518–29. <https://doi.org/10.1002/mrd.22489>.
- [70] Chaumont F de, Dallongeville S, Chenouard N, Hervé N, Pop S, Provoost T, et al. Icy: an open bioimage informatics platform for extended reproducible research. *Nat Methods* 2012;9:690–6. <https://doi.org/10.1038/nmeth.2075>.
- [71] matlab_tutorial.pdf n.d.
- [72] introduction-to-matlab.pdf n.d.
- [73] September 1 P, Share 2005. Image Analysis of Neurite Branching: High-Content Screening at High Speed n.d. <http://americanlaboratory.com/914-Application-Notes/30737-Image-Analysis-of-Neurite-Branching-High-Content-Screening-at-High-Speed/> (accessed June 18, 2019).
- [74] Mendoza F, Lu R. Basics of Image Analysis. In: Park B, Lu R, editors. *Hyperspectral Imaging Technol. Food Agric.*, New York, NY: Springer New York; 2015, p. 9–56. https://doi.org/10.1007/978-1-4939-2836-1_2.
- [75] Urbančič V, Butler R, Richier B, Peter M, Mason J, Livesey FJ, et al. Filopodyan: An open-source pipeline for the analysis of filopodia. *J Cell Biol* 2017;216:3405–22. <https://doi.org/10.1083/jcb.201705113>.

- [76] Meijering E, Jacob M, Sarria J-CF, Steiner P, Hirling H, Unser M. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytom Part J Int Soc Anal Cytol* 2004;58:167–76. <https://doi.org/10.1002/cyto.a.20022>.
- [77] Moroz PE. The cell in the field of gravity and the centrifugal field. *J Theor Biol* 1984;107:303–20.
- [78] What are Fractals? – Fractal Foundation n.d. <https://fractalfoundation.org/resources/what-are-fractals/> (accessed February 15, 2019).
- [79] What is fractal? - Definition from WhatIs.com. WhatIsCom n.d. <https://whatis.techtarget.com/definition/fractal> (accessed February 15, 2019).
- [80] Ranganath A, Mishra J. New Approach for Estimating Fractal Dimension of Both Gray and Color Images. 2017 IEEE 7th Int. Adv. Comput. Conf. IACC, 2017, p. 678–83. <https://doi.org/10.1109/IACC.2017.0142>.
- [81] Napolitano A, Ungania S, Cannata V. Fractal Dimension Estimation Methods for Biomedical Images. *MATLAB - Fundam Tool Sci Comput Eng Appl - Vol 3* 2012. <https://doi.org/10.5772/48760>.
- [82] https://www.nasa.gov/mission_pages/station/research/experiments/explorer/Investigation.html?id=1555
- [83] Masiello MG, Cucina A, Proietti S, Palombo A, Coluccia P, D’Anselmi F, et al. Phenotypic Switch Induced by Simulated Microgravity on MDA-MB-231 Breast Cancer Cells. *BioMed Res Int* 2014. <https://doi.org/10.1155/2014/652434>.
- [84] Higaki T. Quantitative evaluation of cytoskeletal organizations by microscopic image analysis. *Plant Morphol* 2017;29:15–21. <https://doi.org/10.5685/plmorphol.29.15>.
- [85] Borst AG, van Loon JJWA. Technology and Developments for the Random Positioning Machine, RPM. *Microgravity Sci Technol* 2009;21:287–92. <https://doi.org/10.1007/s12217-008-9043-2>.

- [86] Pardo SJ, Patel MJ, Sykes MC, Platt MO, Boyd NL, Sorescu GP, et al. Simulated microgravity using the Random Positioning Machine inhibits differentiation and alters gene expression profiles of 2T3 preosteoblasts. *Am J Physiol-Cell Physiol* 2005;288:C1211–21. <https://doi.org/10.1152/ajpcell.00222.2004>.
- [87] Ferranti F, Caruso M, Cammarota M, Masiello MG, Corano Scheri K, Fabrizi C, et al. Cytoskeleton Modifications and Autophagy Induction in TCam-2 Seminoma Cells Exposed to Simulated Microgravity. *BioMed Res Int* 2014. <https://doi.org/10.1155/2014/904396>.
- [88] Basu S, Liu C, Rohde GK. Localizing and extracting filament distributions from microscopy images. *J Microsc* 2015;258:13–23. <https://doi.org/10.1111/jmi.12209>.
- [89] Kartasalo K, Pölönen R-P, Ojala M, Rasku J, Lekkala J, Aalto-Setälä K, et al. CytoSpectre: a tool for spectral analysis of oriented structures on cellular and subcellular levels. *BMC Bioinformatics* 2015;16:344. <https://doi.org/10.1186/s12859-015-0782-y>.
- [90] Carrillo M, González JM. A new approach to modelling sigmoidal curves. *Technol Forecast Soc Change* 2002;69:233–41. [https://doi.org/10.1016/S0040-1625\(01\)00150-0](https://doi.org/10.1016/S0040-1625(01)00150-0).
- [91] Gangan MS, Athale CA. Threshold effect of growth rate on population variability of Escherichia coli cell lengths. *R Soc Open Sci* 2017;4:160417. <https://doi.org/10.1098/rsos.160417>.
- [92] Ong KH, De J, Cheng L, Ahmed S, Yu W. NeuronCyto II: An automatic and quantitative solution for crossover neural cells in high throughput screening. *Cytometry A* 2016;89:747–54. <https://doi.org/10.1002/cyto.a.22872>.
- [93] Soille P. *Morphological Image Analysis: Principles and Applications*. 2nd ed. Berlin Heidelberg: Springer-Verlag; 2004.
- [94] Gonzalez. *Digital Image Processing Using Matlab*. New Delhi: Dorling Kindersley Pvt Ltd; 2006.
- [95] Bala A. *An Improved Watershed Image Segmentation Technique using MATLAB*, 2012.

- [96] Zhou F, Yang W, Liao Q. A Coarse-to-Fine Subpixel Registration Method to Recover Local Perspective Deformation in the Application of Image Super-Resolution. *Trans Img Proc* 2012;21:53–66. <https://doi.org/10.1109/TIP.2011.2159731>.
- [97] Dutta NA Mohit srivastava, Maitreyee. Comparative Analysis of Pixel-Based and Object-Based Classification of High Resolution Remote Sensing Images – A Review. *Int J Eng Trends Technol - IJETT* n.d.
- [98] El-Sayed MA, Hafeez TA-E. New Edge Detection Technique based on the Shannon Entropy in Gray Level Images. *ArXiv12112502 Cs* 2012.
- [99] Croonenbroeck C. Local entropy based image reconstruction 2012.
- [100] Dubin-Thaler BJ, Hofman JM, Cai Y, Xenias H, Spielman I, Shneidman AV, et al. Quantification of Cell Edge Velocities and Traction Forces Reveals Distinct Motility Modules during Cell Spreading. *PLOS ONE* 2008;3:e3735. <https://doi.org/10.1371/journal.pone.0003735>.
- [101] Aplin JD, Bardsley WG, Niven VM. Kinetic analysis of cell spreading. II. Substratum adhesion requirements of amniotic epithelial (FL) cells. *J Cell Sci* 1983;61:375–88.
- [102] Kauanova S, Vorobjev I, James AP. Automated image segmentation for detecting cell spreading for metastasizing assessments of cancer development. 2017 *Int Conf Adv Comput Commun Inform ICACCI* 2017:2382–7. <https://doi.org/10.1109/ICACCI.2017.8126203>.
- [103] NeuronJ: Manual n.d. <https://imagescience.org/meijering/software/neuronj/manual/> (accessed May 20, 2019).
- [104] Laketa V, Simpson JC, Bechtel S, Wiemann S, Pepperkok R. High-content microscopy identifies new neurite outgrowth regulators. *Mol Biol Cell* 2007;18:242–52. <https://doi.org/10.1091/mbc.e06-08-0666>.
- [105] Verma G, Palombo A, Grigioni M, La Monaca M, D’Avenio G. Systems Biology-Driven Hypotheses Tested In Vivo: The

Need to Advancing Molecular Imaging Tools. *Methods Mol Biol* Clifton NJ 2018;1702:337–59.
https://doi.org/10.1007/978-1-4939-7456-6_17.

List of Publications

- Garima Verma, Maria Laura Luciani, Alessandro Palombo, Linda Metaxa, Giovanna Panzironi, Federica Pediconi, Alessandro Giuliani, Mariano Bizzarri, Virginia Todde. Microcalcification morphological descriptors and parenchyma fractal dimension hierarchically interact in breast cancer: a diagnostic perspective (Published in “Computers in Biology and Medicine”).
- Garima Verma, Alessandro Palombo, Mauro Grigioni, Morena La Monaca, Giuseppe D’Avenio. Systems Biology-driven hypotheses tested in vivo: the need to advancing molecular imaging tool. Springer Methods Molecular Biology, Chapter of book Systems Biology, ISBN 978-1-4939-7455-9.
- Integrated framework for quantitative assessment of Cytoskeleton morphology (Review manuscript being prepared).